



Entwicklerhandbuch

Amazon Simple Workflow Service



API-Version 2012-01-25

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

Amazon Simple Workflow Service: Entwicklerhandbuch

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

Die Marken und Handelsmarken von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, die geeignet ist, Kunden irrezuführen oder Amazon in irgendeiner Weise herabzusetzen oder zu diskreditieren. Alle anderen Marken, die nicht im Besitz von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist Amazon SWF?	1
Workflow-Komponenten	3
Workflow-Komponenten	3
Ihren Workflow ausführen	5
Einrichtung der -Entwicklungsumgebung	6
Entwickeln Sie mit AWS SDKs	6
Betrachten Sie die AWS Flow Framework	7
Erste Schritte	8
Über den Workflow	9
Voraussetzungen	10
Anleitungsschritte	10
Teil 1: Amazon SWF mit dem SDK for Ruby verwenden	10
Schließen Sie ein AWS SDK für Ruby	11
Konfiguration der Sitzung AWS	11
Registrierung einer Amazon SWF-Domain	13
Nächste Schritte	14
Teil 2: Implementieren des Workflows	14
Entwerfen des Workflows	14
Einrichten des Workflow-Codes	16
Registrieren des Workflows	17
Abrufen von Entscheidungen	18
Starten der Workflow-Ausführung	22
Nächste Schritte	24
Teil 3: Implementieren der Aktivitäten	24
Definieren eines Basis-Aktivitätstyps	24
Definieren von GetContactActivity	27
Definieren von SubscribeTopicActivity	29
Definieren von WaitForConfirmationActivity	32
Definieren von SendResultActivity	35
Nächste Schritte	36
Teil 4: Implementieren des Pollers für Aktivitätsaufgaben	37
Ausführen des Workflows	40
Wie geht es weiter?	44
Arbeiten in der Konsole	45

Registrieren einer Domain	45
Workflowtypen registrieren	46
Aktivitätstypen registrieren	47
Einen Workflow starten	47
So starten Sie eine Workflow-Ausführung mithilfe der Konsole	48
Workflow-Ausführungen verwalten	48
Grundkonzepte	52
Einen Workflow erstellen	54
Modellieren des Workflows und der entsprechenden Aktivitäten	54
Workflows ausführen	55
Workflow-Verlauf	56
Objekt-Identifikatoren	61
Domains	61
Akteure	62
Was ist ein Schauspieler in Amazon SWF?	63
Workflow-Starter	64
Entscheider	64
Aktivitäts-Worker	65
Datenaustausch zwischen Akteuren	66
Aufgaben	66
Aufgabenlisten	68
Entscheidungsaufgabenlisten	68
Aktivitätsaufgabenlisten	69
Aufgabenweiterleitung	69
Abschluss der Workflow-Ausführung	70
Lebenszyklus der Workflow-Ausführung	71
Lebenszyklus einer Workflow-Ausführung	71
Abfragen nach Aufgaben	80
Fortgeschrittene Konzepte	81
Versionsverwaltung	81
Signale	82
Workflows für Kinder	84
Marker	86
Tags	88
Verwalten von Tags	88
Taggen Sie Workflow-Ausführungen	89

Steuern Sie den Zugriff auf Domänen mit Tags	90
Exklusive Wahl	91
Timer	94
Aktivitätsaufgaben stornieren	94
Sicherheit	98
Datenschutz	98
Verschlüsselung	99
Identitäts- und Zugriffsverwaltung	100
Zielgruppe	101
Authentifizierung mit Identitäten	101
Verwalten des Zugriffs mit Richtlinien	103
Zugriffskontrolle	105
Richtlinienaktionen	105
Richtlinienressourcen	106
Bedingungsschlüssel für die Richtlinie	106
ACLs	107
ABAC	107
Temporäre Anmeldeinformationen	108
Prinzipalberechtigungen	108
Servicerollen	108
Service-verknüpfte Rollen	108
Identitätsbasierte Richtlinien	109
Ressourcenbasierte Richtlinien	109
So funktioniert Amazon Simple Workflow Service mit IAM	110
Beispiele für identitätsbasierte Richtlinien	111
Grundlegende Prinzipien	114
Amazon SWF IAM-Richtlinien	115
Übersicht über API-Befehle	122
Tagbasierte Richtlinien	132
Amazon VPC-Endpunkte	132
Fehlerbehebung	134
Protokollieren und Überwachen	136
Amazon SWF-Metriken für CloudWatch	137
Amazon SWF-Metriken anzeigen	147
Aufnahme zu CloudTrail	151
EventBridge für Amazon SWF	158

Verwendung AWS-Benutzerbenachrichtigungen mit Amazon SWF	167
Compliance-Validierung	167
Ausfallsicherheit	168
Infrastruktursicherheit	168
Konfigurations- und Schwachstellenanalyse	169
Mit dem AWS CLI	170
Arbeitet mit APIs	172
Senden von HTTP-Anfragen	172
Inhalt des HTTP-Headers	173
HTTP-Textinhalt	175
Beispiel , JSON-Anforderung und -Antwort	175
Berechnen der HMAC-SHA-Signatur	177
Liste der Amazon SWF SWF-Aktionen	179
Aktionen im Zusammenhang mit Aktivitäten	180
Aktionen im Zusammenhang mit Entscheidern	180
Aktionen im Zusammenhang mit Workflow-Ausführungen	180
Aktionen im Zusammenhang mit der Administration	181
Sichtbarkeitsaktionen	182
Registrieren einer Domäne	183
Weitere Informationen finden Sie unter:	183
Timeout-Werte festlegen	183
Kontingente für Timeout-Werte	184
Timeouts für Workflow-Ausführungen und Entscheidungsaufgaben	184
Timeouts für Aktivitätsaufgaben	184
Weitere Informationen finden Sie unter:	185
Registrieren eines Workflow-Typs	185
Weitere Informationen finden Sie unter:	186
Registrieren eines Aktivitätstyps	186
Weitere Informationen finden Sie unter:	187
Lambda-Aufgaben	187
Über AWS Lambda	187
Vorteile und Einschränkungen der Verwendung von Lambda-Aufgaben	188
Verwenden von Lambda-Aufgaben in Ihren Workflows	188
Entwickeln eines Aktivitäts-Workers	193
Abrufen von Aktivitätsaufgaben	194
Verarbeiten von Aktivitätsaufgaben	195

Senden von Heartbeats für Aktivitätsaufgaben	195
Abschließen oder fehlschlagen einer Aktivitätsaufgabe	196
Starten von Aktivitäts-Workern	198
Entwicklung von Entscheidungsträgern	198
Definieren einer Koordinationslogik	200
Abrufen von Entscheidungsaufgaben	200
Anwenden der Koordinationslogik	202
Reagieren auf Entscheidungen	203
Beenden der Workflow-Ausführung	204
Starten von Entscheidern	206
Workflows starten	207
Einstellung der Aufgabenpriorität	208
Einrichten der Aufgabenpriorität für Workflows	209
Einrichten der Aufgabenpriorität für Aktivitäten	211
Aktionen, die Informationen zur Aufgabenpriorität zurückgeben	212
Fehlerbehandlung	212
Validierungsfehler	213
Fehler beim Umsetzen von Aktionen und Entscheidungen	213
Timeouts	214
Fehler durch Benutzercode	214
Fehler beim Schließen einer Workflow-Ausführung	214
Kontingente	216
Allgemeine Kontokontingente für Amazon SWF	216
Kontingente für Workflow-Ausführungen	217
Kontingente für Aufgabenausführungen	218
Drosselungskontingente für Amazon SWF	219
Drosselungsquoten für alle Regionen	220
Entscheidungsquoten für alle Regionen	222
Kontingente auf Workflow-Ebene	222
Beantragen einer Kontingenterhöhung	223
Weitere Ressourcen	224
Zeitüberschreitungstypen	224
Zeitüberschreitungen in Workflow- und Entscheidungsaufgaben	225
Zeitüberschreitungen in Aktivitätsaufgaben	226
Endpunkte	227
Ergänzende -Dokumentation	228

Amazon Simple Workflow Service API-Referenz	228
AWS Flow Framework Dokumentation	228
AWS SDK-Dokumentation	228
AWS CLI Dokumentation	230
Webressourcen	231
Amazon SWF SWF-Forum	231
Häufig gestellte Fragen zu Amazon SWF	231
Amazon SWF SWF-Videos	231
Optionen für Ruby Flow	231
Weitere Nutzung des Ruby Flow Frameworks	232
Migration auf das Java Flow Framework	232
Migration auf Step Functions	233
Verwenden Sie die Amazon SWF SWF-API direkt	234
Dokumentverlauf	235
.....	ccxxxix

Was ist Amazon Simple Workflow Service?

Mit Amazon Simple Workflow Service (Amazon SWF) können Sie Hintergrundjobs mit parallel oder sequentiellen Schritten erstellen, ausführen und skalieren. Sie können die Arbeit über verteilte Komponenten hinweg koordinieren und den Status von Aufgaben verfolgen.

In Amazon SWF stellt eine Aufgabe eine logische Arbeitseinheit dar, die von einer Komponente Ihrer Anwendung ausgeführt wird. Die übergreifende Koordination von Aufgaben umfasst die Verwaltung der Abhängigkeiten zwischen den einzelnen Aufgaben, die Planung und die Parallelität im Ablauf Ihrer Anwendung. Mit Amazon SWF können Sie Aufgaben steuern und koordinieren, ohne sich Gedanken über die zugrunde liegende Komplexität machen zu müssen, wie z. B. die Nachverfolgung des Fortschritts und die Beibehaltung des Aufgabenstatus.

Wenn Sie Amazon SWF verwenden, implementieren Sie Worker zur Ausführung von Aufgaben. Mitarbeiter können entweder in einer Cloud-Infrastruktur wie Amazon Elastic Compute Cloud (Amazon EC2) oder in Ihren eigenen Räumlichkeiten ausgeführt werden. Sie können Aufgaben erstellen, die lange andauern oder die möglicherweise fehlschlagen, eine Zeitüberschreitung haben oder Neustarts erfordern — oder die mit unterschiedlichem Durchsatz und unterschiedlicher Latenz abgeschlossen werden können. Amazon SWF speichert Aufgaben und weist sie Mitarbeitern zu, wenn sie bereit sind, verfolgt den Fortschritt und behält den Status bei, einschließlich Einzelheiten zum Abschluss von Aufgaben.

Um Aufgaben zu koordinieren, schreiben Sie ein Programm, das den neuesten Aufgabenstatus von Amazon SWF abrufen und diesen Status verwendet, um nachfolgende Aufgaben zu initiieren. Amazon SWF hält den Ausführungsstatus einer Anwendung dauerhaft aufrecht, sodass Ihre Anwendung gegen einzelne Komponentenausfälle resistent ist. Mit Amazon SWF können Sie Anwendungskomponenten unabhängig voneinander erstellen, bereitstellen, skalieren und ändern.

Andere AWS Workflow-Dienste

Für die meisten Anwendungsfälle empfehlen wir, Ihre Workflow- und Orchestrierungsanforderungen in Betracht AWS Step Functions zu ziehen.

Mit Step Functions können Sie Workflows, auch Zustandsmaschinen genannt, erstellen, um verteilte Anwendungen zu erstellen, Prozesse zu automatisieren, Microservices zu orchestrieren und Daten- und Machine-Learning-Pipelines zu erstellen. In der Konsole oder im AWS Toolkit von Step Functions in VS Code können Sie das grafische Workflow Studio verwenden, um den Workflow Ihrer Anwendung zu visualisieren, zu bearbeiten, zu testen und zu debuggen.

[Weitere technische Informationen finden Sie im Entwicklerhandbuch.AWS Step Functions](#)

Entwickeln von Workflow-Komponenten mit Amazon SWF

Die Entwicklung verteilter Anwendungen erfordert die Koordination vieler Komponenten und den Umgang mit Latenz und Unzuverlässigkeit, die mit der Fernkommunikation einhergehen.

Mit Amazon Simple Workflow Service (Amazon SWF) können Sie asynchrone und verteilte Anwendungen entwickeln, indem Sie ein Programmiermodell und eine Infrastruktur bereitstellen, um verteilte Komponenten zu koordinieren und ihren Ausführungsstatus auf zuverlässige Weise aufrechtzuerhalten. Wenn Sie sich auf Amazon SWF verlassen, können Sie sich darauf konzentrieren, die Aspekte Ihrer Anwendung zu entwickeln, die sie von anderen abheben.

Komponenten eines Workflows

[Komponenten eines Workflows](#)Das grundlegende Konzept von Amazon SWF ist der Workflow. Ein Workflow ist eine Reihe von Aktivitäten, die ein Ziel verfolgen, zusammen mit der Logik, die die Aktivitäten koordiniert. Ein Workflow könnte beispielsweise eine Kundenbestellung entgegennehmen und alle zur Ausführung der Bestellung erforderlichen Maßnahmen ergreifen.

Jeder Workflow wird in einer als Domäne bezeichneten Ressource ausgeführt, die den Umfang des Workflows steuert. Ein AWS Konto kann mehrere Domänen haben, von denen jede mehrere Workflows enthalten kann, aber Workflows in verschiedenen Domänen können nicht interagieren.

Beim Entwerfen eines Amazon SWF SWF-Workflows definieren Sie jede der erforderlichen Aktivitäten. Anschließend registrieren Sie jede Aktivität bei Amazon SWF als Aktivitätstyp. Sie geben einen Namen, eine Version und Timeout-Werte an. Beispielsweise kann es sein, dass der Kunde erwartet, dass eine Bestellung innerhalb von 24 Stunden geliefert wird.

Im Verlauf der Durchführung des Workflows müssen manche Aktivitäten unter Umständen mehr als nur einmal durchgeführt werden, vielleicht mit unterschiedlichen Eingaben. So haben Sie vielleicht beispielsweise in einem Kundenbestellungs-Workflow eine Aktivität, die erworbene Artikel verarbeitet. Wenn der Kunde mehrere Artikel kauft, müsste diese Aktivität mehrmals ausgeführt werden. Amazon SWF hat das Konzept einer Aktivitätsaufgabe, die einen Aufruf einer Aktivität darstellt. In unserem Beispiel entspricht das Verarbeiten eines einzelnen Artikels einer einzelnen Aktivitätsaufgabe.

Ein Activity Worker ist ein Programm, das Aktivitätsaufgaben empfängt, sie ausführt und Ergebnisse liefert. Die Aufgabe könnte tatsächlich von einer Person ausgeführt werden. Ein statistischer Analyst könnte beispielsweise Datensätze erhalten, die Daten analysieren und dann die Analyse zurücksenden.

Aktivitätsaufgaben und die Aktivitätshelfer, die sie ausführen, können synchron oder asynchron ausgeführt werden. Mitarbeiter können an einem Standort arbeiten oder auf mehrere Computer verteilt sein, möglicherweise in unterschiedlichen geografischen Regionen. Unterschiedliche Aktivitäts-Worker können in verschiedenen Programmiersprachen geschrieben und auf unterschiedlichen Betriebssystemen ausgeführt werden. Beispielsweise kann ein Activity Worker auf einem Server in Asien ausgeführt werden, während ein anderer auf einem mobilen Gerät in Nordamerika ausgeführt wird.

Die Koordinationslogik in einem Workflow ist in einem Softwareprogramm enthalten, das als Entscheider bezeichnet wird. Ein Entscheider plant Aktivitätsaufgaben, gibt Anregungen für die Mitarbeiter, verarbeitet Ereignisse, die während der Ausführung des Workflows eintreffen, und beendet (oder schließt) den Workflow, nachdem das Ziel erreicht wurde.

Die Rolle des Amazon SWF-Service besteht darin, als zuverlässiger zentraler Knotenpunkt zu fungieren, über den Daten zwischen dem Entscheider, den Mitarbeitern der Aktivität und anderen relevanten Stellen wie der Person, die den Workflow verwaltet, ausgetauscht werden. Amazon SWF behält auch den Status jeder Workflow-Ausführung bei, sodass Ihre Anwendung den Status nicht dauerhaft speichern muss.

Der Entscheider steuert den Arbeitsablauf, indem er Entscheidungsaufgaben von Amazon SWF empfängt und Amazon SWF mit Entscheidungen zurückantwortet. Eine Entscheidung stellt eine Aktion oder eine Reihe von Aktionen dar, bei denen es sich um die nächsten Schritte im Arbeitsablauf handelt. Eine typische Entscheidung ist das Planen einer Aktivitätsaufgabe. Entscheidungen können auch verwendet werden, um Aufgaben mit Zeitschaltern zu verzögern, die Stornierung laufender Aufgaben zu beantragen und Workflows abzuschließen.

Der Mechanismus, mit dem sowohl die Aktivitätsarbeiter als auch der Entscheider ihre Aufgaben (Aktivitätsaufgaben bzw. Entscheidungsaufgaben) erhalten, besteht darin, den Amazon SWF-Service abzufragen.

Amazon SWF informiert den Entscheider über den Status des Workflows, indem es jeder Entscheidungsaufgabe eine Kopie des aktuellen Workflow-Ausführungsverlaufs beifügt. Der Workflow-Ausführungsverlauf setzt sich zusammen aus Ereignissen, wobei ein Ereignis eine bedeutende Änderung des Status der Workflow-Ausführung darstellt. Beispiele für Ereignisse sind der Abschluss von Aufgaben, Timeouts für Aufgaben oder der Ablauf eines Timers. Der Verlauf ist eine umfassende, konsistente und autoritative Aufzeichnung des Fortschritts des Workflows.

Amazon SWF Access Control verwendet AWS Identity and Access Management (IAM), sodass Sie den Zugriff auf AWS Ressourcen kontrollieren können. Sie können einem Benutzer beispielsweise

Zugriff auf Ihr Konto gewähren, aber nur, um bestimmte Workflows in einer speziellen Domäne auszuführen.

Ihren Workflow ausführen

Im Folgenden finden Sie einen Überblick über die Schritte, die zur Entwicklung und Ausführung eines Workflows in Amazon SWF erforderlich sind:

1. Schreiben Sie Aktivitätsmitarbeitern, die die Verarbeitungsschritte in Ihrem Workflow ausführen sollen.
2. Schreiben Sie einen Entscheider, um die Koordinationslogik Ihres Workflows zu regeln.
3. Registrieren Sie Ihre Aktivitäten und Ihren Workflow bei Amazon SWF.

Sie können diesen Schritt programmgesteuert oder mithilfe von ausführen. AWS-Managementkonsole

4. Starten Sie Ihre Aktivitäts-Worker und den Entscheider.

Diese Akteure können auf jedem Computergerät ausgeführt werden, das auf einen Amazon SWF SWF-Endpunkt zugreifen kann. Sie könnten beispielsweise Recheninstanzen in der Cloud wie Amazon Elastic Compute Cloud (Amazon EC2), Server in Ihrem Rechenzentrum oder sogar ein Mobilgerät verwenden, um einen Entscheider oder Activity Worker zu hosten. Nach dem Start sollten die Entscheider- und Aktivitätsmitarbeiter damit beginnen, Amazon SWF nach Aufgaben abzufragen.

5. Starten Sie eine oder mehrere Ausführungen Ihres Workflows.

Sie können Workflows programmgesteuert oder über den starten. AWS-Managementkonsole

Jede Ausführung wird unabhängig ausgeführt und Sie können jeder Ausführung einen eigenen Satz von Eingabedaten bereitstellen. Wenn eine Ausführung gestartet wird, plant Amazon SWF die erste Entscheidungsaufgabe. Als Reaktion darauf beginnt Ihr Entscheider mit der Generierung von Entscheidungen, die Aktivitätsaufgaben einleiten. Die Ausführung dauert an, bis Ihr Entscheider die Entscheidung trifft, die Ausführung abzuschließen.

6. Zeigen Sie Workflow-Ausführungen mit dem an. AWS-Managementkonsole

Sie können vollständige Details zu laufenden und abgeschlossenen Ausführungen filtern und anzeigen. Sie können beispielsweise eine offene Ausführung auswählen, um zu sehen, welche Aufgaben abgeschlossen wurden und welche Ergebnisse sie hatten.

Einrichtung der -Entwicklungsumgebung

Sie haben die Möglichkeit, für Amazon SWF in einer der von unterstützten Programmiersprachen zu entwickeln AWS. Für Java-Entwickler AWS Flow Framework ist das ebenfalls verfügbar. Weitere Informationen finden Sie [AWS Flow Framework](#) auf der Website und im [Java Developer Guide](#). AWS Flow Framework

Um die Latenz zu reduzieren und Daten an einem Ort zu speichern, der Ihren Anforderungen entspricht, bietet Amazon SWF Endgeräte in verschiedenen Regionen.

Jeder Endpunkt in Amazon SWF ist völlig unabhängig. Bei Domains, Workflows und Aktivitäten, die Sie in einer Region registriert haben, werden keine Daten oder Attribute mit denen in einer anderen Region geteilt.

Wenn Sie eine Amazon SWF-Domain, einen Workflow oder eine Aktivität registrieren, ist diese nur in der Region vorhanden, in der Sie sie registriert haben. Sie könnten beispielsweise eine Domain registrieren, die SWF-Flows-1 in zwei verschiedenen Regionen benannt ist, aber sie teilen keine Daten oder Attribute miteinander, sodass jede Region als völlig unabhängige Domain fungiert.

Eine Liste der Amazon SWF SWF-Endpunkte finden Sie unter [Regionen und Endpunkte](#).

Entwickeln Sie mit AWS SDKs

Amazon SWF wird von den AWS SDKs für Java, .NET, Node.js, PHP, Python und Ruby unterstützt und bietet so eine bequeme Möglichkeit, die Amazon SWF HTTP-API in der Programmiersprache Ihrer Wahl zu verwenden.

Mithilfe der API, die von diesen Bibliotheken bereitgestellt wird, können Sie Entscheider, Activity Worker oder Workflow-Starter entwickeln. Und Sie können Sichtbarkeitsoperationen über diese Bibliotheken verwenden, um Ihre eigenen Amazon SWF SWF-Überwachungs- und Berichtstools zu entwickeln.

Tools für die Entwicklung und Verwaltung von Anwendungen AWS finden Sie unter SDKs anderem im [Developer Center](#).

Detaillierte Informationen zu den Amazon SWF SWF-Vorgängen in den einzelnen SDKs finden Sie in der sprachspezifischen Referenzdokumentation für das SDK.

Betrachten Sie die AWS Flow Framework

Das AWS Flow Framework ist ein erweitertes SDK zum Schreiben verteilter, asynchroner Programme, die als Workflows auf Amazon SWF ausgeführt werden. Das Framework ist für die Programmiersprache Java verfügbar und bietet Klassen zum Schreiben komplexer verteilter Programme.

Mit dem AWS Flow Framework können Sie vorkonfigurierte Typen verwenden, um die Definition Ihres Workflows direkt den Methoden in Ihrem Programm zuzuordnen. Das AWS Flow Framework unterstützt objektorientierte Standardkonzepte, wie z. B. die ausnahmebasierte Fehlerbehandlung. Programme, die mit dem geschrieben wurden, AWS Flow Framework können vollständig in Ihrem bevorzugten Editor oder Ihrer bevorzugten IDE erstellt, ausgeführt und debuggt werden. Weitere Informationen finden Sie [AWS Flow Framework](#) auf der Website und im [Java Developer Guide](#).AWS Flow Framework

Erste Schritte mit Amazon SWF

Sie können mit der folgenden Amazon Simple Workflow Service-Workflow-Anwendung beginnen, die aus vier Aktivitäten besteht, die nacheinander ausgeführt werden. Das Tutorial behandelt auch die folgenden Themen:

- Einrichten von Standard- und Laufzeit-Workflow- und Aktivitätsoptionen
- Amazon SWF nach Entscheidungs- und Aktivitätsaufgaben abfragen.
- Weitergabe von Daten zwischen den Aktivitäten und dem Workflow mit Amazon SWF.
- Auf menschliche Aufgaben warten und Herzschläge von einer Aktivitätsaufgabe an Amazon SWF melden.
- Verwenden von Amazon SNS, um ein Thema zu erstellen, es für einen Benutzer zu abonnieren und Nachrichten an abonnierten Endpunkten zu veröffentlichen.

Sie können Amazon SWF und Amazon Simple Notification Service (Amazon SNS) zusammen verwenden, um einen „menschlichen Task“-Workflow zu emulieren, bei dem ein menschlicher Mitarbeiter eine Aktion ausführen und dann mit Amazon SWF kommunizieren muss, um die nächste Aktivität im Workflow zu starten.

Da Amazon SWF ein cloudbasierter Webservice ist, kann die Kommunikation mit Amazon SWF von überall aus erfolgen, wo eine Verbindung zum Internet verfügbar ist. In diesem Fall verwenden wir Amazon SNS, um mit dem Benutzer entweder per E-Mail, einer SMS-Textnachricht oder beidem zu kommunizieren.

In diesem Tutorial wird der für den [AWS SDK für Ruby](#) Zugriff auf Amazon SWF und Amazon SNS verwendet. Es stehen jedoch viele Entwicklungsoptionen zur Verfügung, darunter die AWS Flow Framework für Ruby, die eine einfachere Koordination und Kommunikation mit Amazon SWF ermöglicht.

Note

In diesem Tutorial wird der verwendet AWS SDK für Ruby, wir empfehlen jedoch, den [AWS Flow Framework für Java](#) zu verwenden.

Themen

- [Über den Workflow](#)
- [Voraussetzungen](#)
- [Anleitungsschritte](#)
- [Tutorial zum Abonnement-Workflow Teil 1: Verwenden von Amazon SWF mit dem AWS SDK für Ruby](#)
- [Abonnement-Workflow, Anleitung Teil 2: Implementieren des Workflows](#)
- [Abonnement-Workflow, Anleitung Teil 3: Implementieren der Aktivitäten](#)
- [Abonnement-Workflow, Anleitung Teil 4: Implementieren des Pollers für Aktivitätsaufgaben](#)
- [Abonnement-Workflow, Anleitung: Ausführen des Workflows](#)

Über den Workflow

Der Workflow, den wir entwickeln werden, besteht aus vier Hauptschritten:

1. Abrufen einer Abonnementadresse (E-Mail oder SMS) des Benutzers.
2. Erstellen Sie ein SNS-Thema und abonnieren Sie es für die bereitgestellten Endpunkte.
3. Warten Sie darauf, dass der Benutzer das Abonnement bestätigt.
4. Veröffentlichen Sie eine Glückwunschnachricht unter dem Thema, sobald der Benutzer die Bestätigung ausgeführt hat.

Diese Schritte beinhalten Aktivitäten, die vollständig automatisch ablaufen (Schritte 2 und 4), sowie Aktivitäten, bei denen der Workflow darauf warten muss, dass ein menschlicher Bediener Informationen eingibt, bevor der Workflow fortfahren kann (Schritte 1 und 3).

Jeder Schritt basiert auf den Daten, die im vorherigen Schritt erstellt wurden (es wird ein Endpunkt benötigt, bevor er einem Thema zugeordnet werden kann, und es muss ein Themenabonnement vorhanden sein, bevor Sie auf die Bestätigung warten können usw.), In diesem Tutorial erfahren Sie auch, wie Sie Aktivitätsergebnisse nach Abschluss bereitstellen und wie Sie Eingaben an eine geplante Aufgabe weiterleiten. Amazon SWF kümmert sich um die Koordination und Bereitstellung von Informationen zwischen den Aktivitäten und dem Workflow und umgekehrt.

Wir verwenden außerdem sowohl Tastatureingaben als auch Amazon SNS, um die Kommunikation zwischen Amazon SWF und dem Menschen, der Daten für den Workflow bereitstellt, abzuwickeln. In der Praxis können Sie viele verschiedene Techniken verwenden, um mit menschlichen Benutzern zu

kommunizieren. Amazon SNS bietet jedoch eine sehr einfache Möglichkeit, den Benutzer mithilfe von E-Mail- oder Textnachrichten über Ereignisse im Workflow zu informieren.

Voraussetzungen

Für diese Anleitung benötigen Sie Folgendes:

- [Amazon Web Services Services-Konto](#)
- [Einen Ruby-Interpreter](#)
- [AWS SDK für Ruby](#)

Wenn Sie diese Komponenten bereits eingerichtet haben, können Sie fortfahren. Wenn Sie das Beispiel nicht ausführen möchten, können Sie trotzdem dem Tutorial folgen. Ein Großteil des Inhalts dieses Tutorials bezieht sich auf die Verwendung von Amazon SWF und Amazon SNS, unabhängig von der gewählten Entwicklungsoption.

Anleitungsschritte

Diese Anleitung ist in die folgenden Schritte unterteilt:

1. [Tutorial zum Abonnement-Workflow Teil 1: Verwenden von Amazon SWF mit dem AWS SDK für Ruby](#)
2. [Abonnement-Workflow, Anleitung Teil 2: Implementieren des Workflows](#)
3. [Abonnement-Workflow, Anleitung Teil 3: Implementieren der Aktivitäten](#)
4. [Abonnement-Workflow, Anleitung Teil 4: Implementieren des Pollers für Aktivitätsaufgaben](#)
5. [Abonnement-Workflow, Anleitung: Ausführen des Workflows](#)

Tutorial zum Abonnement-Workflow Teil 1: Verwenden von Amazon SWF mit dem AWS SDK für Ruby

Themen

- [Schließen Sie ein AWS SDK für Ruby](#)
- [Konfiguration der Sitzung AWS](#)

- [Registrierung einer Amazon SWF-Domain](#)
- [Nächste Schritte](#)

Schließen Sie ein AWS SDK für Ruby

Erstellen Sie zunächst eine Datei namens `utils.rb`. Der Code in dieser Datei ruft die Amazon SWF-Domain ab, die sowohl vom Workflow- als auch vom Aktivitätscode verwendet wird, oder erstellt sie, falls erforderlich, und bietet einen Ort, an dem Code abgelegt werden kann, der allen unseren Klassen gemeinsam ist.

Zunächst müssen wir die `aws-sdk-v1` Bibliothek in unseren Code aufnehmen, damit wir die vom SDK for Ruby bereitgestellten Funktionen nutzen können.

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

Dadurch erhalten wir Zugriff auf den AWS Namespace, der die Möglichkeit bietet, globale Sitzungsbezogene Werte wie Ihre AWS Anmeldeinformationen und Region festzulegen, und der auch den Zugriff auf den Dienst ermöglicht. AWS APIs

Konfiguration der Sitzung AWS

Wir konfigurieren die AWS Sitzung, indem wir unsere AWS Anmeldeinformationen (die für den Zugriff auf AWS Dienste erforderlich sind) und die zu AWS verwendende Region festlegen.

Es gibt eine Reihe von Möglichkeiten, [AWS Anmeldeinformationen im AWS SDK for Ruby festzulegen](#): indem Sie sie in Umgebungsvariablen (`AWS_ACCESS_KEY_ID` und `AWS_SECRET_ACCESS_KEY`) setzen oder indem Sie sie mit `setenv` setzen. [AWS.config](#) Wir verwenden die letzte Methode und laden sie aus einer YAML-Konfigurationsdatei mit dem Namen `aws-config.txt` hoch, die wie im Folgenden dargestellt aussieht.

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

Erstellen Sie jetzt diese Datei und ersetzen Sie die Zeichenketten, die mit `REPLACE_WITH_` beginnen, durch Ihre Zugriffsschlüssel-ID und Ihren geheimen Zugriffsschlüssel. AWS Informationen zu Ihren AWS Zugriffsschlüsseln finden Sie unter [Wie erhalte ich Sicherheitsanmeldedaten?](#) in der allgemeinen Referenz zu Amazon Web Services.

Wir müssen auch die AWS Region festlegen, die verwendet werden soll. Da wir den [Short Message Service \(SMS\)](#) verwenden werden, um Textnachrichten mit Amazon SNS an das Telefon des Benutzers zu senden, müssen wir sicherstellen, dass wir die von Amazon SNS unterstützte Region verwenden. Weitere Informationen zu [den unterstützten Regionen und Ländern](#) finden Sie im Amazon Simple Notification Service Developer Guide.

Note

Wenn Sie keinen Zugriff auf us-east-1 haben oder das Beispiel nicht mit aktivierter SMS-Messagingfunktion ausführen möchten, können Sie eine andere Region verwenden. Sie können die SMS-Funktionalität aus dem Beispiel entfernen und E-Mail als einzigen Endpunkt verwenden, um das Amazon SNS SNS-Thema zu abonnieren.

Weitere Informationen zum Senden von SMS-Nachrichten finden Sie unter [Senden und Empfangen von SMS-Benachrichtigungen mit Amazon SNS](#) im Amazon Simple Notification Service Developer Guide.

Wir fügen nun Code hinzu, um die Konfigurationsdatei `utils.rb` zu laden, die Anmeldeinformationen des Benutzers abzurufen und dann sowohl die Anmeldeinformationen als auch die Region für anzugeben. [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
```

Registrierung einer Amazon SWF-Domain

Um Amazon SWF verwenden zu können, müssen Sie eine Domain einrichten: eine benannte Entität, die Ihre Workflows und Aktivitäten speichert. Sie können viele Amazon SWF-Domänen registrieren, aber sie müssen alle eindeutige Namen innerhalb Ihres AWS Kontos haben, und Workflows können nicht domänenübergreifend interagieren: Alle Workflows und Aktivitäten für Ihre Anwendung müssen sich in derselben Domain befinden, um miteinander interagieren zu können.

Da wir in unserer gesamten Anwendung dieselbe Domain verwenden, erstellen wir eine Funktion in `utils.rb` called `init_domain`, die die Amazon SWF-Domain mit dem Namen `SWFSampleDomain` abrufen.

Sobald Sie eine Domäne registriert haben, können Sie sie für viele Workflow-Ausführungen wiederverwenden. Allerdings tritt ein Fehler auf, wenn Sie versuchen, eine bereits vorhandene Domäne zu registrieren. Daher prüft unser Code zunächst, ob die Domäne vorhanden ist, und verwendet diese gegebenenfalls. Wenn die Domäne nicht gefunden wird, erstellen wir sie.

Um mit Amazon SWF-Domänen im SDK for Ruby zu arbeiten, verwenden Sie [AWS::Simpleworkflow.Domains](#). Dadurch wird ein [DomainCollection](#) zurückgegeben, das sowohl zum Aufzählen als auch zum Registrieren von Domänen verwendet werden kann:

- [Um zu überprüfen, ob eine Domain bereits registriert ist, können Sie sich die von `.domains.registered` bereitgestellte Liste ansehen. `AWS::Simpleworkflow`](#)
- [Verwenden Sie `.domains.register`, um eine neue Domain zu registrieren. `AWS::Simpleworkflow`](#)

Hier finden Sie den Code für `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
```

Nächste Schritte

Im nächsten Schritt erstellen Sie den Workflow und Ausgangscode in [Abonnement-Workflow, Anleitung Teil 2: Implementieren des Workflows](#).

Abonnement-Workflow, Anleitung Teil 2: Implementieren des Workflows

Bisher war unser Code sehr allgemein. In diesem Teil beginnen wir mit dem Definieren der Vorgehensweise des Workflows und der zu implementierenden Aktivitäten.

Themen

- [Entwerfen des Workflows](#)
- [Einrichten des Workflow-Codes](#)
- [Registrieren des Workflows](#)
- [Abrufen von Entscheidungen](#)
- [Starten der Workflow-Ausführung](#)
- [Nächste Schritte](#)

Entwerfen des Workflows

Wie Sie sich sicherlich erinnern, bestand das ursprüngliche Konzept des Workflows aus den folgenden Schritten:

1. Abrufen einer Abonnementadresse (E-Mail oder SMS) des Benutzers.
2. Erstellen Sie ein SNS-Thema und abonnieren Sie es für die bereitgestellten Endpunkte.
3. Warten Sie darauf, dass der Benutzer das Abonnement bestätigt.
4. Veröffentlichen Sie eine Glückwunschnachricht unter dem Thema, sobald der Benutzer die Bestätigung ausgeführt hat.

Jeder Schritt in unserem Workflow stellt eine Aktivität dar, die ausgeführt werden muss. Der Workflow dient zum Planen der einzelnen Aktivitäten zu den entsprechenden Zeitpunkten und dem Koordinieren des Datentransfers zwischen den Aktivitäten.

Für diesen Workflow generieren wir für die einzelnen Schritte jeweils eine separate Aktivität und benennen sie entsprechend:

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

Diese Aktivitäten werden der Reihenfolge nach ausgeführt. Dabei nutzt der jeweils nachfolgende Schritt die Daten des vorherigen Schritts.

Wir könnten unsere Anwendung so gestalten, dass der gesamte Code in einer Quelldatei existiert, aber das widerspricht der Art und Weise, wie Amazon SWF entworfen wurde. Amazon SWF wurde für Workflows entworfen, die den gesamten Internetbereich umfassen. Deshalb sollte die Anwendung in mindestens zwei separate ausführbare Programme unterteilt werden:

- `swf_sns_workflow.rb` – Enthält den Workflow und den Workflow-Starter.
- `swf_sns_activities.rb` – Enthält die Aktivitäten und den Aktivitäten-Starter.

Der Workflow und das Implementieren der Aktivitäten können in unterschiedlichen Fenstern, auf unterschiedlichen Computern und sogar in unterschiedlichen Teilen der Welt ausgeführt werden. Da Amazon SWF die Details Ihrer Workflows und Aktivitäten verfolgt, kann Ihr Workflow die Planung und Datenübertragung Ihrer Aktivitäten koordinieren, unabhängig davon, wo sie ausgeführt werden.

Einrichten des Workflow-Codes

Als Erstes erstellen wir eine Datei namens `swf_sns_workflow.rb`. Deklarieren Sie in dieser Datei eine Klasse namens `SampleWorkflow`. Hier sehen Sie die Klassendeklaration und den dazugehörigen Konstruktor, die `initialize`-Methode.

```
require_relative 'utils.rb'

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

  attr_accessor :name

  def initialize(workflowId)

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

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

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

    register_workflow
  end
end
```

Wie Sie sehen, behalten wir die folgenden Daten der Klassen-Instance bei:

- `domain` – Den von `init_domain` in `utils.rb` abgerufenen Domänennamen.
- `workflowId` – Die an `initialize` übergebene Aufgabenliste.
- `activity_list` – Die Aktivitätenliste mit den Namen und Versionen der Aktivitäten, die wir ausführen werden.

Der Domainname, der Aktivitätsname und die Aktivitätsversion reichen aus, damit Amazon SWF einen Aktivitätstyp eindeutig identifizieren kann. Das sind also alle Daten, die wir über unsere Aktivitäten speichern müssen, um sie zu planen.

Die Aufgabenliste wird vom Entscheider-Code des Workflows zum Abfragen von Entscheidungsaufgaben und Planen der Aktivitäten verwendet.

Am Ende dieser Funktion rufen wir eine Methode auf, die wir noch nicht definiert haben: `register_workflow`. Wir definieren diese Methode später.

Registrieren des Workflows

Wir müssen einen Workflow-Typ erst registrieren, damit wir ihn verwenden können. Genau wie ein Aktivitätstyp wird auch ein Workflow-Typ anhand seiner Domäne, seines Namens und seiner Version identifiziert. Und genau wie Domänen- und Aktivitätstypen können Sie einen vorhandenen Workflow-Typ nicht erneut registrieren. Wenn Sie den Workflow-Typ ändern müssen, müssen Sie ihn mit einer neuen Version zur Verfügung stellen, wodurch ein neuer Typ generiert wird.

Hier ist der Code für `register_workflow`. Dieser wird entweder zum Abrufen des vorhandenen Workflow-Typs genutzt, den wir während einer vorherigen Ausführung registriert haben, oder zum Registrieren des Workflows, sofern dies noch nicht geschehen ist.

```
# 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 }
  end
end
```

```
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
```

Zunächst prüfen wir, ob der Workflow-Name und die Version bereits registriert sind. Dazu iterieren wir durch die [workflow_types](#)-Sammlung der Domäne. Bei einer Übereinstimmung nutzen wir den bereits registrierten Workflow-Typ.

Wenn wir keine Übereinstimmung finden, wird ein neuer Workflow-Typ registriert (durch Aufrufen von [register](#) in derselben `workflow_types` Sammlung, in der wir nach dem Workflow gesucht haben) mit dem Namen `swf-sns-workflow`, Version `'1'` und den folgenden Optionen.

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

Die während der Registrierung übergebenen Optionen werden zum Festlegen des Standardverhaltens unseres Workflow-Typs genutzt, sodass wir diese Werte nicht bei jeder neuen Workflow-Ausführung einstellen müssen.

In diesem Beispiel legen wir nur einige Timeout-Werte fest: Die maximale Dauer zwischen dem Start und der Beendigung einer Aufgabe (1 Stunde) und die Zeit, die es maximal dauert, bis die Workflow-Ausführung abgeschlossen ist (24 Stunden). Wird eine dieser Zeitgrenzen überschritten, kommt es zu einem Aufgaben- oder Workflow-Timeout.

Weitere Informationen zur Timeout-Werten finden Sie unter [Amazon SWF-Timeout-Typen](#).

Abrufen von Entscheidungen

Das Herzstück einer jeden Workflow-Ausführung stellt der Entscheider dar. Der Entscheider ist für die Verwaltung der Ausführung des Workflows selbst verantwortlich. Er empfängt Entscheidungsaufgaben und reagiert auf diese, indem er entweder neue Aktivitäten plant, Aktivitäten abbricht oder neu startet oder den Status der Workflow-Ausführung auf abgeschlossen, abgebrochen oder fehlgeschlagen setzt.

Der Entscheider nutzt den Namen der Aufgabenliste der Workflow-Ausführung, um Entscheidungsaufgaben abzurufen, auf die er reagieren muss. Zum Abrufen von Entscheidungsaufgaben rufen Sie die Abfrage https://docs.aws.amazon.com/AWSRubySDK/latest/AWS/SimpleWorkflow/DecisionTaskCollection.html#poll-instance_method in der `decision_tasks`-Sammlung der Domäne auf, um nach verfügbaren Entscheidungsaufgaben zu suchen. Sie können neue Ereignisse in der Entscheidungsaufgabe finden, indem Sie eine Iteration über der `new_events`-Sammlung starten.

Bei den zurückgegebenen Ereignissen handelt es sich um `AWS::SimpleWorkflow::HistoryEvent` Objekte, und Sie können den Typ des Ereignisses ermitteln, indem Sie das Element `event_type` des zurückgegebenen Ereignisses verwenden. Eine Liste und Beschreibung der Verlaufsereignistypen finden Sie `HistoryEvent` in der Amazon Simple Workflow Service API-Referenz.

Nachfolgend sehen Sie den Anfang der Logik des Entscheidungsaufgaben-Pollers. Eine neue Methode in unserer Workflow-Klasse namens `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
```

Wir verzweigen nun die Ausführung unseres Entscheiders basierend auf dem empfangenen `event_type`. Der erste, den wir wahrscheinlich erhalten werden, ist `WorkflowExecutionStarted`. Wenn dieses Ereignis empfangen wird, bedeutet dies, dass Amazon SWF Ihrem Entscheider signalisiert, dass es mit der Workflow-Ausführung beginnen soll. Wir beginnen mit dem Planen der ersten Aktivität, indem wir `schedule_activity_task` für die Aufgabe aufrufen, die wir während des Abrufens empfangen haben.

Wir übergeben sie als erste in unserer Aktivitätsliste deklarierten Aktivität, die, da wir die Listenreihenfolge umgedreht haben, sodass sie als Stack verwendet werden kann, die `last` Position in der Liste einnimmt. Die von uns definierten „Aktivitäten“ sind nur Zuordnungen, die aus einem Namen und einer Versionsnummer bestehen. Dies ist jedoch alles, was Amazon SWF benötigt, um die Aktivität für die Planung zu identifizieren, vorausgesetzt, die Aktivität wurde bereits registriert.

```
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" } )
```

Wenn wir eine Aktivität planen, sendet Amazon SWF eine Aktivitätsaufgabe an die Aktivitätsaufgabenliste, die wir bei der Planung übergeben, und signalisiert so, dass die Aufgabe beginnen soll. Wir beschäftigen uns in [Abonnement-Workflow, Anleitung Teil 3: Implementieren der Aktivitäten](#) mit Aktivitätsaufgaben, an dieser Stelle sollte aber erwähnt werden, dass wir hier die Aufgabe nicht ausführen. Wir teilen Amazon SWF nur mit, dass es geplant werden sollte.

Die nächste Aktivität, mit der wir uns befassen müssen, ist das ActivityTaskCompletedEreignis, das eintritt, wenn Amazon SWF die Antwort „Aktivität abgeschlossen“ von einer Aktivitätsaufgabe erhalten hat.

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

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

Da wir unsere Aufgaben linear ausführen und nur eine Aktivität gleichzeitig ausgeführt wird, nutzen wir diese Gelegenheit, um die abgeschlossene Aufgabe aus dem activity_list Stapel zu

nehmen. Führt dies zu einer leeren Liste, wissen wir, dass unser Workflow abgeschlossen ist. In diesem Fall signalisieren wir Amazon SWF, dass unser Workflow abgeschlossen ist, indem wir [complete_workflow_execution](#) für die Aufgabe aufrufen.

Enthält die Liste noch Einträge, planen wir die nächste Aktivität auf der Liste (auch hier an letzter Stelle). Diesmal werden wir jedoch prüfen, ob die vorherige Aktivität nach Abschluss Ergebnisdaten an Amazon SWF zurückgegeben hat, die dem Workflow in den Attributen des Ereignisses im optionalen `result` Schlüssel zur Verfügung gestellt werden. Wurde durch die Aktivität ein Ergebnis generiert, übergeben wir dieses als `input`-Option zusammen mit der Aktivitätsaufgabenliste an die nächste geplante Aktivität.

Durch Abruf der `result`-Werte abgeschlossener Aktivitäten und Festlegen der `input`-Werte geplanter Aktivitäten, können wir Daten von einer Aktivität an die nächste übergeben oder Daten aus einer Aktivität nutzen, um das Verhalten unseres Entscheiders basierend auf den Ergebnissen einer Aktivität zu ändern.

Im Rahmen dieser Anleitung stellen diese beiden Ereignistypen die wichtigsten Typen in Bezug auf das Definieren des Workflow-Verhaltens dar. Eine Aktivität kann jedoch auch andere Ereignisse auslösen als `ActivityTaskCompleted`. Wir schließen unseren Entscheidungscode ab, indem wir Demonstrationshandlercode für die `ActivityTaskFailed` Ereignisse `ActivityTaskTimedOut` und für das `WorkflowExecutionCompleted` Ereignis bereitstellen. Dieser Code wird generiert, wenn Amazon SWF den `complete_workflow_execution` Aufruf verarbeitet, den wir tätigen, wenn uns die auszuführenden Aktivitäten ausgehen.

```
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
```

Starten der Workflow-Ausführung

Ehe Entscheidungsaufgaben für den Workflow generiert werden, die abgerufen werden können, müssen wir die Workflow-Ausführung starten.

Um die Workflow-Ausführung zu starten, rufen Sie [start_execution](#) für Ihren registrierten Workflow-Typ () auf. [AWS::SimpleWorkflow::WorkflowType](#) Wir definieren einen kleinen Wrapper darum herum, um das `workflow_type`-Instance-Mitglied zu nutzen, das wir im Klassen-Konstruktor abgerufen haben.

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

Sobald der Workflow ausgeführt wird, erscheinen Entscheidungsereignisse in der Aufgabenliste des Workflows, die als Workflow-Ausführungsoption in [start_execution](#) übergeben wird.

Im Gegensatz zu Optionen, die bereitgestellt werden, wenn der Workflow-Typ registriert wird, gelten Optionen, die an `start_execution` übergeben werden, nicht als Teil des Workflow-Typs. Sie können dies pro Workflow-Ausführung ändern, ohne die Workflow-Version ändern zu müssen.

Da wir möchten, dass der Workflow mit der Ausführung der Datei beginnt, fügen Sie Code hinzu, der die Klasse instanziiert und dann die `start_execution` Methode aufruft, die wir gerade definiert haben.

```
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
```

Um zu vermeiden, dass Konflikte mit Aufgabenlistenamen auftreten, verwenden wir `SecureRandom.uuid`, um eine zufällig UUID zu generieren, die wir als Aufgabenlistenamen verwenden können. So wird sichergestellt, dass für jede Workflow-Ausführung ein anderer Aufgabenlistenamenname genutzt wird.

Note

Aufgabenlisten zeichnen Ereignisse zur Workflow-Ausführung auf. Wenn Sie also dieselbe Aufgabenliste für mehrere Ausführungen desselben Workflow-Typs nutzen, erhalten Sie möglicherweise Ereignisse, die während der vorherigen Ausführung generiert wurden. Das gilt insbesondere dann, wenn Sie diese dicht nacheinander ausführen. Dies ist oftmals der Fall, wenn neuer Code ausprobiert oder Tests durchgeführt werden.

Zur Vermeidung des Problems mit Artefakten aus vorherigen Ausführungen kann für jede Ausführung eine neue Aufgabenliste verwendet werden, die angegeben wird, wenn mit der Workflow-Ausführung begonnen wird.

Hier ist auch etwas Code vorhanden, der Anleitungen für den ausführenden Benutzer (möglicherweise Sie) sowie die Aktivitätsversion der Aufgabenliste bereitstellt. Der Entscheider nutzt den Aufgabenlistennamen zum Planen der Aktivitäten für den Workflow. Durch die Aktivitätenimplementierung wird nach Aktivitätsereignissen in diesem Aufgabenlistennamen gesucht, damit die Anwendung weiß, wann die geplante Aktivität beginnen soll und um Aktualisierungen zur Aktivitätsausführung bereitzustellen.

Der Code wartet zudem darauf, dass der Benutzer mit der Ausführung des Aktivitäten-Starters beginnt, bevor er die Workflow-Ausführung startet. Somit ist der Aktivitäten-Starter bereit, sobald Aktivitätsaufgaben auf der bereitgestellten Aufgabenliste erscheinen.

Nächste Schritte

Die Implementierung des Workflows ist abgeschlossen. Als Nächstes definieren Sie die Aktivitäten sowie einen Aktivitäten-Starter in [Abonnement-Workflow, Anleitung Teil 3: Implementieren der Aktivitäten](#).

Abonnement-Workflow, Anleitung Teil 3: Implementieren der Aktivitäten

Wir implementieren nun die einzelnen Aktivitäten in unserem Workflow. Dazu beginnen wir mit einer Basisklasse, die einige gängige Funktionen für den Aktivitätscode bereitstellt.

Themen

- [Definieren eines Basis-Aktivitätstyps](#)
- [Definierend GetContactActivity](#)
- [Definierend SubscribeTopicActivity](#)
- [Definierend WaitForConfirmationActivity](#)
- [Definierend SendResultActivity](#)
- [Nächste Schritte](#)

Definieren eines Basis-Aktivitätstyps

Beim Entwerfen des Workflows haben wir die folgenden Aktivitäten identifiziert:

- `get_contact_activity`

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

Jede dieser Aktivitäten werden wir nun implementieren. Da unsere Aktivitäten einige Funktionen gemeinsam haben werden, sollten wir ein wenig Grundlagenarbeit leisten und gemeinsamen Code erstellen, den sie gemeinsam nutzen können. Wir rufen es auf `BasicActivity` und definieren es in einer neuen Datei namens `basic_activity.rb`.

Wie bei den anderen Quelldateien fügen wir `utils.rb` hinzu, um auf die Funktion `init_domain` zuzugreifen und die Beispieldomäne einzurichten.

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

Als Nächstes deklarieren wir die Basis-Aktivitätsklasse und einige gemeinsame Daten für die Aktivitäten. Wir speichern die [AWS::SimpleWorkflow::ActivityType](#) Instanz, den Namen und die Ergebnisse der Aktivität in den Attributen der Klasse.

```
class BasicActivity

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

Diese Attribute greifen auf Instanzdaten zu, die in der `initialize` Methode der Klasse definiert sind, die einen Aktivitätsnamen und eine optionale Version und Zuordnung von Optionen verwendet, die bei der Registrierung der Aktivität bei Amazon SWF verwendet werden.

```
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
```

Wie bei der Registrierung des Workflow-Typs können wir einen bereits registrierten Aktivitätstyp abrufen, indem wir uns die Sammlung [activity_types](#) der Domäne anschauen. Wird die Aktivität nicht gefunden, wird sie registriert.

Ebenso wie bei den Workflow-Typen können Sie Standardoptionen festlegen, die mit Ihrem Aktivitätstyp bei dessen Registrierung gespeichert werden.

Im letzten Schritt legen wir eine konsistente Ausführungsmethode für unsere Basisaktivität fest. Wir definieren eine `do_activity`-Methode, die eine Aktivitätsaufgabe verwendet. Wie oben dargestellt können wir die übergebene Aktivitätsaufgabe verwenden, um Daten über ihr Instance-Attribut `input` zu empfangen.

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

Damit ist die Klasse abgeschlossen. `BasicActivity` Wir können sie nun verwenden, um unsere Aktivitäten auf einfache und konsistente Weise zu definieren.

Definierend GetContactActivity

Die erste Aktivität, die während einer Workflow-Ausführung ausgeführt wird `get_contact_activity`, ist, die Abonnementinformationen für Amazon SNS-Themen des Benutzers abzurufen.

Erstellen Sie eine neue Datei mit dem Namen `get_contact_activity.rbyaml`, und `require both`, die wir verwenden, um eine Zeichenfolge für die Übergabe an Amazon SWF vorzubereiten `basic_activity.rb`, und die wir als Grundlage für diese `GetContactActivity` Klasse verwenden werden.

```
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
```

Da wir den Registrierungscode für die Aktivität eingegeben haben `BasicActivity`, `GetContactActivity` ist die `initialize` Methode dafür ziemlich einfach. Wir rufen einfach den Basisklassenkonstruktor mit dem Aktivitätsnamen `get_contact_activity` auf. Dies ist alles, was für die Registrierung unserer Aktivität erforderlich ist.

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

Wir definieren jetzt die `do_activity` Methode, die zur Eingabe der and/or E-Mail-Telefonnummer des Benutzers auffordert.

```
def do_activity(task)
  puts ""
  puts "Please enter either an email address or SMS message (mobile phone) number
to"
  puts "receive SNS notifications. You can also enter both to use both address
types."
  puts ""
  puts "If you enter a phone number, it must be able to receive SMS messages, and
must"
```

```
puts "be 11 digits (such as 12065550101 to represent the number
1-206-555-0101)."  
input_confirmed = false
while !input_confirmed
  puts ""
  print "Email: "
  email = $stdin.gets.strip  
  
  print "Phone: "
  phone = $stdin.gets.strip  
  
  puts ""
  if (email == '') && (phone == '')
    print "You provided no subscription information. Quit? (y/n)"
    confirmation = $stdin.gets.strip.downcase
    if confirmation == 'y'
      return false
    end
  else
    puts "You entered:"
    puts "  email: #{email}"
    puts "  phone: #{phone}"
    print "\nIs this correct? (y/n): "
    confirmation = $stdin.gets.strip.downcase
    if confirmation == 'y'
      input_confirmed = true
    end
  end
end  
  
# make sure that @results is a single string. YAML makes this easy.
@results = { :email => email, :sms => phone }.to_yaml
return true
end
end
```

Am Ende von `do_activity` platzieren wir die vom Benutzer erhaltene E-Mail-Adresse und Telefonnummer in eine Map und konvertieren diese mithilfe von `to_yaml` in eine YAML-Zeichenfolge. Dafür gibt es einen wichtigen Grund: Alle Ergebnisse, die Sie nach Abschluss einer Aktivität an Amazon SWF weitergeben, dürfen nur Zeichenkettendaten sein. Die Fähigkeit von Ruby,

Objekte einfach in YAML-Zeichenfolgen und dann wieder zurück in Objekte zu konvertieren, ist für diesen Zweck sehr gut geeignet.

Damit ist die Implementierung von `get_contact_activity` abgeschlossen. Diese Daten werden bei der `subscribe_topic_activity`-Implementierung als Nächstes verwendet.

Definierend SubscribeTopicActivity

Wir werden uns nun mit Amazon SNS befassen und eine Aktivität erstellen, die die von generierten Informationen verwendet, `get_contact_activity` um den Benutzer für ein Amazon SNS SNS-Thema zu abonnieren.

Erstellen Sie eine neue Datei mit dem Namen `subscribe_topic_activity.rb`. Fügen Sie die gleichen Anforderungen wie bei `get_contact_activity` hinzu, deklarieren Sie Ihre Klasse und stellen Sie ihre `initialize`-Methode bereit.

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

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

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

Nachdem wir nun den Code für die Einrichtung und Registrierung der Aktivität eingerichtet und registriert haben, werden wir etwas Code hinzufügen, um ein Amazon SNS SNS-Thema zu erstellen. Dazu verwenden wir die Methode [AWS::SNS::Client#create_topic](#) des Objekts.

Fügen Sie die `create_topic` Methode zu Ihrer Klasse hinzu, die ein übergebenes Amazon SNS SNS-Client-Objekt verwendet.

```
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!
  end
end
```

```

    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

```

Sobald wir den Amazon-Ressourcennamen (ARN) des Themas haben, können wir ihn mit der Methode [set_topic_attributes](#) des Amazon SNS-Clients verwenden, um das Thema festzulegen `DisplayName`, was für den Versand von SMS-Nachrichten mit Amazon SNS erforderlich ist.

Schließlich definieren wir die `do_activity`-Methode. Zunächst sammeln wir alle Daten, die bei Terminierung der Aktivität über die `input`-Option übergeben wurden. Wie zuvor erwähnt muss diese als Zeichenfolge, erstellt mit `to_yaml`, übergeben werden. Beim Abrufen verwenden wir `YAML.load`, um die Daten in Ruby-Objekte umzuwandeln.

Hier ist der Anfang von `do_activity`, in den wir die Eingabedaten abrufen.

```

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))
```

Wenn wir keine Eingabedaten empfangen haben, gibt es nichts zu verarbeiten, also schlägt die Aktivität fehl.

Vorausgesetzt, dass alles in Ordnung ist, füllen wir unsere `do_activity` Methode weiter aus, rufen einen Amazon SNS SNS-Client mit dem ab und übergeben es an unsere `create_topic` Methode AWS SDK für Ruby, um das Amazon SNS SNS-Thema zu erstellen.

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

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

An dieser Stelle sei auf Folgendes hingewiesen:

- Wir verwenden [AWS.config.with](#), um die Region für unseren Amazon SNS SNS-Client festzulegen. Da wir SMS-Nachrichten versenden wollen, verwenden wir die SMS-fähige Region, die wir in `utils.rb` deklariert haben.
- Wir speichern den ARN des Themas in unserer Map `activity_data`. Dies ist der Teil der Daten, die an die nächste Aktivität in unserem Workflow übergeben werden.

Schließlich abonniert diese Aktivität den Benutzer mithilfe der übergebenen Endpunkte (E-Mail und SMS) für das Amazon SNS-Thema. Wir fordern nicht vom Benutzer, dass er beide Endpunkte angibt, wir benötigen aber mindestens einen.

```
# 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
```

```
end
```

[AWS::SNS::Client.subscribe](#) verwendet das Thema ARN, das Protokoll (das wir geschickt als Map-Schlüssel für den entsprechenden Endpunkt getarnt haben `activity_data`).

Schließlich verpacken wir die Informationen für die nächste Aktivität erneut im YAML-Format, sodass wir sie an Amazon SWF zurücksenden können.

```
# 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
```

Damit ist die Implementierung von `subscribe_topic_activity` abgeschlossen. Als Nächstes definieren wir `wait_for_confirmation_activity`.

Definierend WaitForConfirmationActivity

Sobald ein Benutzer ein Amazon SNS SNS-Thema abonniert hat, muss er die Abonnementanfrage noch bestätigen. In diesem Fall warten wir, bis der Benutzer entweder per E-Mail oder SMS bestätigt.

Die Aktivität, mit der der Benutzer das Abonnement bestätigt, heißt `wait_for_confirmation_activity`. Sie wird hier definiert. Erstellen Sie zunächst eine neue Datei namens `wait_for_confirmation_activity.rb`. Richten Sie sie so wie die vorherigen Aktivitäten ein.

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

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

```
class WaitForConfirmationActivity < BasicActivity

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

Als Nächstes beginnen wir mit der Definition der `do_activity`-Methode und rufen alle Eingabedaten in eine lokale Variable namens `subscription_data` ab.

```
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)
```

Jetzt, wo wir das Thema ARN haben, können wir das Thema abrufen, indem wir eine neue Instanz von `create` erstellen [AWS::SNS::Topic](#) und ihr den ARN übergeben.

```
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
```

Als Nächstes überprüfen wir das Thema, um zu schauen, ob der Benutzer das Abonnement über einen der Endpunkte bestätigt hat. Wir verlangen nur, dass ein Endpunkt bestätigt wurde, um die Aktivität als erfolgreich zu betrachten.

Ein Amazon SNS SNS-Thema verwaltet eine Liste der [Abonnements](#) für dieses Thema, und wir können überprüfen, ob der Benutzer ein bestimmtes Abonnement bestätigt hat oder nicht, indem wir überprüfen, ob der ARN des Abonnements auf etwas anderes als `PendingConfirmation` gesetzt ist.

```
# 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

```

Wenn wir für das Abonnement einen ARN erhalten, speichern wir diesen in den Ergebnisdaten der Aktivität, konvertieren ihn in YAML und geben von `do_activity` „true“ zurück. Dies weist darauf hin, dass die Aktivität erfolgreich abgeschlossen wurde.

Da das Warten auf die Bestätigung eines Abonnements eine Weile dauern kann, rufen wir gelegentlich die Aktivitätsaufgabe `record_heartbeat` auf. Dies signalisiert Amazon SWF, dass die Aktivität noch verarbeitet wird, und kann auch verwendet werden, um Updates über den Fortschritt der Aktivität bereitzustellen (wenn Sie etwas tun, z. B. Dateien verarbeiten, für das Sie den Fortschritt melden können).

```

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

```

Damit endet unsere `while`-Schleife. Wenn wir irgendwie aus der `While`-Schleife geraten, melden wir einen Fehler und beenden die `do_activity`-Methode.

```

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
```

Damit ist die Implementierung von `wait_for_confirmation_activity` abgeschlossen. Wir müssen nur noch eine Aktivität definieren: `send_result_activity`.

Definierend `SendResultActivity`

Wenn der Workflow so weit fortgeschritten ist, haben wir den Benutzer erfolgreich für ein Amazon SNS SNS-Thema angemeldet und der Benutzer hat das Abonnement bestätigt.

Unsere letzte Aktivität, `send_result_activity`, sendet dem Benutzer eine Bestätigung, dass er das Thema erfolgreich abonniert hat. Dazu verwendet sie dieses Thema sowie den Endpunkt, über das der Benutzer das Abonnement bestätigt hat.

Erstellen Sie eine neue Datei namens `send_result_activity.rb`. Richten Sie sie so wie alle anderen bisherigen Aktivitäten ein.

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

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

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

Unsere `do_activity` Methode beginnt ebenfalls auf ähnliche Weise, indem wir die Eingabedaten aus dem Workflow abrufen, sie aus YAML konvertieren und dann das Thema ARN verwenden, um eine [AWS::SNS::Topic](#) Instanz zu erstellen.

```
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
```

Wenn wir das Thema haben, [veröffentlichen](#) wir eine Nachricht dafür (und stellen sie auf dem Bildschirm dar).

```
@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
```

Wenn Sie unter einem Amazon SNS SNS-Thema veröffentlichen, wird die von Ihnen bereitgestellte Nachricht an alle abonnierten und bestätigten Endpunkte gesendet, die für dieses Thema existieren. Wenn der Benutzer also mit einer E-Mail-Adresse und einer Rufnummer für SMS bestätigt hat, erhält er zwei Bestätigungsmeldungen (eine an jedem Endpunkt).

Nächste Schritte

Damit ist die Implementierung von `send_result_activity` abgeschlossen. Nun verbinden Sie alle diese Aktivitäten in einer Aktivitätsanwendung, die die Aktivitätsaufgaben verarbeitet und als Reaktion darauf Aktivitäten starten kann (siehe [Abonnement-Workflow, Anleitung Teil 4: Implementieren des Pollers für Aktivitätsaufgaben](#)).

Abonnement-Workflow, Anleitung Teil 4: Implementieren des Pollers für Aktivitätsaufgaben

In Amazon SWF werden Aktivitätsaufgaben für eine laufende Workflow-Ausführung in der Aktivitätsaufgabenliste angezeigt, die angezeigt wird, wenn Sie eine Aktivität im Workflow planen.

Wir implementieren einen einfachen Aktivitätsabfrage, um diese Aufgaben für unseren Workflow zu erledigen, und verwenden ihn, um unsere Aktivitäten zu starten, wenn Amazon SWF eine Aufgabe auf die Aktivitätsaufgabenliste setzt, um die Aktivität zu starten.

Erstellen Sie zu Beginn eine neue Datei mit dem Namen `swf_sns_activities.rb`. Anhand dieser Datei können Sie folgende Aktivitäten ausführen:

- Instanzieren der von uns erstellten Aktivitätenklassen.
- Registrieren Sie jede Aktivität bei Amazon SWF.
- Abfragen zu Aktivitäten und Aufrufen von `do_activity` für die einzelnen Aktivitäten, wenn deren Name auf der Aktivitätsaufgabenliste erscheint.

Fügen Sie unter `swf_sns_activities.rb` die folgenden Anweisungen hinzu, um die einzelnen Aktivitätsklassen anzufordern, die von uns festgelegt wurden.

```
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'
```

Jetzt erstellen wir die Klasse und stellen den Initialisierungscode bereit.

```
class ActivitiesPoller  
  
  def initialize(domain, workflowId)  
    @domain = domain  
    @workflowId = workflowId  
    @activities = {}  
  
    # These are the activities we'll run  
    activity_list = [  
      GetContactActivity,  
      SubscribeTopicActivity,  
    ]  
  end  
  
end
```

```

    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

```

Zusätzlich zum Speichern der vergangenen Aktivitäten in der Domäne und Aufgabenliste, instanziiert dieser Code alle der von uns erstellten Aktivitätsklassen. Da jede Klasse ihre zugehörige Aktivität registriert (siehe, `basic_activity.rb` falls Sie diesen Code überprüfen müssen), reicht dies aus, um Amazon SWF über alle Aktivitäten zu informieren, die wir ausführen werden.

Jede instanziierte Aktivität wird anhand des Aktivitätsnamens (beispielsweise `get_contact_activity`) als Schlüssel auf einer Map gespeichert, damit sie problemlos im Aktivitäten-Poller-Code, den wir als Nächstes definieren werden, nachgeschlagen werden kann.

Erstellen Sie eine neue Methode mit dem Namen `poll_for_activities` und rufen Sie die Abfrage https://docs.aws.amazon.com/AWSRubySDK/latest/AWS/SimpleWorkflow/ActivityTaskCollection.html#poll-instance_method für die auf der Domäne befindlichen [activity_tasks](#) auf, um die Aktivitätsaufgaben zu erhalten.

```

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

```

Der Aktivitätsname kann vom [activity_type](#)-Element der Aufgabe abgerufen werden. Als Nächstes verwenden wir den Aktivitätsnamen, der dieser Aufgabe zugeordnet ist, um die Klasse nachzuschlagen, auf der `do_activity` ausgeführt wird, und um die Aufgabe weiterzuleiten (dabei sind auch die Eingabedaten eingeschlossen, die an diese Aktivität weitergeleitet werden sollen).

```

# 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
end
else
  puts "couldn't find key in @activities list: #{activity_name}"
  puts "contents: #{@activities.keys}"
end
end
end
end
```

Der Code wartet darauf, dass `do_activity` abgeschlossen wird und markiert die Aufgabe entweder als [abgeschlossen!](#) oder [fehlschlagen!](#). Dies ist hängt vom Rückgabecode ab.

Note

Dieser Code wird aus dem Poller beendet, sobald die letzte Aktivität gestartet wurde, weil er seine Mission erfüllt und alle Aktivitäten gestartet hat. Wenn Ihre Aktivitäten in Ihrem eigenen Amazon SWF SWF-Code erneut ausgeführt werden, sollten Sie den Aktivitätsumfrage möglicherweise auf unbestimmte Zeit laufen lassen.

Das ist das Ende des Codes für unsere `ActivitiesPoller`-Klasse, aber wir fügen am Ende der Datei etwas mehr Code hinzu, damit der Benutzer ihn von der Befehlszeile aus ausführen kann.

```
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
```

Wenn der Benutzer die Datei von der Befehlszeile aus ausführt (und ihr eine Aktivitätsaufgabenliste als erstes Argument weiterleitet), instanziiert dieser Code die Poller-Klasse und startet diese durch die Abfrage nach Aktivitäten. Nachdem der Poller seine Aufgabe abgeschlossen hat (nach dem Start der letzten Aktivität), drucken wir einfach eine Nachricht aus und beenden ihn.

Das ist alles zum Aktivitäten-Poller. Jetzt müssen Sie nur noch den Code ausführen und in [Abonnement-Workflow, Anleitung: Ausführen des Workflows](#) sehen, wie er funktioniert.

Abonnement-Workflow, Anleitung: Ausführen des Workflows

Nun, da Sie die Implementierung Ihres Workflows, Ihrer Aktivitäten sowie der Workflow- und Aktivitäten-Poller abgeschlossen haben, können Sie den Workflow ausführen.

Falls Sie dies noch nicht getan haben, müssen Sie Ihre AWS Zugangsschlüssel in der `aws-config.txt` Datei angeben, wie [Konfiguration der Sitzung AWS](#) in Teil 1 des Tutorials beschrieben.

Gehen Sie jetzt zu Ihrer Befehlszeile und ändern Sie das Verzeichnis, in dem sich die Quelldateien der Anleitung befinden. Es sollten folgende Dateien vorhanden sein:

```
.
|-- 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
```

Beginnen Sie nun den Workflow mit dem folgenden Befehl.

```
ruby swf_sns_workflow.rb
```

Dies startet den Workflow. Außerdem sollte eine Meldung ausgegeben werden, mit einer Zeile, die Sie kopieren und in einem separaten Befehlszeilenfenster (oder sogar auf einem anderen Computer, wenn Sie die Quelldateien der Anleitung darauf kopiert haben) einfügen können.

```
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...

Der Workflow-Code wartet, bis Sie den Aktivitäten-Poller in einem separaten Fenster gestartet haben.

Öffnen Sie ein neues Befehlszeilenfenster, ändern Sie erneut das Verzeichnis, in dem sich die Quelldateien befinden, und verwenden Sie dann den von der Datei `swf_sns_workflow.rb` bereitgestellten Befehl, um den Aktivitäten-Poller zu starten. Wenn Sie beispielsweise die vorherige Ausgabemeldung erhalten haben, geben (oder fügen) Sie Folgendes ein.

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

Sobald Sie die Ausführung Ihres Aktivitäten-Pollers gestartet haben, beginnt er, Informationen über die Aktivitätenregistrierung auszugeben.

```
** 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
```

Sie können jetzt zu Ihrem ursprünglichen Befehlszeilenfenster zurückkehren und auf die Eingabetaste drücken, um die Workflow-Ausführung zu starten. Sie registriert den Workflow und plant die erste Aktivität.

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

Gehen Sie zurück zu dem anderen Fenster, in dem Ihr Aktivitäten-Poller ausgeführt wird. Das Ergebnis der ersten ausgeführten Aktivität wird angezeigt und eine Eingabeaufforderung für die Eingabe Ihrer E-Mail-Adresse oder SMS-Telefonnummer wird bereitgestellt. Geben Sie eine dieser Informationen oder auch beide ein und bestätigen Sie dann die Texteingabe.

```
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

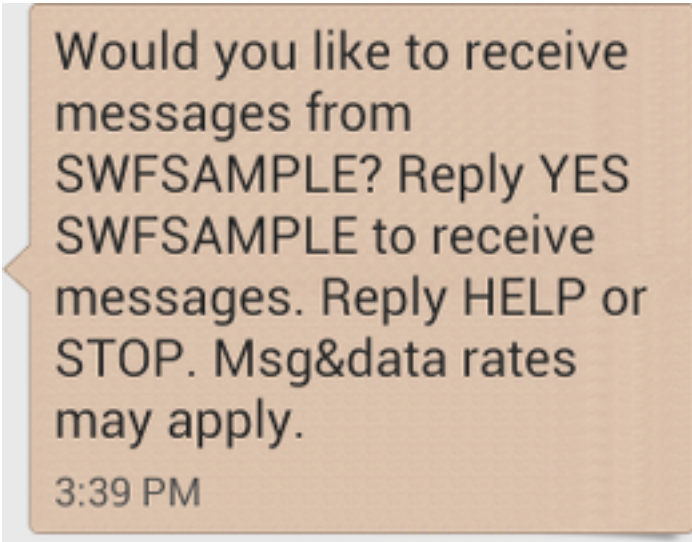
phone: 12065550101

Is this correct? (y/n): y

Note

Die hier gezeigte Telefonnummer ist rein fiktiv und wird nur zu Demonstrationszwecken verwendet. Verwenden Sie an dieser Stelle Ihre eigene Telefonnummer bzw. E-Mail-Adresse!

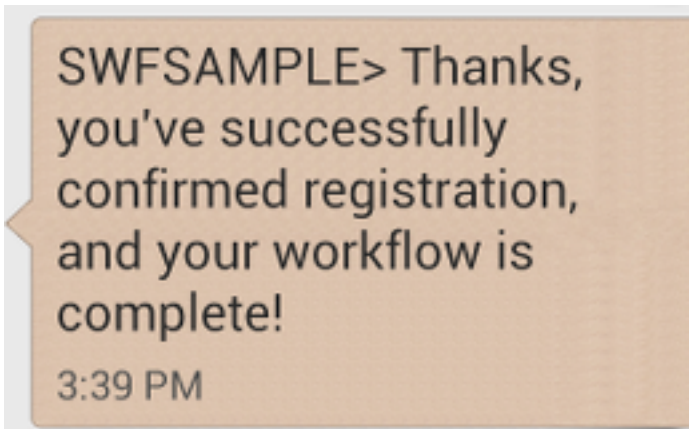
Kurz nach Eingabe dieser Informationen sollten Sie eine E-Mail oder Textnachricht von Amazon SNS erhalten, in der Sie aufgefordert werden, Ihr Themenabonnement zu bestätigen. Wenn Sie eine SMS-Telefonnummer eingegeben haben, wird eine Meldung ähnlich der folgenden auf Ihrem Mobiltelefon angezeigt.



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

Wenn Sie auf diese Nachricht mit YES antworten, erhalten Sie die in `send_result_activity` bereitgestellte Antwort.



Während all dies ablief, haben Sie gesehen, was in Ihrem Befehlszeilenfenster passiert ist? Sowohl die Workflow- als auch die Aktivitäten-Poller haben intensiv gearbeitet.

Hier sehen Sie die Ausgabe des Workflow-Pollers.

```
** 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...
```

Und dies ist die Ausgabe des Aktivitäten-Pollers, der zur selben Zeit in einem anderen Befehlszeilenfenster ablief.

```
++ 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!
```

Glückwunsch, Ihr Workflow ist abgeschlossen – ebenso wie diese Anleitung!

Sie können den Workflow erneut ausführen, um zu sehen, wie Timeouts funktionieren, oder um unterschiedliche Daten einzugeben. Denken Sie nur daran: Sobald Sie ein Thema abonnieren, haben Sie es so lange abonniert, bis Sie das Abonnement beenden. Wenn Sie den Workflow erneut ausführen, bevor Sie sich von Themen abmelden, wird dies wahrscheinlich automatisch zum Erfolg führen, da dort angezeigt `wait_for_confirmation_activity` wird, dass Ihr Abonnement bereits bestätigt wurde.

So melden Sie sich vom Amazon SNS SNS-Thema ab

- Antworten Sie verneinend auf die SMS (senden Sie STOP).
- Wählen Sie den Link zum Beenden des Abonnements aus, den Sie per E-Mail erhalten haben.

Sie können nun das Thema erneut abonnieren.

Wie geht es weiter?

In diesem Tutorial wurden viele Themen behandelt, aber es gibt noch viel mehr AWS SDK für Ruby, was Sie über Amazon SWF oder Amazon SNS erfahren können. Weitere Informationen und viele weitere Beispiele finden Sie in der jeweiligen offiziellen Dokumentation:

- [AWS SDK für Ruby -Dokumentation](#)
- [Dokumentation zu Amazon Simple Notification Service](#)
- [Dokumentation zum Amazon Simple Workflow Service](#)

In der Amazon SWF SWF-Konsole arbeiten

Die Amazon SWF SWF-Konsole bietet Optionen zur Konfiguration, Initiierung und Verwaltung von Workflow-Ausführungen.

Mit der Amazon SWF SWF-Konsole können Sie:

- Workflow-Domänen registrieren.
- Workflow-Typen und Aktivitätstypen registrieren.
- Workflow-Ausführungen starten, anzeigen, signalisieren, abrechnen, beenden und neu starten.

Registrieren einer Domain

Workflows werden in einer als Domäne bezeichneten AWS Ressource ausgeführt, die den Umfang des Workflows steuert. Ein AWS Konto kann mehrere Domänen haben, von denen jede mehrere Workflows enthalten kann, aber Workflows in verschiedenen Domänen können nicht interagieren.

Die Domainregistrierung ist die einzige Funktionalität, die ursprünglich in der Konsole verfügbar war. Nachdem mindestens eine Domain registriert wurde, können Sie die folgenden Aktionen für die Domain ausführen:

- Registrieren Sie Workflow- und Aktivitätstypen.
- Workflow-Ausführungen initiieren.
- Signale für laufende Workflow-Ausführungen abrechnen, beenden und senden.
- Geschlossene Workflow-Ausführungen neu starten.

Sie können auch Aktionen zur Domänenverwaltung durchführen, z. B. Domänen als veraltet oder nicht mehr als gültig deklarieren.

Nachdem Sie eine Domain als veraltet markiert haben, können Sie sie nicht mehr verwenden, um neue Workflow-Ausführungen zu erstellen oder neue Workflows zu registrieren. Wenn eine Domäne als veraltet eingestuft wird, werden auch alle Aktivitäten und Workflows, die in der Domäne registriert sind, als veraltet angesehen. Ausführungen, die gestartet wurden, bevor die Domain als veraltet galt, werden weiterhin ausgeführt.

Nachdem Sie eine zuvor als veraltet eingestufte Domäne nicht mehr unterstützt haben, können Sie die Domäne wieder verwenden, um Workflowtypen zu registrieren und neue Workflow-Ausführungen zu starten.

Weitere Informationen zu diesen Domänenverwaltungsaktionen finden Sie unter und.

[DeprecateDomainUndeprecateDomain](#)

Workflowtypen registrieren

Sie können Workflow-Typen in der Amazon SWF SWF-Konsole registrieren, nachdem Sie mindestens eine Domain registriert haben.

Ein Workflow-Typ besteht aus einer Reihe von Aktivitätstypen, die ein Ziel verfolgen und die Logik enthalten, mit der die Aktivitäten koordiniert werden. Workflowtypen koordinieren und verwalten die Ausführung von Aktivitäten, die asynchron auf mehreren Computergeräten ausgeführt werden können und sowohl sequentielle als auch parallel Verarbeitungsmethoden bieten.

So registrieren Sie einen Amazon SWF SWF-Workflowtyp mithilfe der Konsole

1. Öffnen Sie die Domain, in der Sie einen Workflow registrieren möchten.
2. Wählen Sie Registrieren und dann Workflow registrieren aus.
3. Geben Sie auf der Seite Workflow registrieren den Workflow-Namen und die Workflow-Version ein. Optional können Sie auch eine [Standardaufgabenliste](#) angeben, die verwendet wird, um Entscheidungsaufgaben für die Ausführung dieses Workflows zu planen.
4. (Optional) Wählen Sie Erweiterte Optionen, um die folgenden Details für Ihren Workflow anzugeben:
 - [Standard-Aufgabenpriorität](#) — Die Standard-Aufgabenpriorität, die dem Workflow zugewiesen werden soll.
 - [Standardzeitlimit für Ausführungsbeginn bis Ende](#) — Die standardmäßige maximale Dauer für Ausführungen dieses Workflows.
 - [Standard-Timeout für Start bis Ende](#) von Aufgaben — Die standardmäßige maximale Dauer von Entscheidungsaufgaben für diesen Workflow.
 - [Standardrichtlinie für untergeordnete](#) Arbeitsabläufe — Die Standardrichtlinie, die für die Ausführung untergeordneter Workflows verwendet wird.
 - [Standard-Lambda-Rolle](#) — Die Standard-IAM-Rolle, die diesem Workflow zugeordnet ist.
5. Wählen Sie Workflow registrieren aus.

Aktivitätstypen registrieren

Aktivitäten sind Aufgaben, die Ihr Workflow-Typ koordinieren und ausführen soll (z. B.: Kundenbestellung verifizieren, Kreditkarte belasten usw.). Die Reihenfolge, in der Aktivitäten ausgeführt werden, wird durch die Koordinationslogik des Workflow-Typs bestimmt.

Sie können Aktivitätstypen registrieren, nachdem mindestens eine Domain registriert wurde.

So registrieren Sie einen Amazon SWF SWF-Aktivitätstyp mit der Konsole

1. Öffnen Sie die Domain, in der Sie eine Aktivität registrieren möchten.
2. Wählen Sie Registrieren und anschließend Aktivität registrieren aus.
3. Geben Sie auf der Seite Aktivität registrieren den [Namen der Aktivität](#) und die [Version der Aktivität](#) ein. Optional können Sie auch eine [Standard-Aufgabenliste](#) angeben, die zur Planung der Aufgaben dieser Aktivität verwendet wird.
4. (Optional) Wählen Sie Erweiterte Optionen, um die folgenden Details für Ihre Aktivität anzugeben:
 - [Standard-Aufgabenpriorität](#) — Die Standard-Aufgabenpriorität, die der Aktivität zugewiesen werden soll.
 - [Standard-Zeitlimit für den Start von Aufgaben](#) — Die standardmäßige maximale Dauer, die eine Aufgabe dieser Aktivität abwarten kann, bevor sie einer Arbeitskraft zugewiesen wird.
 - [Standardzeitlimit für den Start bis zum Abschluss von Aufgaben](#) — Die standardmäßige maximale Dauer, die ein Mitarbeiter für die Bearbeitung von Aufgaben dieser Aktivität benötigen kann.
 - [Standardzeitplan für das Schließen von Aufgaben](#) — Die standardmäßige maximale Dauer für eine Aufgabe dieser Aktivität.
 - [Standard-Timeout für Aufgaben](#) — Die [standardmäßige maximale Zeitspanne](#), vor der ein Mitarbeiter, der eine Aufgabe dieses Typs bearbeitet, den Fortschritt per Anruf melden muss. [RecordActivityTaskHeartbeat](#)
5. Wählen Sie Aktivität registrieren.

Einen Workflow starten

Sie können eine Workflow-Ausführung von der Amazon SWF SWF-Konsole aus starten. Sie können eine Workflow-Ausführung erst starten, wenn Sie mindestens einen Workflow registriert haben.

So starten Sie eine Workflow-Ausführung mithilfe der Konsole

1. Öffnen Sie die Amazon SWF SWF-Konsole und wählen Sie im linken Navigationsbereich Domains aus.
2. Wählen Sie unter dem Domainnamen die Option Workflows aus.
3. Wählen Sie auf der Seite Workflows den Workflow aus, den Sie ausführen möchten.
4. Wählen Sie Start execution (Ausführung starten) aus.
5. Geben Sie auf der Seite Ausführung starten den [Workflow-Namen](#) und die Ausführungs-ID ein, um Ihre Ausführung anhand eines Namens zu identifizieren. Optional können Sie auch eine [Aufgabenliste](#) angeben, die für die Entscheidungsaufgaben verwendet wird, die für diese Workflow-Ausführung generiert wurden.
6. (Optional) Wählen Sie Erweiterte Optionen, um die folgenden Details für Ihre Workflow-Ausführung anzugeben:
 - [Aufgabenpriorität](#) — Die Aufgabenpriorität, die für diese Workflow-Ausführung verwendet werden soll.
 - [Timeout für den Start bis zum Ende der Ausführung](#) — Die Gesamtdauer dieser Workflow-Ausführung.
 - [Timeout für Start bis Ende der Aufgabe](#) — Die maximale Dauer von Entscheidungsaufgaben für diese Workflow-Ausführung.
 - [Untergeordnete Richtlinie](#) — Die Richtlinie, die für die untergeordneten Workflow-Ausführungen dieser Workflow-Ausführung verwendet werden soll, wenn diese durch expliziten Aufruf der [TerminateWorkflowExecution](#)Aktion oder aufgrund eines abgelaufenen Timeouts beendet wird.
 - [Lambda-Rolle](#) — Die IAM-Rolle, die dieser Workflow-Ausführung zugewiesen werden soll.
7. Wählen Sie Start execution (Ausführung starten) aus.

Workflow-Ausführungen verwalten

Sie können Ihre Workflow-Ausführungen nach Name, Status, ID und Tag filtern. Sie können Signale mit Eingaben in aktive Workflow-Ausführungen senden. Wenn Sie einen Workflow abbrechen oder beenden müssen, können Sie die Option Try-Cancel verwenden. Das Abbrechen ist dem Beenden einer Workflow-Ausführung vorzuziehen, da das Abbrechen dem Workflow die Möglichkeit gibt, alle Bereinigungsaufgaben auszuführen und dann ordnungsgemäß zu beenden.

In der Konsole können Sie die Workflow-Ausführungen verwalten, die derzeit geschlossen ausgeführt werden. and/or

Um Ihre Workflow-Ausführungen zu verwalten

1. Öffnen Sie eine Domain, um deren Workflow-Ausführungen zu verwalten.
2. Wählen Sie Find Execution aus.
3. Wählen Sie auf der Seite Workflow-Ausführungen die Option Ausführungen nach Eigenschaft filtern aus, und wählen Sie dann unter Eigenschaften einen der folgenden Filter aus:

Klicken Sie auf	Um diesen Filter anzuwenden
Workflow	<p>Wählen Sie diesen Filter, um Ausführungen eines bestimmten Workflows aufzulisten. Gehen Sie beispielsweise wie folgt vorfiction-books-order-workflow , um Ausführungen von anzuzeigen:</p> <ol style="list-style-type: none"> 1. Wählen Sie Workflow. 2. Wählen Sie unter Operatoren die Option Equals aus. 3. Wählen Sie unter Workflows die Option fiction-books-order-workflow. 4. (Optional) Wählen Sie Filter löschen, um den Filter zu entfernen und eine neue Suche nach Ausführungen zu starten.
Status	<p>Wählen Sie diesen Filter, um Ausführungen mit einem bestimmten Status aufzulisten. Gehen Sie beispielsweise wie folgt vor, um Ausführungen mit dem Status Fehlgeschlagen anzuzeigen:</p> <ol style="list-style-type: none"> 1. Wählen Sie Status. 2. Wählen Sie unter Operatoren die Option Equals aus. 3. Wählen Sie unter Status die Option Fehlgeschlagen aus. 4. (Optional) Wählen Sie Filter löschen, um den Filter zu entfernen und eine neue Suche nach Ausführungen zu starten.
Ausführungs-ID	<p>Wählen Sie diesen Filter, um eine Workflow-Ausführung anhand ihrer ID anzuzeigen. Gehen Sie beispielsweise wie folgt vorfiction-b</p>

Klicken Sie auf	Um diesen Filter anzuwenden
	<p>books-order-category1 , um die Ausführung mit ID anzuzeigen:</p> <ol style="list-style-type: none"> 1. Wählen Sie Ausführungs-ID. 2. Wählen Sie unter Operatoren die Option Equals aus. 3. Wählen Sie unter Ausführung IDs die Option fiction-books-order-category1 aus. 4. (Optional) Wählen Sie Filter löschen, um den Filter zu entfernen und eine neue Suche nach Ausführungen zu starten.
Markierung	<p>Wählen Sie diesen Filter, um Ausführungen mit einem bestimmten Tag aufzulisten. Um beispielsweise Ausführungen mit dem Status anzuzeigenpurchaseOrder , gehen Sie wie folgt vor:</p> <ol style="list-style-type: none"> 1. Wählen Sie Tag. 2. Wählen Sie unter Operatoren die Option Equals aus. 3. Wählen Sie unter Tag die Option PurchaseOrder aus. 4. (Optional) Wählen Sie Filter löschen, um den Filter zu entfernen und eine neue Suche nach Ausführungen zu starten.

4. (Optional) Nachdem Sie den erforderlichen Filter angewendet haben, um Workflow-Ausführungen aufzulisten, können Sie die folgenden Operationen für eine aktive Ausführung ausführen:
 - Signal — Verwenden Sie diese Option, um einer laufenden Workflow-Ausführung zusätzliche Daten zu senden. So gehen Sie vor:
 1. Wählen Sie die Ausführung aus, an die Sie zusätzliche Daten senden möchten.
 2. Wählen Sie Signal und geben Sie dann die Daten im Dialogfeld Signalausführung an.
 3. Wählen Sie Signal.
 - Try-Cancel — Verwenden Sie diese Option, um zu versuchen, eine Workflow-Ausführung abzubrechen. Eine Workflow-Ausführung wird besser abgebrochen als beendet. Durch das Abbrechen der Workflow-Ausführung wird zunächst eine Bereinigung durchgeführt und die Workflow-Ausführung dann korrekt geschlossen.
 1. Wählen Sie die Ausführung aus, die Sie abbrechen möchten.

2. Wählen Sie Try-Cancel.
- Beenden — Verwenden Sie diese Option, um die Ausführung eines Workflows zu beenden. Beachten Sie, dass Workflow-Ausführungen besser abgebrochen als beendet werden sollten.
 1. Wählen Sie die Ausführung aus, die Sie beenden möchten.
 2. Stellen Sie sicher, dass unter Richtlinie für Kinder die Option Terminieren ausgewählt ist.
 3. (Optional) Geben Sie den Grund und die Details für die Beendigung der Ausführung an.
 4. Wählen Sie Beenden.
5. (Optional) Erneut ausführen — Verwenden Sie diese Option, um eine abgeschlossene Workflow-Ausführung erneut auszuführen.
 1. Wählen Sie in der Liste der Workflow-Ausführungen die gewünschte Ausführung aus. Wenn Sie eine geschlossene Ausführung auswählen, wird die Schaltfläche Erneut ausführen aktiviert. Wählen Sie Erneut ausführen.
 2. Geben Sie auf der Seite Ausführung erneut ausführen die Details für die Workflow-Ausführung an, wie unter beschrieben. [Einen Workflow starten](#)

Grundlegende Workflow-Konzepte in Amazon SWF

Note

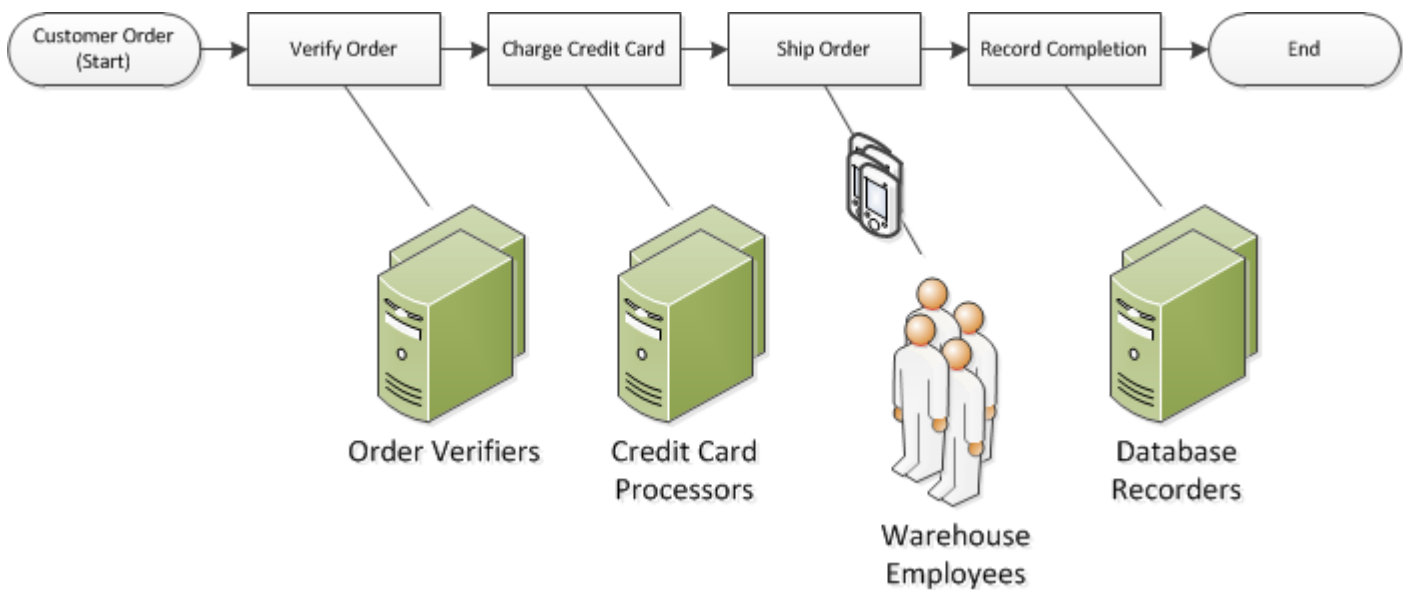
Die Konzepte in diesem Kapitel bieten einen Überblick über den Amazon Simple Workflow Service und beschreiben seine wichtigsten Funktionen. Wenn Sie nach Beispielen suchen, finden Sie weitere Informationen unter [Arbeiten mit Amazon SWF APIs](#).

Mit dem Amazon Simple Workflow Service (Amazon SWF) können Sie verteilte, asynchrone Anwendungen als Workflows implementieren. Über Workflows wird die Ausführung von Aktivitäten koordiniert und verwaltet, die asynchron über mehrere Geräte ausgeführt werden können und sowohl eine sequentielle als auch eine parallele Verarbeitung anbieten.

Beim Entwerfen eines Workflows analysieren Sie Ihre Anwendung, um die zugehörigen Aufgaben (Tasks) der Komponenten zu ermitteln. In Amazon SWF werden diese Aufgaben durch Aktivitäten dargestellt. In welcher Reihenfolge die Aktivitäten ausgeführt werden, wird von der Koordinationslogik des Workflows bestimmt.

Beispiel-Workflow für eine E-Commerce-Anwendung

Die folgende Abbildung zeigt einen Workflow zur Auftragsabwicklung im E-Commerce, an dem sowohl Personen als auch automatisierte Prozesse beteiligt sind:



Der Workflow für E-Commerce-Anwendungen beginnt, wenn ein Kunde eine Bestellung aufgibt, und umfasst vier Aufgaben:

1. Die Bestellung wird überprüft.
2. Wenn die Bestellung gültig ist, wird sie dem Kunden berechnet.
3. Nach erfolgter Zahlung wird die Bestellung ausgeliefert.
4. Wenn die Bestellung ausgeliefert wird, werden die Bestelldetails gespeichert.

Die Aufgaben in diesem Workflow werden nacheinander ausgeführt (sequenziell): Vor Belastung einer Kreditkarte muss eine Bestellung überprüft worden sein, vor Auslieferung einer Bestellung muss eine Kreditkarte erfolgreich belastet worden sein und vor Aufzeichnung einer Bestellung muss sie ausgeliefert worden sein. Da Amazon SWF verteilte Prozesse unterstützt, können diese Aufgaben jedoch an verschiedenen Orten ausgeführt werden. Wenn es sich um programmgesteuerte Aufgaben handelt, können sie auch in unterschiedlichen Programmiersprachen oder mithilfe unterschiedlicher Tools geschrieben werden.

Neben der sequentiellen Verarbeitung von Aufgaben unterstützt Amazon SWF auch Workflows mit parallel Verarbeitung von Aufgaben. Parallele Aufgaben werden zur gleichen Zeit ausgeführt. Außerdem können sie unabhängig voneinander durch unterschiedliche Anwendungen oder Benutzer ausgeführt werden. Ihr Workflow entscheidet, wie es weitergeht, wenn eine oder mehrere der parallelen Aufgaben abgeschlossen wurden.

Zusätzliche Konzepte

- [Einen Workflow in Amazon SWF erstellen](#)
- [Workflows in Amazon SWF ausführen](#)
- [Workflow-Verlauf in Amazon SWF](#)
- [Objektbezeichner in Amazon SWF](#)
- [Domänen in Amazon SWF](#)
- [Schauspieler in Amazon SWF](#)
- [Aufgaben in Amazon SWF](#)
- [Aufgabenlisten in Amazon SWF](#)
- [Abschluss der Workflow-Ausführung in Amazon SWF](#)
- [Lebenszyklus eines Amazon SWF SWF-Workflows](#)
- [Abfragen für Aufgaben in Amazon SWF](#)

Einen Workflow in Amazon SWF erstellen

Das Erstellen eines einfachen Workflows umfasst die folgenden Schritte.

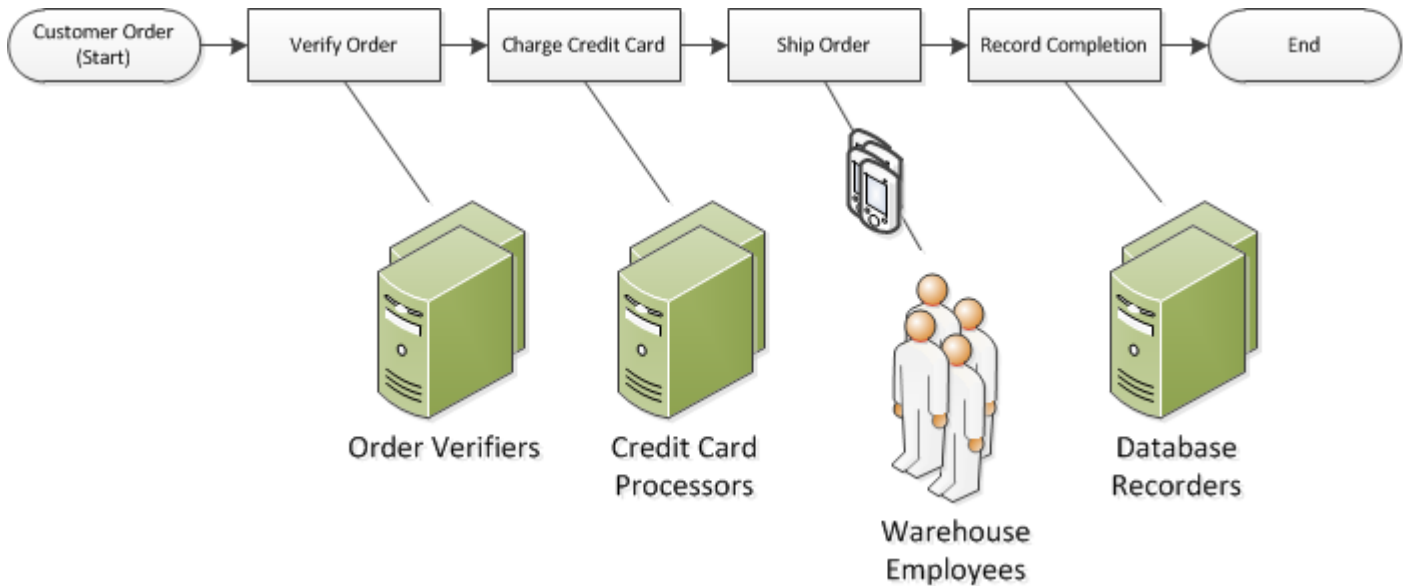
- Modellieren eines Workflows, Registrieren des entsprechenden Typs und der Aktivitätstypen
- Entwickeln und Starten von Aktivitäts-Workern, die die Aktivitätsaufgaben ausführen
- Entwickeln und Starten von Entscheidern, die anhand des Workflow-Verlaufs den nächsten Schritt festlegen
- Entwickeln und Starten von Workflow-Startern, d. h. Anwendungen, die Workflow-Ausführungen starten

Modellieren des Workflows und der entsprechenden Aktivitäten

Um Amazon SWF zu verwenden, modellieren Sie die logischen Schritte in Ihrer Anwendung als Aktivitäten. Eine Aktivität repräsentiert einen einzelnen logischen Schritte oder eine Aufgabe in Ihrem Workflow. Das Autorisieren einer Kreditkarte ist beispielsweise eine Aktivität, die das Angeben einer Kreditkartennummer und anderer Informationen sowie das Empfangen eines Genehmigungscode oder einer Nachricht umfasst, dass die Karte abgelehnt wurde.

Zusätzlich zu den Aktivitäten müssen Sie außerdem die Koordinationslogik für die Entscheidungen definieren. Die Koordinationslogik plant z. B. eine andere Folgeaktivität, je nachdem, ob die Kreditkarte genehmigt oder abgelehnt wurde.

Die folgende Abbildung zeigt ein Beispiel eines sequenziellen Kundenbestellungs-Workflows mit vier Aktivitäten (Bestellung prüfen, Kreditkarte belasten, Bestellung absenden und Vorgang schließen).



Workflows in Amazon SWF ausführen

Nachdem die Koordinationslogik und die Aktivitäten entworfen wurden, registrieren Sie diese Komponenten als Workflow- und Aktivitätstypen bei Amazon SWF. Bei der Registrierung geben Sie einen Namen, eine Version und Standardkonfigurationswerte für jeden Typ an.

Nur registrierte Workflow- und Aktivitätstypen können mit Amazon SWF verwendet werden. Im E-Commerce-Beispiel würden Sie den CustomerOrder Workflow-Typ und die RecordCompletion Aktivitätstypen VerifyOrder ChargeCreditCard ShipOrder,, und registrieren.

Nachdem Sie den Workflowtyp registriert haben, können Sie ihn beliebig oft ausführen. Eine Workflow-Ausführung ist eine laufende Instance eines Workflows.

Eine Workflow-Ausführung kann von beliebigen Prozessen oder Anwendungen, sogar von einer anderen Workflow-Ausführung, gestartet werden. Bei dem E-Commerce-Beispiel wird bei jeder Kundenbestellung eine neue Workflow-Ausführung gestartet. Die Art der Anwendung, die den Workflow initiiert, hängt davon ab, wie der Kunde die Bestellung aufgibt. Der Workflow könnte von einer Website oder mobilen Anwendung oder von einem Kundendienstmitarbeiter über eine interne Unternehmensanwendung initiiert werden.

Mit Amazon SWF können Sie Ihren Workflow-Ausführungen eine Kennung — eine so genannte `workflowId` — zuordnen, sodass Sie Ihre vorhandenen Geschäftskennungen in Ihren Workflow integrieren können. Bei dem E-Commerce-Beispiel lässt sich jede Workflow-Ausführung anhand der Rechnungsnummer identifizieren.

Zusätzlich zu der von Ihnen angegebenen ID ordnet Amazon SWF jeder Workflow-Ausführung eine eindeutige, vom System generierte Kennung — eine `runId` — zu. Amazon SWF lässt zu, dass jeweils nur eine Workflow-Ausführung mit dieser ID ausgeführt wird. Sie können zwar mehrere Workflow-Ausführungen desselben Workflow-Typs haben, aber jede Workflow-Ausführung hat eine eigene `runId`.

Workflow-Verlauf in Amazon SWF

Amazon SWF zeichnet den Fortschritt jeder Workflow-Ausführung im Workflow-Verlauf auf — eine detaillierte, vollständige und konsistente Aufzeichnung aller Ereignisse, die seit Beginn der Workflow-Ausführung aufgetreten sind.

Ein Ereignis stellt eine diskrete Änderung des Status Ihrer Workflow-Ausführung dar, z. B. wenn eine neue Aktivität geplant ist oder eine laufende Aktivität abgeschlossen wird. Der Workflow-Verlauf enthält alle Ereignisse, die eine Änderung des Status der Workflow-Ausführung verursachen. Dazu zählen geplante und abgeschlossene Aktivitäten, Aufgaben-Timeouts und Signale.

Operationen, die den Status einer Workflow-Ausführung nicht ändern, werden in der Regel nicht im Workflow-Verlauf erfasst. So enthält der Workflow-Verlauf beispielsweise keine Abrufversuche oder Daten zur Verwendung von Sichtbarkeitsoperationen.

Der Workflow-Verlauf bietet mehrere wichtige Vorteile:

- Anwendungen können statuslos sein, da alle Informationen über eine Workflow-Ausführung in ihrem Workflow-Verlauf gespeichert werden.
- Der Verlauf stellt für jede Workflow-Ausführung einen Datensatz mit den geplanten Aktivitäten, deren aktuellen Status und deren Ergebnisse zur Verfügung. Anhand dieser Informationen werden die nächsten Schritte der Workflow-Ausführung bestimmt.
- Der Verlauf enthält detaillierte Audit-Trail-Daten, die Sie zum Überwachen von Workflow-Ausführungen und zum Prüfen abgeschlossener Workflow-Ausführungen nutzen können.

Im Folgenden finden Sie einen konzeptionellen Überblick über den Verlauf des E-Commerce-Workflows:

```
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
```

Im vorherigen Beispiel ist die Bestellung bereit für den Versand. Im folgenden Beispiel ist die Bestellung abgeschlossen. Da der Workflow-Verlauf kumulativ ist, werden die neueren Ereignisse angefügt:

```
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
```

Programmgesteuert werden die Ereignisse im Verlauf der Workflow-Ausführung als JSON-Objekte (JavaScript Object Notation) dargestellt. Beim Verlauf selbst handelt es sich um ein JSON-Array dieser Objekte. Jedes Ereignis verfügt über Folgendes:

- Ein Typ, wie oder [WorkflowExecutionStartedActivityTaskCompleted](#)

- Einen Zeitstempel im Unix-Zeitformat
- Eine ID, über die das Ereignis eindeutig identifiziert wird

Darüber hinaus verfügt jeder Ereignistyp über einen eindeutigen Satz an beschreibenden Attributen, die dem Typ entsprechen. Das `ActivityTaskCompleted` Ereignis hat beispielsweise Attribute, die die IDs für die Ereignisse stehen, die dem Zeitpunkt entsprechen, zu dem die Aktivitätsaufgabe geplant und gestartet wurde, sowie ein Attribut, das Ergebnisdaten enthält.

Mithilfe der [GetWorkflowExecutionHistory](#) Aktion können Sie eine Kopie des aktuellen Status des Workflow-Ausführungsverlaufs abrufen. Darüber hinaus erhält der Entscheider im Rahmen der Interaktion zwischen Amazon SWF und dem Entscheider für Ihren Workflow regelmäßig Kopien des Verlaufs.

Nachfolgend sehen Sie ein Beispiel für einen Workflow-Ausführungsverlauf im JSON-Format.

```
[ {
  "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"
  }
]

```

Eine detaillierte Liste der verschiedenen Ereignistypen, die im Verlauf der Workflow-Ausführung erscheinen können, finden Sie in der Amazon Simple Workflow Service API-Referenz zum [HistoryEvent](#)Datentyp.

Amazon SWF speichert den vollständigen Verlauf aller Workflow-Ausführungen für eine konfigurierbare Anzahl von Tagen nach Abschluss der Ausführung. Dieser Zeitraum, der auch als Aufbewahrungszeitraum des Workflow-Verlaufs bezeichnet wird, wird festgelegt, wenn Sie eine

Domäne für Ihren Workflow registrieren. Domänen werden an späterer Stelle in diesem Abschnitt detaillierter erläutert.

Objektbezeichner in Amazon SWF

In der folgenden Liste wird beschrieben, wie Amazon SWF SWF-Objekte, wie Workflow-Ausführungen, eindeutig identifiziert werden.

- **Workflow-Typ** — Ein registrierter Workflow-Typ wird anhand seiner Domäne, seines Namens und seiner Version identifiziert. Workflow-Typen werden im Aufruf `RegisterWorkflowType` angegeben.
- **Aktivitätstyp** — Ein registrierter Aktivitätstyp wird anhand seiner Domäne, seines Namens und seiner Version identifiziert. Aktivitätstypen werden im Aufruf `RegisterActivityType` angegeben.
- **Entscheidungsaufgaben und Aktivitätsaufgaben** — Jede Entscheidungsaufgabe und Aktivitätsaufgabe wird durch ein eindeutiges Aufgabentoken identifiziert. Das Aufgaben-Token wird von Amazon SWF generiert und zusammen mit anderen Informationen über die Aufgabe in der Antwort von `PollForDecisionTask` oder zurückgegeben `PollForActivityTask`. Das Token wird zwar meist vom Prozess verwendet, der die Aufgabe empfangen hat, dieser Prozess kann das Token jedoch auch an einen anderen Prozess übergeben, der dann wiederum den Abschluss oder Fehlschlag der Aufgabe berichtet.
- **Workflow-Ausführung** — Eine einzelne Ausführung eines Workflows wird anhand der Domäne, der Workflow-ID und der Ausführungs-ID identifiziert. Die ersten beiden sind Parameter, an die übergeben werden [StartWorkflowExecution](#). Die Ausführungs-ID wird von `StartWorkflowExecution` zurückgegeben.

Domänen in Amazon SWF

Workflows werden in einer als Domain bezeichneten AWS Ressource ausgeführt und bieten eine Möglichkeit, Amazon SWF SWF-Ressourcen innerhalb Ihres AWS Kontos abzugrenzen. Alle Komponenten eines Workflows, wie beispielsweise der Workflow-Typ und die Aktivitätstypen, müssen einer Domäne zugewiesen sein.

Ein AWS Konto kann mehrere Domänen haben, von denen jede mehrere Workflows enthalten kann, aber Workflows in verschiedenen Domänen können nicht interagieren.

Sofern noch nicht geschehen, müssen Sie beim Einrichten eines neuen Workflows zunächst eine Domäne registrieren, bevor Sie die anderen Workflow-Komponenten einrichten können.

Wenn Sie eine Domäne registrieren, geben Sie einen Aufbewahrungszeitraum für den Workflow-Verlauf an. Die Aufbewahrungsfrist ist der Zeitraum, für den Amazon SWF weiterhin Informationen über die Workflow-Ausführung aufbewahrt, nachdem die Workflow-Ausführung abgeschlossen ist.

Die Domainregistrierung ist die einzige Funktion, die ursprünglich in der Konsole verfügbar war. Nachdem mindestens eine Domain registriert wurde, können Sie die folgenden Aktionen für die Domain ausführen:

- Registrieren Sie Workflow- und Aktivitätstypen.
- Workflow-Ausführungen initiieren.
- Signale für laufende Workflow-Ausführungen abrechnen, beenden und senden.
- Geschlossene Workflow-Ausführungen neu starten.

Sie können auch Aktionen zur Domänenverwaltung durchführen, z. B. Domänen als veraltet oder nicht mehr als gültig deklarieren.

Nachdem Sie eine Domain als veraltet markiert haben, können Sie sie nicht mehr verwenden, um neue Workflow-Ausführungen zu erstellen oder neue Workflows zu registrieren. Wenn eine Domäne als veraltet eingestuft wird, werden auch alle Aktivitäten und Workflows, die in der Domäne registriert sind, als veraltet angesehen. Ausführungen, die gestartet wurden, bevor die Domain als veraltet galt, werden weiterhin ausgeführt.

Nachdem Sie eine zuvor als veraltet eingestufte Domäne nicht mehr unterstützt haben, können Sie die Domäne wieder verwenden, um Workflowtypen zu registrieren und neue Workflow-Ausführungen zu starten.

Weitere Informationen zu diesen Domänenverwaltungsaktionen finden Sie unter und.

[DeprecateDomainUndeprecateDomain](#)

Schauspieler in Amazon SWF

Themen

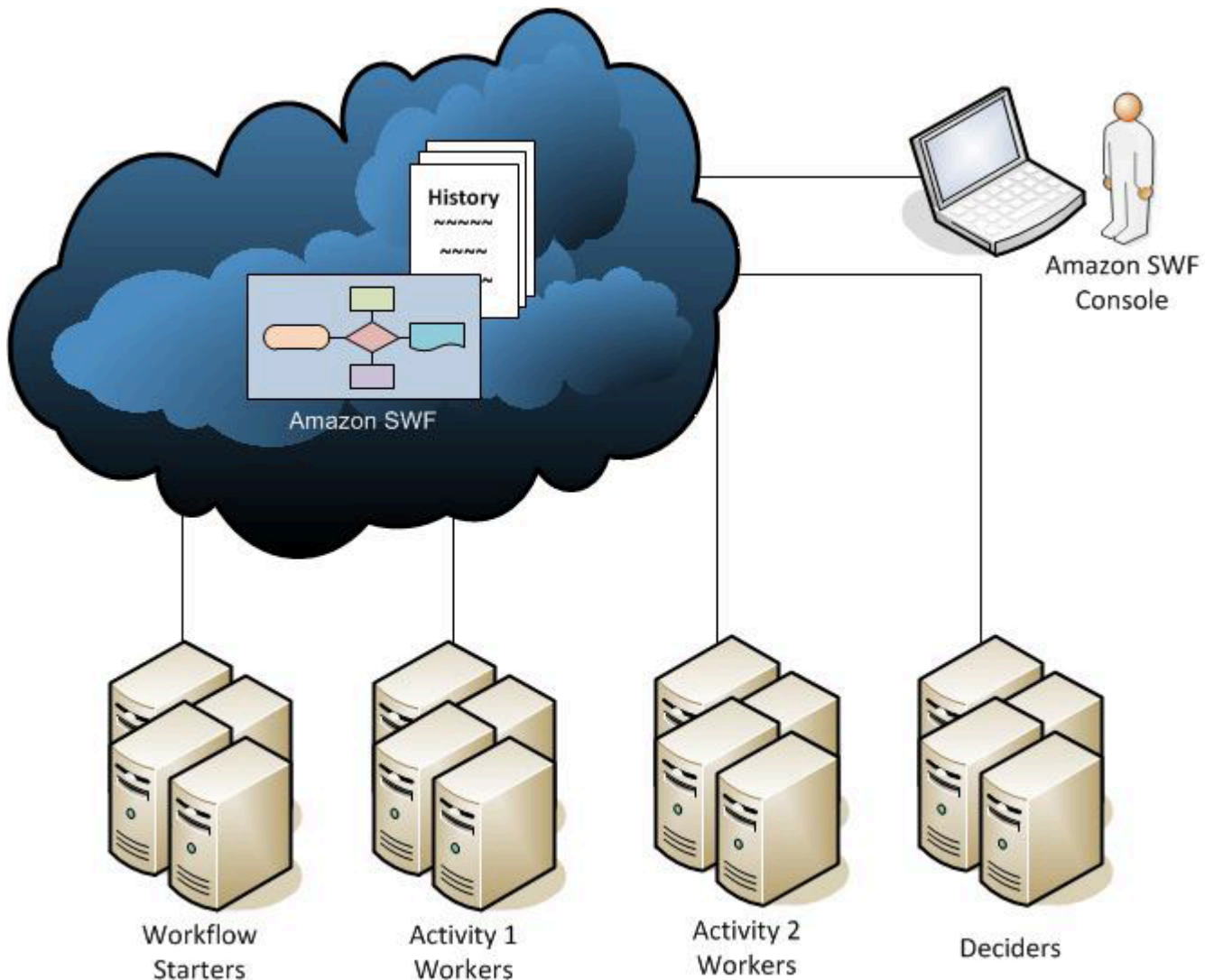
- [Was ist ein Schauspieler in Amazon SWF?](#)
- [Workflow-Starter](#)

- [Entscheider](#)
- [Aktivitäts-Worker](#)
- [Datenaustausch zwischen Akteuren](#)

Was ist ein Schauspieler in Amazon SWF?

Im Laufe seiner Geschäftstätigkeit interagiert Amazon SWF mit einer Reihe verschiedener Arten von programmatischen Akteuren. Akteure können [Workflow-Starter](#), [Entscheider](#) oder [Aktivitäts-Worker](#) sein. Diese Akteure kommunizieren über die API mit Amazon SWF. Sie können diese Akteure in jeder Programmiersprache entwickeln.

Das folgende Diagramm zeigt die Amazon SWF-Architektur, einschließlich Amazon SWF und seiner Akteure.



Workflow-Starter

Ein Workflow-Starter ist eine beliebige Anwendung, die Workflow-Ausführungen initiieren kann. Im E-Commerce beispielsweise könnte ein Workflow-Starter eine Website sein, auf der ein Kunde eine Bestellung aufgibt. Ein anderer Workflow-Starter könnte eine mobile Anwendung oder ein mobiles System sein, das von einem Kundendienstmitarbeiter verwendet wird, um die Bestellung im Auftrag des Kunden aufzugeben.

Entscheider

Ein Entscheider ist eine Implementierung der Koordinationslogik eines Workflows. Entscheider steuern den Ablauf von Aktivitätsaufgaben in einer Workflow-Ausführung. Immer wenn während einer Workflow-Ausführung eine Änderung eintritt, wie etwa der Abschluss einer Aufgabe, wird eine Entscheidungsaufgabe, die den gesamten Workflow-Verlauf enthält, an einen Entscheider übergeben. Wenn der Entscheider die Entscheidungsaufgabe von Amazon SWF erhält, analysiert er den Verlauf der Workflow-Ausführung, um die nächsten geeigneten Schritte in der Workflow-Ausführung zu bestimmen. Der Entscheider teilt diese Schritte mithilfe von Entscheidungen an Amazon SWF zurück. Eine Entscheidung ist ein Amazon SWF SWF-Datentyp, der verschiedene nächste Aktionen darstellen kann. Eine Liste der möglichen Entscheidungen finden Sie unter [Entscheidung](#) in der Amazon Simple Workflow Service API-Referenz.

Hier ist ein Beispiel für eine Entscheidung im JSON-Format, dem Format, in dem sie an Amazon SWF übertragen wird. Diese Entscheidung plant eine neue Aktivitätsaufgabe.

```
{
  "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"
  }
}
```

```
}  
}
```

Ein Entscheider erhält eine Entscheidungsaufgabe, wenn die Workflow-Ausführung gestartet wird, und immer dann, wenn in der Workflow-Ausführung eine Statusänderung stattfindet. Entscheider treiben die Workflow-Ausführung weiter voran, indem sie Entscheidungsaufgaben erhalten und Amazon SWF mit weiteren Entscheidungen antworten, bis der Entscheider feststellt, dass die Workflow-Ausführung abgeschlossen ist. Er antwortet dann mit der Entscheidung, die Workflow-Ausführung abzuschließen. Nach Abschluss der Workflow-Ausführung plant Amazon SWF keine weiteren Aufgaben für diese Ausführung.

Im E-Commerce-Beispiel bestimmt der Entscheider, ob die einzelnen Schritte ordnungsgemäß ausgeführt wurden. Dann plant er entweder den nächsten Schritt oder verwaltet mögliche Fehlerbedingungen.

Ein Entscheider repräsentiert einen einzelnen Computerprozess oder Thread. Mehrere Entscheider können Aufgaben für denselben Workflow-Typ ausführen.

Aktivitäts-Worker

Ein Aktivitäts-Worker ist ein Prozess oder Thread, der die Aktivitätsaufgaben durchführt, die Teil Ihres Workflows sind. Die Aktivitätsaufgabe stellt eine der Aufgaben dar, die Sie in Ihrer Anwendung angegeben haben.

Um eine Aktivitätsaufgabe in Ihrem Workflow zu verwenden, müssen Sie sie entweder über die Amazon SWF SWF-Konsole oder über die [RegisterActivityType](#)Aktion registrieren.

Jeder Aktivitätsarbeiter fragt Amazon SWF nach neuen Aufgaben ab, die für diesen Aktivitätsarbeiter geeignet sind. Bestimmte Aufgaben können nur von bestimmten Aktivitätsmitarbeitern ausgeführt werden. Nach Erhalt einer Aufgabe bearbeitet der Activity Worker die Aufgabe bis zum Abschluss und meldet dann Amazon SWF, dass die Aufgabe abgeschlossen wurde, und stellt das Ergebnis bereit. Der Aktivitäts-Worker führt dann eine Abfrage für eine neue Aufgabe durch. Die einer Workflow-Ausführung zugeordneten Aktivitäts-Worker fahren auf diese Weise fort, indem sie Aufgaben verarbeiten, bis die Workflow-Ausführung selbst abgeschlossen ist. Im E-Commerce-Beispiel sind Aktivitäts-Worker unabhängige Prozesse und Anwendungen, die von Menschen verwendet werden, wie etwa von Kreditkartenverarbeitern und Lagermitarbeitern, die einzelne Schritte im Prozess ausführen.

Ein Aktivitäts-Worker stellt einen einzelnen Computerprozess (oder Thread) dar. Mehrere Aktivitäts-Worker können Aufgaben desselben Aktivitätstyps verarbeiten.

Datenaustausch zwischen Akteuren

Eingabedaten können einer Workflow-Ausführung bei ihrem Start bereitgestellt werden. Ebenso können Eingabedaten Aktivitäts-Workern bereitgestellt werden, wenn sie Aktivitätsaufgaben planen. Wenn eine Aktivitätsaufgabe abgeschlossen ist, kann der Aktivitätsarbeiter Ergebnisse an Amazon SWF zurückgeben. Ebenso kann ein Entscheider die Ergebnisse einer Workflow-Ausführung melden, wenn die Ausführung abgeschlossen ist. Jeder Akteur kann Daten über Zeichenketten, deren Form benutzerdefiniert ist, an Amazon SWF senden und Daten von Amazon SWF empfangen. Je nach Größe und Sensitivität der Daten können Sie Daten direkt übergeben oder einen Zeiger auf Daten übergeben, die auf einem anderen System oder Dienst (wie Amazon S3 oder DynamoDB) gespeichert sind. Sowohl die direkt übergebenen Daten als auch die Verweise auf andere Datenspeicher werden im Workflow-Ausführungsverlauf aufgezeichnet. Amazon SWF kopiert oder speichert jedoch keine Daten aus externen Speichern als Teil des Verlaufs.

Da Amazon SWF den vollständigen Ausführungsstatus jeder Workflow-Ausführung beibehält, einschließlich der Eingaben und Ergebnisse von Aufgaben, können alle Akteure zustandslos sein. Folglich ist die Workflow-Verarbeitung in hohem Maße skalierbar. Wenn die Auslastung Ihres Systems wächst, können Sie einfach weitere Akteure hinzufügen, um die Kapazität zu erhöhen.

Aufgaben in Amazon SWF

Amazon SWF interagiert mit Mitarbeitern und Entscheidungsträgern, indem es ihnen Arbeitsaufträge, sogenannte Aufgaben, zur Verfügung stellt. In Amazon SWF gibt es drei Arten von Aufgaben:

- **Aktivitätsaufgabe** — Eine Aktivitätsaufgabe weist einen Aktivitätsmitarbeiter an, seine Funktion auszuführen, z. B. den Lagerbestand zu überprüfen oder eine Kreditkarte zu belasten. Der Aktivitätsaufgabe enthält alle Informationen, die der Aktivitäts-Worker zum Ausüben seiner Funktion benötigt.
- **Lambda-Aufgabe** — Eine Lambda-Aufgabe ähnelt einer Aktivitätsaufgabe, führt jedoch eine Lambda-Funktion anstelle einer herkömmlichen Amazon SWF SWF-Aktivität aus. Weitere Informationen zum Definieren einer Lambda-Aufgabe finden Sie unter [AWS Lambda Aufgaben in Amazon SWF](#).
- **Entscheidungsaufgabe** — Eine Entscheidungsaufgabe teilt einem Entscheider mit, dass sich der Status der Workflow-Ausführung geändert hat, sodass der Entscheider die nächste Aktivität bestimmen kann, die ausgeführt werden muss. Die Entscheidungsaufgabe enthält den aktuellen Workflow-Verlauf.

Amazon SWF plant eine Entscheidungsaufgabe, wenn der Workflow gestartet wird und wann immer sich der Status des Workflows ändert, z. B. wenn eine Aktivitätsaufgabe abgeschlossen ist. Jede Entscheidungsaufgabe enthält eine paginierte Ansicht des gesamten Workflow-Ausführungsverlaufs. Der Entscheider analysiert den Verlauf der Workflow-Ausführung und antwortet Amazon SWF mit einer Reihe von Entscheidungen, die festlegen, was als Nächstes bei der Workflow-Ausführung geschehen soll. Im Grunde bietet jede Entscheidungsaufgabe dem Entscheider die Möglichkeit, den Arbeitsablauf zu bewerten und Amazon SWF Anweisungen zu geben.

Um sicherzustellen, dass keine widersprüchlichen Entscheidungen verarbeitet werden, weist Amazon SWF jede Entscheidungsaufgabe genau einem Entscheider zu und lässt zu, dass jeweils nur eine Entscheidungsaufgabe in einer Workflow-Ausführung aktiv ist.

Die folgende Tabelle zeigt die Beziehung zwischen den verschiedenen Konstrukten in Bezug auf Workflows und Entscheider.

Logisches Design	Registriert als	Durchgeführt von	Empfängt und führt durch	Generiert
Workflow	Workflow-Typ	Entscheider	Entscheidungs- aufgaben	Entscheidungen

Wenn ein Aktivitätsmitarbeiter die Aktivitätsaufgabe abgeschlossen hat, meldet er Amazon SWF, dass die Aufgabe abgeschlossen wurde, und enthält alle relevanten Ergebnisse, die generiert wurden. Amazon SWF aktualisiert den Verlauf der Workflow-Ausführung mit einem Ereignis, das angibt, dass die Aufgabe abgeschlossen wurde, und plant dann eine Entscheidungsaufgabe, um den aktualisierten Verlauf an den Entscheider zu übertragen.

Amazon SWF weist jede Aktivitätsaufgabe genau einem Aktivitätsarbeiter zu. Sobald die Aufgabe zugewiesen ist, kann kein anderer Aktivitäts-Worker die Aufgabe beanspruchen oder durchführen.

Die folgende Tabelle zeigt die Beziehung zwischen den verschiedenen Konstrukten in Bezug auf Aktivitäten.

Logisches Design	Registriert als	Durchgeführt von	Empfängt und führt durch	Generiert
Aktivität	Aktivitätstyp	Aktivitäts-Worker	Aufgaben	Ergebnisdaten

Aufgabenlisten in Amazon SWF

Aufgabenlisten sind eine Möglichkeit, die verschiedenen, einem Workflow zugeordneten Aufgaben zu organisieren. Aufgabenlisten sind vergleichbar mit dynamischen Warteschlangen. Wenn eine Aufgabe in Amazon SWF geplant ist, können Sie eine Warteschlange (Aufgabenliste) angeben, in die sie gestellt werden soll. Wenn Sie Amazon SWF nach einer Aufgabe abfragen, geben Sie auf ähnliche Weise an, aus welcher Warteschlange (Aufgabenliste) die Aufgabe abgerufen werden soll.

Aufgabenlisten sind ein flexibler Mechanismus zur Verteilung von Aufgaben an Worker je nach Anwendungsfall. Sie sind dynamisch, d. h. Sie müssen eine Aufgabenliste nicht registrieren oder explizit mit einer Aktion erstellen: Wenn Sie eine Aufgabe planen, wird die Aufgabenliste erstellt, falls sie noch nicht vorhanden ist.

Es gibt eigene Listen für Aktivitätsaufgaben und Entscheidungsaufgaben. Eine Aufgabe wird immer nur für eine Aufgabenliste und nicht listenübergreifend geplant. Darüber hinaus sind Aufgabenlisten wie Aktivitäten und Workflows auf eine bestimmte AWS Region und Amazon SWF-Domain beschränkt.

Themen

- [Entscheidungsaufgabenlisten](#)
- [Aktivitätsaufgabenlisten](#)
- [Aufgabenweiterleitung](#)

Entscheidungsaufgabenlisten

Jede Workflow-Ausführung ist einer bestimmten Entscheidungsaufgabenliste zugeordnet. Wenn ein Workflow-Typ registriert ist ([RegisterWorkflowType](#)Aktion), können Sie eine Standard-Aufgabenliste für Ausführungen dieses Workflow-Typs angeben. Wenn der Workflow-Starter die Workflow-Ausführung (mit der Aktion `StartWorkflowExecution`) startet, kann eine andere Aufgabenliste für diese Workflow-Ausführung angegeben werden.

Wenn ein Entscheider eine neue Entscheidungsaufgabe (mit der Aktion `PollForDecisionTask`) abrufen, gibt er die Entscheidungsaufgabenliste an, aus der die Aufgabe abgerufen werden soll. Ein einzelner Entscheider kann durch mehrfaches Aufrufen von `PollForDecisionTask` mit mehreren Workflow-Ausführungen arbeiten und in jedem Aufruf eine andere Aufgabenliste verwenden. Jede Aufgabenliste ist dabei explizit einer bestimmten Workflow-Ausführung zugeordnet. Alternativ kann der Entscheider eine einzelne Entscheidungsaufgabenliste abfragen, die Entscheidungsaufgaben für

mehrere Workflow-Ausführungen enthält. Mehrere Entscheider können mit einer einzelnen Workflow-Ausführung arbeiten, indem alle die Aufgabenliste dieser Workflow-Ausführung abfragen.

Aktivitätsaufgabenlisten

Eine einzelne Aktivitätsaufgabenliste kann Aufgaben mit unterschiedlichen Aktivitätstypen enthalten. Die Aufgaben werden der Reihe nach in der Aufgabenliste geplant. Amazon SWF gibt die Aufgaben aus der Liste in der Reihenfolge nach bestem Wissen zurück. Es kann jedoch vorkommen, dass Aufgaben in anderer Reihenfolge zurückgegeben werden.

Wenn ein Aktivitätstyp registriert ist ([RegisterActivityType](#)Aktion), können Sie eine Standardaufgabenliste für diesen Aktivitätstyp angeben. Standardmäßig werden Aktivitätsaufgaben dieses Typs auf der angegebenen Aufgabenliste geplant. Wenn der Entscheider jedoch eine Aktivitätsaufgabe plant ([ScheduleActivityTask](#)Entscheidung), kann der Entscheider optional eine andere Aufgabenliste angeben, auf der die Aufgabe geplant werden soll. Wenn der Entscheider keine Aufgabenliste angibt, wird die Standardaufgabenliste verwendet. So können Sie Aktivitätsaufgaben je nach den Attributen einer Aufgabe in unterschiedlichen Aufgabenlisten platzieren. Sie können beispielsweise alle Instances einer Aktivitätsaufgabe für einen bestimmten Kreditkartentyp in eine bestimmte Aufgabenliste einordnen.

Aufgabenweiterleitung

Wenn ein Aktivitätsmanager nach einer neuen Aufgabe ([PollForActivityTask](#)Aktion) fragt, kann er eine Aktivitätsaufgabenliste angeben, aus der er schöpfen kann. In diesem Fall akzeptiert der Aktivitäts-Worker nur Aufgaben aus dieser Liste. So können Sie sicherstellen, dass bestimmte Aufgaben nur ausgewählten Aktivitäts-Workern zugewiesen werden. Sie können beispielsweise eine Aufgabenliste für Aufgaben erstellen, für die ein leistungsfähiger Computer erforderlich ist. Nur Aktivitäts-Worker, die auf geeigneter Hardware ausgeführt werden, fragen diese Aufgabenliste ab. Ein weiteres Beispiel ist eine Aufgabenliste für eine bestimmte geografische Region. Sie können dann sicherstellen, dass nur Worker in dieser Region Aufgaben aus dieser Liste abrufen. Sie können auch eine Aufgabenliste für wichtige Bestellungen erstellen und diese Liste grundsätzlich zuerst abfragen.

Wenn Sie bestimmte Aufgaben auf diese Weise bestimmten Aktivitäts-Workern zuweisen, wird dies als Aufgabenweiterleitung bezeichnet. Die Aufgabenweiterleitung ist optional. Wenn Sie beim Planen einer Aktivitätsaufgabe keine Aufgabenliste angeben, wird die Aufgabe automatisch der Standardaufgabenliste zugewiesen.

Abschluss der Workflow-Ausführung in Amazon SWF

Sobald Sie eine Workflow-Ausführung starten, ist sie geöffnet. Eine offene Workflow-Ausführung kann mit dem Status "abgeschlossen", "abgebrochen", "fehlgeschlagen" oder "Zeitüberschreitung" geschlossen werden. Die Ausführung kann auch als neue Ausführung fortgesetzt oder beendet werden. Eine Workflow-Ausführung kann vom Entscheider, von der Person, die den Workflow verwaltet, oder von Amazon SWF geschlossen werden.

Wenn der Entscheider feststellt, dass die Aktivitäten des Workflows abgeschlossen sind, sollte er die Workflow-Ausführung mit der Aktion [RespondDecisionTaskCompleted](#) als abgeschlossen schließen und die Entscheidung [CompleteWorkflowExecution](#) übergeben.

Alternativ kann ein Entscheider die Workflow-Ausführung auch als abgebrochen oder fehlgeschlagen schließen. Um die Ausführung abzubrechen, muss der Entscheider die Aktion `RespondDecisionTaskCompleted` verwenden und die Entscheidung [CancelWorkflowExecution](#) übergeben.

Die Workflow-Ausführung sollte als fehlgeschlagen geschlossen werden, wenn sie mit einem anderen Status als dem normalen Abschluss endet. Um die Ausführung als fehlgeschlagen zu schließen, muss der Entscheider die Aktion `RespondDecisionTaskCompleted` verwenden und die Entscheidung [FailWorkflowExecution](#) übergeben.

Amazon SWF überwacht Workflow-Ausführungen, um sicherzustellen, dass sie keine benutzerdefinierten Timeout-Einstellungen überschreiten. Wenn bei der Ausführung eines Workflows ein Timeout auftritt, schließt Amazon SWF ihn automatisch. Weitere Informationen zu den Werten für die Zeitüberschreitung finden Sie im Abschnitt [Amazon SWF-Timeout-Typen](#).

Ein Entscheider kann die Ausführung auch schließen und mit dem Befehl `RespondDecisionTaskCompleted` logisch als neue Ausführung fortsetzen und die Entscheidung [ContinueAsNewWorkflowExecution](#) übergeben. Diese Strategie ist vor allem für langfristig ausgeführte Workflow-Ausführungen sinnvoll, deren Verlauf sonst zu umfangreich werden würde.

Schließlich können Sie Workflow-Ausführungen direkt von der Amazon SWF SWF-Konsole aus oder programmgesteuert mithilfe der API beenden. [TerminateWorkflowExecution](#) Durch das Beenden wird ein Schließen der Workflow-Ausführung erzwungen. Ein Abbruch ist dem Beenden vorzuziehen, da Ihre Entscheider das Schließen der Workflow-Ausführung verwalten können.

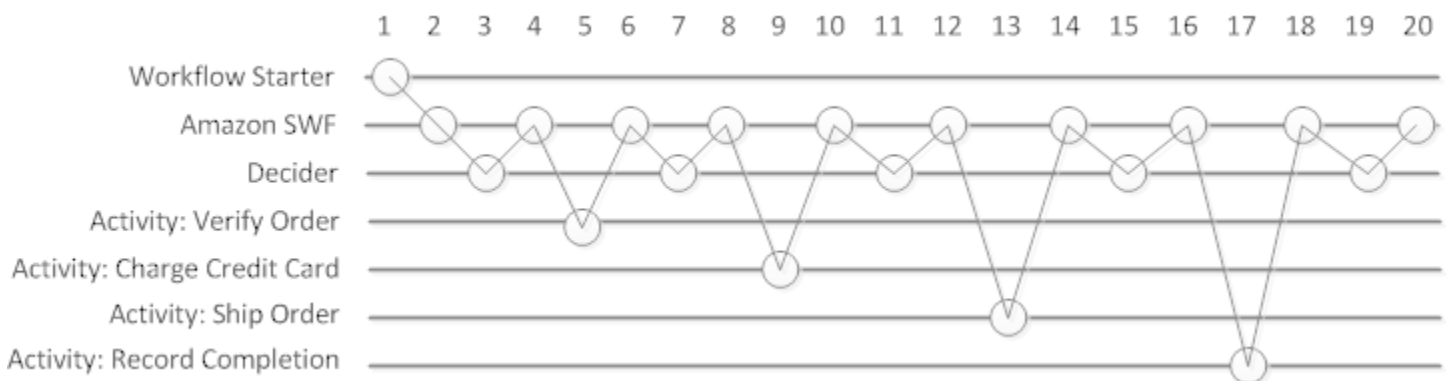
Amazon SWF beendet eine Workflow-Ausführung, wenn die Ausführung bestimmte vom Service definierte Grenzwerte überschreitet. Amazon SWF beendet einen untergeordneten Workflow, wenn

der übergeordnete Workflow beendet wurde und die entsprechende untergeordnete Richtlinie besagt, dass der untergeordnete Workflow ebenfalls beendet werden sollte.

Lebenszyklus eines Amazon SWF SWF-Workflows

Vom Beginn einer Workflow-Ausführung bis zu deren Abschluss interagiert Amazon SWF mit den Akteuren, indem es ihnen entsprechende Aufgaben zuweist, entweder Aktivitätsaufgaben oder Entscheidungsaufgaben.

Die folgende Abbildung zeigt den Lebenszyklus einer Workflow-Ausführung zur Bestellabwicklung aus der Perspektive der Komponenten, die darauf aufbauen.



Lebenszyklus einer Workflow-Ausführung

In der folgenden Tabelle werden die einzelnen Aufgaben aus der vorherigen Abbildung erläutert.

Beschreibung	Aktion, Entscheidung oder Ereignis
1. Der Workflow-Starter ruft die entsprechende Amazon SWF SWF-Aktion auf, um die Workflow-Ausführung für eine Bestellung zu starten, und stellt die Bestellinformationen bereit.	Aktion StartWorkflowExecution

Beschreibung	Aktion, Entscheidung oder Ereignis
2. Amazon SWF empfängt die Anfrage zum Starten der Workflow-Ausführung und plant dann die erste Entscheidungsaufgabe.	Ereignis WorkflowExecutionStarted und Ereignis DecisionTaskScheduled
3. Der Entscheider erhält die Aufgabe von Amazon SWF, überprüft den Verlauf, wendet die Koordinationslogik an, um festzustellen, dass keine vorherigen Aktivitäten stattgefunden haben, trifft die Entscheidung, die Aktivität „Bestellung überprüfen“ mit den Informationen zu planen, die der Aktivitätsmitarbeiter zur Bearbeitung der Aufgabe benötigt, und sendet die Entscheidung an Amazon SWF zurück.	PollForDecisionTask -Aktion. RespondDecisionTaskCompleted -Aktion und ScheduleActivityTask -Entscheidung.

Beschreibung	Aktion, Entscheidung oder Ereignis
4. Amazon SWF erhält die Entscheidung, plant die Aktivität saufgabe „Bestellung verifizieren“ und wartet, bis die Aktivität saufgabe abgeschlossen ist oder das Timeout überschritten wird.	ActivityTaskScheduled event
5. Ein Aktivität smitarbeiter, der die Aktivität „Bestellung verifizieren“ ausführen kann, empfängt die Aufgabe, führt sie aus und sendet die Ergebnisse an Amazon SWF zurück.	Aktion PollForActivityTask und Aktion RespondActivityTaskCompleted
6. Amazon SWF empfängt die Ergebnisse der Aktivität „Bestellung verifizieren“, fügt sie dem Workflow-Verlauf hinzu und plant eine Entscheidungsaufgabe.	Ereignis ActivityTaskCompleted und Ereignis DecisionTaskScheduled

Beschreibung	Aktion, Entscheidung oder Ereignis
<p>7. Der Entscheider erhält die Aufgabe von Amazon SWF, überprüft den Verlauf, wendet die Koordinationslogik an, entscheidet, eine ChargeCreditCard Aktivitätsaufgabe mit den Informationen zu planen, die der Aktivitätshelfer zur Bearbeitung der Aufgabe benötigt, und sendet die Entscheidung an Amazon SWF zurück.</p>	<p>PollForDecisionTask -Aktion. RespondDecisionTaskCompleted -Aktion mit ScheduleActivityTask -Entscheidung.</p>
<p>8. Amazon SWF empfängt die Entscheidung, plant die ChargeCreditCard Aktivitätsaufgabe und wartet, bis sie abgeschlossen ist oder das Timeout überschritten wird.</p>	<p>Ereignis DecisionTaskCompleted und Ereignis ActivityTaskScheduled</p>

Beschreibung	Aktion, Entscheidung oder Ereignis
9. Ein Aktivitätssarbeiter, der die ChargeCreditCard Aktivität ausführen kann, empfängt die Aufgabe, führt sie aus und sendet die Ergebnisse an Amazon SWF zurück.	Aktion PollForActivityTask und Aktion RespondActivityTaskCompleted
10. Amazon SWF empfängt die Ergebnisse der ChargeCreditCard Aktivitätsaufgabe, fügt sie dem Workflow-Verlauf hinzu und plant eine Entscheidungsaufgabe.	Ereignis ActivityTaskCompleted und Ereignis DecisionTaskScheduled

Beschreibung	Aktion, Entscheidung oder Ereignis
<p>11. Der Entscheider erhält die Aufgabe von Amazon SWF, überprüft den Verlauf, wendet die Koordinationslogik an, entscheidet, eine ShipOrder Aktivitätsaufgabe mit den Informationen zu planen, die der Aktivitätshelfer zur Ausführung der Aufgabe benötigt, und gibt die Entscheidung an Amazon SWF zurück.</p>	<p>PollForDecisionTask -Aktion. RespondDecisionTaskCompleted mit ScheduleActivityTask -Entscheidung.</p>
<p>12. Amazon SWF empfängt die Entscheidung, plant eine ShipOrder Aktivitätsaufgabe und wartet, bis sie abgeschlossen ist oder das Timeout überschritten wird.</p>	<p>Ereignis DecisionTaskCompleted und Ereignis ActivityTaskScheduled</p>

Beschreibung	Aktion, Entscheidung oder Ereignis
13. Ein Aktivität sarbeiter, der die ShipOrder Aktivität ausführen kann, empfängt die Aufgabe, führt sie aus und sendet die Ergebnisse an Amazon SWF zurück.	Aktion PollForActivityTask und Aktion RespondActivityTaskCompleted
14. Amazon SWF empfängt die Ergebnisse der ShipOrder Aktivität saufgabe, fügt sie dem Workflow-Verlauf hinzu und plant eine Entscheidungsaufgabe.	Ereignis ActivityTaskCompleted und Ereignis DecisionTaskScheduled

Beschreibung	Aktion, Entscheidung oder Ereignis
<p>15. Der Entscheider erhält die Aufgabe von Amazon SWF, überprüft den Verlauf, wendet die Koordinationslogik an, entscheidet, eine RecordCompletion Aktivitätsaufgabe mit den Informationen zu planen, die der Aktivitätshelfer zur Ausführung der Aufgabe benötigt, und gibt die Entscheidung an Amazon SWF zurück.</p>	<p>PollForDecisionTask -Aktion. RespondDecisionTaskCompleted -Aktion mit ScheduleActivityTask -Entscheidung.</p>
<p>16. Amazon SWF empfängt die Entscheidung, plant eine RecordCompletion Aktivitätsaufgabe und wartet, bis sie abgeschlossen ist oder das Timeout überschritten wird.</p>	<p>Ereignis DecisionTaskCompleted und Ereignis ActivityTaskScheduled</p>

Beschreibung	Aktion, Entscheidung oder Ereignis
17. Ein Aktivitätssarbeiter, der die RecordCompletion Aktivität ausführen kann, empfängt die Aufgabe, führt sie aus und sendet die Ergebnisse an Amazon SWF zurück.	Aktion PollForActivityTask und Aktion RespondActivityTaskCompleted
18. Amazon SWF empfängt die Ergebnisse der RecordCompletion Aktivitätsaufgabe, fügt sie dem Workflow-Verlauf hinzu und plant eine Entscheidungsaufgabe.	Ereignis ActivityTaskCompleted und Ereignis DecisionTaskScheduled
19. Der Entscheider erhält die Aufgabe von Amazon SWF, überprüft den Verlauf, wendet die Koordinationslogik an, entscheidet, die Workflow-Ausführung zu beenden, und sendet die Entscheidung zusammen mit allen Ergebnissen an Amazon SWF zurück.	PollForDecisionTask -Aktion. RespondDecisionTaskCompleted -Aktion mit CompleteWorkflowExecution -Entscheidung.

Beschreibung	Aktion, Entscheidung oder Ereignis
20. Amazon SWF schließt die Workflow-Ausführung ab und archiviert den Verlauf zum future Nachschlagen.	WorkflowExecutionCompleted event.

Abfragen für Aufgaben in Amazon SWF

Entscheider und Mitarbeiter von Aktivitäten kommunizieren mit Amazon SWF über lange Abfragen. Der Entscheider oder Activity Worker initiiert in regelmäßigen Abständen die Kommunikation mit Amazon SWF, benachrichtigt Amazon SWF über seine Verfügbarkeit, um eine Aufgabe anzunehmen, und gibt dann eine Aufgabenliste an, von der Aufgaben abgerufen werden sollen.

Wenn eine Aufgabe auf der angegebenen Aufgabenliste verfügbar ist, gibt Amazon SWF sie sofort in der Antwort zurück. Wenn keine Aufgabe verfügbar ist, hält Amazon SWF die TCP-Verbindung für bis zu 60 Sekunden geöffnet, sodass eine Aufgabe, falls sie während dieser Zeit verfügbar wird, über dieselbe Verbindung zurückgegeben werden kann. Wenn innerhalb von 60 Sekunden keine Aufgabe verfügbar wird, wird eine leere Antwort zurückgegeben und die Verbindung geschlossen. (Eine leere Antwort ist eine Aufgabenstruktur, in der der Wert taskToken eine leere Zeichenfolge ist.) Wenn dies geschieht, sollte der Entscheider oder Aktivitäts-Worker eine erneut Abfrage durchführen.

Langes Abfragen ist für die Aufgabenverarbeitung in großem Umfang ideal. Entscheider und Aktivitäts-Worker können ihre eigene Kapazität verwalten und die Benutzerfreundlichkeit ist gewährleistet, wenn sich Entscheider und Aktivitäts-Worker hinter einer Firewall befinden.

Weitere Informationen erhalten Sie unter [Abrufen von Entscheidungsaufgaben](#) und [Abrufen von Aktivitätsaufgaben](#).

Fortgeschrittene Workflow-Konzepte in Amazon SWF

Das E-Commerce-Beispiel im Abschnitt [???](#) stellt ein vereinfachtes Workflow-Szenario dar. In Wirklichkeit sollte Ihr Workflow mehrere Aufgaben gleichzeitig ausführen (eine Bestätigungs-E-Mail senden und gleichzeitig eine Kreditkarte autorisieren), wichtige Ereignisse aufzeichnen (alle Artikel wurden verpackt), Änderungen an der Bestellung vornehmen (Artikel hinzufügen oder entfernen) und andere, weitreichendere Entscheidungen als Teil der Workflow-Ausführung durchführen. In diesem Abschnitt werden erweiterte Workflow-Konzepte beschrieben, mit denen Sie Ihre Workflows erstellen können.

Fortgeschrittene Konzepte

- [Versionsverwaltung](#)
- [Signale](#)
- [Workflows für Kinder in Amazon SWF](#)
- [Markierungen in Amazon SWF](#)
- [Schlagworte in Amazon SWF](#)
- [Implementierung von Exclusive Choice mit Amazon SWF](#)
- [Timer in Amazon SWF](#)
- [Abbrechen von Aktivitätsaufgaben in Amazon SWF](#)

Versionsverwaltung

Geschäftliche Anforderungen machen es häufig erforderlich, dass Sie verschiedene Implementierungen oder Variationen desselben Workflows oder derselben Aktivität gleichzeitig ausführen. Beispielsweise wenn Sie eine neue Implementierung eines Workflows testen möchten, während eine andere Implementierung in der Produktion verwendet wird. Oder Sie möchten zwei verschiedene Implementierungen mit zwei verschiedenen Funktionssätzen ausführen, wie zum Beispiel eine grundlegende und eine Premium-Implementierung. Durch das Versioning sind Sie in der Lage, mehrere Implementierungen der Workflows und Aktivitäten gleichzeitig auszuführen und so Ihren speziellen Anforderungen entgegenzukommen.

Den Workflow- und Aktivitätstypen ist jeweils eine Version zugeordnet, die während der Registrierung festgelegt wird. Bei der Version handelt es sich um eine formfreie Zeichenfolge und Sie können Ihr eigenes Versioning-Schema auswählen. Um eine neue Version eines registrierten Typs zu erstellen, müssen Sie sie mit dem gleichen Namen und einer anderen Version registrieren.

[Aufgabenlisten in Amazon SWF](#) kann Ihnen, wie bereits beschrieben, bei der Implementierung des Versioning behilflich sein. Angenommen, Sie haben langlaufende Workflow-Ausführungen eines bestimmten Typs, die bereits in Bearbeitung sind, und die Umstände erfordern es, dass Sie den Workflow überarbeiten, beispielsweise, um eine neue Funktion hinzuzufügen. Sie könnten die neue Funktion anhand neuer Versionen der Aktivitätstypen und -Worker sowie eines neuen Entscheiders erstellen. Anschließend könnten Sie Ausführungen der neuen Workflow-Version unter Verwendung verschiedener Aufgabenlisten starten. Auf diese Weise könnten Sie Workflow-Ausführungen verschiedener Versionen gleichzeitig ausführen, ohne dass diese sich gegenseitig beeinträchtigen.

Signale

Signale ermöglichen es Ihnen, Informationen in eine laufende Workflow-Ausführung einzufügen. In manchen Szenarien empfiehlt es sich, einer laufenden Workflow-Ausführung Informationen hinzuzufügen, um ihr mitzuteilen, dass Änderungen aufgetreten sind oder sie über ein externes Ereignis zu informieren. Jeder Prozess kann einer offenen Workflow-Ausführung ein Signal senden. So kann beispielsweise eine Workflow-Ausführung einer anderen ein Signal senden.

Note

Der Versuch ein Signal an eine Workflow-Ausführung zu senden, die nicht offen ist, resultiert in ein Fehlschlagen von `SignalWorkflowExecution` aufgrund von `UnknownResourceFault`.

Um Signale zu verwenden, definieren Sie den Signalnamen und die Daten, die an das Signal weitergegeben werden sollen, sofern vorhanden. Programmieren Sie dann den Entscheider so, dass er das Signalereignis ([WorkflowExecutionSignaled](#)) in der Historie erkennt und entsprechend verarbeitet. Wenn ein Prozess eine Workflow-Ausführung signalisieren möchte, ruft er Amazon SWF auf (mithilfe der [SignalWorkflowExecution](#)Aktion oder, im Fall eines Entscheiders, mithilfe der [SignalExternalWorkflowExecution](#)Entscheidung), der die Kennung für die Ziel-Workflow-Ausführung, den Signalnamen und die Signaldaten angibt. Amazon SWF empfängt dann das Signal, zeichnet es im Verlauf der Ausführung des Ziel-Workflows auf und plant eine Entscheidungsaufgabe dafür. Wenn der Entscheider die Entscheidungsaufgabe empfängt, empfängt er auch das Signal im Workflow-Ausführungsverlauf. Daraufhin kann der Entscheider basierend auf dem Signal und dessen Daten die entsprechenden Aktionen einleiten.

Manchmal empfiehlt es sich, auf ein Signal zu warten. Beispielsweise könnte ein Benutzer eine Bestellung stornieren, indem er ein Signal sendet, jedoch nur innerhalb einer Stunde nach der

Bestellung. Amazon SWF verfügt nicht über ein Primitiv, das es einem Entscheider ermöglicht, auf ein Signal vom Service zu warten. Die Pausenfunktionalität muss vom Entscheider selbst implementiert werden. Um Pausieren zu können, muss der Entscheider mithilfe der `StartTimer`-Entscheidung einen Timer starten, der die Dauer festlegt, die der Entscheider auf das Signal wartet, während er gleichzeitig Entscheidungsaufgaben abfragt. Wenn der Entscheider eine Entscheidungsaufgabe empfängt, sollte er den Verlauf überprüfen, um zu sehen, ob das Signal empfangen oder der Timer aktiviert wurde. Wurde das Signal empfangen, sollte der Entscheider den Timer deaktivieren. Wird der Timer jedoch aktiviert, heißt das, dass das Signal nicht innerhalb des festgelegten Zeitraums ankam. Befolgen Sie zusammenfassend die folgenden Schritte, um auf ein bestimmtes Signal zu warten.

1. Erstellen Sie einen Timer für die Zeitspanne, die der Entscheider warten soll.
2. Wenn eine Entscheidungsaufgabe empfangen wurde, sollten Sie den Verlauf überprüfen, um zu sehen, ob das Signal eingetroffen ist oder der Timer aktiviert wurde.
3. Ist das Signal eingetroffen, deaktivieren Sie den Timer anhand einer `CancelTimer`-Entscheidung und bearbeiten Sie das Signal. Abhängig vom Zeitpunkt kann der Verlauf sowohl `TimerFired`- als auch `WorkflowExecutionSignaled`-Ereignisse umfassen. In diesen Fällen können Sie anhand der relativen Reihenfolge der Ereignisse im Verlauf feststellen, welches Ereignis zuerst eintrat.
4. Wenn der Timer vor dem Empfang eines Signals aktiviert wurde, bedeutet das, dass der Zeitraum überschritten wurde, die der Entscheider auf das Signal wartet. Sie können die Ausführung fehlschlagen lassen oder entsprechend Ihrem Anwendungsfall eine andere Logik anwenden.

In Fällen, in denen ein Workflow storniert werden sollte — zum Beispiel, wenn die Bestellung selbst vom Kunden storniert wurde —, sollte die `RequestCancelWorkflowExecution` Aktion verwendet werden, anstatt ein Signal an den Workflow zu senden.

Nachfolgend finden Sie einige Anwendungsbeispiele für Signale:

- Pausieren der Workflow-Ausführungen bis ein Signal empfangen wird (z. B. auf eine Bestandslieferung warten).
- Bereitstellen von Informationen für eine Workflow-Ausführung, die die Logik beeinflusst, mit der der Entscheider Entscheidungen tätigt. Dies empfiehlt sich für Workflows, die von externen Ereignissen beeinflusst werden (z. B. der Verkauf von Aktien nachdem der Markt geschlossen ist).
- Aktualisieren der Workflow-Ausführung, wenn Sie Änderungen erwarten (z. B. Ändern der Bestellmengen nachdem eine Bestellung aufgegeben wurde und bevor sie versendet wird).

Im folgenden Beispiel wird bei der Ausführung des Workflows ein Signal zur Stornierung einer Bestellung gesendet.

```
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"}
```

Wenn die Workflow-Ausführung das Signal empfängt, gibt Amazon SWF eine erfolgreiche HTTP-Antwort zurück, die der folgenden ähnelt. Amazon SWF generiert eine Entscheidungsaufgabe, um den Entscheider darüber zu informieren, das Signal zu verarbeiten.

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

Workflows für Kinder in Amazon SWF

Komplizierte Workflows können durch die Verwendung von untergeordneten Workflows in kleinere, überschaubarere und möglicherweise wiederverwendbare Komponenten aufgeteilt werden. Bei einem untergeordneten Workflow handelt es sich um eine Workflow-Ausführung, die von einer anderen (übergeordneten) Workflow-Ausführung initiiert wird. Um einen untergeordneten Workflow zu initiieren, verwendet der übergeordnete Workflow die `StartChildWorkflowExecution`-Entscheidung. Eingabedaten, die mit dieser Entscheidung festgelegt wurden, werden dem untergeordneten Workflow während des gesamten Verlaufs verfügbar gemacht.

Die Attribute für die `StartChildWorkflowExecution` Entscheidung geben auch die untergeordnete Richtlinie an, d. h., wie Amazon SWF mit der Situation umgehen soll, in der die Ausführung des übergeordneten Workflows vor der Ausführung des untergeordneten Workflows beendet wird. Hierfür gibt es drei mögliche Werte:

- **BEENDEN**: Amazon SWF beendet die untergeordneten Ausführungen.
- **REQUEST_CANCEL**: Amazon SWF versucht, die untergeordnete Ausführung abubrechen, indem ein `WorkflowExecutionCancelRequested` Ereignis in den Workflow-Ausführungsverlauf des Kindes aufgenommen wird.

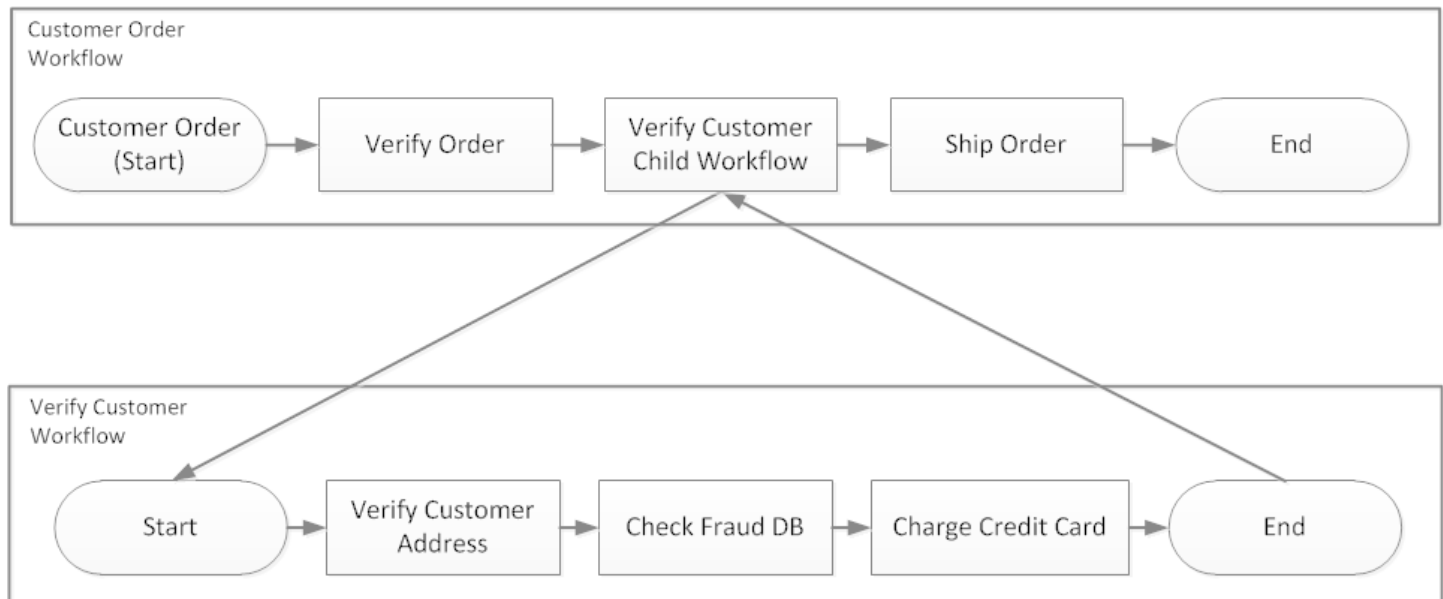
- **AUFGEBEN:** Amazon SWF ergreift keine Maßnahmen; die untergeordneten Hinrichtungen werden weiterhin ausgeführt.

Nachdem die untergeordnete Ausführung gestartet wurde, wird sie wie eine reguläre Ausführung durchgeführt. Nach Abschluss zeichnet Amazon SWF den Abschluss zusammen mit den Ergebnissen im Workflow-Verlauf der übergeordneten Workflow-Ausführung auf. Nachfolgend finden Sie Beispiele untergeordneter Workflows:

- Untergeordneter Workflow zur Kreditkartenabwicklung, der von Workflows auf verschiedenen Websites verwendet wird
- Untergeordneter E-Mail-Workflow, der die E-Mail-Adresse des Kunden bestätigt, die Abmeldeliste überprüft, die E-Mail sendet und sicherstellt, dass sie zugestellt wird bzw. nicht fehlschlägt.
- Untergeordneter Workflow zur Datenbankspeicherung und Wiederherstellung, der Verbindung, Einrichtung, Transaktion und Verifizierung kombiniert.
- Untergeordneter Workflow zur Kompilierung des Quellcodes, der Erstellung, Paketerstellung und Verifizierung kombiniert.

Im E-Commerce-Beispiel empfiehlt es sich, die Aktivität zur Belastung der Kreditkarte als einen untergeordneten Workflow einzurichten. Registrieren Sie dazu einen neuen Workflow zur Überprüfung des Kunden, registrieren Sie die Aktivitäten zur Verifizierung der Kundenadresse und zum Überprüfen der Datenbank auf Betrug und definieren Sie die Koordinationslogik für die Aufgabe. Anschließend kann ein Entscheider im Kundenauftrag-Workflow einen untergeordneten Workflow zum Überprüfen des Kunden initiieren, indem die `StartChildWorkflowExecution`-Entscheidung geplant wird, die diesen Workflow-Typ festlegt.

Die folgende Abbildung stellt einen Kundenauftrag-Workflow dar, der einen untergeordneten Workflow zum Überprüfen des Kunden einschließt, über den die Kundenadresse und die Betrugsdatenbank überprüft und die Kreditkarte belastet werden.



Mehrere Workflows können unter Verwendung des gleichen Workflow-Typs untergeordnete Workflow-Ausführungen erstellen. So könnte beispielsweise der untergeordnete Workflow zur Überprüfung des Kunden auch in anderen Bereichen der Organisation verwendet werden. Die Ereignisse für einen untergeordneten Workflow sind in ihrem eigenen Workflow-Verlauf und nicht im übergeordneten Workflow-Verlauf eingeschlossen.

Da es sich bei untergeordneten Workflows um einfache Workflow-Ausführungen handelt, die von einem Entscheider initiiert werden, können sie auch als normale eigenständige Workflow-Ausführungen gestartet werden.

Markierungen in Amazon SWF

Manchmal empfiehlt es sich, die Informationen im Workflow-Verlauf einer Workflow-Ausführung aufzuzeichnen, die speziell auf Ihren Anwendungsfall zutreffen. Markierungen ermöglichen es Ihnen, Informationen im Verlauf der Workflow-Ausführung aufzuzeichnen, die Sie für alle benutzerdefinierten oder szenariospezifischen Zwecke verwenden können.

Um Markierungen zu verwenden, verwendet ein Entscheider die `RecordMarker` Entscheidung, benennt die Markierung, hängt die gewünschten Daten an die Entscheidung an und benachrichtigt Amazon SWF über die Aktion. `RespondDecisionTaskCompleted` Amazon SWF empfängt die Anfrage, zeichnet die Markierung im Workflow-Verlauf auf und trifft alle anderen Entscheidungen in der Anfrage. Ab diesem Zeitpunkt kann der Entscheider die Markierung im Workflow-Verlauf sehen und auf alle die von Ihnen programmierten Weisen verwenden.

Durch das Aufzeichnen einer Markierung selbst wird noch keine Entscheidungsaufgabe initiiert. Damit die Workflow-Ausführung nicht hängen bleibt, muss etwas passieren, um die Ausführung des Workflows fortzusetzen. Das kann beispielsweise sein, dass der Entscheider eine weitere Aktivitätsaufgabe plant, dass die Workflow-Ausführung ein Signal empfängt oder eine zuvor geplante Aktivitätsaufgabe abgeschlossen wird.

Nachfolgend finden Sie Beispiele von Markierungen:

- Ein Zähler, der die Anzahl der Loops in einem rekursiven Workflow angibt.
- Fortschritt der Workflow-Ausführung basierend auf den Ergebnissen der Aktivitäten.
- Zusammenfassung der Informationen aus früheren Ereignissen des Workflow-Verlaufs.

Im E-Commerce-Beispiel ist es eventuell sinnvoll eine Aktivität hinzuzufügen, die den Bestand jeden Tag überprüft und die Zählung jedes Mal in einer Markierung inkrementiert. Anschließend könnten Sie Entscheidungslogik hinzufügen, die dem Kunden eine E-Mail sendet oder einen Manager benachrichtigt, wenn die Anzahl "Fünf" überschritten wird – ohne dass der gesamte Verlauf überprüft werden muss.

Im folgenden Beispiel schließt der Entscheider eine Entscheidungsaufgabe ab und antwortet mit der Aktion `RespondDecisionTaskCompleted`, die ein Entscheidungs `RecordMarker` enthält.

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

Wenn Amazon SWF die Markierung erfolgreich aufzeichnet, gibt es eine erfolgreiche HTTP-Antwort zurück, die der folgenden ähnelt.

```
HTTP/1.1 200 OK
Content-Length: 0
```

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

Schlagworte in Amazon SWF

Amazon SWF unterstützt das Markieren einer Workflow-Ausführung. Dies ist vor allem nützlich, wenn Sie viele Ressourcen haben.

Amazon SWF unterstützt das Markieren einer Workflow-Ausführung mit bis zu fünf Tags. Bei jedem Tag handelt es sich um eine formfreie Zeichenfolge, die bis zu 256 Zeichen lang sein kann. Wenn Sie Tags verwenden möchten, müssen Sie sie beim Start der Workflow-Ausführung zuordnen. Nachdem die Workflow-Ausführung gestartet wurde, können Sie einem Workflow keine Tags mehr hinzufügen. Darüber hinaus können Sie die Tags, die einer Workflow-Ausführung zugeordnet wurden, auch nicht mehr bearbeiten oder entfernen.

IAM unterstützt die Steuerung des Zugriffs auf Amazon SWF-Domains anhand von Tags. Um den Zugriff auf Grundlage von Tags zu steuern, geben Sie Informationen über die Tags im Bedingungelement einer IAM-Richtlinie an.

Verwalten von Tags

Verwalten Sie Amazon Simple Workflow Service-Tags mithilfe der AWS SDKs oder, indem Sie direkt mit der Amazon SWF SWF-API interagieren. Mit der API können Sie beim Registrieren einer Domäne Tags hinzufügen, Tags für eine vorhandene Domäne auflisten und Tags für eine vorhandene Domäne hinzufügen oder löschen.

Note

Es gilt eine Beschränkung von 50 Tags pro Ressource. Siehe [Allgemeine Kontokontingente für Amazon SWF](#)

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

Weitere Informationen finden Sie [Arbeiten mit Amazon SWF APIs](#) unter und [Amazon Simple Workflow Service API-Referenz](#).

Taggen Sie Workflow-Ausführungen

Mit Amazon SWF können Sie Workflow-Ausführungen Tags zuordnen und dann anhand dieser Tags Workflow-Ausführungen abfragen. Sie können die Listen filtern, wenn Sie die Sichtbarkeitsoperationen verwenden. Wenn Sie die Tags, die Sie einer Ausführung zuweisen, sorgfältig auswählen, können Sie sie verwenden, um aussagekräftige Auflistungen zu erstellen.

Angenommen, Sie betreiben mehrere Fulfillment-Center. Mithilfe von Stichwörtern können Sie die Prozesse auflisten, die in einem bestimmten Versandzentrum ablaufen. Oder, wenn ein Kunde verschiedene Arten von Mediendateien konvertiert, könnten Tags auf unterschiedliche Prozesse bei der Konvertierung von Video-, Audio- und Bilddateien hinweisen.

Sie können bis zu fünf Tags mit einer Workflow-Ausführung verknüpfen, wenn Sie die Ausführung mit der `StartWorkflowExecution`-Aktion, der `StartChildWorkflowExecution`-Entscheidung oder der `ContinueAsNewWorkflowExecution`-Entscheidung starten. Wenn Sie Sichtbarkeitsaktionen verwenden, um Workflow-Ausführungen aufzulisten oder zu zählen, können Sie die Ergebnisse anhand Ihrer Tags filtern.

So setzen Sie Tags ein

1. Entwerfen Sie eine Tag-Strategie. Berücksichtigen Sie dabei Ihre Geschäftsanforderungen und erstellen Sie eine Liste mit für Sie wichtigen Tags. Legen Sie fest, welche Ausführungen welche Tags erhalten sollen. In Ihrer Bibliothek kann eine beliebige Anzahl an Tags enthalten sein, auch wenn pro Ausführung nur maximal fünf Tags zugewiesen werden können. Da jedes Tag ein beliebiger Zeichenfolgenwert von bis zu 256 Zeichen sein kann, kann mittels eines Tags nahezu jedes Geschäftskonzept beschrieben werden.
2. Taggen Sie eine Ausführung mit bis zu fünf Tags, wenn Sie diese generieren.
3. Listen Sie die Ausführungen auf oder zählen Sie solche, die mit einem bestimmten Tag gekennzeichnet sind, indem Sie den `tagFilter`-Parameter mit den Aktionen `ListOpenWorkflowExecutions`, `ListClosedWorkflowExecutions`, `CountOpenWorkflowExecutions` und `CountClosedWorkflowExecutions` angeben. Die Aktion filtert die Ausführung basierend auf den angegebenen Tags.

Wenn Sie einen Tag mit einer Workflow-Ausführung verknüpfen, ist die Verknüpfung permanent und kann nicht entfernt werden.

Sie können nur einen Tag im `tagFilter`-Parameter mit `ListWorkflowExecutions` angeben. Darüber hinaus wird bei der Tag-Übereinstimmung die Groß- und Kleinschreibung beachtet. Nur exakte Übereinstimmungen geben ein Ergebnis zurück.

Angenommen, Sie haben bereits zwei Ausführungen eingerichtet, die wie folgt mit Tags versehen sind.

Ausführungsname	Zugewiesene Tags
Execution-One	Consumer, 2011-February
Execution-Two	Wholesale, 2011-March

Sie können die Ausführungsliste, die von `ListOpenWorkflowExecutions` zurückgegeben wird, nach dem Consumer-Tag filtern. Die Werte `oldestDate` und `latestDate` sind als [Unix-Zeit](#)-Werte angegeben.

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

Steuern Sie den Zugriff auf Domänen mit Tags

Sie können den Zugriff auf Amazon Simple Workflow Service-Domains steuern, indem Sie in IAM auf Tags verweisen, die Amazon SWF-Domänen zugeordnet sind.

Sie könnten beispielsweise Amazon SWF-Domains, die ein Tag mit dem Schlüssel `environment` und dem Wert `production` enthalten, `production` mit der folgenden Bedingung einschränken:

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

}

Weitere Informationen finden Sie unter:

- [Zugriffssteuerung mit IAM-Tags](#)
- [Tagbasierte Richtlinien](#)

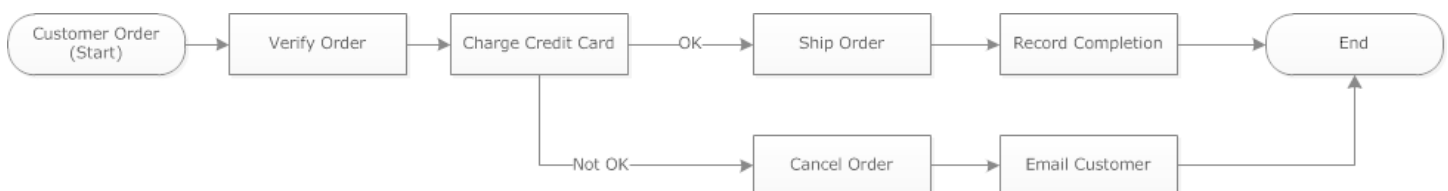
Implementierung von Exclusive Choice mit Amazon SWF

In einigen Szenarios kann es sinnvoll sein, abhängig vom Ergebnis einer vorherigen Aktivität unterschiedliche Folgeaktivitäten zu planen. Mit dem exklusiven Auswahlmuster können Sie flexible Workflows erstellen, die den komplexen Anforderungen Ihrer Anwendung gerecht werden.

Amazon SWF verfügt nicht über eine spezielle exklusive Auswahlaktion. Um Exclusive Choice zu implementieren, müssen Sie Ihre Entscheidungslogik so schreiben, dass Entscheidungen auf der Grundlage der Ergebnisse einer früheren Aktivität getroffen werden. Nachfolgend finden Sie einige Anwendungsbeispiele für die exklusive Wahl:

- Durchführen von Bereinigungsaktivitäten, wenn eine vorherige Aktivität nicht erfolgreich abgeschlossen wurde
- Planen unterschiedlicher Aktivitäten abhängig davon, ob der Kunde einen Basis- oder Premiumplan gekauft hat
- Ausführen von unterschiedlichen Benutzerauthentifizierungsaktivitäten abhängig vom Bestellverlauf des Kunden

In dem E-Commerce-Beispiel können Sie mit der exklusiven Wahl eine Bestellung abhängig vom Ergebnis der Kreditkartenbelastung eine Bestellung entweder versenden oder stornieren. In der folgenden Abbildung plant der Entscheider die Aktivitätsaufgaben "Bestellung versenden" und "Vorgang schließen", wenn die Kreditkarte erfolgreich belastet werden konnte. Andernfalls plant er die Aktivitätsaufgaben "Bestellung stornieren" und "E-Mail an Kunde".



Der Entscheider plant die Aktivität `ShipOrder`, wenn die Kreditkarte erfolgreich belastet werden konnte. Andernfalls plant er die Aktivität `CancelOrder`.

Programmieren Sie den Entscheider in diesem Fall so, dass er den Verlauf ausliest und entscheidet, ob die Kreditkarte erfolgreich belastet wurde. Hierfür können Sie eine Logik wie die folgende verwenden.

```
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
```

Wenn die Kreditkarte erfolgreich belastet wurde, sollte der Entscheider mit `RespondDecisionTaskCompleted` antworten, um die Aktivität `ShipOrder` zu planen.

```
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"
      }
    }
  ]
}
```

Wenn die Kreditkarte nicht erfolgreich belastet wurde, sollte der Entscheider mit `RespondDecisionTaskCompleted` antworten, um die Aktivität `CancelOrder` zu planen.

```
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"
    }
}
]
```

Wenn Amazon SWF in der Lage ist, die Daten in der RespondDecisionTaskCompleted Aktion zu validieren, gibt Amazon SWF eine erfolgreiche HTTP-Antwort zurück, die der folgenden ähnelt.

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

Timer in Amazon SWF

Mit einem Timer können Sie Ihren Entscheider benachrichtigen, wenn eine bestimmte Zeit abgelaufen ist.

Als Reaktion auf eine Entscheidungsaufgabe hat der Entscheider die Möglichkeit, mit einer StartTimer-Entscheidung zu antworten. Diese Entscheidung gibt die Zeitdauer an, nach der der Timer ausgelöst werden soll. Nach Ablauf der angegebenen Zeit fügt Amazon SWF dem Workflow-Ausführungsverlauf ein TimerFired Ereignis hinzu und plant eine Entscheidungsaufgabe. Anhand dieser Informationen kann der Entscheider dann weitere Entscheidungen zugrunde legen. Ein häufiges Anwendungsszenario für einen Timer ist, die Ausführung einer Aktivitätsaufgabe zu verzögern. So könnte ein Kunde beispielsweise wünschen, dass die Auslieferung eines Artikels verzögert wird.

Abbrechen von Aktivitätsaufgaben in Amazon SWF

Durch das Stornieren von Aktivitätsaufgaben wird der Entscheider darüber informiert, dass Aktivitäten, die nicht mehr ausgeführt werden müssen, beendet werden. Amazon SWF verwendet

einen kooperativen Stornierungsmechanismus und unterbricht laufende Aktivitätsaufgaben nicht gewaltsam. Sie müssen Ihre Aktivitäts-Worker so programmieren, dass sie Abbruchanforderungen abhandeln können.

Der Entscheider kann eine Aktivitätsaufgabe abbrechen, während diese eine Entscheidungsaufgabe verarbeitet. Der Entscheider beendet eine Aktivitätsaufgabe mit der Aktion `RespondDecisionTaskCompleted` mit der Entscheidung `RequestCancelActivityTask`.

Wenn die Aktivitätsaufgabe vom Aktivitäts-Worker noch nicht übernommen wurde, bricht der Service die Aufgabe ab. Beachten Sie, dass möglicherweise eine Race Condition (Wettlaufsituation) vorliegt, da ein Aktivitäts-Worker die Aufgabe jederzeit übernehmen könnte. Wenn die Aufgabe bereits einem Aktivitäts-Worker zugewiesen wurde, wird eine Anforderung zum Abbrechen der Aufgabe an den Aktivitäts-Worker gesendet.

In diesem Beispiel wurde der Workflow-Ausführung ein Signal zum Stornieren einer Bestellung gesendet.

```
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"}
```

Wenn die Workflow-Ausführung das Signal empfängt, gibt Amazon SWF eine erfolgreiche HTTP-Antwort zurück, die der folgenden ähnelt. Amazon SWF generiert eine Entscheidungsaufgabe, um den Entscheider darüber zu informieren, das Signal zu verarbeiten.

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

Wenn der Entscheider die Entscheidungsaufgabe verarbeitet und das Signal im Verlauf findet, versucht er, die offene Aktivität mit der Aktivitäts-ID `ShipOrderActivity0001` abzuberechnen. Die Aktivitäts-ID wird im Workflow-Verlauf durch die Aktivitätsplanung bereitgestellt.

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

```
RespondDecisionTaskCompleted
{
  "taskToken": "12342e17-80f6-FAKE-TASK-TOKEN32f0223",
  "decisions": [{
    "decisionType": "RequestCancelActivityTask",
    "RequestCancelActivityTaskDecisionAttributes": {
      "ActivityID": "ShipOrderActivity0001"
    }
  ]
}
```

Wenn Amazon SWF die Stornierungsanfrage erfolgreich empfängt, gibt es eine erfolgreiche HTTP-Antwort zurück, die der folgenden ähnelt:

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

Der Abbruchversuch wird im Verlauf als `ActivityTaskCancelRequested`-Ereignis protokolliert.

Wenn die Aufgabe erfolgreich storniert wurde — was durch ein `ActivityTaskCanceled` Ereignis angezeigt wird —, programmieren Sie Ihren Entscheider so, dass er nach dem Abbrechen der Aufgabe die entsprechenden Schritte ergreift, z. B. die Workflow-Ausführung zu beenden.

Wenn die Aktivitätsaufgabe nicht abgebrochen werden konnte, z. B. wenn die Aufgabe abgeschlossen ist, fehlschlägt oder ein Timeout eintritt, anstatt sie abubrechen, sollte Ihr Entscheider die Ergebnisse der Aktivität akzeptieren oder alle aufgrund Ihres Anwendungsfalls erforderlichen Säuberungs- oder Abhilfemaßnahmen durchführen.

Wenn die Aktivitätsaufgabe bereits einem Aktivitäts-Worker zugewiesen wurde, wird die Abbruchanforderung über den Task-Heartbeat-Mechanismus übermittelt. Mithilfe `RecordActivityTaskHeartbeat` dieser Aktivität können Mitarbeiter Amazon SWF regelmäßig melden, dass die Aufgabe noch in Bearbeitung ist.

Beachten Sie, dass Aktivitäts-Worker keinen Heartbeat senden müssen, auch wenn dies für langfristige Aufgaben empfohlen wird. Für den Aufgabenabbruch sind regelmäßige Heartbeats erforderlich. Wenn der Worker keinen Heartbeat sendet, kann die Aufgabe nicht abgebrochen werden.

Wenn der Entscheider den Abbruch der Aufgabe anfordert, setzt Amazon SWF den Wert des `cancelRequest` Objekts auf `true`. Das Objekt `cancelRequest` gehört zum Objekt `ActivityTaskStatus`, das vom Service als Antwort auf `RecordActivityTaskHeartbeat` zurückgegeben wird.

Amazon SWF verhindert nicht den erfolgreichen Abschluss einer Aktivitätsaufgabe, deren Stornierung beantragt wurde. Es liegt an der Aktivität, zu entscheiden, wie mit der Stornierungsanfrage umgegangen werden soll. Programmieren Sie abhängig von Ihren Anforderungen den Aktivitäts-Worker so, dass die Aktivitätsaufgabe entweder abgebrochen oder Abbruchanforderung ignoriert wird.

Wenn Sie möchten, dass der Aktivitäts-Worker bestätigt, dass die Aktivitätsaufgabe abgebrochen wurde, programmieren Sie ihn so, dass die Antwort `RespondActivityTaskCanceled` gesendet wird. Wenn Sie möchten, dass der Aktivitäts-Worker die Aufgabe abschließt, programmieren Sie ihn so, dass die Standardantwort `RespondActivityTaskCompleted` gesendet wird.

Wenn Amazon SWF die `RespondActivityTaskCompleted` oder `RespondActivityTaskCanceled` Antwort erhält, aktualisiert es den Verlauf der Workflow-Ausführung und plant eine Entscheidungsaufgabe, um den Entscheider zu informieren.

Programmieren Sie den Entscheider so, dass die Entscheidungsaufgabe verarbeitet wird und alle zusätzlichen Entscheidungen zurückgegeben werden. Wenn die Aktivitätsaufgabe erfolgreich abgebrochen wurde, programmieren Sie den Entscheider so, dass die für das Fortführen oder Schließen der Workflow-Ausführung erforderlichen Aufgaben ausgeführt werden. Wenn die Aktivitätsaufgabe nicht erfolgreich abgebrochen wurde, programmieren Sie den Entscheider so, dass die Ergebnisse akzeptiert oder ignoriert werden oder eine Bereinigung geplant wird.

Sicherheit im Amazon Simple Workflow Service

Dieser Abschnitt enthält Informationen zur Sicherheit und Authentifizierung von Amazon Simple Workflow Service.

Themen

- [Datenschutz bei Amazon Simple Workflow Service](#)
- [Identity and Access Management in Amazon Simple Workflow Service](#)
- [Protokollieren und Überwachen](#)
- [Konformitätsprüfung für Amazon Simple Workflow Service](#)
- [Resilienz im Amazon Simple Workflow Service](#)
- [Infrastruktursicherheit in Amazon Simple Workflow Service](#)
- [Konfiguration und Schwachstellenanalyse in Amazon Simple Workflow Service](#)

Amazon SWF verwendet IAM, um den Zugriff auf andere AWS Dienste und Ressourcen zu kontrollieren. Eine Übersicht der Funktionsweise von IAM finden Sie unter [Übersicht über die Zugriffsverwaltung](#) im IAM-Benutzerhandbuch. Eine Übersicht der Sicherheitsanmeldeinformationen finden Sie unter [AWS -Sicherheitsanmeldeinformationen](#) in der Allgemeine Amazon Web Services-Referenz.

Datenschutz bei Amazon Simple Workflow Service

Das AWS [Modell](#) der gilt für den Datenschutz in Amazon Simple Workflow Service. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der alle Systeme laufen AWS Cloud. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter [Häufig gestellte Fragen zum Datenschutz](#). Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag [AWS -Modell der geteilten Verantwortung und in der DSGVO](#) im AWS -Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Wird verwendet SSL/TLS , um mit AWS Ressourcen zu kommunizieren. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein AWS CloudTrail. Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter [Arbeiten mit CloudTrail Pfaden](#) im AWS CloudTrail Benutzerhandbuch.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter [Federal Information Processing Standard \(FIPS\) 140-3](#).

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit Amazon SWF oder anderen AWS-Services über die Konsole AWS CLI, API oder arbeiten AWS SDKs. Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Verschlüsselung im Amazon Simple Workflow Service

Verschlüsselung im Ruhezustand

Amazon SWF verschlüsselt Ihre Daten im Ruhezustand immer. Daten in Amazon Simple Workflow Service werden im Ruhezustand mit transparenter serverseitiger Verschlüsselung verschlüsselt. Dieser Service reduziert den Ausführungsaufwand und die Komplexität, die mit dem Schutz sensibler Daten verbunden sind. Dank der Verschlüsselung von Daten im Ruhezustand können Sie sicherheitsrelevante Anwendungen erstellen, die die Verschlüsselungs-Compliance und gesetzliche Bestimmungen einhalten.

Verschlüsselung während der Übertragung

Alle Daten, die zwischen Amazon SWF und anderen Diensten übertragen werden, werden mit Transport Layer Security (TLS) verschlüsselt.

Identity and Access Management in Amazon Simple Workflow Service

Für den Zugriff auf Amazon SWF sind Anmeldeinformationen erforderlich, mit denen Sie Ihre Anfragen authentifizieren AWS können. Diese Anmeldeinformationen müssen über Berechtigungen für den Zugriff auf AWS Ressourcen verfügen, z. B. für das Abrufen von Ereignisdaten aus anderen AWS Ressourcen. In den folgenden Abschnitten erfahren Sie, wie Sie [AWS Identity and Access Management \(IAM\)](#) und Amazon SWF verwenden können, um Ihre Ressourcen zu schützen, indem Sie den Zugriff darauf kontrollieren.

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service, den Zugriff auf Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Amazon SWF SWF-Ressourcen zu verwenden. IAM ist ein Programm AWS-Service, das Sie ohne zusätzliche Kosten nutzen können.

Themen

- [Zielgruppe](#)
- [Authentifizierung mit Identitäten](#)
- [Verwalten des Zugriffs mit Richtlinien](#)
- [Zugriffskontrolle](#)
- [Richtlinienmaßnahmen für Amazon SWF](#)
- [Richtlinienressourcen für Amazon SWF](#)
- [Schlüssel für Richtlinienbedingungen für Amazon SWF](#)
- [ACLs bei Amazon SWF](#)
- [ABAC mit Amazon SWF](#)
- [Temporäre Anmeldeinformationen mit Amazon SWF verwenden](#)
- [Serviceübergreifende Prinzipalberechtigungen für Amazon SWF](#)
- [Servicerollen für Amazon SWF](#)

- [Servicebezogene Rollen für Amazon SWF](#)
- [Identitätsbasierte Richtlinien für Amazon SWF](#)
- [Ressourcenbasierte Richtlinien in Amazon SWF](#)
- [So funktioniert Amazon Simple Workflow Service mit IAM](#)
- [Beispiele für identitätsbasierte Richtlinien für Amazon Simple Workflow Service](#)
- [Grundlegende Prinzipien](#)
- [Amazon SWF IAM-Richtlinien](#)
- [Übersicht über API-Befehle](#)
- [Tagbasierte Richtlinien](#)
- [Amazon VPC-Endpunkte für Amazon SWF](#)
- [Fehlerbehebung bei Identität und Zugriff auf Amazon Simple Workflow Service](#)

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von Ihrer Rolle ab:

- Servicebenutzer – Fordern Sie von Ihrem Administrator Berechtigungen an, wenn Sie nicht auf Features zugreifen können (siehe [Fehlerbehebung bei Identität und Zugriff auf Amazon Simple Workflow Service](#)).
- Serviceadministrator – Bestimmen Sie den Benutzerzugriff und stellen Sie Berechtigungsanfragen (siehe [So funktioniert Amazon Simple Workflow Service mit IAM](#)).
- IAM-Administrator – Schreiben Sie Richtlinien zur Zugriffsverwaltung (siehe [Beispiele für identitätsbasierte Richtlinien für Amazon Simple Workflow Service](#)).

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen sich als IAM-Benutzer authentifizieren oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich als föderierte Identität anmelden, indem Sie Anmeldeinformationen aus einer Identitätsquelle wie AWS IAM Identity Center (IAM Identity Center), Single Sign-On-Authentifizierung oder Anmeldeinformationen verwenden. Google/Facebook Weitere Informationen zum Anmelden

finden Sie unter [So melden Sie sich bei Ihrem AWS-Konto an](#) im Benutzerhandbuch für AWS-Anmeldung .

AWS Bietet für den programmatischen Zugriff ein SDK und eine CLI zum kryptografischen Signieren von Anfragen. Weitere Informationen finden Sie unter [AWS Signature Version 4 for API requests](#) im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie ein erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, dem sogenannten AWS-Konto Root-Benutzer, der vollständigen Zugriff auf alle AWS-Services Ressourcen hat. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Eine Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter [Tasks that require root user credentials](#) im IAM-Benutzerhandbuch.

Verbundidentität

Es hat sich bewährt, dass menschliche Benutzer für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen einen Verbund mit einem Identitätsanbieter verwenden müssen.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensverzeichnis, Ihrem Directory Service Web-Identitätsanbieter oder der AWS-Services mithilfe von Anmeldeinformationen aus einer Identitätsquelle zugreift. Verbundene Identitäten übernehmen Rollen, die temporäre Anmeldeinformationen bereitstellen.

Für die zentrale Zugriffsverwaltung empfehlen wir AWS IAM Identity Center. Weitere Informationen finden Sie unter [Was ist IAM Identity Center?](#) im AWS IAM Identity Center -Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein [IAM-Benutzer](#) ist eine Identität mit bestimmten Berechtigungen für eine einzelne Person oder Anwendung. Wir empfehlen die Verwendung temporärer Anmeldeinformationen anstelle von IAM-Benutzern mit langfristigen Anmeldeinformationen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter [Erfordern, dass menschliche Benutzer den Verbund mit einem Identitätsanbieter verwenden müssen, um AWS mithilfe temporärer Anmeldeinformationen darauf zugreifen zu können](#).

Eine [IAM-Gruppe](#) spezifiziert eine Sammlung von IAM-Benutzern und erleichtert die Verwaltung von Berechtigungen für große Gruppen von Benutzern. Weitere Informationen finden Sie unter [Anwendungsfälle für IAM-Benutzer](#) im IAM-Benutzerhandbuch.

IAM-Rollen

Eine [IAM-Rolle](#) ist eine Identität mit spezifischen Berechtigungen, die temporäre Anmeldeinformationen bereitstellt. Sie können eine Rolle übernehmen, indem Sie [von einer Benutzer- zu einer IAM-Rolle \(Konsole\) wechseln](#) AWS CLI oder einen AWS API-Vorgang aufrufen. Weitere Informationen finden Sie unter [Methoden, um eine Rolle zu übernehmen](#) im IAM-Benutzerhandbuch.

IAM-Rollen sind nützlich für den Verbundbenutzer-Zugriff, temporäre IAM-Benutzerberechtigungen, kontoübergreifenden Zugriff, serviceübergreifenden Zugriff und Anwendungen, die auf Amazon EC2 laufen. Weitere Informationen finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie definiert Berechtigungen, wenn sie mit einer Identität oder Ressource verknüpft sind. AWS bewertet diese Richtlinien, wenn ein Principal eine Anfrage stellt. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu JSON-Richtliniendokumenten finden Sie unter [Übersicht über JSON-Richtlinien](#) im IAM-Benutzerhandbuch.

Mit Hilfe von Richtlinien legen Administratoren fest, wer Zugriff auf was hat, indem sie definieren, welches Prinzipal welche Aktionen auf welchen Ressourcen und unter welchen Bedingungen durchführen darf.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator erstellt IAM-Richtlinien und fügt sie zu Rollen hinzu, die die Benutzer dann übernehmen können. IAM-Richtlinien definieren Berechtigungen unabhängig von der Methode, die zur Ausführung der Operation verwendet wird.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität (Benutzer, Gruppe oder Rolle) anfügen können. Diese Richtlinien steuern, welche Aktionen Identitäten für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien](#) im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können Inline-Richtlinien (direkt in eine einzelne Identität eingebettet) oder verwaltete Richtlinien (eigenständige Richtlinien, die mit mehreren Identitäten verbunden sind)

sein. Informationen dazu, wie Sie zwischen verwalteten und Inline-Richtlinien wählen, finden Sie unter [Choose between managed policies and inline policies](#) im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele hierfür sind Vertrauensrichtlinien für IAM-Rollen und Amazon S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#).

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Weitere Richtlinientypen

AWS unterstützt zusätzliche Richtlinientypen, mit denen die maximalen Berechtigungen festgelegt werden können, die durch gängigere Richtlinientypen gewährt werden:

- **Berechtigungsgrenzen** – Eine Berechtigungsgrenze legt die maximalen Berechtigungen fest, die eine identitätsbasierte Richtlinie einer IAM-Entität erteilen kann. Weitere Informationen finden Sie unter [Berechtigungsgrenzen für IAM-Entitäten](#) im IAM-Benutzerhandbuch.
- **Richtlinien zur Dienstkontrolle (SCPs)** — Geben Sie die maximalen Berechtigungen für eine Organisation oder Organisationseinheit in an AWS Organizations. Weitere Informationen finden Sie unter [Service-Kontrollrichtlinien](#) im AWS Organizations -Benutzerhandbuch.
- **Richtlinien zur Ressourcenkontrolle (RCPs)** — Legen Sie die maximal verfügbaren Berechtigungen für Ressourcen in Ihren Konten fest. Weitere Informationen finden Sie im AWS Organizations Benutzerhandbuch unter [Richtlinien zur Ressourcenkontrolle \(RCPs\)](#).
- **Sitzungsrichtlinien** – Sitzungsrichtlinien sind erweiterte Richtlinien, die als Parameter übergeben werden, wenn Sie eine temporäre Sitzung für eine Rolle oder einen Verbundbenutzer erstellen. Weitere Informationen finden Sie unter [Sitzungsrichtlinien](#) im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn für eine Anfrage mehrere Arten von Richtlinien gelten, sind die daraus resultierenden Berechtigungen schwieriger zu verstehen. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie unter [Bewertungslogik für Richtlinien](#) im IAM-Benutzerhandbuch.

Zugriffskontrolle

Sie können über gültige Anmeldeinformationen verfügen, um Ihre Anfragen zu authentifizieren, aber ohne die entsprechenden Berechtigungen können Sie keine Amazon SWF SWF-Ressourcen erstellen oder darauf zugreifen. Sie benötigen beispielsweise Berechtigungen zum Aufrufen AWS Lambda von Zielen für Amazon Simple Notification Service (Amazon SNS) und Amazon Simple Queue Service (Amazon SQS), die mit Ihren Amazon SWF SWF-Regeln verknüpft sind.

In den folgenden Abschnitten wird beschrieben, wie Sie Berechtigungen für Amazon SWF verwalten. Wir empfehlen Ihnen, zunächst die Übersicht zu lesen.

- [Grundlegende Prinzipien](#)
- [Amazon SWF IAM-Richtlinien](#)
- [Richtlinien für Amazon SWF schreiben](#)

Richtlinienmaßnahmen für Amazon SWF

Unterstützt Richtlinienaktionen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Nehmen Sie Aktionen in eine Richtlinie auf, um Berechtigungen zur Ausführung des zugehörigen Vorgangs zu erteilen.

Eine Liste der Amazon SWF SWF-Aktionen finden Sie unter [Von Amazon Simple Workflow Service definierte Ressourcen in der Service](#) Authorization Reference.

Richtlinienaktionen in Amazon SWF verwenden vor der Aktion das folgende Präfix:

```
swf
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

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

```
"swf:action2"  
]
```

Beispiele für identitätsbasierte Amazon SWF SWF-Richtlinien finden Sie unter. [Beispiele für identitätsbasierte Richtlinien für Amazon Simple Workflow Service](#)

Richtlinienressourcen für Amazon SWF

Unterstützt Richtlinienressourcen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Als Best Practice geben Sie eine Ressource mit dem zugehörigen [Amazon-Ressourcennamen \(ARN\)](#) an. Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

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

Eine Liste der Amazon SWF SWF-Ressourcentypen und ihrer Eigenschaften ARNs finden Sie unter [Von Amazon Simple Workflow Service definierte Aktionen](#) in der Service Authorization Reference. Informationen darüber, mit welchen Aktionen Sie den ARN jeder Ressource angeben können, finden Sie unter [Von Amazon Simple Workflow Service definierte Ressourcen](#).

Beispiele für identitätsbasierte Amazon SWF SWF-Richtlinien finden Sie unter. [Beispiele für identitätsbasierte Richtlinien für Amazon Simple Workflow Service](#)

Schlüssel für Richtlinienbedingungen für Amazon SWF

Unterstützt servicespezifische Richtlinienbedingungsschlüssel: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Condition` gibt an, wann Anweisungen auf der Grundlage definierter Kriterien ausgeführt werden. Sie können bedingte Ausdrücke erstellen, die [Bedingungsoperatoren](#)

verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter [Kontextschlüssel für AWS globale Bedingungen](#) im IAM-Benutzerhandbuch.

Eine Liste der Amazon SWF SWF-Bedingungsschlüssel finden Sie unter [Bedingungsschlüssel für Amazon Simple Workflow Service](#) in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter [Von Amazon Simple Workflow Service definierte Ressourcen](#).

Beispiele für identitätsbasierte Amazon SWF SWF-Richtlinien finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon Simple Workflow Service](#)

ACLs bei Amazon SWF

Unterstützt ACLs: Nein

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

ABAC mit Amazon SWF

Unterstützt ABAC (Tags in Richtlinien): Teilweise

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen, auch als Tags bezeichnet, definiert werden. Sie können Tags an IAM-Entitäten und AWS -Ressourcen anhängen und dann ABAC-Richtlinien entwerfen, um Operationen zu ermöglichen, wenn das Tag des Prinzipals mit dem Tag auf der Ressource übereinstimmt.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungelement einer [Richtlinie Tag-Informationen](#) an, indem Sie die Schlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter [Definieren von Berechtigungen mit ABAC-Autorisierung](#) im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe [Attributbasierte Zugriffskontrolle \(ABAC\)](#) verwenden im IAM-Benutzerhandbuch.

Temporäre Anmeldeinformationen mit Amazon SWF verwenden

Unterstützt temporäre Anmeldeinformationen: Ja

Temporäre Anmeldeinformationen ermöglichen kurzfristigen Zugriff auf AWS Ressourcen und werden automatisch erstellt, wenn Sie den Verbund verwenden oder die Rollen wechseln. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter [Temporäre Anmeldeinformationen in IAM](#) und [AWS-Services , die mit IAM funktionieren](#) im IAM-Benutzerhandbuch.

Serviceübergreifende Prinzipalberechtigungen für Amazon SWF

Unterstützt Forward Access Sessions (FAS): Ja

Forward Access Sessions (FAS) verwenden die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-Service zu stellen. Einzelheiten zu den Richtlinien für FAS-Anforderungen finden Sie unter [Zugriffssitzungen weiterleiten](#).

Servicerollen für Amazon SWF

Unterstützt Servicerollen: Ja

Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch.

Warning

Durch das Ändern der Berechtigungen für eine Servicerolle kann die Amazon SWF SWF-Funktionalität beeinträchtigt werden. Bearbeiten Sie Servicerollen nur, wenn Amazon SWF Sie dazu anleitet.

Servicebezogene Rollen für Amazon SWF

Unterstützt serviceverknüpfte Rollen: Ja

Eine serviceverknüpfte Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Details zum Erstellen oder Verwalten von serviceverknüpften Rollen finden Sie unter [AWS -Services, die mit IAM funktionieren](#). Suchen Sie in der Tabelle nach einem Service mit einem Yes in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

Identitätsbasierte Richtlinien für Amazon SWF

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien](#) im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der [IAM-Referenz für JSON-Richtlinienelemente](#) im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für Amazon SWF

Beispiele für identitätsbasierte Amazon SWF SWF-Richtlinien finden Sie unter. [Beispiele für identitätsbasierte Richtlinien für Amazon Simple Workflow Service](#)

Ressourcenbasierte Richtlinien in Amazon SWF

Unterstützt ressourcenbasierte Richtlinien: Nein

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und

Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Weitere Informationen finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.

So funktioniert Amazon Simple Workflow Service mit IAM

Bevor Sie IAM verwenden, um den Zugriff auf Amazon SWF zu verwalten, sollten Sie sich darüber informieren, welche IAM-Funktionen für Amazon SWF verfügbar sind.

IAM-Funktionen, die Sie mit Amazon Simple Workflow Service verwenden können

IAM-Feature	Amazon SWF SWF-Unterstützung
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Nein
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Richtlinienbedingungsschlüssel (servicespezifisch)	Ja
ACLs	Nein
ABAC (Tags in Richtlinien)	Teilweise
Temporäre Anmeldeinformationen	Ja
Prinzipalberechtigungen	Ja
Servicerollen	Ja

IAM-Feature	Amazon SWF SWF-Unterstützung
Service-verknüpfte Rollen	Nein

Einen allgemeinen Überblick darüber, wie Amazon SWF und andere AWS Dienste mit den meisten IAM-Funktionen funktionieren, finden Sie im [IAM-Benutzerhandbuch unter AWS Dienste, die mit IAM funktionieren](#).

Beispiele für identitätsbasierte Richtlinien für Amazon Simple Workflow Service

Standardmäßig sind Benutzer und Rollen nicht berechtigt, Amazon SWF SWF-Ressourcen zu erstellen oder zu ändern. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter [Erstellen von IAM-Richtlinien \(Konsole\)](#) im IAM-Benutzerhandbuch.

Einzelheiten zu den von Amazon SWF definierten Aktionen und Ressourcentypen, einschließlich des Formats ARNs für die einzelnen Ressourcentypen, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Simple Workflow Service in der Service Authorization Reference](#).

Themen

- [Best Practices für Richtlinien](#)
- [Verwenden der Amazon SWF SWF-Konsole](#)
- [Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer](#)

Best Practices für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand Amazon SWF SWF-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder diese löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Wenn Sie identitätsbasierte Richtlinien erstellen oder bearbeiten, befolgen Sie diese Richtlinien und Empfehlungen:

- Erste Schritte mit AWS verwalteten Richtlinien und Umstellung auf Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für

viele gängige Anwendungsfälle gewähren, um damit zu beginnen, Ihren Benutzern und Workloads Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter [Von AWS verwaltete Richtlinien](#) oder [Von AWS verwaltete Richtlinien für Auftragsfunktionen](#) im IAM-Benutzerhandbuch.

- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter [Richtlinien und Berechtigungen in IAM](#) im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. CloudFormation B. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingung](#) im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter [Richtlinienvvalidierung mit IAM Access Analyzer](#) im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter [Sicherer API-Zugriff mit MFA](#) im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter [Best Practices für die Sicherheit in IAM](#) im IAM-Benutzerhandbuch.

Verwenden der Amazon SWF SWF-Konsole

Um auf die Amazon Simple Workflow Service-Konsole zugreifen zu können, benötigen Sie ein Mindestmaß an Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den Amazon SWF SWF-Ressourcen in Ihrem aufzulisten und anzuzeigen AWS-Konto. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Sie müssen Benutzern, die nur die API AWS CLI oder die AWS API aufrufen, keine Mindestberechtigungen für die Konsole gewähren. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Um sicherzustellen, dass Benutzer und Rollen die Amazon SWF-Konsole weiterhin verwenden können, fügen Sie den Entitäten auch die Amazon SWF *ConsoleAccess* - oder *ReadOnly* AWS verwaltete Richtlinie hinzu. Weitere Informationen finden Sie unter [Hinzufügen von Berechtigungen zu einem Benutzer](#) im IAM-Benutzerhandbuch.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie beinhaltet Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der API AWS CLI oder AWS .

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

Grundlegende Prinzipien

Die Amazon SWF SWF-Zugriffskontrolle basiert hauptsächlich auf zwei Arten von Berechtigungen:

- **Ressourcenberechtigungen:** Auf welche Amazon SWF SWF-Ressourcen ein Benutzer zugreifen kann.

Sie können Ressourcenberechtigungen nur für Domänen erteilen.

- **API-Berechtigungen:** Welche Amazon SWF SWF-Aktionen ein Benutzer aufrufen kann.

Die einfachste Methode besteht darin, vollen Kontozugriff zu gewähren — jede Amazon SWF SWF-Aktion in einer beliebigen Domain aufzurufen — oder den Zugriff vollständig zu verweigern. IAM unterstützt jedoch einen detaillierteren Ansatz zur Zugriffskontrolle, der oft nützlicher ist. So können Sie beispielsweise:

- Erlaubt einem Benutzer, jede Amazon SWF SWF-Aktion ohne Einschränkungen aufzurufen, jedoch nur in einer bestimmten Domain. Solch eine Richtlinie eignet sich, um Workflow-Anwendungen, die sich noch in der Entwicklung befinden, die Möglichkeit zu bieten, alle Aktionen zu verwenden, allerdings nur in einer "Sandbox"-Domäne.
- einem Benutzer Zugriff auf alle Domänen zu gewähren, jedoch die Verwendung der API einzuschränken. Solch eine Richtlinie eignet sich, um einer "Auditor"-Anwendung die Berechtigung zu erteilen, die API in allen Domänen aufzurufen, ihr allerdings nur Lesezugriff zu gewähren.

- einem Benutzer die Berechtigung erteilen, eine begrenzte Anzahl an Aktionen in bestimmten Domänen aufzurufen. Solch eine Richtlinie eignet sich, um einem Workflow-Starter die Berechtigung zu erteilen, nur die `StartWorkflowExecution`-Aktion in einer bestimmten Domäne aufzurufen.

Die Amazon SWF SWF-Zugriffskontrolle basiert auf den folgenden Prinzipien:

- Entscheidungen zur Zugriffskontrolle basieren ausschließlich auf IAM-Richtlinien. Die gesamte Überprüfung und Bearbeitung von Richtlinien erfolgt über IAM.
- Das Zugriffskontrollmodell verwendet eine deny-by-default Richtlinie. Jeder Zugriff, der nicht ausdrücklich erlaubt ist, wird verweigert.
- Sie kontrollieren den Zugriff auf Amazon SWF SWF-Ressourcen, indem Sie den Akteuren des Workflows entsprechende IAM-Richtlinien zuordnen.
- Ressourcenberechtigungen können nur für Domänen erteilt werden.
- Sie können die Verwendung für einige Aktionen weiter einschränken, indem Sie einem oder mehreren Parametern Bedingungen hinzufügen.
- Wenn Sie die Erlaubnis zur Nutzung erteilen [RespondDecisionTaskCompleted](#), können Sie Genehmigungen für die Liste der in dieser Aktion enthaltenen Entscheidungen erteilen.

Jede Entscheidung umfasst einen oder mehrere Parameter, ähnlich einem regulären API-Aufruf. Damit die Richtlinien so gut wie möglich zu lesen sind, können Sie die Berechtigungen für Entscheidungen so erteilen, als würde es sich dabei um API-Aufrufe handeln, einschließlich dem Hinzufügen von Bedingungen zu einigen Parametern. Diese Art von Berechtigungen werden als Pseudo-API-Berechtigungen bezeichnet.

Eine Zusammenfassung der regulären und Pseudo-API-Parameter, die durch Bedingungen eingeschränkt werden können, finden Sie unter [Übersicht über API-Befehle](#).

Amazon SWF IAM-Richtlinien

Eine IAM-Richtlinie enthält ein oder mehrere Statement Elemente, von denen jedes eine Reihe von Elementen enthält, die die Richtlinie definieren. Eine vollständige Liste der Elemente und eine allgemeine Erläuterung der Erstellung von Richtlinien finden Sie unter [The Access Policy Language](#). Die Amazon SWF SWF-Zugriffskontrolle basiert auf den folgenden Elementen:

Auswirkung

(Erforderlich) Der Effekt der Anweisung: deny oder allow.

Note

Sie müssen den Zugriff explizit zulassen. IAM verweigert den Zugriff standardmäßig.

Ressource

(Erforderlich) Die Ressource — eine Entität in einem AWS Dienst, mit der ein Benutzer interagieren kann —, für die sich die Anweisung bezieht.

Sie können Ressourcenberechtigungen nur für Domänen erteilen. Beispielsweise kann eine Richtlinie nur Zugriff auf bestimmte Domänen in Ihrem Konto erteilen. Um Berechtigungen für eine Domain auszudrücken, legen Sie Resource den Amazon-Ressourcennamen (ARN) der Domain fest, der das Format „arn:aws:swf::*Region*:*AccountID*/domain/“ hat. *DomainName* *Region* ist die AWS Region, *AccountID* ist die Konto-ID ohne Bindestriche und ist der Domainname. *DomainName*

Action

(Erforderlich) Die Aktion, für die sich die Aussage bezieht und auf die Sie sich beziehen, indem Sie das folgende Format verwenden *serviceId:action*. Stellen Sie für Amazon SWF *serviceID* auf `swf`. `swf:StartWorkflowExecution` bezieht sich beispielsweise auf die [StartWorkflowExecution](#) Aktion und wird verwendet, um zu steuern, welche Benutzer Workflows starten dürfen.

Wenn Sie die Nutzungsberechtigung erteilen [RespondDecisionTaskCompleted](#), können Sie auch den Zugriff auf die enthaltene Entscheidungsliste kontrollieren, indem Sie die Berechtigungen für die Pseudo-API Action zum Ausdruck bringen. Da IAM standardmäßig den Zugriff verweigert, muss die Entscheidung eines Entscheiders ausdrücklich zugelassen werden, andernfalls wird sie nicht akzeptiert. Mithilfe des *-Werts können Sie alle Entscheidungen erlauben.

Bedingung

(Optional) Drückt eine Einschränkung für einen oder mehrere Parameter einer Aktion aus. Hierdurch werden die zulässigen Werte eingeschränkt.

Amazon SWF SWF-Aktionen haben oft einen großen Umfang, den Sie mithilfe von IAM-Bedingungen reduzieren können. Um beispielsweise einzuschränken, auf welche Aufgabenlisten

die [PollForActivityTask](#)Aktion zugreifen darf, fügen Sie a hinzu `Condition` und geben mit der `swf:taskList.name` Taste die zulässigen Listen an.

Sie können für die folgenden Entitäten Einschränkungen hinzufügen.

- Der Workflow-Typ. Der Name und die Version verfügen über separate Schlüssel.
- Der Aktivitätstyp. Der Name und die Version verfügen über separate Schlüssel.
- Aufgabenlisten.
- Tags. Sie können für manche Aktionen mehrere Tags angeben. In diesem Fall verfügt jedes Tag über einen separaten Schlüssel.

Note

Bei Amazon SWF handelt es sich bei den Werten ausschließlich um Zeichenketten, sodass Sie einen Parameter einschränken, indem Sie einen Zeichenkettenoperator wie `StringEquals`, verwenden, der den Parameter auf eine bestimmte Zeichenfolge beschränkt. Die regulären Zeichenfolgenoperatoren wie `StringEquals` erfordern es jedoch, dass alle Anfragen den Parameter enthalten. Wenn Sie den Parameter nicht ausdrücklich einschließen und kein Standardwert vorliegt, wie beispielsweise die während der Typenregistrierung bereitgestellte standardmäßige Aufgabenliste, wird der Zugriff verweigert.

Es empfiehlt sich häufig, Bedingungen als optional zu behandeln, so können Sie eine Aktion auch ohne den zugeordneten Parameter aufrufen. Beispielsweise könnten Sie einem Entscheider die Möglichkeit geben, eine Reihe von [RespondDecisionTaskCompleted](#)Entscheidungen zu spezifizieren, aber auch, nur eine davon für einen bestimmten Anruf anzugeben. In diesem Fall beschränken Sie die entsprechenden Parameter mithilfe des `StringEqualsIfExists`-Operators, der den Zugriff gewährt, wenn der Parameter der Bedingung entspricht, bei Abwesenheit des Parameters den Zugriff jedoch verweigert.

Eine vollständige Liste der einschränkbaren Parameter sowie der zugeordneten Schlüssel finden Sie unter [Übersicht über API-Befehle](#).

Der folgende Abschnitt enthält Beispiele für die Erstellung von Amazon SWF SWF-Richtlinien. Weitere Informationen finden Sie unter [Zeichenfolgenbedingungen](#).

Richtlinien für Amazon SWF schreiben

Ein Workflow besteht aus mehreren Akteuren — Aktivitäten, Entscheidern usw. Sie können den Zugriff für jeden Akteur steuern, indem Sie eine entsprechende IAM-Richtlinie anhängen.

Mit der folgenden Aktion erhält der Akteur vollen Kontozugriff in allen Regionen:

- Aktion: `swf:*`
- Ressource: `arn:aws:swf:*:123456789012:/domain/*`

Sie können Platzhalter verwenden, damit ein einzelner Wert mehrere Ressourcen, Aktionen oder Regionen repräsentiert.

- Der erste Platzhalter (*) im Resource Wert gibt an, dass die Ressourcenberechtigungen für alle Regionen gelten.

Um die Berechtigungen auf eine einzelne Region zu beschränken, ersetzen Sie den Platzhalter durch die entsprechende Regionszeichenfolge, z. B. `us-east-1`.

- Der zweite Platzhalter (*) im Resource-Wert ermöglicht es dem Akteur, auf jede der zum Konto gehörenden Domänen innerhalb der angegebenen Regionen zuzugreifen.
- Der Platzhalter (*) im Action Wert ermöglicht es dem Akteur, jede Amazon SWF SWF-Aktion aufzurufen.

Weitere Informationen zur Verwendung von Platzhaltern finden Sie unter [Beschreibungen der Elemente](#).

Domänenberechtigungen

Um die Arbeitsabläufe einer Abteilung auf eine bestimmte Domäne zu beschränken, könnten Sie einem Akteur eine Genehmigung erteilen, die es einem Akteur ermöglicht, jede Aktion auszurufen, jedoch nur für eine bestimmte Abteilung.

Um einem Akteur Zugriff auf mehr als eine Domäne zu gewähren, geben Sie für jede Domäne die ausdrückliche Genehmigung in Form einer Liste von Anweisungen aus:

- Aktion: `swf:*`
- Ressource: `arn:aws:swf:*:123456789012:/domain/department1`
- Ressource: `arn:aws:swf:*:123456789012:/domain/department2`

Sie können einem Akteur erlauben, jede Amazon SWF SWF-Aktion in den `department2` Domänen `department1` und zu verwenden. Manchmal können Sie auch Platzhalter verwenden, um mehrere Domänen zu repräsentieren.

API-Berechtigungen und -Einschränkungen

Sie steuern, welche Aktionen ein Akteur verwenden kann, indem Sie die Aktion im `Action Element` angeben.

Mit der folgenden Aktion kann ein Akteur Workflows nur aufrufen, `StartWorkflowExecution` um sie zu starten. Er kann keine anderen Aktionen verwenden.

- Aktion: `swf:StartWorkflowExecution`

Bedingungen

Sie können optional die zulässigen Parameterwerte der Aktion mithilfe eines `Condition Element` einschränken.

Um einzuschränken, welche Workflows ein Akteur starten kann, schränken Sie einen oder mehrere der `StartWorkflowExecution` Parameterwerte wie folgt ein:

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

Ein Akteur mit den vorherigen Einschränkungen kann nur `version2` von ausgeführt werden, `workflow1` und beide Parameter müssen in der Anforderung enthalten sein.

Sie können einen Parameter aber auch beschränken, ohne dass er in einer Anfrage enthalten sein muss, indem Sie einen `StringEqualsIfExists`-Operator verwenden. Gehen Sie dazu folgendermaßen vor:

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

Ein Akteur mit der vorherigen Richtlinie kann optional eine Aufgabenliste angeben, wenn er eine Workflow-Ausführung startet.

Sie können für manche Aktionen eine Reihe von Tags beschränken. Jedes Tag hat einen eigenen Schlüssel, mit `swf:tagList.member.0` dem Sie das erste Tag in der Liste einschränken, `swf:tagList.member.1` das zweite Tag in der Liste einschränken usw., bis zu einem Maximum von 5.

Sie müssen vorsichtig sein, wie Sie Taglisten einschränken. Beispielsweise wird die folgende Bedingung nicht empfohlen.

Die folgende Bedingung wird nicht empfohlen, da Sie damit optional entweder `some_ok_tag` oder angeben können `another_ok_tag`. Die Bedingung schränkt jedoch nur das erste Element der Tagliste ein. Die Liste könnte zusätzliche Elemente mit beliebigen Werten enthalten, die alle zulässig wären, weil die Bedingung keine Bedingungen für `swf:tagList.member.1` `swf:tagList.member.2`, usw. erfüllt.

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

Eine Möglichkeit, das vorherige Problem zu lösen, besteht darin, die Verwendung von Taglisten zu verbieten.

Die folgende Richtlinie stellt sicher, dass nur `some_ok_tag` oder `another_ok_tag` zulässig sind, da die Liste nur ein Element besitzen darf.

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

Pseudo-API-Berechtigungen und -Einschränkungen

Um die verfügbaren Entscheidungen einzuschränken `RespondDecisionTaskCompleted`, müssen Sie dem Akteur zunächst erlauben, anzurufen `RespondDecisionTaskCompleted`. Anschließend

geben Sie die Berechtigungen für die entsprechenden Pseudo-API-Mitglieder mit derselben Syntax wie für die reguläre API wie folgt aus:

- Aussage 1

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

Aktion: `swf:RespondDecisionTaskCompleted`

- Aussage 2

Ressource: `*`

Aktion: `swf:ScheduleActivityTask`

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

Die erste Statement ermöglicht es dem Schauspieler, anzurufen `RespondDecisionTaskCompleted`. Die zweite Anweisung ermöglicht es dem Akteur, die `ScheduleActivityTask` Entscheidung zu nutzen, Amazon SWF anzuweisen, eine Aktivitätsaufgabe zu planen. Um alle Entscheidungen zuzulassen, ersetzen Sie „`swf:ScheduleActivityTask`“ durch „`swf: *`“.

Sie können Bedingungsoperatoren verwenden, um Parameter wie bei der regulären API einzuschränken. Der `StringEquals` Operator im vorherigen Beispiel `Condition` ermöglicht es `RespondDecisionTaskCompleted`, eine Aktivitätsaufgabe für die `SomeActivityType` Aktivität zu planen, und er muss diese Aufgabe planen. Wenn Sie `RespondDecisionTaskCompleted` die Möglichkeit geben wollen, einen Parameterwert zu verwenden, dies aber nicht erforderlich sein soll, können Sie stattdessen den `StringEqualsIfExists`-Operator verwenden.

AWS verwaltete Richtlinie: `SimpleWorkflowFullAccess`

Sie können die `SimpleWorkflowFullAccess`-Richtlinie an Ihre IAM-Identitäten anfügen.

Diese Richtlinie bietet vollen Zugriff auf den Amazon SWF SWF-Konfigurationsservice.

Einschränkungen des Servicemodells in Bezug auf IAM-Richtlinien

Bei der Erstellung von IAM-Richtlinien müssen Sie Einschränkungen des Servicemodells berücksichtigen. Es ist möglich, eine syntaktisch gültige IAM-Richtlinie zu erstellen, die eine ungültige

Amazon SWF SWF-Anfrage darstellt. Eine Anforderung, die im Hinblick auf die Zugriffskontrolle zulässig ist, kann dennoch fehlschlagen, da es sich um eine ungültige Anforderung handelt.

Das Amazon SWF-Servicemodell erlaubt beispielsweise nicht, dass die `tagFilter` Parameter `typeFilter` und in derselben [ListOpenWorkflowExecutions](#) Anfrage verwendet werden. Die folgende Bedingung würde Aufrufe zulassen, die der Service ablehnt — indem er sie ausgibt `ValidationException` — als ungültige Anfrage:

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

Übersicht über API-Befehle

In diesem Abschnitt wird kurz beschrieben, wie Sie mithilfe von IAM-Richtlinien steuern können, wie ein Akteur jede API und Pseudo-API für den Zugriff auf Amazon SWF SWF-Ressourcen verwenden kann.

- Sie können für alle Aktionen mit Ausnahme von `RegisterDomain` und `ListDomains` auf jede oder alle Domänen eines Kontos Zugriff gewähren, indem Sie die Berechtigungen für die Domänenressource erteilen.
- Sie können jedem Mitglied einer regulären API Berechtigungen erteilen bzw. verweigern. Berechtigungen zum Aufrufen von [RespondDecisionTaskCompleted](#) können Sie jedem Mitglied einer Pseudo-API gewähren.
- Sie können eine Bedingung verwenden, um die zulässigen Werte eines Parameters einzuschränken

Die folgenden Abschnitte listen die Parameter auf, die für die einzelnen Elemente einer regulären und einer Pseudo-API eingeschränkt werden können. Darüber hinaus finden Sie die zugeordneten Schlüssel sowie alle Einschränkungen, die Sie bei der Kontrolle über den Domänenzugriff beachten müssen.

Normale API

Dieser Abschnitt listet die regulären API-Elemente auf und beschreibt kurz die Parameter, die eingeschränkt werden können sowie die zugeordneten Schlüssel. Er beschreibt auch alle Einschränkungen, die Sie bei der Kontrolle über den Domänenzugriff beachten müssen.

[CountClosedWorkflowExecutions](#)

- `tagFilter.tag`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:tagFilter.tag`
- `typeFilter.name`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:typeFilter.name`.
- `typeFilter.version`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:typeFilter.version`.

Note

`CountClosedWorkflowExecutions` erfordert es, dass `typeFilter` und `tagFilter` sich gegenseitig ausschließen.

[CountOpenWorkflowExecutions](#)

- `tagFilter.tag`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:tagFilter.tag`
- `typeFilter.name`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:typeFilter.name`.
- `typeFilter.version`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:typeFilter.version`.

Note

`CountOpenWorkflowExecutions` erfordert es, dass `typeFilter` und `tagFilter` sich gegenseitig ausschließen.

[CountPendingActivityTasks](#)

- `taskList.name`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:taskList.name`.

[CountPendingDecisionTasks](#)

- `taskList.name`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:taskList.name`.

[DeleteActivityType](#)

- `activityType.name`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:activityType.name`.
- `activityType.version`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:activityType.version`.

[DeprecateActivityType](#)

- `activityType.name`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:activityType.name`.
- `activityType.version`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:activityType.version`.

[DeprecateDomain](#)

- Sie können die Parameter für diese Aktion nicht einschränken.

[DeleteWorkflowType](#)

- `workflowType.name`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:workflowType.name`.
- `workflowType.version`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:workflowType.version`.

[DeprecateWorkflowType](#)

- `workflowType.name`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:workflowType.name`.
- `workflowType.version`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:workflowType.version`.

[DescribeActivityType](#)

- `activityType.name`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:activityType.name`.
- `activityType.version`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:activityType.version`.

[DescribeDomain](#)

- Sie können die Parameter für diese Aktion nicht einschränken.

[DescribeWorkflowExecution](#)

- Sie können die Parameter für diese Aktion nicht einschränken.

[DescribeWorkflowType](#)

- `workflowType.name`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:workflowType.name`.
- `workflowType.version`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:workflowType.version`.

[GetWorkflowExecutionHistory](#)

- Sie können die Parameter für diese Aktion nicht einschränken.


[ListActivityTypes](#)

- Sie können die Parameter für diese Aktion nicht einschränken.

[ListClosedWorkflowExecutions](#)

- `tagFilter.tag`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:tagFilter.tag`
- `typeFilter.name`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:typeFilter.name`.

- `typeFilter.version`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:typeFilter.version`.

 Note


`ListClosedWorkflowExecutions` erfordert es, dass `typeFilter` und `tagFilter` sich gegenseitig ausschließen.

[ListDomains](#)

- Sie können die Parameter für diese Aktion nicht einschränken.

[ListOpenWorkflowExecutions](#)

- `tagFilter.tag`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:tagFilter.tag`
- `typeFilter.name`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:typeFilter.name`.
- `typeFilter.version`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:typeFilter.version`.

 Note

`ListOpenWorkflowExecutions` erfordert es, dass `typeFilter` und `tagFilter` sich gegenseitig ausschließen.

[ListWorkflowTypes](#)

- Sie können die Parameter für diese Aktion nicht einschränken.

[PollForActivityTask](#)

- `taskList.name`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:taskList.name`.

[PollForDecisionTask](#)

- `taskList.name`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:taskList.name`.

[RecordActivityTaskHeartbeat](#)

- Sie können die Parameter für diese Aktion nicht einschränken.

[RegisterActivityType](#)

- `defaultTaskList.name`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:defaultTaskList.name`.
- `name`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:name`.
- `version`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:version`.

[RegisterDomain](#)

- `name`— Der Name der Domain, die registriert wird, ist als Ressource für diese Aktion verfügbar.

[RegisterWorkflowType](#)

- `defaultTaskList.name`— Zeichenkettenbeschränkung. Der Schlüssel lautet `swf:defaultTaskList.name`.
- `name`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:name`.
- `version`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:version`.

[RequestCancelWorkflowExecution](#)

- Sie können die Parameter für diese Aktion nicht einschränken.

[RespondActivityTaskCanceled](#)

- Sie können die Parameter für diese Aktion nicht einschränken.

[RespondActivityTaskCompleted](#)

- Sie können die Parameter für diese Aktion nicht einschränken.

[RespondActivityTaskFailed](#)

- Sie können die Parameter für diese Aktion nicht einschränken.

[RespondDecisionTaskCompleted](#)

- `decisions.member.N`— Indirekt durch Pseudo-API-Berechtigungen eingeschränkt. Details hierzu finden Sie unter [Pseudo-API](#).

[SignalWorkflowExecution](#)

- Sie können die Parameter für diese Aktion nicht einschränken.

[StartWorkflowExecution](#)

- `tagList.member.0`— Zeichenkettenbeschränkung. Der Schlüssel lautet `swf:tagList.member.0`
- `tagList.member.1`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:tagList.member.1`
- `tagList.member.2`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:tagList.member.2`
- `tagList.member.3`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:tagList.member.3`
- `tagList.member.4`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:tagList.member.4`
- `taskList.name`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:taskList.name`.
- `workflowType.name`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:workflowType.name`.
- `workflowType.version`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:workflowType.version`.

Note

Sie können nicht mehr als fünf Tags einschränken.

[TerminateWorkflowExecution](#)

- Sie können die Parameter für diese Aktion nicht einschränken.

Pseudo-API

In diesem Abschnitt werden Elemente der Pseudo-API aufgelistet, die die unter [RespondDecisionTaskCompleted](#) eingeschlossenen Entscheidungen repräsentieren. Wenn Sie die Berechtigung erteilt haben, RespondDecisionTaskCompleted zu verwenden, kann Ihre Richtlinie die Berechtigungen für die Elemente dieser API auf die gleiche Art und Weise erteilen wie für die reguläre API. Sie können die Elemente für einige Elemente der Pseudo-API weiter einschränken, indem Sie für einen oder mehrere Parameter Bedingungen einrichten. Dieser Abschnitt listet die Pseudo-API-Elemente auf und beschreibt kurz die Parameter, die eingeschränkt werden können sowie die zugeordneten Schlüssel.

Note

Die Schlüssel `aws:SourceIP`, `aws:UserAgent` und `aws:SecureTransport` stehen für die Pseudo-API nicht zur Verfügung. Wenn Ihre vorgesehene Sicherheitsrichtlinie diese Schlüssel benötigt, um den Zugriff auf die Pseudo-API zu kontrollieren, können Sie sie über die RespondDecisionTaskCompleted-Aktion verwenden.

CancelTimer

- Sie können die Parameter für diese Aktion nicht einschränken.

CancelWorkflowExecution


- Sie können die Parameter für diese Aktion nicht einschränken.

CompleteWorkflowExecution

- Sie können die Parameter für diese Aktion nicht einschränken.

ContinueAsNewWorkflowExecution

- `tagList.member.0`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:tagList.member.0`
- `tagList.member.1`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:tagList.member.1`
- `tagList.member.2`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:tagList.member.2`
- `tagList.member.3`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:tagList.member.3`
- `tagList.member.4`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:tagList.member.4`
- `taskList.name`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:taskList.name`.
- `workflowTypeVersion`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:workflowTypeVersion`.

 Note

Sie können nicht mehr als fünf Tags einschränken.

FailWorkflowExecution

- Sie können die Parameter für diese Aktion nicht einschränken.

RecordMarker

- Sie können die Parameter für diese Aktion nicht einschränken.

RequestCancelActivityTask

- Sie können die Parameter für diese Aktion nicht einschränken.

RequestCancelExternalWorkflowExecution

- Sie können die Parameter für diese Aktion nicht einschränken.

ScheduleActivityTask

- `activityType.name`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:activityType.name`.
- `activityType.version`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:activityType.version`.
- `taskList.name`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:taskList.name`.

SignalExternalWorkflowExecution

- Sie können die Parameter für diese Aktion nicht einschränken.

StartChildWorkflowExecution

- `tagList.member.0`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:tagList.member.0`
- `tagList.member.1`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:tagList.member.1`
- `tagList.member.2`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:tagList.member.2`
- `tagList.member.3`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:tagList.member.3`
- `tagList.member.4`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:tagList.member.4`
- `taskList.name`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:taskList.name`.
- `workflowType.name`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:workflowType.name`.
- `workflowType.version`— Zeichenketteneinschränkung. Der Schlüssel lautet `swf:workflowType.version`.

Note

Sie können nicht mehr als fünf Tags einschränken.

StartTimer

- Sie können die Parameter für diese Aktion nicht einschränken.

Tagbasierte Richtlinien

Amazon SWF unterstützt Richtlinien, die auf Tags basieren. Sie könnten beispielsweise Amazon SWF-Domains, die ein Tag mit dem Schlüssel `environment` und dem Wert `production` enthalten, mit der folgenden Bedingung einschränken:

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

Weitere Informationen zum Markieren finden Sie unter:

- [Schlagworte in Amazon SWF](#)
- [Zugriffssteuerung mit IAM-Tags](#)

Amazon VPC-Endpunkte für Amazon SWF

Note

AWS PrivateLink Support ist derzeit nur in den Regionen AWS Top Secret — Ost, AWS Secret Region und China verfügbar.

Wenn Sie Amazon Virtual Private Cloud (Amazon VPC) zum Hosten Ihrer AWS Ressourcen verwenden, können Sie eine Verbindung zwischen Ihren Amazon VPC- und Amazon Simple Workflow Service-Workflows herstellen. Sie können diese Verbindung mit Ihren Amazon SWF SWF-Workflows verwenden, ohne das öffentliche Internet nutzen zu müssen.

Mit Amazon VPC können Sie AWS Ressourcen in einem benutzerdefinierten virtuellen Netzwerk starten. Mit einer VPC können Sie Netzwerkeinstellungen, wie IP-Adressbereich, Subnetze, Routing-Tabellen und Netzwerk-Gateways, steuern. Weitere Informationen VPCs finden Sie im [Amazon VPC-Benutzerhandbuch](#).

Um Ihre Amazon VPC mit Amazon SWF zu verbinden, müssen Sie zunächst einen VPC-Schnittstellen-Endpunkt definieren, über den Sie Ihre VPC mit anderen verbinden können. AWS-Services Der Endpunkt bietet eine zuverlässige, skalierbare Konnektivität, ohne dass ein Internet-Gateway, eine NAT-Instance (Network Address Translation) oder eine VPN-Verbindung erforderlich ist. Weitere Informationen finden Sie unter [Schnittstellen-VPC-Endpunkte \(AWS PrivateLink\)](#) im Amazon-VPC-Benutzerhandbuch.

Erstellen des Endpunkts

Sie können einen Amazon SWF SWF-Endpunkt in Ihrer VPC mit dem AWS-Managementkonsole, dem AWS Command Line Interface (AWS CLI), einem AWS SDK, der Amazon SWF SWF-API oder CloudFormation erstellen.

Informationen zum Erstellen und Konfigurieren eines Endpunkts über die Amazon-VPC-Konsole oder die AWS CLI finden Sie unter [Creating an Interface Endpoint](#) (Erstellen eines Schnittstellenendpunkts) im Amazon-VPC-Benutzerhandbuch.

Note

Wenn Sie einen Endpunkt erstellen, geben Sie Amazon SWF als den Service an, zu dem Ihre VPC eine Verbindung herstellen soll. In der Amazon VPC-Konsole variieren die Servicenamen je nach AWS Region. In der Region AWS Top Secret — East lautet der Servicenamen für Amazon SWF beispielsweise `com.amazonaws.us-iso-east-1.swf`.

Informationen zum Erstellen und Konfigurieren eines Endpunkts mithilfe CloudFormation von finden Sie in der VPC-Endpoint Ressource [AWSEC2:::](#) im CloudFormation Benutzerhandbuch.

Amazon VPC-Endpunkttrichtlinien

Um den Konnektivitätszugriff auf Amazon SWF zu kontrollieren, können Sie beim Erstellen eines Amazon VPC-Endpunkts eine AWS Identity and Access Management (IAM-) Endpunkttrichtlinie anhängen. Sie können komplexe IAM-Regeln erstellen, indem Sie mehrere Endpunkttrichtlinien anhängen. Weitere Informationen finden Sie unter:

- [Amazon Virtual Private Cloud Cloud-Endpunkttrichtlinien für Amazon SWF](#)
- [Kontrollieren des Zugriffs auf Services mit VPC-Endpunkten](#)

Amazon Virtual Private Cloud Cloud-Endpunktrichtlinien für Amazon SWF

Sie können eine Amazon VPC-Endpunktrichtlinie für Amazon SWF erstellen, in der Sie Folgendes angeben:

- Der Principal, der Aktionen ausführen kann.
- Aktionen, die ausgeführt werden können
- Die Ressourcen, auf denen die Aktionen ausgeführt werden können.

Das folgende Beispiel fügt einer Richtlinie eine bestimmte IAM-Rolle hinzu:

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

- Weitere Informationen zum Erstellen von Endpunktrichtlinien finden Sie unter [Steuern des Zugriffs auf Dienste mit VPC-Endpunkten](#).
- Informationen darüber, wie Sie IAM verwenden können, um den Zugriff auf Ihre AWS und Amazon SWF SWF-Ressourcen zu kontrollieren, finden Sie unter [Identity and Access Management in Amazon Simple Workflow Service](#).

Fehlerbehebung bei Identität und Zugriff auf Amazon Simple Workflow Service

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit Amazon SWF und IAM auftreten können.

Themen

- [Ich bin nicht berechtigt, eine Aktion in Amazon SWF durchzuführen](#)
- [Ich bin nicht berechtigt, iam auszuführen: PassRole](#)
- [Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine Amazon SWF SWF-Ressourcen ermöglichen](#)

Ich bin nicht berechtigt, eine Aktion in Amazon SWF durchzuführen

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zum Durchführen einer Aktion autorisiert sind, müssen Ihre Richtlinien aktualisiert werden, um die Aktion durchführen zu können.

Der folgende Beispielfehler tritt auf, wenn der `mateojackson`-Benutzer versucht, die Konsole zum Anzeigen von Details zu einer fiktiven `my-example-widget`-Ressource zu verwenden, jedoch nicht über `swf:GetWidget`-Berechtigungen verfügt.

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

In diesem Fall muss die Mateo-Richtlinie aktualisiert werden, damit er mit der `swf:GetWidget`-Aktion auf die `my-example-widget`-Ressource zugreifen kann.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich bin nicht berechtigt, iam auszuführen: PassRole

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht berechtigt sind, die `iam:PassRole` Aktion durchzuführen, müssen Ihre Richtlinien aktualisiert werden, damit Sie eine Rolle an Amazon SWF übergeben können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Service zu übergeben, anstatt eine neue Servicerolle oder eine dienstbezogene Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in Amazon SWF auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

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

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine Amazon SWF SWF-Ressourcen ermöglichen

Sie können eine Rolle erstellen, mit der Benutzer mit anderen Konten oder Personen außerhalb Ihrer Organisation auf Ihre Ressourcen zugreifen können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Für Dienste, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (ACLs) unterstützen, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob Amazon SWF diese Funktionen unterstützt, finden Sie unter [So funktioniert Amazon Simple Workflow Service mit IAM](#).
- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie im IAM-Benutzerhandbuch unter [Gewähren des Zugriffs auf einen IAM-Benutzer in einem anderen AWS-Konto, den Sie besitzen](#).
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie [AWS-Konten im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte](#).
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter [Gewähren von Zugriff für extern authentifizierte Benutzer \(Identitätsverbund\)](#) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.

Protokollieren und Überwachen

Dieser Abschnitt enthält Informationen zur Protokollierung und Überwachung von Amazon SWF.

Themen

- [Amazon SWF-Metriken für CloudWatch](#)
- [Amazon SWF-Metriken für die CloudWatch Verwendung von anzeigen AWS-Managementkonsole](#)
- [API-Aufrufe aufzeichnen mit AWS CloudTrail](#)

- [EventBridge für Änderungen des Amazon SWF SWF-Ausführungsstatus](#)
- [Verwendung AWS-Benutzerbenachrichtigungen mit Amazon Simple Workflow Service](#)

Amazon SWF-Metriken für CloudWatch

Amazon SWF bietet jetzt Metriken CloudWatch , mit denen Sie Ihre Workflows und Aktivitäten verfolgen und Alarme für von Ihnen gewählte Schwellenwerte einrichten können. Sie können Metriken mit dem anzeigen AWS-Managementkonsole. Weitere Informationen finden Sie unter [Amazon SWF-Metriken für die CloudWatch Verwendung von anzeigen AWS-Managementkonsole](#).

Themen

- [Berichtseinheiten für Amazon SWF-Metriken](#)
- [API- und Entscheidungsereignismetriken](#)
- [Amazon SWF-Metriken](#)
- [Namen und Abmessungen von Amazon SWF SWF-Ressourcen ohne ASCII CloudWatch](#)

Berichtseinheiten für Amazon SWF-Metriken

Metriken, die ein Zeitintervall berichten

Einige der Amazon SWF-Metriken für CloudWatch sind Zeitintervalle, die immer in Millisekunden gemessen werden. Die CloudWatch Einheit wird als `Time` gemeldet. Diese Metriken entsprechen in der Regel Phasen Ihrer Workflow-Ausführung, für die Sie Workflow- und Aktivitäts-Timeouts festlegen können, und besitzen ähnliche Namen.

Die Metrik `DecisionTaskStartToCloseTime` misst beispielsweise die Zeit, die die Entscheidungsaufgabe nach dem Ausführen bis zum Abschluss benötigte. Dies entspricht dem Zeitraum, für den Sie einen Wert `DecisionTaskStartToCloseTimeout` festlegen können.

Diagramme dieser Workflow-Stadien sowie weitere Informationen zu deren Auftreten im Workflow- und Aktivitätslebenszyklus finden Sie unter [Amazon SWF-Timeout-Typen](#).

Metriken, die eine Anzahl berichten

Einige der Amazon SWF-Metriken für CloudWatch Berichtsergebnisse als Zählung. `WorkflowsCanceled` erfasst beispielsweise ein Ergebnis entweder als Eins oder Null, was angibt,

ob der Workflow abgebrochen wurde oder nicht. Der Wert Null gibt nicht an, dass die Metrik nicht gemeldet wurde, sondern nur, dass die von der Metrik beschriebene Bedingung nicht auftrat.

Bei einigen der Amazon SWF-Metriken für CloudWatch diesen Bericht a Count in CloudWatch handelt es sich um eine Anzahl pro Sekunde. Beispielsweise entspricht der Wert `ProvisionedRefillRate`, der als Ein Count gemeldet wird CloudWatch, einer Rate Count von Anfragen pro Sekunde.

Bei Zahl-Metriken beträgt der Mindest- und Maximalwert stets entweder Null oder Eins, doch der Durchschnittswert ist stets ein Wert zwischen Null bis Eins.

API- und Entscheidungsereignismetriken

Sie können sowohl API- als auch Decision-Ereignisse überwachen CloudWatch , um einen Einblick in Ihre Nutzung und Kapazität zu erhalten. Weitere Informationen finden [Sie im Grundlegende Workflow-Konzepte in Amazon SWF Abschnitt „Entscheider“](#) und das Thema „[Entscheidung](#)“ in der [Amazon Simple Workflow Service API-Referenz](#).

Sie können diese Grenzwerte auch überwachen, um einen Alarm auszulösen, wenn Sie sich Ihren Amazon SWF-Drosselungsgrenzwerten nähern. Unter [Drosselungskontingente für Amazon SWF](#) finden Sie eine Beschreibung dieser Limits und ihrer Standardeinstellungen. Durch diese Limits soll verhindert werden, dass fehlerhafte Workflows Systemressourcen übermäßig verbrauchen. Informationen zur Erhöhung des Limits finden Sie unter [???](#).

Als bewährte Methode sollten Sie CloudWatch Alarme für etwa 60% Ihrer API- oder Decision Events-Kapazität konfigurieren. Auf diese Weise können Sie entweder Ihren Workflow anpassen oder eine Erhöhung des Servicelimits beantragen, bevor die Amazon SWF-Drosselung aktiviert wird. Abhängig von der [Ruckartigkeit](#) Ihrer Aufrufe können Sie unterschiedliche Warnungen konfigurieren, die Sie informieren, wenn Sie sich Ihren Servicelimits nähern:

- Wenn Sie erhebliche Spitzen in Ihrem Datenverkehr haben, richten Sie eine Warnung bei 60 % Ihrer `ProvisionedBucketSize`-Limits ein.
- Wenn Aufrufe relativ gleichmäßig eingehen, richten Sie eine Warnung bei 60 % Ihres `ProvisionedRefillRate`-Limits für die zugehörigen API- und Entscheidungsereignisse ein.

Amazon SWF-Metriken

Die folgenden Metriken sind für Amazon SWF verfügbar:

Metrik	Description
DecisionTaskScheduleToStartTime	<p>Zeitintervall in Millisekunden zwischen der Zeit, zu der die Entscheidungsaufgabe geplant worden ist, und der Zeit, zu der sie von einem Worker abgeholt und gestartet wird</p> <p>CloudWatch Einheiten: Time</p> <p>Maße: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Gültige Statistiken: Average, Minimum, Maximum</p>
DecisionTaskStartToCloseTime	<p>Zeitintervall in Millisekunden zwischen der Start- und Abschlusszeit der Entscheidungsaufgabe</p> <p>CloudWatch Einheiten: Time</p> <p>Maße: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Gültige Statistiken: Average, Minimum, Maximum</p>
DecisionTasksCompleted	<p>Anzahl der abgeschlossenen Entscheidungs-Tasks.</p> <p>CloudWatch Einheiten: Count</p> <p>Maße: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Gültige Statistiken: Sum</p>
PendingTasks	<p>Die Anzahl der ausstehenden Aufgaben in einem 1-Minuten-Intervall für eine bestimmte Aufgabenliste.</p> <p>CloudWatch Einheiten: Count</p> <p>Maße: Domain, TaskListName</p> <p>Gültige Statistiken: Sum</p>

Metrik	Description
<code>StartedDecisionTasksTimedOutOnClose</code>	<p>Anzahl der Entscheidungs-Tasks, die begonnen wurden, aber das Zeitlimit beim Abschließen überschritten haben.</p> <p>CloudWatch Einheiten: Count</p> <p>Maße: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Gültige Statistiken: Sum</p>
<code>WorkflowStartToCloseTime</code>	<p>Zeit in Millisekunden zwischen der Start- und Abschlusszeit des Workflows</p> <p>CloudWatch Einheiten: Time</p> <p>Maße: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Gültige Statistiken: Average, Minimum, Maximum</p>
<code>WorkflowsCanceled</code>	<p>Anzahl der abgebrochenen Workflows.</p> <p>CloudWatch Einheiten: Count</p> <p>Maße: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Gültige Statistiken: Sum</p>
<code>WorkflowsCompleted</code>	<p>Anzahl der abgeschlossenen Workflows.</p> <p>CloudWatch Einheiten: Count</p> <p>Maße: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Gültige Statistiken: Sum</p>

Metrik	Description
<code>WorkflowsContinuedAsNew</code>	<p>Anzahl der Workflows, die als neue Workflows fortgesetzt worden sind.</p> <p>CloudWatch Einheiten: Count</p> <p>Maße: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Gültige Statistiken: Sum</p>
<code>WorkflowsFailed</code>	<p>Anzahl der fehlgeschlagenen Workflows</p> <p>CloudWatch Einheiten: Count</p> <p>Maße: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Gültige Statistiken: Sum</p>
<code>WorkflowsTerminated</code>	<p>Anzahl der beendeten Workflows</p> <p>CloudWatch Einheiten: Count</p> <p>Maße: Cause, Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Gültige Statistiken: Sum</p>
<code>WorkflowsTimedOut</code>	<p>Anzahl der Workflows, die aus irgendeinem Grund das Zeitlimit überschritten haben.</p> <p>CloudWatch Einheiten: Count</p> <p>Maße: Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Gültige Statistiken: Sum</p>

Metrik	Description
ActivityTaskScheduleToCloseTime	<p>Zeitintervall in Millisekunden zwischen der Zeit, zu der die Aktivität geplant worden ist, und der Abschlusszeit der Aktivität</p> <p>CloudWatch Einheiten: Time</p> <p>Maße: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Gültige Statistiken: Average, Minimum, Maximum</p>
ActivityTaskScheduleToStartTime	<p>Zeitintervall in Millisekunden zwischen der Zeit, zu der die Aktivitäts-Task geplant worden ist, und der Startzeit der Aktivitäts-Task.</p> <p>CloudWatch Einheiten: Time</p> <p>Maße: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Gültige Statistiken: Average, Minimum, Maximum</p>
ActivityTaskStartToCloseTime	<p>Zeitintervall in Millisekunden zwischen der Start- und Abschlusszeit der Aktivitätsaufgabe</p> <p>CloudWatch Einheiten: Time</p> <p>Maße: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Gültige Statistiken: Average, Minimum, Maximum</p>
ActivityTasksCancelled	<p>Anzahl der abgebrochenen Aktivitäts-Tasks.</p> <p>CloudWatch Einheiten: Count</p> <p>Maße: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Gültige Statistiken: Sum</p>

Metrik	Description
ActivityTasksCompleted	Anzahl der abgeschlossenen Aktivitäts-Tasks. CloudWatch Einheiten: Count Maße: Domain, ActivityTypeName, ActivityTypeVersion Gültige Statistiken: Sum
ActivityTasksFailed	Anzahl der fehlgeschlagenen Aktivitäts-Tasks. CloudWatch Einheiten: Count Maße: Domain, ActivityTypeName, ActivityTypeVersion Gültige Statistiken: Sum
ScheduledActivityTasksTimedOutOnClose	Anzahl der Aktivitäts-Tasks, die geplant wurden, aber das Zeitlimit beim Abschließen überschritten haben. CloudWatch Einheiten: Count Maße: Domain, ActivityTypeName, ActivityTypeVersion Gültige Statistiken: Sum
ScheduledActivityTasksTimedOutOnStart	Anzahl der Aktivitäts-Tasks, die geplant wurden, aber das Zeitlimit beim Starten überschritten haben. CloudWatch Einheiten: Count Maße: Domain, ActivityTypeName, ActivityTypeVersion Gültige Statistiken: Sum

Metrik	Description
StartedActivityTasksTimedOutOnClose	<p>Anzahl der Aktivitäts-Tasks, die begonnen wurden, aber das Zeitlimit beim Abschließen überschritten haben.</p> <p>CloudWatch Einheiten: Count</p> <p>Maße: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Gültige Statistiken: Sum</p>
StartedActivityTasksTimedOutOnHeartbeat	<p>Anzahl der Aktivitäts-Tasks, die begonnen wurden, aber das Zeitlimit infolge einer Heartbeat-Zeitüberschreitung überschritten haben.</p> <p>CloudWatch Einheiten: Count</p> <p>Maße: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Gültige Statistiken: Sum</p>
ThrottledEvents	<p>Anzahl der gedrosselten Anforderungen</p> <p>CloudWatch Einheiten: Count</p> <p>Maße: APIName, DecisionName, ThrottlingScope</p> <p>Gültige Statistiken: Sum</p>
ProvisionedBucketSize	<p>Anzahl der verfügbaren Anforderungen pro Sekunde</p> <p>Maße: APIName, DecisionName</p> <p>Gültige Statistiken: Minimum</p>

Metrik	Description
ConsumedCapacity	Anzahl der Anforderungen pro Sekunde CloudWatch Einheiten: Count Maße: APIName, DecisionName Gültige Statistiken: Sum
ConsumedLimit	Die Menge des generellen Grenzwerts, der verbraucht wurde. Maße: GeneralLimitType
ProvisionedRefillRate	Anzahl der Anforderungen pro Sekunde für den Bucket Maße: APIName, DecisionName Gültige Statistiken: Minimum
ProvisionedLimit	Der Betrag des allgemeinen Limits, der dem Konto zugewiesen wurde. Maße: GeneralLimitType

Dimension	Description
Domain	Filtert Daten in die Amazon SWF-Domain, in der der Workflow oder die Aktivität ausgeführt wird.
ActivityTypeName	Filtert Daten nach dem Namen des Aktivitätstyps
ActivityTypeVersion	Filtert Daten nach der Version des Aktivitätstyps
WorkflowTypeName	Filtert Daten nach dem Namen des Workflow-Typs für diese Workflow-Ausführung
WorkflowTypeVersion	Filtert Daten nach der Version des Workflow-Typs für diese Workflow-Ausführung

Dimension	Description
APIName	Filtert Daten nach der API mit dem angegebenen API-Namen
DecisionName	Filtert Daten nach dem angegebenen Entscheidungsnamen
TaskListName	Filtert Daten auf den angegebenen Namen der Aufgabenliste.
TaskListClassification	Filtert Daten entsprechend der Klassifizierung der Aufgabenliste. Der Wert ist „D“ für Aufgabenlisten mit Entscheidungen und „A“ für Aufgabenlisten mit Aktivitäten.
ThrottlingScope	Filtert Daten bis zum angegebenen Einschränkungsbereich. Der Wert ist „Konto“, wenn das Kontingent auf Kontoebene überschritten wird, oder „Workflow“, wenn das Kontingent auf Workflow-Ebene überschritten wird.

Namen und Abmessungen von Amazon SWF SWF-Ressourcen ohne ASCII CloudWatch

Amazon SWF erlaubt Nicht-ASCII-Zeichen in Ressourcennamen wie und. TaskList DomainName Die Dimensionswerte von CloudWatch Metriken können jedoch nur druckbare ASCII-Zeichen enthalten. Um sicherzustellen, dass Amazon SWF Dimensionswerte verwendet, die mit den [CloudWatch Anforderungen](#) kompatibel sind, werden Amazon SWF SWF-Ressourcennamen, die diese Anforderungen nicht erfüllen, konvertiert und es wird eine Prüfsumme wie folgt angehängt:

- Jedes Nicht-ASCII-Zeichen wird durch ersetzt. ?
- Die Eingabezeichenfolge oder die konvertierte Zeichenfolge wird, falls erforderlich, gekürzt. Dadurch wird sichergestellt, dass beim Anhängen der Prüfsumme die neue Zeichenkettenlänge das Maximum nicht überschreitet. CloudWatch
- Da alle Nicht-ASCII-Zeichen in konvertiert werden, können einige CloudWatch metrische Dimensionswerte?, die vor der Konvertierung unterschiedlich waren, nach der Konvertierung identisch erscheinen. Zur besseren Unterscheidung wird an den Ressourcennamen ein Unterstrich (_) gefolgt von den ersten 16 Zeichen der SHA256 Prüfsumme des ursprünglichen Ressourcennamens angehängt.

Beispiele für Konvertierungen:

- test àpplewürde umgewandelt werden in test ?pp1e_82cc5b8e3a771d12
- àòàwürde umgewandelt werden in??_2fec5edbb2c05c22.
- Die TaskList Namen àpplé und beide âpplè würden umgewandelt werden und wären identisch. ?pp1? Das Anhängen der Prüfsumme gibt unterschiedliche Werte zurück, und?pp1?_f39a36df9d85a69d. ?pp1?_da3efb4f11dd0f7f

i Tip

Sie können Ihre eigene SHA256 Prüfsumme generieren. Um beispielsweise das shasum Befehlszeilentool zu verwenden:

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

Amazon SWF-Metriken für die CloudWatch Verwendung von anzeigen AWS-Managementkonsole

Amazon CloudWatch bietet eine Reihe von sichtbaren Metriken für Amazon SWF SWF-Workflows und -Aktivitäten. Sie können die Metriken anzeigen und Alarmer für Ihre Amazon SWF SWF-Workflow-Ausführungen einrichten, indem Sie [AWS-Managementkonsole](#) Sie müssen bei der Konsole angemeldet sein, um fortzufahren.

Eine Beschreibung der verfügbaren Metriken finden Sie unter [Amazon SWF-Metriken für CloudWatch](#).

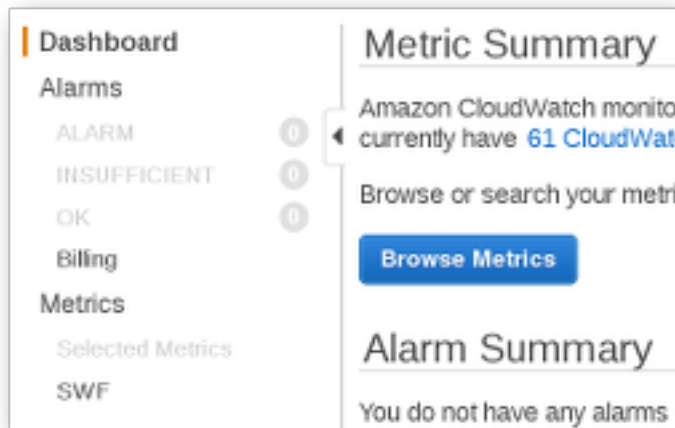
Themen

- [Anzeigen von -Metriken](#)
- [Festlegen von Alarmen](#)

Anzeigen von -Metriken

So zeigen Sie Ihre Messwerte für Amazon SWF an

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudWatch Konsole unter <https://console.aws.amazon.com/cloudwatch/>.
2. Wählen Sie im Navigationsbereich unter Metrics die Option SWF aus.



Wenn Sie vor Kurzem erst Workflow-Ausführungen gestartet haben, sehen Sie zwei Listen mit Metriken: Metriken vom Typ Workflow und Metriken vom Typ Aktivität.

Domain	WorkflowTypeName	WorkflowTypeVersion	Metric Name
HelloWorld	HelloWorldWorkflow.hello_workflow	1.0	WorkflowStartToCloseTime
HelloWorld	HelloWorldWorkflow.hello_workflow	1.0	WorkflowsCompleted

Domain	ActivityTypeName	ActivityTypeVersion	Metric Name
Booking	BookingActivity.reserve_airline	1.0	ActivityTaskScheduleToStartTime
Booking	BookingActivity.reserve_airline	1.0	ActivityTaskStartToCloseTime

Note

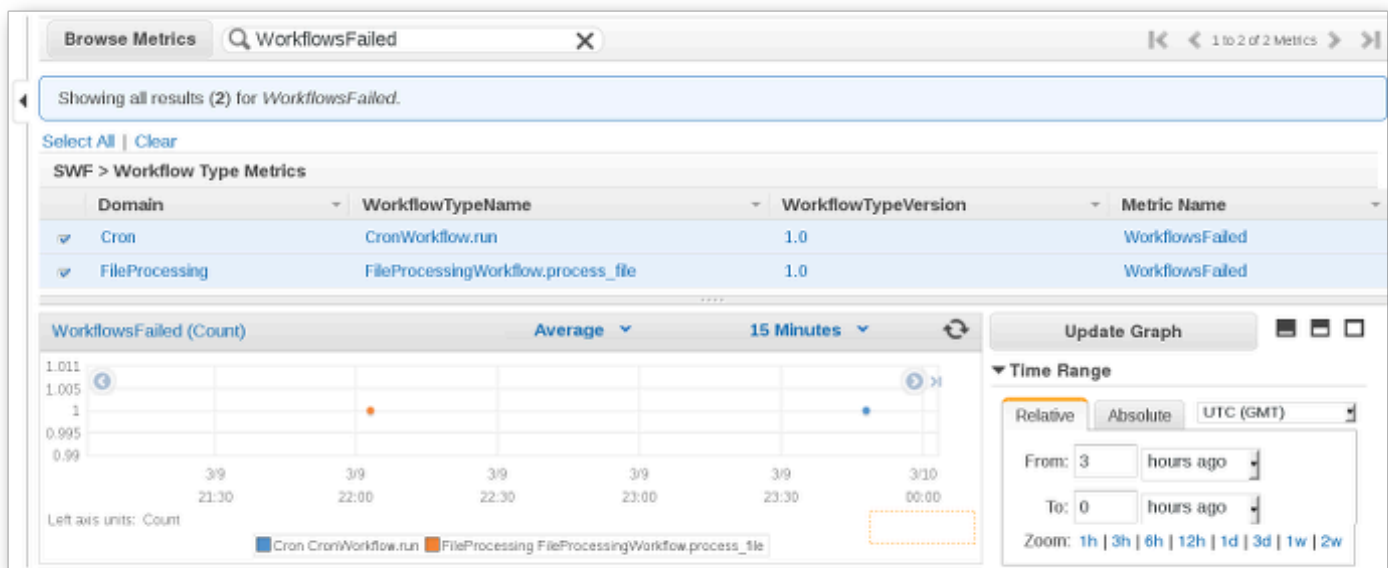
Es kann sein, dass Ihnen zuerst nur die Metriken vom Typ Workflow angezeigt werden. Die Metriken vom Typ Aktivität erscheinen in der gleichen Ansicht, ggf. müssen Sie aber nach unten scrollen, um sie zu sehen.

Es werden bis zu 50 der letzten Metriken gleichzeitig angezeigt (aufgegliedert nach Workflow- und Aktivitätsmetriken).

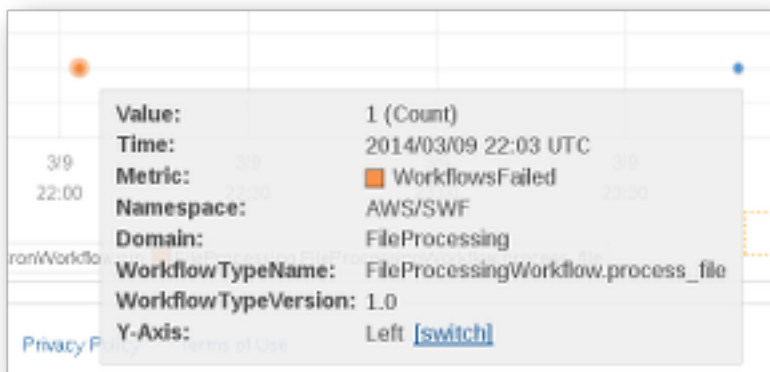
Über die interaktiven Überschriften über jeder Spalte in der Liste können Sie Ihre Metriken anhand einer der bereitgestellten Dimensionen sortieren. Für Workflows lauten die Dimensionen Domain, WorkflowTypeNameWorkflowTypeVersion, und Metric Name. Für Aktivitäten lauten die Dimensionen Domain, ActivityTypeNameActivityTypeVersion, und Metrikname.

Die verschiedenen Arten von Metriken werden unter [Amazon SWF-Metriken für CloudWatch](#) beschrieben.

Sie können Diagramme für Metriken anzeigen, indem Sie die Felder neben der Metrikzeile in der Liste auswählen und die Diagrammparameter mithilfe der Zeitraum-Steurelemente rechts neben der Diagrammansicht verwenden.



Um weitere Details zu einem Punkt auf dem Diagramm anzuzeigen, platzieren Sie den Cursor darauf. Es werden alle Details der Dimensionen des Punktes angezeigt.



Weitere Informationen zur Arbeit mit CloudWatch Metriken finden Sie unter [Metriken anzeigen, grafisch darstellen und veröffentlichen](#) im CloudWatch Amazon-Benutzerhandbuch.

Festlegen von Alarmen

Sie können CloudWatch Alarme verwenden, um Aktionen auszuführen, z. B. Sie zu benachrichtigen, wenn ein Alarmschwellenwert erreicht wird. Sie können beispielsweise einen Alarm festlegen, um eine Benachrichtigung zu einem SNS-Thema oder eine E-Mail zu senden, wenn die Metrik `WorkflowsFailed` einen bestimmten Schwellenwert überschreitet.

So legen Sie einen Alarm für eine Ihrer Metriken fest

1. Wählen Sie eine Metrik durch Auswahl des zugehörigen Felds aus.
2. Wählen Sie rechts neben dem Diagramm unter Tools die Option Create Alarm.
3. Geben Sie auf dem Bildschirm Define Alarm den Wert für den Alarmschwellenwert, die Parameter für den Zeitraum und die auszuführenden Aktionen an.

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:

Send notification to: New list

Email list:

+ Notification + AutoScaling Action + EC2 Action

Weitere Informationen zum Einstellen und Verwenden von CloudWatch Alarmen finden Sie unter [CloudWatch Amazon-Alarme erstellen](#) im CloudWatch Amazon-Benutzerhandbuch.

API-Aufrufe aufzeichnen mit AWS CloudTrail

Amazon Simple Workflow Service ist in einen Service integriert [AWS CloudTrail](#), der eine Aufzeichnung der von einem Benutzer, einer Rolle oder einem ausgeführten Aktionen bereitstellt AWS-Service. CloudTrail erfasst alle API-Aufrufe für Amazon SWF als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der Amazon SWF SWF-Konsole und Code-Aufrufe der Amazon SWF SWF-API-Operationen. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an Amazon SWF gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, den Zeitpunkt der Anfrage und weitere Details ermitteln.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anfrage mit Anmeldeinformationen des Root-Benutzers oder des Benutzers gestellt wurde.
- Die Anforderung wurde im Namen eines IAM-Identity-Center-Benutzers erstellt.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anforderung aus einem anderen AWS-Service gesendet wurde.

CloudTrail ist in Ihrem aktiv AWS-Konto , wenn Sie das Konto erstellen, und Sie haben automatisch Zugriff auf den CloudTrail Eventverlauf. Der CloudTrail Ereignisverlauf bietet eine einsehbare, durchsuchbare, herunterladbare und unveränderliche Aufzeichnung der aufgezeichneten Verwaltungsereignisse der letzten 90 Tage in einem. AWS-Region Weitere Informationen finden Sie im AWS CloudTrail Benutzerhandbuch unter [Arbeiten mit dem CloudTrail Ereignisverlauf](#). Für die Anzeige des Eventverlaufs CloudTrail fallen keine Gebühren an.

Für eine fortlaufende Aufzeichnung der Ereignisse in AWS-Konto den letzten 90 Tagen erstellen Sie einen Trail- oder [CloudTrailLake-Event-Datenspeicher](#).

CloudTrail Pfade

Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Alle mit dem erstellten Pfade AWS-Managementkonsole sind regionsübergreifend. Sie können mithilfe von AWS CLI einen Einzel-Region- oder einen Multi-Region-Trail erstellen. Es wird empfohlen, einen Trail mit mehreren Regionen zu erstellen, da Sie alle Aktivitäten AWS-

Regionen in Ihrem Konto erfassen. Wenn Sie einen Einzel-Region-Trail erstellen, können Sie nur die Ereignisse anzeigen, die im AWS-Region des Trails protokolliert wurden. Weitere Informationen zu Trails finden Sie unter [Erstellen eines Trails für Ihr AWS-Konto](#) und [Erstellen eines Trails für eine Organisation](#) im AWS CloudTrail -Benutzerhandbuch.

Sie können eine Kopie Ihrer laufenden Verwaltungsereignisse kostenlos an Ihren Amazon S3 S3-Bucket senden, CloudTrail indem Sie einen Trail erstellen. Es fallen jedoch Amazon S3 S3-Speichergebühren an. Weitere Informationen zur CloudTrail Preisgestaltung finden Sie unter [AWS CloudTrail Preise](#). Informationen zu Amazon-S3-Preisen finden Sie unter [Amazon S3 – Preise](#).

CloudTrail Datenspeicher für Ereignisse in Lake

CloudTrail Mit Lake können Sie SQL-basierte Abfragen für Ihre Ereignisse ausführen. CloudTrail [Lake konvertiert bestehende Ereignisse im zeilenbasierten JSON-Format in das Apache ORC-Format](#). ORC ist ein spaltenförmiges Speicherformat, das für den schnellen Abruf von Daten optimiert ist. Die Ereignisse werden in Ereignisdatenspeichern zusammengefasst, bei denen es sich um unveränderliche Sammlungen von Ereignissen handelt, die auf Kriterien basieren, die Sie mit Hilfe von [erweiterten Ereignisselektoren](#) auswählen. Die Selektoren, die Sie auf einen Ereignisdatenspeicher anwenden, steuern, welche Ereignisse bestehen bleiben und für Sie zur Abfrage verfügbar sind. Weitere Informationen zu CloudTrail Lake finden Sie unter [Arbeiten mit AWS CloudTrail Lake](#) im AWS CloudTrail Benutzerhandbuch.

CloudTrail Für das Speichern und Abfragen von Ereignisdaten in Lake fallen Kosten an. Beim Erstellen eines Ereignisdatenspeichers wählen Sie die [Preisoption](#) aus, die für den Ereignisdatenspeicher genutzt werden soll. Die Preisoption bestimmt die Kosten für die Erfassung und Speicherung von Ereignissen sowie die standardmäßige und maximale Aufbewahrungsdauer für den Ereignisdatenspeicher. Weitere Informationen zur CloudTrail Preisgestaltung finden Sie unter [AWS CloudTrail Preise](#).

Datenereignisse in CloudTrail

[Datenereignisse](#) liefern Informationen über die Ressourcenoperationen, die auf oder in einer Ressource ausgeführt werden (z. B. Lesen oder Schreiben in ein Amazon-S3-Objekt). Sie werden auch als Vorgänge auf Datenebene bezeichnet. Datenereignisse sind oft Aktivitäten mit hohem Volume. Protokolliert standardmäßig CloudTrail keine Datenereignisse. Der CloudTrail Ereignisverlauf zeichnet keine Datenereignisse auf.

Für Datenereignisse werden zusätzliche Gebühren fällig. Weitere Informationen zur CloudTrail Preisgestaltung finden Sie unter [AWS CloudTrail Preisgestaltung](#).

Sie können Datenereignisse für die Amazon SWF SWF-Ressourcentypen mithilfe der CloudTrail Konsole oder CloudTrail API-Operationen protokollieren. AWS CLI Weitere Informationen zum Protokollieren von Datenereignissen finden Sie unter [Protokollieren von Datenereignissen mit dem AWS-Managementkonsole](#) und [Protokollieren von Datenereignissen mit dem AWS Command Line Interface](#) im AWS CloudTrail -Benutzerhandbuch.

In der folgenden Tabelle sind die Amazon SWF SWF-Ressourcentypen aufgeführt, für die Sie Datenereignisse protokollieren können. In der Spalte Datenereignistyp wird der Wert angezeigt, den Sie in der Liste Datenereignistyp auf der CloudTrail Konsole auswählen können. In der Wertspalte `resources.type` wird der `resources.type` Wert angezeigt, den Sie angeben würden, wenn Sie erweiterte Event-Selektoren mithilfe von oder konfigurieren würden. AWS CLI CloudTrail APIs In der CloudTrail Spalte APIs Protokolierte Daten werden die API-Aufrufe angezeigt, die CloudTrail für den Ressourcentyp protokolliert wurden.

Sie können erweiterte Event-Selektoren so konfigurieren, dass sie nach den Feldern `eventName`, `readOnly` und `resources.ARN` filtern, sodass nur die Ereignisse protokolliert werden, die für Sie wichtig sind. Weitere Informationen zu diesen Kontingenten finden Sie unter [AdvancedFieldSelector](#) in der AWS CloudTrail -API-Referenz.

Typ des Datenereignisses	<code>resources.type</code> -Wert	Daten APIs wurden protokolliert CloudTrail
SWF-Domäne	<code>AWS::SWF::Domain</code>	Workflow-Ereignisse <ul style="list-style-type: none"> • CountClosedWorkflowExecutions • CountOpenWorkflowExecutions • DescribeWorkflowExecution • ListClosedWorkflowExecutions • ListOpenWorkflowExecutions • GetWorkflowExecutionHistory • RequestCancelWorkflowExecution

Typ des Datenereignisses	resources.type-Wert	Daten APIs wurden protokolliert CloudTrail
		<ul style="list-style-type: none"> • SignalWorkflowExecution • StartWorkflowExecution • TerminateWorkflowExecution <p>Aufgabenereignisse</p> <ul style="list-style-type: none"> • CountPendingActivityTasks • PollForDecisionTask • PollForActivityTask • RecordActivityTaskHeartbeat • RespondActivityTaskCanceled • RespondActivityTaskCompleted • RespondActivityTaskFailed • RespondDecisionTaskCompleted <p>Entscheidungsereignisse</p> <ul style="list-style-type: none"> • CancelTimer • CancelWorkflowExecution • CompleteWorkflowExecution • ContinueAsNewWorkflowExecution • FailWorkflowExecution • RecordMarker • RequestCancelActivityTask

Typ des Datenereignisses	resources.type-Wert	Daten APIs wurden protokolliert CloudTrail
		<ul style="list-style-type: none"> • RequestCancelExternalWorkflowExecution • ScheduleActivityTask • ScheduleLambdaFunction • SignalExternalWorkflowExecution • StartChildWorkflowExecution • StartTimer

CloudTrail Ereignisse und RespondDecisionTaskCompleted

Die [RespondDecisionTaskCompleted](#)Aktion trifft eine Liste von Entscheidungen in der Nutzlast der Anfrage. Ein abgeschlossener Aufruf gibt N+1 CloudTrail Datenereignisse aus, eines für jede Entscheidung und eines für den API-Aufruf selbst. Die Datenereignisse und das API-Ereignis haben alle dieselbe Anforderungs-ID.

Verwaltungsereignisse in CloudTrail

[Verwaltungsereignisse](#) enthalten Informationen zu Verwaltungsvorgängen, die an Ressourcen in Ihrem ausgeführt werden AWS-Konto. Sie werden auch als Vorgänge auf Steuerebene bezeichnet. CloudTrail protokolliert standardmäßig Verwaltungsereignisse.

Amazon Simple Workflow Service protokolliert die folgenden Vorgänge auf der Kontrollebene CloudTrail als Verwaltungsereignisse.

Domänenereignisse

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

- [UndeprecateDomain](#)

Aktivitätsereignisse

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

WorkflowType --Ereignisse

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

Tag-Ereignisse

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

Beispielereignis

Ein Ereignis stellt eine einzelne Anfrage aus einer beliebigen Quelle dar und enthält Informationen über den angeforderten API-Vorgang, Datum und Uhrzeit des Vorgangs, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass Ereignisse nicht in einer bestimmten Reihenfolge angezeigt werden.

Das folgende Beispiel zeigt ein CloudTrail Ereignis, das den `CountClosedWorkflowExecutions` Vorgang demonstriert.

```
{
  "eventVersion": "1.09",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "1234567890abcdef02345:admin",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/admin",
    "accountId": "111122223333",
    "accessKeyId": "abcdef01234567890abc",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "1234567890abcdef02345",
        "arn": "arn:aws:iam::111122223333:role/Admin",
        "accountId": "111122223333",
        "userName": "Admin"
      },
      "attributes": {
        "creationDate": "2023-11-23T16:37:38Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2023-11-23T17:52:46Z",
  "eventSource": "swf.amazonaws.com",
  "eventName": "CountClosedWorkflowExecutions",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "198.51.100.42",
  "userAgent": "aws-internal/3 aws-sdk-java/1.11.42",
  "requestParameters": {
    "domain": "nsg-domain",
    "closeTimeFilter": {
      "oldestDate": "Nov 23, 2023 5:52:46 PM",
      "latestDate": "Nov 23, 2023 5:52:46 PM"
    }
  },
  "responseElements": null,
  "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa",
  "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLEebbbbb",
  "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"
}
}
```

Informationen zu CloudTrail Datensatzinhalten finden Sie im AWS CloudTrail Benutzerhandbuch unter [CloudTrailDatensatzinhalt](#).

EventBridge für Änderungen des Amazon SWF SWF-Ausführungsstatus

Sie verwenden Amazon EventBridge, um auf Statusänderungen oder Ereignisse in einer AWS Ressource zu reagieren. Wenn Amazon SWF ein Ereignis ausgibt, wird es immer an den EventBridge Standardereignisbus für Ihr Konto weitergeleitet. Sie können eine Regel für Ereignisse erstellen, sie dem Standard-Event-Bus zuordnen und eine Zielaktion angeben, die ausgeführt werden soll, wenn ein Ereignis EventBridge empfangen wird, das der Regel entspricht. Auf diese Weise können Sie Ihre Workflows überwachen, ohne ständig Abfragen über die [GetWorkflowExecutionHistory](#) API durchführen zu müssen. Basierend auf Änderungen an den Workflow-Ausführungen können Sie ein EventBridge Ziel verwenden, um AWS Lambda Funktionen aufzurufen, Nachrichten zu Amazon Simple Notification Service (Amazon SNS) -Themen zu veröffentlichen und vieles mehr.

Den vollständigen Inhalt eines Ereignisses zur Änderung des Ausführungsstatus finden Sie unter [DescribeWorkflowExecution](#).

Weitere Informationen finden Sie im [EventBridge Amazon-Benutzerhandbuch](#).

EventBridge Ereignisse

Die Verlaufsereignistypen enthalten die Änderungen des Ausführungsstatus. Der `detail` Abschnitt jedes Ereignisses enthält mindestens die folgenden Parameter:

- `eventId`: die Ereignis-ID, die von `GetWorkflowExecutionHistory` angezeigt wird.
- `workflowExecutionDetail`: der Status des Workflows, als das Ereignis ausgelöst wurde.
- `eventType`: der Ereignistyp des Verlaufs, einer der folgenden:
 - `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`

Beispiele für Amazon SWF SWF-Ereignisse

Im Folgenden finden Sie Beispiele dafür, wie Amazon SWF Ereignisse sendet an EventBridge:

Themen

- [Die Ausführung wurde gestartet](#)
- [Ausführung abgeschlossen](#)
- [Die Ausführung ist fehlgeschlagen](#)
- [Bei der Ausführung wurde das Timeout überschritten](#)
- [Die Ausführung wurde beendet](#)

In jedem Fall bietet der Abschnitt `detail` in den Ereignisdaten die gleichen Informationen wie die [DescribeWorkflowExecution](#) API. Das `executionStatus` Feld gibt den Status der Ausführung zum Zeitpunkt des Sendens des Ereignisses an, entweder `OPEN` oder `CLOSED`.

Die Ausführung wurde gestartet

```
{
  "version": "0",
  "id": "44444444444444",
  "detail-type": "Simple Workflow Execution State Change",
  "source": "aws.swf",
  "account": "44444444444444",
  "time": "2020-05-08T15:57:38Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:swf:us-east-1:44444444444444:/domain/SimpleWorkflowUserSimulator"
  ],
  "detail": {
    "eventId": 1,
    "eventType": "WorkflowExecutionStarted",
    "workflowExecutionDetail": {
      "executionInfo": {
```

```

    "execution": {
      "workflowId": "123456789012",
      "runId": "AKIAIOSFODNN7EXAMPLE"
    },
    "workflowType": {
      "name": "SimpleWorkflowUserSimulator",
      "version": "myWorkflow"
    },
    "startTimestamp": 1588953458484,
    "closeTimestamp": null,
    "executionStatus": "OPEN",
    "closeStatus": null,
    "parent": null,
    "parentExecutionArn": null,
    "tagList": null,
    "cancelRequested": false
  },
  "executionConfiguration": {
    "taskStartToCloseTimeout": "60",
    "executionStartToCloseTimeout": "1000",
    "taskList": {
      "name": "44444444444444"
    },
    "taskPriority": null,
    "childPolicy": "ABANDON",
    "lambdaRole": "arn:aws:iam::444444444444:role/BasicSWFLambdaExecution"
  },
  "openCounts": {
    "openActivityTasks": 0,
    "openDecisionTasks": 1,
    "openTimers": 0,
    "openChildWorkflowExecutions": 0,
    "openLambdaFunctions": 0
  },
  "latestActivityTaskTimestamp": null,
}
}
}

```

Ausführung abgeschlossen

```

{
  "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,
}
}
}

```

Die Ausführung ist fehlgeschlagen

```

{
  "version": "0",
  "id": "1111-2222-3333",
  "detail-type": "Simple Workflow Execution State Change",
  "source": "aws.swf",
  "account": "444455556666",
  "time": "2020-05-08T15:57:38Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:swf:us-east-1:444455556666:/domain/SimpleWorkflowUserSimulator"
  ],
  "detail": {
    "eventId": 11,
    "eventType": "WorkflowExecutionFailed",
    "workflowExecutionDetail": {
      "executionInfo": {
        "execution": {
          "workflowId": "1234-5678-9012",
          "runId": "777788889999"
        },
        "workflowType": {
          "name": "SimpleWorkflowUserSimulator",
          "version": "myWorkflow"
        },
        "startTimestamp": 1588953158481,
        "closeTimestamp": 1588953458560,
        "executionStatus": "CLOSED",
        "closeStatus": "FAILED",
        "parent": null,
        "parentExecutionArn": null,
        "tagList": null,
        "cancelRequested": false
      }
    }
  }
}

```

```

    },
    "executionConfiguration": {
      "taskStartToCloseTimeout": "60",
      "executionStartToCloseTimeout": "1000",
      "taskList": {
        "name": "1111-1111-1111"
      },
      "taskPriority": null,
      "childPolicy": "ABANDON",
      "lambdaRole": "arn:aws:iam::444455556666:role/BasicSWFLambdaExecution"
    },
    "openCounts": {
      "openActivityTasks": 0,
      "openDecisionTasks": 0,
      "openTimers": 0,
      "openChildWorkflowExecutions": 0,
      "openLambdaFunctions": 0
    },
    "latestActivityTaskTimestamp": null,
  }
}
}
}

```

Bei der Ausführung wurde das Timeout überschritten

```

{
  "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,
}
}
}

```

Die Ausführung wurde beendet

```

{
  "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,
}
}
```

Verwendung AWS-Benutzerbenachrichtigungen mit Amazon Simple Workflow Service

Sie können Lieferkanäle einrichten [AWS-Benutzerbenachrichtigungen](#), um über Amazon Simple Workflow Service-Ereignisse informiert zu werden. Sie erhalten eine Benachrichtigung, wenn ein Ereignis einer von Ihnen angegebenen Regel entspricht. Sie können Benachrichtigungen für Ereignisse über mehrere Kanäle erhalten, darunter E-Mail, [Amazon Q Developer in Chat-Anwendungen](#), Chat-Benachrichtigungen oder [AWS Console Mobile Application](#) Push-Benachrichtigungen. Sie können Benachrichtigungen auch im [Konsolen-Benachrichtigungscenter](#) anzeigen. Benutzerbenachrichtigungen unterstützt die Aggregation, wodurch die Anzahl der Benachrichtigungen, die Sie bei bestimmten Ereignissen erhalten, verringert werden kann.

Konformitätsprüfung für Amazon Simple Workflow Service

Externe Prüfer bewerten die Sicherheit und Konformität von Amazon Simple Workflow Service im Rahmen mehrerer AWS Compliance-Programme. Hierzu zählen unter anderem SOC, PCI, FedRAMP und HIPAA.

Eine Liste der AWS Services im Rahmen bestimmter Compliance-Programme finden Sie unter [AWS Services im Umfang nach Compliance-Programmen AWS](#). Allgemeine Informationen finden Sie unter [AWS Compliance-Programme AWS](#).

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter [Berichte in AWS Artifact herunterladen Berichte in AWS Artifact](#).

Ihre Compliance-Verantwortung bei der Nutzung von Amazon SWF hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung der Vorschriften unterstützen:

- Schnellstartanleitungen zu [Sicherheit und Compliance Schnellstartanleitungen](#) zu — In diesen Bereitstellungshandbüchern werden architektonische Überlegungen erörtert und Schritte für die

Implementierung von sicherheits- und Compliance-orientierten Basisumgebungen beschrieben.
AWS

- Whitepaper „[Architecting for HIPAA Security and Compliance](#)“ — In diesem Whitepaper wird beschrieben, wie Unternehmen HIPAA-konforme Anwendungen entwickeln können. AWS
- [AWS Compliance-Ressourcen](#) — Diese Sammlung von Arbeitsmapen und Leitfäden kann auf Ihre Branche und Ihren Standort zutreffen.
- [Bewertung von Ressourcen anhand von Regeln](#) im AWS Config Entwicklerhandbuch — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- [AWS Security Hub CSPM](#)— Dieser AWS Service bietet einen umfassenden Überblick über Ihren Sicherheitsstatus, sodass Sie überprüfen können AWS, ob Sie die Sicherheitsstandards und Best Practices der Branche einhalten.

Resilienz im Amazon Simple Workflow Service

Die AWS globale Infrastruktur basiert auf AWS Regionen und Availability Zones. AWS Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu AWS Regionen und Availability Zones finden Sie unter [AWS Globale Infrastruktur](#).

Zusätzlich zur AWS globalen Infrastruktur bietet Amazon SWF mehrere Funktionen, um Ihre Anforderungen an Datenstabilität und Datensicherung zu erfüllen.

Infrastruktursicherheit in Amazon Simple Workflow Service

Als verwalteter Service ist er durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS Sicherheitsdiensten und zum AWS Schutz der Infrastruktur finden Sie unter [AWS Cloud-Sicherheit](#). Informationen zum Entwerfen Ihrer AWS Umgebung unter Verwendung der bewährten Methoden für die Infrastruktursicherheit finden Sie unter [Infrastructure Protection](#) in Security Pillar AWS Well-Architected Framework.

Sie verwenden AWS veröffentlichte API-Aufrufe für den Zugriff über das Netzwerk. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Sie können diese API-Operationen von jedem Netzwerkstandort aus aufrufen, Amazon SWF unterstützt jedoch ressourcenbasierte Zugriffsrichtlinien, die Einschränkungen basierend auf der Quell-IP-Adresse beinhalten können. Sie können auch Amazon SWF SWF-Richtlinien verwenden, um den Zugriff von bestimmten Amazon Virtual Private Cloud (Amazon VPC) -Endpunkten oder bestimmten zu kontrollieren. VPCs Dadurch wird der Netzwerkzugriff auf eine bestimmte Amazon SWF SWF-Ressource effektiv nur von der spezifischen VPC innerhalb des AWS Netzwerks isoliert.

Konfiguration und Schwachstellenanalyse in Amazon Simple Workflow Service

Konfiguration und IT-Steuerung liegen in der gemeinsamen Verantwortung AWS von Ihnen, unserem Kunden. Weitere Informationen finden Sie im [Modell der AWS gemeinsamen Verantwortung](#).

Verwenden des Simple Workflow Service AWS CLI mit Amazon

Auf viele Funktionen von Amazon Simple Workflow Service kann über die zugegriffen werden AWS CLI. Das AWS CLI bietet eine Alternative zur Verwendung von Amazon SWF mit der AWS-Managementkonsole oder in einigen Fällen zur Programmierung mit der Amazon SWF SWF-API und der AWS Flow Framework.

Sie können zum Beispiel den verwenden, AWS CLI um einen neuen Workflow-Typ zu registrieren:

```
aws swf register-workflow-type --domain MyDomain --name "MySimpleWorkflow" --workflow-version "v1"
```

Sie können Ihre registrierten Workflow-Typen auch auflisten:

```
aws swf list-workflow-types --domain MyDomain --registration-status REGISTERED
```

Im Folgenden sehen Sie ein Beispiel der Standardausgabe 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"
      }
    }
  ]
}
```

```
}
```

Die Amazon SWF SWF-Befehle AWS CLI bieten die Möglichkeit, Workflow-Ausführungen zu starten und zu verwalten, Aktivitätsaufgaben abzufragen, Aufgaben-Heartbeats aufzuzeichnen und vieles mehr! Eine vollständige Liste der Amazon SWF SWF-Befehle mit Beschreibungen der verfügbaren Argumente und Beispielen zu ihrer Verwendung finden Sie unter [Amazon SWF SWF-Befehle](#) in der AWS CLI Befehlsreferenz.

Die AWS CLI Befehle orientieren sich eng an der Amazon SWF SWF-API, sodass Sie die verwenden können, AWS CLI um mehr über die zugrunde liegende Amazon SWF SWF-API zu erfahren. Sie können auch Ihre vorhandenen API-Kenntnisse nutzen, um Code zu prototypisieren oder Amazon SWF SWF-Aktionen in der Befehlszeile auszuführen.

Weitere Informationen zu AWS CLI finden Sie im [AWS Command Line Interface Benutzerhandbuch](#).

Arbeiten mit Amazon SWF APIs

Sie können nicht nur AWS SDKs die unter beschriebenen Funktionen verwenden [Entwickeln Sie mit AWS SDKs](#), sondern auch die HTTP-API direkt verwenden.

Zur Verwendung der API senden Sie HTTP-Anforderungen an den [SWF-Endpunkt](#) für die Region, die Sie für Ihre Domänen, Workflows und Aktivitäten verwenden möchten. Weitere Informationen zum Stellen von HTTP-Anfragen für Amazon SWF finden Sie unter [Übermitteln von HTTP-Anforderungen an Amazon SWF](#).

Dieser Abschnitt enthält grundlegende Informationen zur Verwendung der HTTP-API zur Entwicklung Ihrer Workflows mit Amazon SWF. Fortgeschrittenere Funktionen, wie z. B. die Verwendung von Timern, das Protokollieren CloudTrail und Markieren Ihrer Workflows, finden Sie im Abschnitt. [Grundlegende Workflow-Konzepte in Amazon SWF](#)

Themen

- [Übermitteln von HTTP-Anforderungen an Amazon SWF](#)
- [Liste der Amazon SWF SWF-Aktionen nach Kategorie](#)
- [Registrierung einer Domain bei Amazon SWF](#)
- [Timeout-Werte in Amazon SWF festlegen](#)
- [Registrieren eines Workflow-Typs bei Amazon SWF](#)
- [Registrierung eines Aktivitätstyps bei Amazon SWF](#)
- [AWS Lambda Aufgaben in Amazon SWF](#)
- [Entwicklung eines Activity Workers in Amazon SWF](#)
- [Entwicklung von Entscheidern in Amazon SWF](#)
- [Workflows in Amazon SWF starten](#)
- [Aufgabenpriorität in Amazon SWF festlegen](#)
- [Behandlung von Fehlern in Amazon SWF](#)

Übermitteln von HTTP-Anforderungen an Amazon SWF

Wenn Sie keinen der verwenden AWS SDKs, können Sie Amazon Simple Workflow Service (Amazon SWF) -Operationen über HTTP mit der POST-Anforderungsmethode ausführen. Bei der POST-

Methode müssen Sie den Vorgang im Header der Anforderung festlegen und im Anforderungstext die Daten für den Vorgang im JSON-Format angeben.

Inhalt des HTTP-Headers

Amazon SWF benötigt die folgenden Informationen im Header einer HTTP-Anfrage:

- `host` Der Amazon SWF SWF-Endpunkt.
- `x-amz-date` Sie müssen den Zeitstempel entweder im Date HTTP-Header oder im angeben `AWS x-amz-date header` (in einigen HTTP-Clientbibliotheken können Sie den Date Header nicht festlegen). Ist der Header `x-amz-date` vorhanden, ignoriert das System bei der Anforderungsauthentifizierung alle Header des Typs `Date`.

Das Datum muss in einem der folgenden drei Formate angegeben werden, wie in HTTP/1.1 RFC festgelegt:

- Sun, 06 Nov 1994 08:49:37 GMT (RFC 822, aktualisiert durch RFC 1123)
- Sunday, 06-Nov-94 08:49:37 GMT (RFC 850, abgelöst durch RFC 1036)
- Sun Nov 6 08:49:37 1994 (ANSI C `asctime()`-Format)
- `x-amzn-authorization` Die Parameter der signierten Anforderung im Format:

```
AWS3 AWSAccessKeyId=####,Algorithm=HmacSHA256, [,SignedHeaders=Header1;Header2;...]
Signature=S(StringToSign)
```

`AWS3`— Dies ist ein AWS implementierungsspezifisches Tag, das die Authentifizierungsversion angibt, die zum Signieren der Anfrage verwendet wurde (derzeit ist dieser Wert für Amazon SWF immer). `AWS3`

`AWSAccessKeyId`— Ihre AWS Zugangsschlüssel-ID.

`Algorithm`— Der Algorithmus, der zur Erstellung des HMAC-SHA-Werts von verwendet wurde `string-to-sign`, z. B. oder. `HmacSHA256 HmacSHA1`

`Signature`— Base64 (Algorithmus (`StringToSign`)). SigningKey Details dazu finden Sie unter [Berechnung der HMAC-SHA-Signatur für Amazon SWF](#).

`SignedHeaders`— (Optional) Muss, falls vorhanden, eine Liste aller HTTP-Header enthalten, die in der `HttpHeaders` kanonisierten Berechnung verwendet wurden. Die Listeneinträge müssen durch ein einzelnes Semikolon (;) (ASCII-Zeichen 59) getrennt werden.

- `x-amz-target`— Der Zieldienst der Anfrage und die Operation für die Daten im Format

```
com.amazonaws.swf.service.model.SimpleWorkflowService. + <action>
```

Beispiel:

```
com.amazonaws.swf.service.model.SimpleWorkflowService.RegisterDomain
```

- `content-type`— Der Typ muss JSON und den Zeichensatz angeben, als `application/json; charset=UTF-8`

Nachfolgend finden Sie einen Beispiel-Header für eine HTTP-Anforderung zum Erstellen einer Domäne.

```
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"}
```

Das folgende Beispiel enthält die entsprechende HTTP-Antwort.

```
HTTP/1.1 200 OK
Content-Length: 0
Content-Type: application/json
x-amzn-RequestId: 4ec4ac3f-3e16-11e1-9b11-7182192d0b57
```

HTTP-Textinhalt

Der Textkörper einer HTTP-Anforderung enthält die Daten für den Vorgang, der im Header der HTTP-Anforderung festgelegt ist. Verwenden Sie das JSON-Datenformat zur gleichzeitigen Übermittlung von Datenwerten und -strukturen. Elemente können mit der Klammerschreibweise innerhalb anderer Elemente verschachtelt werden. Das Folgende zeigt beispielsweise eine Anforderung, alle Workflow-Ausführungen aufzulisten, die zwischen zwei angegebenen Zeitpunkten gestartet wurden. Dabei wird die Unix-Zeitnotation verwendet.

```
{
  "domain": "867530901",
  "startTimeFilter":
  {
    "oldestDate": 1325376070,
    "latestDate": 1356998399
  },
  "tagFilter":
  {
    "tag": "music purchase"
  }
}
```

Beispiel für eine Amazon SWF JSON-Anfrage und -Antwort

Das folgende Beispiel zeigt eine Anfrage an Amazon SWF nach einer Beschreibung der Domain, die wir zuvor erstellt haben. Dann wird die Amazon SWF SWF-Antwort angezeigt.

HTTP-POST-Anforderung

```
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"}
```

Antwort von 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"}
}
```

Beachten Sie, dass auf das Protokoll (HTTP/1.1) ein Statuscode (200) folgt. Der Codewert 200 gibt an, dass ein Vorgang erfolgreich war.

Amazon SWF serialisiert keine Nullwerte. Wenn Ihr JSON-Parser so eingestellt ist, dass er Nullwerte für Anfragen serialisiert, ignoriert Amazon SWF sie.

Berechnung der HMAC-SHA-Signatur für Amazon SWF

Jede Anfrage an Amazon SWF muss authentifiziert werden. Sie signieren Ihre Anfragen AWS SDKs automatisch und verwalten Ihre tokenbasierte Authentifizierung. Wenn Sie jedoch eigene HTTP POST-Anforderungen schreiben möchten, müssen Sie einen `x-amzn-authorization`-Wert für den HTTP POST Header-Inhalt als Teil Ihrer Anforderungsauthentifizierung erstellen.

Weitere Informationen zum Formatieren von Headern finden Sie unter [Inhalt des HTTP-Headers](#). Informationen zur AWS SDK für Java Implementierung der Signierung von AWS Version 3 finden Sie in der [AWSSigner.java-Klasse](#).

Erstellen einer Anforderungssignatur

Bevor Sie eine HMAC-SHA-Anforderungssignatur erstellen, müssen Sie Ihre AWS - Anmeldeinformationen (Zugriffsschlüssel-ID und geheimer Schlüssel) abrufen.

Important

Sie können entweder SHA1 oder verwenden, um Ihre Anfragen SHA256 zu signieren. Stellen Sie jedoch sicher, dass Sie im gesamten Signaturprozess die gleiche Methode verwenden. Die jeweilige Methode muss mit dem Wert des `Algorithm`-Namens im HTTP-Header übereinstimmen.

So erstellen Sie die Anforderungssignatur

1. Erstellen Sie eine kanonische Form der HTTP-Anforderungsheader. Die kanonische Form des HTTP-Headers umfasst Folgendes:
 - `host`
 - Ein beliebiges Header-Element, das mit `x-amz-` beginnt

Weitere Informationen zu diesen Headern finden Sie unter [Inhalt des HTTP-Headers](#).

- a. Konvertieren Sie für jedes Header-Name/Wert-Paar den Header-Namen (nicht den Header-Wert) in Kleinbuchstaben.
- b. Erstellen Sie eine Zuordnung des Header-Namens zu durch Komma getrennten Header-Werten.

```
x-amz-example: value1
x-amz-example: value2 => x-amz-example:value1,value2
```

Weitere Informationen finden Sie unter [Section 4.2 of RFC 2616](#).

- c. Konvertieren Sie für jedes Header-Name/Wert-Paar das Name/Wert-Paar in eine Zeichenfolge im Format `headerName:headerValue`. Schneiden Sie Leerzeichen am Anfang und am Ende von `headerName` und `headerValue` ab, sodass keine Leerzeichen vor und nach dem Doppelpunkt stehen.

```
x-amz-example1:value1,value2
x-amz-example2:value3
```


- d. Fügen Sie nach jeder konvertierten Zeile, einschließlich der letzten, eine neue Zeile (U+000A) ein.
 - e. Sortieren Sie die konvertierten Zeichenfolgen alphabetisch nach dem Headernamen.
2. Erstellen Sie einen string-to-sign Wert, der die folgenden Elemente enthält:
 - Zeile 1: Die HTTP-Methode (POST), gefolgt von einem Zeilenumbruch.
 - Zeile 2: Die Anforderungs-URI (/), gefolgt von einem Zeilenumbruch.
 - Zeile 3: Eine leere Zeichenfolge, gefolgt von einem Zeilenumbruch.

Note

Normalerweise wird die Abfragezeichenfolge hier angezeigt, Amazon SWF verwendet jedoch keine Abfragezeichenfolge.

- Zeilen 4–n: Die Zeichenfolge, die die kanonisierten Anforderungs-Header darstellt, die Sie in Schritt 1 berechnet haben, gefolgt von einem Zeilenumbruch. Mit diesem Zeilenumbruch wird eine leere Zeile zwischen den Headern und dem Text der HTTP-Anforderung eingefügt. Weitere Informationen finden Sie unter [RFC 2616](#).
 - Der Anforderungstext, nicht gefolgt von einem Zeilenumbruch.
3. Berechnet den SHA256 oder SHA1 Digest des string-to-sign Werts. Verwenden Sie die gleiche SHA-Methode für den gesamten Prozess.
 4. Berechnen und Base64-kodieren Sie den HMAC-SHA mithilfe eines SHA256 oder eines SHA1 Digest (je nach verwendeter Methode) des Werts, der sich aus dem vorherigen Schritt ergibt, und

des temporären geheimen Zugriffsschlüssels aus dem Security Token Service mithilfe der AWS API-Aktion. [GetSessionToken](#)

 Note

Amazon SWF erwartet ein Gleichheitszeichen (=) am Ende des Base64-codierten HMAC-SHA-Werts. Wenn Ihre Base64-Codierung kein angefügtes Gleichheitszeichen umfasst, fügen Sie ein solches Zeichen am Ende des Werts an.

Weitere Informationen zur Verwendung temporärer Sicherheitsanmeldedaten mit Amazon SWF und anderen AWS Diensten finden Sie unter [AWS Services That Work with IAM](#) im IAM-Benutzerhandbuch.

5. Platzieren Sie den resultierenden Wert als Wert für den Signature Namen im `x-amzn-authorization` Header der HTTP-Anfrage an Amazon SWF.
6. Amazon SWF verifiziert die Anfrage und führt den angegebenen Vorgang aus.

Liste der Amazon SWF SWF-Aktionen nach Kategorie

In diesem Abschnitt werden die Referenzthemen für Amazon SWF SWF-Aktionen in der Amazon SWF SWF-API (Application Programming Interface) aufgeführt. Diese werden nach der funktionalen Kategorie angegeben.

Eine alphabetische Liste der Aktionen finden Sie in der [Amazon Simple Workflow Service API-Referenz](#).

Themen

- [Aktionen im Zusammenhang mit Aktivitäten](#)
- [Aktionen im Zusammenhang mit Entscheidern](#)
- [Aktionen im Zusammenhang mit Workflow-Ausführungen](#)
- [Aktionen im Zusammenhang mit der Administration](#)
- [Sichtbarkeitsaktionen](#)

Aktionen im Zusammenhang mit Aktivitäten

Aktivitäts-Worker verwenden `PollForActivityTask`, um neue Aktivitäts-Tasks abzurufen. Nachdem ein Worker einen Aktivitäts-Task aus Amazon SWF empfangen hat, führt er den Task aus und antwortet mit `RespondActivityTaskCompleted`, wenn die Aufgabe erfolgreich ausgeführt wurde bzw. mit `RespondActivityTaskFailed`, wenn ein Fehler aufgetreten ist.

Die folgenden Aktionen werden von Aktivitäts-Workern ausgeführt.

- [PollForActivityTask](#)
- [RespondActivityTaskCompleted](#)
- [RespondActivityTaskFailed](#)
- [RespondActivityTaskCanceled](#)
- [RecordActivityTaskHeartbeat](#)

Aktionen im Zusammenhang mit Entscheidern

Entscheider nutzen `PollForDecisionTask`, um Entscheidungs-Tasks abzurufen. Wenn ein Entscheider eine Entscheidungs-Aufgabe aus Amazon SWF erhält, untersucht er dessen Workflow-Ausführungsverlauf und entscheidet, was als Nächstes zu tun ist. Er ruft `RespondDecisionTaskCompleted` auf, um den Entscheidungs-Task abzuschließen, und stellt null oder mehr weitere Entscheidungen bereit.

Die folgenden Aktionen werden von Entscheidern ausgeführt.

- [PollForDecisionTask](#)
- [RespondDecisionTaskCompleted](#)

Aktionen im Zusammenhang mit Workflow-Ausführungen

Die folgenden Aktionen werden für eine Workflow-Ausführung verwendet.

- [RequestCancelWorkflowExecution](#)
- [StartWorkflowExecution](#)
- [SignalWorkflowExecution](#)
- [TerminateWorkflowExecution](#)

Aktionen im Zusammenhang mit der Administration

Sie können Verwaltungsaufgaben zwar von der Amazon SWF SWF-Konsole aus ausführen, aber Sie können die Aktionen in diesem Abschnitt verwenden, um Funktionen zu automatisieren oder Ihre eigenen Verwaltungstools zu erstellen.

Aktivitätsverwaltung

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

Workflow-Verwaltung

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

Domänenverwaltung

Mit diesen Aktionen können Sie Amazon SWF-Domains registrieren und als veraltet markieren.

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

Weitere Informationen und Beispiele zu diesen Domänenverwaltungsaktionen finden Sie unter [Registrierung einer Domain bei Amazon SWF](#).

Workflow-Ausführungsverwaltung

- [RequestCancelWorkflowExecution](#)
- [TerminateWorkflowExecution](#)

Sichtbarkeitsaktionen

Sie können zwar Sichtbarkeitsaktionen von der Amazon SWF SWF-Konsole aus ausführen, aber Sie können die Aktionen in diesem Abschnitt verwenden, um Ihre eigene Konsole oder Verwaltungstools zu erstellen.

Aktivitätssichtbarkeit

- [ListActivityTypes](#)
- [DescribeActivityType](#)

Workflow-Sichtbarkeit

- [ListWorkflowTypes](#)
- [DescribeWorkflowType](#)

Workflow-Ausführungssichtbarkeit

- [DescribeWorkflowExecution](#)
- [ListOpenWorkflowExecutions](#)
- [ListClosedWorkflowExecutions](#)
- [CountOpenWorkflowExecutions](#)
- [CountClosedWorkflowExecutions](#)
- [GetWorkflowExecutionHistory](#)

Domänensichtbarkeit

- [ListDomains](#)
- [DescribeDomain](#)

Task-Listensichtbarkeit

- [CountPendingActivityTasks](#)
- [CountPendingDecisionTasks](#)

Registrierung einer Domain bei Amazon SWF

Ihr Workflow und die Aktivitätstypen sowie die eigentliche Workflow-Ausführung sind auf eine Domäne beschränkt. Domänen trennen eine Reihe von Typen, Ausführungen und Aufgabenliste von anderen innerhalb des gleichen Kontos.

Sie können eine Domain registrieren, indem Sie die `RegisterDomain` Aktion AWS-Managementkonsole oder in der Amazon SWF SWF-API verwenden. Im folgenden Beispiel wird API verwendet.

```
https://swf.us-east-1.amazonaws.com
RegisterDomain
{
  "name" : "867530901",
  "description" : "music",
  "workflowExecutionRetentionPeriodInDays" : "60"
}
```

Die Parameter werden im Format JavaScript Object Notation (JSON) angegeben. Der Aufbewahrungszeitraum beträgt 60 Tage. Während des Aufbewahrungszeitraums sind alle Informationen über die Workflow-Ausführung über Sichtbarkeitsoperationen verfügbar, die entweder die Amazon SWF API AWS-Managementkonsole oder die Amazon SWF API verwenden.

Nach dem Registrieren der Domäne sollten Sie den Workflow-Typ und die vom Workflow verwendeten Aktivitätstypen registrieren. Sie müssen die Domäne zuerst registrieren, da der Name einer registrierten Domäne beim Erstellen des Workflows und der Aktivitätstypen erforderlich ist.

Weitere Informationen finden Sie unter:

[RegisterDomain](#) in der Amazon Simple Workflow Service API-Referenz

Timeout-Werte in Amazon SWF festlegen

Themen

- [Kontingente für Timeout-Werte](#)
- [Timeouts für Workflow-Ausführungen und Entscheidungsaufgaben](#)
- [Timeouts für Aktivitätsaufgaben](#)
- [Weitere Informationen finden Sie unter:](#)

Kontingente für Timeout-Werte

Timeout-Werte werden immer in Sekunden deklariert und können auf eine beliebige Anzahl von Sekunden bis zu einem Jahr (31536000 Sekunden) festgelegt werden — das maximale Ausführungslimit für jeden Workflow oder jede Aktivität. Der spezielle Wert NONE wird verwendet, um einen Timeout-Parameter so festzulegen, dass es kein Timeout gibt, oder um ihn auf endlos festzulegen. Das maximale Limit von einem Jahr gilt jedoch weiterhin.

Timeouts für Workflow-Ausführungen und Entscheidungsaufgaben

Sie können Timeout-Werte für Ihren Workflow und Entscheidungsaufgaben festlegen, wenn Sie den Workflow-Typen registrieren. Zum Beispiel:

```
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"
}
```

Diese Registrierung des Workflow-Typs legt [defaultTaskStartToCloseTimeout](#) auf 600 Sekunden (10 Minuten) und [defaultExecutionStartToCloseTimeout](#) auf 3 600 Sekunden (1 Stunde) fest.

Weitere Informationen zur Registrierung von Workflow-Typen finden Sie unter [Registrieren eines Workflow-Typs bei Amazon SWF](#) und [RegisterWorkflowType](#) in der Amazon Simple Workflow Service API-Referenz.

Sie können den für `defaultExecutionStartToCloseTimeout` festgelegten Wert überschreiben, indem Sie [executionStartToCloseTimeout](#) angeben.

Timeouts für Aktivitätsaufgaben

Sie können Timeout-Werte für Ihre Aktivitätsaufgaben festlegen, wenn Sie den Aktivitätstyp registrieren. Zum Beispiel:

```
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"
}
```

Diese Aktivitätstypregistrierung legt [defaultTaskStartToCloseTimeout](#) auf 600 Sekunden (10 Minuten), [defaultTaskHeartbeatTimeout](#) auf 120 Sekunden (2 Minuten), [defaultTaskScheduleToStartTimeout](#) auf 1 800 Sekunden (30 Minuten) und [defaultTaskScheduleToCloseTimeout](#) auf 5 400 Sekunden (1,5 Stunden) fest.

Weitere Informationen zur Registrierung von Aktivitätstypen finden Sie unter [Registrierung eines Aktivitätstyps bei Amazon SWF](#) und [RegisterActivityType](#) in der Amazon Simple Workflow Service API-Referenz.

Sie können den für `defaultTaskStartToCloseTimeout` festgelegten Wert überschreiben, indem Sie bei der Planung der Aktivitätsaufgabe [taskStartToCloseTimeout](#) angeben.

Weitere Informationen finden Sie unter:

[Amazon SWF-Timeout-Typen](#)

Registrieren eines Workflow-Typs bei Amazon SWF

Das in diesem Abschnitt besprochene Beispiel registriert einen Workflow-Typ mithilfe der Amazon SWF SWF-API. Der Name und die Version, die Sie während der Registrierung angeben, bilden einen eindeutigen Bezeichner für den Workflow-Typ. Die angegebene Domäne muss bereits mithilfe der [RegisterDomain](#)-API-Aktion registriert worden sein.

Die Timeout-Parameter im folgenden Beispiel sind Werte für die Dauer, angegeben in Sekunden. Für den `defaultTaskStartToCloseTimeout`-Parameter können Sie den Dauerspezifizierer NONE

verwenden, um anzugeben, dass es keine Zeitüberschreitung gibt. Sie können jedoch nicht den Wert `NONE` für `defaultExecutionStartToCloseTimeout` festlegen. Es besteht ein maximales Limit von einem Jahr für den Zeitraum, in dem eine Workflow-Ausführung ausgeführt werden kann. Eine Überschreitung dieses Limits führt immer zu einem Timeout der Workflow-Ausführung. Wenn Sie für `defaultExecutionStartToCloseTimeout` einen Wert festlegen, der größer ist als ein Jahr, schlägt die Registrierung fehl.

```
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"
}
```

Weitere Informationen finden Sie unter:

[RegisterWorkflowType](#) in der Amazon Simple Workflow Service API-Referenz

Registrierung eines Aktivitätstyps bei Amazon SWF

Das folgende Beispiel registriert einen Aktivitätstyp mithilfe der Amazon SWF SWF-API. Der Name und die Version, die Sie während der Registrierung angeben, bilden einen eindeutigen Bezeichner für den Aktivitätstyp innerhalb der Domäne. Die angegebene Domäne muss bereits mithilfe der Aktion `RegisterDomain` registriert worden sein.

Die Zeitüberschreitungsparameter in diesem Beispiel sind Werte für die Dauer, angegeben in Sekunden. Sie können den Dauerspezifizierer `NONE` verwenden, um anzugeben, dass es keine Zeitüberschreitung gibt.

```
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"
}
```

Weitere Informationen finden Sie unter:

[RegisterActivityType](#) in der Amazon Simple Workflow Service API-Referenz

AWS Lambda Aufgaben in Amazon SWF

Themen

- [Über AWS Lambda](#)
- [Vorteile und Einschränkungen der Verwendung von Lambda-Aufgaben](#)
- [Verwenden von Lambda-Aufgaben in Ihren Workflows](#)

Über AWS Lambda

AWS Lambda ist ein vollständig verwalteter Rechenservice, der Ihren Code als Reaktion auf Ereignisse ausführt, die durch benutzerdefinierten Code oder durch verschiedene AWS Dienste wie Amazon S3, DynamoDB, Amazon Kinesis, Amazon SNS und Amazon Cognito generiert wurden. Weitere Informationen zu Lambda finden Sie im [AWS Lambda Entwicklerhandbuch](#).

Amazon Simple Workflow Service bietet eine Lambda-Aufgabe, sodass Sie Lambda-Funktionen anstelle von oder zusammen mit herkömmlichen Amazon SWF SWF-Aktivitäten ausführen können.

Important

Ihr AWS Konto wird für Lambda-Ausführungen (Anfragen) belastet, die von Amazon SWF in Ihrem Namen ausgeführt werden. Einzelheiten zu den Lambda-Preisen finden Sie unter <https://aws.amazon.com/lambda/pricing/>.

Vorteile und Einschränkungen der Verwendung von Lambda-Aufgaben

Die Verwendung von Lambda-Aufgaben anstelle einer herkömmlichen Amazon SWF SWF-Aktivität bietet eine Reihe von Vorteilen:

- Lambda-Aufgaben müssen nicht wie Amazon SWF SWF-Aktivitätstypen registriert oder versioniert werden.
- Sie können alle vorhandenen Lambda-Funktionen verwenden, die Sie bereits in Ihren Workflows definiert haben.
- Lambda-Funktionen werden direkt von Amazon SWF aufgerufen. Sie müssen kein Worker-Programm implementieren, um sie auszuführen, wie dies bei herkömmlichen Aktivitäten der Fall ist.
- Lambda stellt Ihnen Metriken und Protokolle zur Verfügung, mit denen Sie Ihre Funktionsausführungen verfolgen und analysieren können.

Bei Lambda-Aufgaben sind jedoch einige Einschränkungen zu beachten:

- Lambda-Aufgaben können nur in AWS Regionen ausgeführt werden, die Lambda unterstützen. Einzelheiten zu den derzeit unterstützten [Regionen für Lambda finden Sie unter Lambda Regions and Endpoints](#) in der Amazon Web Services General Reference.
- Lambda-Aufgaben werden derzeit nur von der SWF-Basis-SWF-HTTP-API und in der AWS Flow Framework für Java unterstützt. Derzeit gibt es keine Unterstützung für Lambda-Aufgaben in der AWS Flow Framework für Ruby.

Verwenden von Lambda-Aufgaben in Ihren Workflows

Um Lambda-Aufgaben in Ihren Amazon SWF SWF-Workflows zu verwenden, müssen Sie:

1. Richten Sie IAM-Rollen ein, um Amazon SWF die Berechtigung zum Aufrufen von Lambda-Funktionen zu gewähren.
2. die IAM-Rollen Ihren Workflows anfügen.
3. Ihre Lambda-Funktion während der Workflow-Ausführung aufrufen.

Einrichten einer IAM-Rolle

Bevor Sie Lambda-Funktionen von Amazon SWF aufrufen können, müssen Sie eine IAM-Rolle bereitstellen, die den Zugriff auf Lambda von Amazon SWF aus ermöglicht. Führen Sie dazu einen der folgenden Schritte aus:

- Wählen Sie eine vordefinierte Rolle, `AWSLambdaRole`, um Ihren Workflows die Erlaubnis zu geben, alle Lambda-Funktionen aufzurufen, die mit Ihrem Konto verknüpft sind.
- Definieren Sie Ihre eigene Richtlinie und die zugehörige Rolle, um Workflows die Erlaubnis zu erteilen, bestimmte Lambda-Funktionen aufzurufen, die durch ihre Amazon-Ressourcennamen () ARNs spezifiziert sind.

Beschränken Sie die Berechtigungen für eine IAM-Rolle

Sie können die Berechtigungen für eine IAM-Rolle, die Sie Amazon SWF zur Verfügung stellen, einschränken, indem Sie die `SourceAccount` Kontextschlüssel `SourceArn` und in Ihrer `Resource Trust Policy` verwenden. Diese Schlüssel schränken die Verwendung einer IAM-Richtlinie ein, sodass sie nur für Amazon Simple Workflow Service-Ausführungen verwendet wird, die zum angegebenen Domain-ARN gehören. Wenn Sie beide Kontextschlüssel für globale Bedingungen verwenden, müssen der `aws:SourceAccount` Wert und das Konto, auf das im `aws:SourceArn` Wert verwiesen wird, dieselbe Konto-ID verwenden, wenn sie in derselben Richtlinienerklärung verwendet werden.

Im folgenden Beispiel schränkt der `SourceArn` Kontextschlüssel die IAM-Servicerolle so ein, dass sie nur in Amazon Simple Workflow Service-Ausführungen verwendet wird, die zu dem Konto `someDomain` gehören, `123456789012`

- Aussage 1

Schulleiter: `"Service": "swf.amazonaws.com"`

Aktion: `sts:AssumeRole`

```
"Condition": {
  "ArnLike": {
    "aws:SourceArn": "arn:aws:swf:*:123456789012:/domain/someDomain"
  }
}
```

Im folgenden Beispiel schränkt der SourceAccount Kontextschlüssel die IAM-Servicerolle so ein, dass sie nur in Amazon Simple Workflow Service-Ausführungen im Konto, verwendet wird. 123456789012

```
"Condition": {
  "StringLike": {
    "aws:SourceAccount": "123456789012"
  }
}
```

Amazon SWF Zugriff zum Aufrufen beliebiger Lambda-Rollen gewähren

Sie können die vordefinierte Rolle Rolle verwenden, AWSLambda um Ihren Amazon SWF SWF-Workflows die Möglichkeit zu geben, jede Lambda-Funktion aufzurufen, die mit Ihrem Konto verknüpft ist.

So verwenden Sie AWSLambda Role, um Amazon SWF Zugriff zum Aufrufen von Lambda-Funktionen zu gewähren

1. Öffnen Sie die [Amazon IAM-Konsole](#).
2. Wählen Sie Roles und anschließend Create New Role aus.
3. Geben Sie einen Namen für die Rolle ein, z. B. swf-lambda, und klicken Sie auf Next Step.
4. Wählen Sie unter AWS Service Roles Amazon SWF und dann Next Step aus.
5. Wählen Sie auf dem Bildschirm „Richtlinie anhängen“ die Option AWSLambdaRolle aus der Liste aus.
6. Klicken Sie auf Next Step und auf Create Role, sobald Sie die Rolle überprüft haben.


Definition einer IAM-Rolle für den Zugriff auf den Aufruf einer bestimmten Lambda-Funktion

Wenn Sie Zugriff zum Aufrufen einer bestimmten Lambda-Funktion aus Ihrem Workflow gewähren möchten, müssen Sie Ihre eigene IAM-Richtlinie definieren.

Um eine IAM-Richtlinie zu erstellen, um Zugriff auf eine bestimmte Lambda-Funktion zu gewähren

1. Öffnen Sie die [Amazon IAM-Konsole](#).
2. Wählen Sie Policies und dann Create Policy aus.

3. Wählen Sie „AWS Verwaltete Richtlinie kopieren“ und wählen Sie „AWSLambdaRolle“ aus der Liste aus. Es wird eine Richtlinie erstellt. Sie können ihren Namen und die Beschreibung nach Bedarf ändern.
4. Fügen Sie im Feld Ressource des Richtliniendokuments den ARN Ihrer Lambda-Funktion (en) hinzu. Zum Beispiel:
 - Ressource: `arn:aws:lambda:us-east-1:111122223333:function:hello_lambda_function`

 Note

Eine vollständige Beschreibung der Angabe von Ressourcen in einer IAM-Rolle finden Sie unter [Überblick über IAM-Richtlinien in Using IAM](#).

5. Wählen Sie Create policy aus, um Ihre Richtlinie zu erstellen.

Sie können diese Richtlinie dann auswählen, wenn Sie eine neue IAM-Rolle erstellen, und diese Rolle verwenden, um Aufrufzugriff auf Ihre Amazon SWF SWF-Workflows zu gewähren. Dieses Verfahren ist dem Erstellen einer Rolle mit der Rollenrichtlinie sehr ähnlich. Wählen Sie stattdessen Ihre eigene Richtlinie, wenn Sie die AWSLambdaRolle erstellen.

So erstellen Sie eine Amazon SWF SWF-Rolle mithilfe Ihrer Lambda-Richtlinie

1. Öffnen Sie die [Amazon IAM-Konsole](#).
2. Wählen Sie Roles und anschließend Create New Role aus.
3. Geben Sie einen Namen für die Rolle ein, z. B. `swf-lambda-function`, und klicken Sie auf Next Step.
4. Wählen Sie unter AWS Service Roles Amazon SWF und dann Next Step aus.
5. Wählen Sie auf dem Bildschirm Attach Policy (Richtlinie anhängen) Ihre funktionspezifische Lambda-Richtlinie aus der Liste aus.
6. Klicken Sie auf Next Step und auf Create Role, sobald Sie die Rolle überprüft haben.

Anfügen der IAM-Rolle an Ihren Workflow

Nachdem Sie Ihre IAM-Rolle definiert haben, müssen Sie sie an den Workflow anhängen, der sie zum Aufrufen der Lambda-Funktionen verwendet, auf die Sie Amazon SWF Zugriff gewährt haben.

Sie können die Rolle an zwei Stellen an Ihren Workflow anfügen.

- Während der Workflow-Typ registriert wird. Diese Rolle kann als die Standard-Lambda-Rolle für jede Ausführung dieses Workflow-Typs verwendet werden.
- Beim Start einer Workflow-Ausführung. Diese Rolle wird nur während der Ausführung dieses Workflows (und während der gesamten Ausführung) verwendet.

So stellen Sie eine Standard-Lambda-Rolle für einen Workflow-Typ bereit

- Stellen Sie das `defaultLambdaRole` Feld beim Aufrufen `RegisterWorkflowType` auf den ARN der Rolle ein, die Sie definiert haben.

So stellen Sie eine Lambda-Rolle während der Workflow-Ausführung bereit

- Stellen Sie beim Aufrufen `StartWorkflowExecution` das Feld `LambdaRole` auf den ARN der Rolle ein, die Sie definiert haben.

Note

Wenn das Konto aufruft `RegisterWorkflowType` oder `StartWorkflowExecution` nicht berechtigt ist, die angegebene Rolle zu verwenden, schlägt der Aufruf mit einer `OperationNotPermittedFault` fehl.

Rufen Sie Ihre Lambda-Funktion von einem Amazon SWF SWF-Workflow aus auf

Sie können den `ScheduleLambdaFunctionDecisionAttributes` Datentyp verwenden, um die Lambda-Funktion zu identifizieren, die während einer Workflow-Ausführung aufgerufen werden soll.

Stellen Sie während eines Aufrufs eine Liste `ScheduleLambdaFunctionDecisionAttributes` mit Ihren Entscheidungen bereit. `RespondDecisionTaskCompleted` Zum Beispiel:

```
{
  "decisions": [{
    "ScheduleLambdaFunctionDecisionAttributes": {
      "id": "lambdaTaskId",
      "name": "myLambdaFunctionName",
      "input": "inputToLambdaFunction",
```

```
    "startToCloseTimeout": "30"  
  },  
}],  
}
```

Legen Sie die folgenden Parameter fest:

- `id` mit einem Bezeichner für die Lambda-Aufgabe. Dies muss eine Zeichenfolge mit 1 bis 256 Zeichen sein und darf weder die Zeichen : (Doppelpunkt), / (Schrägstrich), | (senkrechter Strich), noch Steuerzeichen (`\u0000` - `\u001f` und `\u007f` - `\u009f`) oder die Literalzeichenfolge `arn` enthalten.
- `name` mit dem Namen Ihrer Lambda-Funktion. Ihr Amazon SWF SWF-Workflow muss mit einer IAM-Rolle ausgestattet sein, die ihm Zugriff auf den Aufruf der Lambda-Funktion gewährt. Der angegebene Name muss den Einschränkungen für den `FunctionName` Parameter entsprechen, wie in der Lambda Invoke-Aktion.
- `input` mit optionalen Eingabedaten für die Funktion. Falls gesetzt, muss dies den Einschränkungen für den `ClientContext` Parameter entsprechen, wie in der Lambda Invoke-Aktion.
- `startToCloseTimeout` mit einem optionalen Höchstzeitraum in Sekunden, den die Ausführung der Funktion benötigen kann, bevor die Aufgabe aufgrund einer Timeout-Ausnahme fehlschlägt. Der Wert `NONE` kann dafür verwendet werden, eine unbegrenzte Dauer festzulegen.

[Weitere Informationen finden Sie unter Aufgaben implementieren AWS Lambda](#)

Entwicklung eines Activity Workers in Amazon SWF

Ein Aktivitäts-Worker implementiert einen oder mehrere Aktivitätstypen. Ein Activity Worker kommuniziert mit Amazon SWF, um Aktivitätsaufgaben entgegenzunehmen und auszuführen. Sie können mehrere Aktivitäts-Worker für Aktivitätsaufgaben desselben Aktivitätstyps erstellen.

Amazon SWF stellt Aktivitätsmitarbeitern eine Aktivitätsaufgabe zur Verfügung, wenn der Entscheider die Aktivitätsaufgabe plant. Wenn ein Entscheider eine Aktivitätsaufgabe plant, stellt er die Daten bereit (die Sie festlegen), die der für die Aktivität zuständige Mitarbeiter benötigt, um die Aktivitätsaufgabe auszuführen. Amazon SWF fügt diese Daten in die Aktivitätsaufgabe ein, bevor sie an den Activity Worker gesendet werden.

Sie selbst verwalten die Aktivitäts-Worker. Sie können in einer beliebigen Programmiersprache geschrieben werden. Ein Worker kann überall ausgeführt werden, sofern er über die API mit Amazon

SWF kommunizieren kann. Da Amazon SWF alle Informationen bereitstellt, die für die Ausführung einer Aktivitätsaufgabe erforderlich sind, können alle Activity Worker staatenlos sein. Dadurch sind Ihre Workflows extrem skalierbar; um höhere Kapazitäten verarbeiten zu können, fügen Sie einfach weitere Aktivitäts-Worker hinzu.

In diesem Abschnitt wird erläutert, wie Sie einen Aktivitäts-Worker implementieren. Aktivitäts-Worker sollten folgende Aktivitäten regelmäßig ausführen.

1. Fragen Sie Amazon SWF nach einer Aktivitätsaufgabe ab.
2. Mit der Ausführung der Aufgabe beginnen
3. Melden Sie Amazon SWF regelmäßig einen Heartbeat, wenn es sich um eine langlebige Aufgabe handelt.
4. Melden Sie, dass die Aufgabe abgeschlossen oder fehlgeschlagen ist, und senden Sie die Ergebnisse an Amazon SWF zurück.

Themen

- [Abrufen von Aktivitätsaufgaben](#)
- [Verarbeiten von Aktivitätsaufgaben](#)
- [Senden von Heartbeats für Aktivitätsaufgaben](#)
- [Abschließen oder fehlschlagen einer Aktivitätsaufgabe](#)
- [Starten von Aktivitäts-Workern](#)

Abrufen von Aktivitätsaufgaben

Um Aktivitätsaufgaben auszuführen, muss jeder Activity Worker Amazon SWF abfragen, indem er die `PollForActivityTask` Aktion regelmäßig aufruft.

Im folgenden Beispiel sendet der Aktivitäts-Worker `ChargeCreditCardWorker01` eine Anfrage für die erste Aufgabe in der Aufgabenliste `ChargeCreditCard-v0.1`. Wenn keine Aktivitätsaufgaben verfügbar sind, sendet Amazon SWF nach 60 Sekunden eine leere Antwort zurück. Eine leere Antwort ist eine Task-Struktur, in der der Wert von `taskToken` eine leere Zeichenfolge ist.

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

```
"domain" : "867530901",  
"taskList" : { "name": "ChargeCreditCard-v0.1" },  
"identity" : "ChargeCreditCardWorker01"  
}
```

Wenn eine Aktivitätsaufgabe verfügbar wird, gibt Amazon SWF sie an den Activity Worker zurück. Die Aufgabe enthält die Daten, die der Entscheider beim Planen der Aktivität festgelegt hat.

Sobald ein Aktivitäts-Worker eine Aktivitätsaufgabe empfängt, kann er mit der Verarbeitung beginnen. Im nächsten Abschnitt wird die Verarbeitung von Aktivitätsaufgaben beschrieben.

Verarbeiten von Aktivitätsaufgaben

Nachdem ein Aktivitäts-Worker eine Aktivitätsaufgabe empfangen hat, kann er mit der Verarbeitung beginnen.

So verarbeiten Sie eine Aktivitätsaufgabe

1. Programmieren Sie ihren Aktivitäts-Worker so, dass der Inhalt im Eingabefeld der Aufgabe analysiert wird. Dieses Feld enthält die Daten, die der Entscheider beim Planen der Aktivität festgelegt hat.
2. Programmieren Sie den Aktivitäts-Worker so, dass er mit der Verarbeitung der Daten und der Ausführung der programmierten Logik beginnt.

Im nächsten Abschnitt wird beschrieben, wie Sie Ihre Activity Worker so programmieren, dass sie Statusaktualisierungen für Amazon SWF für lang andauernde Aktivitäten bereitstellen.

Senden von Heartbeats für Aktivitätsaufgaben

Wenn für einen Aktivitätstyp eine Heartbeat-Zeitüberschreitung registriert wurde, muss der Aktivitäts-Worker einen Heartbeat senden, bevor die Zeitüberschreitung abläuft. Wenn eine Aktivitätsaufgabe innerhalb des Timeouts keinen Heartbeat liefert, tritt bei der Aufgabe ein Timeout auf. Amazon SWF schließt sie und plant eine neue Entscheidungsaufgabe, um einen Entscheider über den Timeout zu informieren. Der Entscheider kann die Aktivitätsaufgabe dann neu planen oder eine andere Aktion ausführen.

Wenn der Activity Worker nach einem Timeout versucht, Amazon SWF zu kontaktieren, z. B. durch einen `AnrufRespondActivityTaskCompleted`, gibt Amazon SWF einen `UnknownResource` Fehler zurück.

In diesem Abschnitt wird erläutert, wie Sie einen Aktivitäts-Heartbeat senden.

Um einen Heartbeat für eine Aktivitätsaufgabe aufzuzeichnen, programmieren Sie Ihren Aktivitäts-Worker so, dass die Aktion `RecordActivityTaskHeartbeat` aufgerufen wird. Diese Aktion stellt auch ein Zeichenfolgenfeld bereit, in dem Sie Freitextdaten speichern können, um den Fortschritt in einer für Ihre Anwendung geeigneten Form anzugeben.

In diesem Beispiel meldet der Activity Worker Heartbeat an Amazon SWF und verwendet das Detailfeld, um zu melden, dass die Aktivitätsaufgabe zu 40 Prozent abgeschlossen ist. Um einen Heartbeat zu senden, muss der Aktivitäts-Worker das Aufgabentoken der Aktivitätsaufgabe angeben.

```
https://swf.us-east-1.amazonaws.com
RecordActivityTaskHeartbeat
{
  "taskToken" : "12342e17-80f6-FAKE-TASK-TOKEN32f0223",
  "details" : "40"
}
```

Durch diese Aktion selbst wird noch kein Ereignis im Verlauf der Workflow-Ausführung erstellt. Wenn jedoch eine Zeitüberschreitung für die Aufgabe auftritt, enthält der Verlauf der Workflow-Ausführung ein Ereignis `ActivityTaskTimedOut` mit den Informationen des letzten Heartbeats, den der Aktivitäts-Worker gesendet hat.

Abschließen oder fehlschlagen einer Aktivitätsaufgabe

Nachdem eine Aufgabe ausgeführt wurde, sollte der Aktivitäts-Worker melden, ob die Aktivitätsaufgabe abgeschlossen wurde oder fehlgeschlagen ist.

Abschließen einer Aktivitätsaufgabe

Um eine Aktivitätsaufgabe abzuschließen, programmieren Sie Ihren Aktivitäts-Worker so, dass er unter Angabe des Aufgabentokens die Aktion `RespondActivityTaskCompleted` aufruft, nachdem eine Aktivitätsaufgabe erfolgreich abgeschlossen wurde.

In diesem Beispiel teilt der Aktivitäts-Worker mit, dass die Aufgabe erfolgreich abgeschlossen wurde.

```
https://swf.us-east-1.amazonaws.com
RespondActivityTaskCompleted
{
  "taskToken": "12342e17-80f6-FAKE-TASK-TOKEN32f0223",
```

```
"results": "40"  
}
```

Wenn die Aktivität abgeschlossen ist, plant Amazon SWF eine neue Entscheidungsaufgabe für die Workflow-Ausführung, mit der die Aktivität verknüpft ist.

Programmieren Sie den Aktivitäts-Worker so, dass er nach Abschluss der aktuellen Aufgabe eine neue Aktivitätsaufgabe anfordert. So entsteht eine Schleife, in der der Aktivitäts-Worker fortlaufend Aufgaben anfordert und abschließt.

Wenn die Aktivität nicht innerhalb des `StartToCloseTimeout`Zeitraums reagiert oder wenn sie überschritten `ScheduleToCloseTimeout`wurde, beendet Amazon SWF das Timeout für die Aktivitätsaufgabe und plant eine Entscheidungsaufgabe. So kann ein Entscheider eine angemessene Aktion wie das erneute Planen der Aufgabe ausführen.

Wenn beispielsweise eine EC2 Amazon-Instance eine Aktivitätsaufgabe ausführt und die Instance ausfällt, bevor die Aufgabe abgeschlossen ist, erhält der Entscheider ein Timeout-Ereignis in der Workflow-Ausführungshistorie. Wenn die Aktivitätsaufgabe einen Heartbeat verwendet, empfängt der Entscheider das Ereignis, wenn die Aufgabe den nächsten Heartbeat nach dem Ausfall der EC2 Amazon-Instance nicht zustellt. Andernfalls empfängt der Entscheider das Ereignis, wenn die Aktivitätsaufgabe fehlschlägt, bevor einer der allgemeinen Zeitüberschreitungswerte erreicht wurde. Der Entscheider muss die Aufgabe dann neu zuweisen oder eine andere Aktion ausführen.

Fehlschlagen einer Aktivitätsaufgabe

Wenn ein Activity Worker aus irgendeinem Grund eine Aktivitätsaufgabe nicht ausführen kann, er aber trotzdem mit Amazon SWF kommunizieren kann, können Sie ihn so programmieren, dass die Aufgabe fehlschlägt.

Um einen Aktivitäts-Worker so zu programmieren, dass eine Aktivitätsaufgabe fehlschlägt, lassen Sie den Aktivitäts-Worker die Aktion `RespondActivityTaskFailed` unter Angabe des Aufgabentokens der Aufgabe aufrufen.

```
https://swf.us-east-1.amazonaws.com  
RespondActivityTaskFailed  
{  
  "taskToken" : "12342e17-80f6-FAKE-TASK-TOKEN32f0223",  
  "reason" : "CC-Invalid",  
  "details" : "Credit Card Number Checksum Failed"  
}
```

Als Entwickler legen Sie in den Grund- und Detailfeldern die Werte fest, die gespeichert werden. Dabei handelt es sich um Zeichenketten in freier Form. Sie können alle Fehlercodekonventionen verwenden, die für Ihre Anwendung gelten. Amazon SWF verarbeitet diese Werte nicht. Amazon SWF kann diese Werte jedoch in der Konsole anzeigen.

Wenn eine Aktivitätsaufgabe fehlschlägt, plant Amazon SWF eine Entscheidungsaufgabe für die Workflow-Ausführung, mit der die Aktivitätsaufgabe verknüpft ist, um den Entscheider über den Fehler zu informieren. Programmieren Sie Ihren Entscheider so, dass er abhängig von der Art des Fehlers fehlgeschlagene Aktivitäten abarbeitet, beispielsweise durch erneutes Planen der Aktivität oder Fehlschlagen der Workflow-Ausführung.

Starten von Aktivitäts-Workern

Um Aktivitäts-Worker zu starten, packen Sie Ihre Logik in ein ausführbares Programm, das von Ihrer Aktivitäts-Worker-Plattform unterstützt wird. Sie können beispielsweise den Aktivitätscode in eine ausführbare Java-Datei einschließen, die sowohl auf Linux- als auch auf Windows-Servern ausgeführt werden kann.

Nach dem Start beginnen Ihre Worker mit dem Abrufen von Aufgaben. Bis der Entscheider jedoch Aktivitätsaufgaben plant, treten für diese Abfragen Zeitüberschreitungen auf und die Worker fragen weitere Aufgaben ab.

Da es sich bei Umfragen um ausgehende Anfragen handelt, kann Activity Worker in jedem Netzwerk ausgeführt werden, das Zugriff auf den Amazon SWF SWF-Endpunkt hat.

Sie können beliebig viele Aktivitäts-Worker starten. Wenn der Entscheider Aktivitätsaufgaben plant, verteilt Amazon SWF die Aktivitätsaufgaben automatisch an die Mitarbeiter der Abfrageaktivität.

Entwicklung von Entscheidern in Amazon SWF

Ein Entscheider ist eine Implementierung der Koordinationslogik Ihres Workflow-Typs, die während der Workflow-Ausführung ausgeführt wird. Sie können mehrere Entscheider für einen einzelnen Workflow-Typ ausführen.

Da der Ausführungsstatus einer Workflow-Ausführung in ihrem Workflow-Verlauf gespeichert ist, können Entscheider zustandslos sein. Amazon SWF verwaltet den Verlauf der Workflow-Ausführung und stellt ihn bei jeder Entscheidungsaufgabe einem Entscheider zur Verfügung. Dadurch können Sie nach Bedarf Entscheider dynamisch hinzufügen oder entfernen. So wird die Verarbeitung Ihrer Workflows in hohem Maße skalierbar. Wenn die Auslastung Ihres Systems wächst, fügen Sie einfach

weitere Entscheider hinzu, um die gestiegene Kapazität zu kompensieren. Beachten Sie jedoch, dass bei einer gegebenen Workflow-Ausführung immer nur jeweils eine Entscheidungsaufgabe geöffnet sein kann.

Jedes Mal, wenn bei einer Workflow-Ausführung eine Statusänderung erfolgt, plant Amazon SWF eine Entscheidungsaufgabe. Empfängt ein Entscheider eine Entscheidungsaufgabe, geschieht Folgendes:

- Er interpretiert den Workflow-Ausführungsverlauf, der mit der Entscheidungsaufgabe bereitgestellt wird.
- Er wendet die Koordinationslogik basierend auf dem Workflow-Ausführungsverlauf an und entscheidet, was als Nächstes zu tun ist. Jede Entscheidung wird durch eine Entscheidungsstruktur dargestellt.
- Schließt die Entscheidungsaufgabe ab und stellt Amazon SWF eine Liste der Entscheidungen zur Verfügung.

In diesem Abschnitt wird die Entwicklung eines Entscheiders beschrieben. Dazu gehört:

- Programmieren des Entscheiders zum Abrufen von Entscheidungsaufgaben
- Programmieren des Entscheiders zum Interpretieren des Workflow-Ausführungsverlaufs und zum Treffen von Entscheidungen
- Programmieren des Entscheiders zum Reagieren auf eine Entscheidungsaufgabe

Die Beispiele in diesem Abschnitt zeigen die Programmierung eines Entscheiders für den E-Commerce-Beispiel-Workflow.

Sie können den Decider in jeder beliebigen Sprache implementieren und ihn überall ausführen, sofern er über seine Service-API mit Amazon SWF kommunizieren kann.

Themen

- [Definieren einer Koordinationslogik](#)
- [Abrufen von Entscheidungsaufgaben](#)
- [Anwenden der Koordinationslogik](#)
- [Reagieren auf Entscheidungen](#)
- [Beenden der Workflow-Ausführung](#)
- [Starten von Entscheidern](#)

Definieren einer Koordinationslogik

Für die Entwicklung eines Entscheiders ist zunächst einmal das Definieren der Koordinationslogik notwendig. Beim E-Commerce-Beispiel sieht die Koordinationslogik, die jede Aktivität nach Abschluss der vorherigen Aktivität plant, in etwa wie folgt aus:

```
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
```

Der Entscheider wendet die Koordinationslogik auf den Workflow-Ausführungsverlauf an und erstellt eine Liste mit Entscheidungen, wenn die Entscheidungsaufgabe mit der `RespondDecisionTaskCompleted`-Aktion abgeschlossen wird.

Abrufen von Entscheidungsaufgaben

Jeder Entscheider ruft Entscheidungsaufgaben ab. Jede Entscheidungsaufgabe enthält Informationen anhand derer der Entscheider Entscheidungen, beispielsweise das Planen von Aktivitätsaufgaben, generiert. Der Entscheider ruft Entscheidungen mit der `PollForDecisionTask`-Aktion ab.

In diesem Beispiel ruft der Entscheider eine Entscheidungsaufgabe ab, die die `customerOrderWorkflow-0.1`-Aufgabenliste angibt.

```
https://swf.us-east-1.amazonaws.com
PollForDecisionTask
{
  "domain": "867530901",
```

```
"taskList": {"name": "customerOrderWorkflow-v0.1"},
"identity": "Decider01",
"maximumPageSize": 50,
"reverseOrder": true
}
```

Wenn eine Entscheidungsaufgabe in der angegebenen Aufgabenliste verfügbar ist, gibt Amazon SWF sie sofort zurück. Wenn keine Entscheidungsaufgabe verfügbar ist, hält Amazon SWF die Verbindung bis zu 60 Sekunden lang geöffnet und gibt eine Aufgabe zurück, sobald sie verfügbar ist. Wenn keine Aufgabe verfügbar ist, gibt Amazon SWF eine leere Antwort zurück. Eine leere Antwort ist eine Task-Struktur, in der der Wert von `taskToken` eine leere Zeichenfolge ist. Stellen Sie sicher, dass Ihr Entscheider so programmiert ist, dass er bei einer leeren Antwort eine neue Aufgabe abrufen.

Wenn eine Entscheidungsaufgabe verfügbar ist, gibt Amazon SWF eine Antwort zurück, die die Entscheidungsaufgabe sowie eine paginierte Ansicht des Workflow-Ausführungsverlaufs enthält.

In diesem Beispiel gibt der Typ des letzten Ereignisses an, dass die Workflow-Ausführung gestartet wurde. Das Eingabeelement enthält die für die Ausführung der ersten Aufgabe notwendigen Informationen.

```
{
  "events": [
    {
      "decisionTaskStartedEventAttributes": {
        "identity": "Decider01",
        "scheduledEventId": 2
      },
      "eventId": 3,
      "eventTimestamp": 1326593394.566,
      "eventType": "DecisionTaskStarted"
    }, {
      "decisionTaskScheduledEventAttributes": {
        "startToCloseTimeout": "600",
        "taskList": { "name": "specialTaskList" }
      },
      "eventId": 2,
      "eventTimestamp": 1326592619.474,
      "eventType": "DecisionTaskScheduled"
    }, {
      "eventId": 1,
      "eventTimestamp": 1326592619.474,
      "eventType": "WorkflowExecutionStarted",

```

```
"workflowExecutionStartedEventAttributes": {
  "childPolicy" : "TERMINATE",
  "executionStartToCloseTimeout" : "3600",
  "input" : "data-used-decider-for-first-task",
  "parentInitiatedEventId": 0,
  "tagList" : ["music purchase", "digital", "ricoh-the-dog"],
  "taskList": { "name": "specialTaskList" },
  "taskStartToCloseTimeout": "600",
  "workflowType": {
    "name": "customerOrderWorkflow",
    "version": "1.0"
  }
}
},
],
...
}
```

Nach dem Erhalt des Workflow-Ausführungsverlaufs interpretiert der Entscheider den Verlauf und trifft basierend auf seiner Koordinationslogik Entscheidungen.

Da die Anzahl der Workflow-Verlaufereignisse für eine einzelne Workflow-Ausführung möglicherweise hoch ist, wird das zurückgegebene Ergebnis möglicherweise auf mehrere Seiten umgebrochen. Um nachfolgende Seiten abzurufen, führen Sie zusätzliche Aufrufe durch, um die `PollForDecisionTask` beim ersten Aufruf `nextPageToken` zurückgegebenen Seiten zu verwenden. Beachten Sie, dass Sie damit nicht anrufen `GetWorkflowExecutionHistory nextPageToken`. Rufen Sie stattdessen `PollForDecisionTask` erneut auf.

Anwenden der Koordinationslogik

Sie müssen den Entscheider so programmieren, dass er, nachdem er eine Entscheidungsaufgabe empfangen hat, den Workflow-Ausführungsverlauf interpretiert, um festzustellen, was bisher geschehen ist. Anhand dieser Informationen sollte er dann eine Liste von Entscheidungen generieren.

Beim E-Commerce-Beispiel geht es primär um das letzte Ereignis im Workflow-Verlauf. Deshalb definieren wir die folgende Logik.

```
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
```

Ist das `lastEvent` eine `CompleteVerifyOrderActivity`-Aktivität, fügen Sie die `ScheduleChargeCreditCardActivity`-Aktivität zur Liste der Entscheidungen hinzu.

Nachdem der Entscheider die zu treffende (n) Entscheidung (en) festgelegt hat, kann er Amazon SWF mit entsprechenden Entscheidungen antworten.

Reagieren auf Entscheidungen

Nach der Interpretation des Workflow-Verlaufs und der Generierung einer Liste von Entscheidungen ist der Entscheider bereit, Amazon SWF mit diesen Entscheidungen zu antworten.

Programmieren Sie den Entscheider so, dass er die benötigten Daten aus dem Workflow-Ausführungsverlauf extrahieren kann. Erstellen Sie dann Entscheidungen, die die nächsten passenden Aktionen für den Workflow angeben. Der Entscheider überträgt diese Entscheidung mithilfe der `RespondDecisionTaskCompleted` Aktion zurück an Amazon SWF. Eine Liste der verfügbaren [Entscheidungstypen](#) finden Sie in der Amazon Simple Workflow Service API-Referenz.

Wenn der Entscheider im E-Commerce-Beispiel mit einer Reihe von Entscheidungen reagiert, die er erzeugt hat, schließt er auch die Kreditkartendaten aus dem Workflow-Ausführungsverlauf ein. Der Aktivitäts-Worker verfügt dann über die Informationen, die er zum Ausführen der Aktivitätsaufgabe benötigt.

Sind alle Aktivitäten der Workflow-Ausführung abgeschlossen, beendet der Entscheider die Ausführung.

```
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"
    }
  }
]
```

Beenden der Workflow-Ausführung

Stellt der Entscheider fest, dass der Geschäftsprozess abgeschlossen ist, also keine weiteren Aktivitäten ausgeführt werden müssen, generiert er eine Entscheidung, um die Workflow-Ausführung zu beenden.

Zum Beenden einer Workflow-Ausführung programmieren Sie den Entscheider so, dass er Ereignisse im Workflow-Verlauf interpretiert, um festzustellen, was bisher während der Ausführung passiert ist und um zu erkennen, ob die Ausführung des Workflows beendet werden sollte.

Wurde der Workflow erfolgreich abgeschlossen, beenden Sie dessen Ausführung, indem Sie `RespondDecisionTaskCompleted` mit der `CompleteWorkflowExecution`-Entscheidung aufrufen. Alternativ können Sie eine fehlerhafte Ausführung mit der `FailWorkflowExecution`-Entscheidung fehlschlagen lassen.

Im E-Commerce-Beispiel prüft der Entscheider den Verlauf und fügt basierend auf der Koordinationslogik eine Entscheidung zum Beenden der Workflow-Ausführung zu seiner Entscheidungsliste hinzu. Zudem initiiert er eine `RespondDecisionTaskCompleted`-Aktion durch eine "Workflow beenden"-Entscheidung.

Note

Es gibt Fälle, denen das Beenden einer Workflow-Ausführung fehlschlägt. Wird beispielsweise ein Signal empfangen, während der Entscheider die Workflow-Ausführung beendet, schlägt die Beenden-Entscheidung fehl. Sorgen Sie deshalb dafür, dass der Entscheider weiterhin Entscheidungsaufgaben abfragt. Stellen Sie außerdem sicher, dass der Entscheider, der die nächste Entscheidungsaufgabe erhält, auf das Ereignis reagiert — in diesem Fall ein Signal —, das den Abschluss der Ausführung verhindert hat.

Sie können auch das Abbrechen von Workflow-Ausführungen unterstützen. Dies kann insbesondere bei Workflows hilfreich sein, deren Ausführung lange dauert. Zur Unterstützung eines Abbruchs sollte Ihr Entscheider das `WorkflowExecutionCancelRequested`-Ereignis im Verlauf verarbeiten. Dieses Ereignis zeigt an, dass ein Abbruch der Ausführung angefordert wurde. Der Entscheider sollte entsprechende Aufräumaktionen durchführen. Dazu zählen das Abbrechen laufender Aktivitätsaufgaben und das Schließen des Workflows durch Aufruf der `RespondDecisionTaskCompleted`-Aktion mit der `CancelWorkflowExecution`-Entscheidung.

Im folgenden Beispiel wird `RespondDecisionTaskCompleted` aufgerufen, um anzugeben, dass die aktuelle Workflow-Ausführung abgebrochen wurde.

```
https://swf.us-east-1.amazonaws.com
RespondDecisionTaskCompleted
{
  "taskToken" : "12342e17-80f6-FAKE-TASK-TOKEN32f0223",
  "decisions" : [
    {
      "decisionType":"CancelWorkflowExecution",
      "CancelWorkflowExecutionAttributes":{"
        "Details": "Customer canceled order"
      }
    }
  ]
}
```

Amazon SWF überprüft, ob die Entscheidung, die Workflow-Ausführung zu beenden oder abzurechnen, die letzte Entscheidung ist, die vom Entscheider gesendet wurde. Sind nach der Entscheidung, den Workflow zu schließen, noch andere Entscheidungen vorhanden, ist dies ungültig.

Starten von Entscheidern

Nach Abschluss der Entscheiderentwicklung können Sie einen oder mehrere Entscheider starten.

Packen Sie dazu Ihre Koordinationslogik in ein ausführbares Programm, das von Ihrer Entscheiderplattform unterstützt wird. Sie können beispielsweise den Entscheidercode in eine ausführbare Java-Datei einschließen, die sowohl auf Linux- als auch auf Windows-Computern ausgeführt werden kann.

Nach dem Start sollten Ihre Entscheider damit beginnen, Amazon SWF nach Aufgaben abzufragen. Solange Sie nicht mit der Ausführung von Workflows beginnen und Amazon SWF Entscheidungsaufgaben plant, treten bei diesen Umfragen eine Zeitbegrenzung auf und es werden leere Antworten angezeigt. Eine leere Antwort ist eine Task-Struktur, in der der Wert von `taskToken` eine leere Zeichenfolge ist. Ihre Entscheider sollten einfach mit dem Abrufen fortfahren.

Amazon SWF stellt sicher, dass immer nur eine Entscheidungsaufgabe für eine Workflow-Ausführung aktiv sein kann. So werden Probleme, beispielsweise Konflikte zwischen Entscheidungen, verhindert. Darüber hinaus stellt Amazon SWF sicher, dass eine einzelne Entscheidungsaufgabe einem einzelnen Entscheider zugewiesen wird, unabhängig von der Anzahl der laufenden Entscheider.

Wenn etwas passiert, das eine Entscheidungsaufgabe generiert, während ein Entscheider eine andere Entscheidungsaufgabe verarbeitet, stellt Amazon SWF die neue Aufgabe in die Warteschlange, bis die aktuelle Aufgabe abgeschlossen ist. Nachdem die aktuelle Aufgabe abgeschlossen ist, stellt Amazon SWF die neue Entscheidungsaufgabe zur Verfügung. Außerdem werden Entscheidungsaufgaben in dem Sinne gebündelt, dass Amazon SWF, wenn mehrere Aktivitäten abgeschlossen werden, während ein Entscheider eine Entscheidungsaufgabe bearbeitet, nur eine einzige neue Entscheidungsaufgabe erstellt, um die Abschlüsse mehrerer Aufgaben zu berücksichtigen. Allerdings wird jedem Abschluss einer Aufgabe ein separates Ereignis im Workflow-Ausführungsverlauf zugewiesen.

Da es sich bei Umfragen um ausgehende Anfragen handelt, können Entscheider in jedem Netzwerk ausgeführt werden, das Zugriff auf den Amazon SWF SWF-Endpunkt hat.

Damit Workflow-Ausführungen voranschreiten, muss mindestens ein Entscheider ausgeführt werden. Sie können so viele Entscheider starten, wie Sie möchten. Amazon SWF unterstützt die Abfrage mehrerer Entscheider auf derselben Aufgabenliste.

Workflows in Amazon SWF starten

Sie können eine Workflow-Ausführung eines registrierten Workflow-Typs mit der Aktion `StartWorkflowExecution` aus beliebigen Anwendungen heraus starten. Beim Starten der Ausführung ordnen Sie dieser eine ID, die sogenannte `workflowId`, zu. Bei der `workflowId` kann es sich um eine Zeichenfolge passend zu Ihrer Anwendung handeln, beispielsweise die Bestellnummer in einer Anwendung zur Abwicklung von Bestellvorgängen. Sie können dieselbe `workflowId` nicht für mehrere offene Workflow-Ausführungen innerhalb derselben Domäne verwenden. Wenn Sie beispielsweise zwei Workflow-Ausführungen mit der `workflowId` `Customer Order 01` starten, wird die zweite Workflow-Ausführung nicht gestartet und die Anforderung schlägt fehl. Sie können jedoch eine geschlossene Ausführung wiederverwenden. `workflowId` Amazon SWF ordnet außerdem jeder Workflow-Ausführung eine eindeutige, vom System generierte Kennung `runId`, die, zu.

Nachdem der Workflow und die Aktivitätstypen registriert wurden, können Sie den Workflow mit der Aktion `StartWorkflowExecution` starten. Der Wert des Parameters `input` kann eine durch die Anwendung, aus der heraus der Workflow gestartet wird, festgelegte Zeichenfolge sein. `executionStartToCloseTimeout` ist die Zeitdauer in Sekunden, die die Workflow-Ausführung vom Starten bis zum Schließen maximal dauern kann. Eine Überschreitung dieses Limits führt zu einer Zeitüberschreitung der Workflow-Ausführung. Im Gegensatz zu einigen anderen Timeout-Parametern in Amazon SWF können Sie `NONE` für dieses Timeout keinen Wert von angeben. Es gibt eine Höchstdauer von einem Jahr für die Dauer, für die eine Workflow-Ausführung ausgeführt werden kann. Ähnlich verhält `taskStartToCloseTimeout` sich mit der Zeitspanne in Sekunden, die eine mit dieser Workflow-Ausführung verbundene Entscheidungsaufgabe bis zum Timeout in Anspruch nehmen kann.

```
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"  
}
```

Wenn die `StartWorkflowExecution` Aktion erfolgreich ist, gibt Amazon SWF den `runId` für die Workflow-Ausführung zurück. Die `runId` für eine Workflow-Ausführung ist innerhalb einer bestimmten Region eindeutig. Speichern Sie das für den `runId` Fall, dass Sie diese Workflow-Ausführung später in einem Aufruf von Amazon SWF angeben müssen. Verwenden Sie die `runId` beispielsweise, wenn Sie später ein Signal an die Workflow-Ausführung senden müssen.

```
{"runId": "9ba33198-4b18-4792-9c15-7181fb3a8852"}
```

Aufgabenpriorität in Amazon SWF festlegen

Standardmäßig werden Aufgaben in einer Aufgabenliste basierend auf ihrer Ankunftszeit bereitgestellt: Aufgaben, die zuerst geplant wurden, werden möglichst zuerst ausgeführt. Indem Sie eine optionale Aufgabenpriorität festlegen, können Sie bestimmten Aufgaben Priorität einräumen: Amazon SWF versucht, Aufgaben mit höherer Priorität auf einer Aufgabenliste vor Aufgaben mit niedrigerer Priorität zuzuweisen.

Note

Aufgaben, die zuerst geplant wurden, werden in der Regel zuerst ausgeführt. Eine Garantie gibt es dafür jedoch nicht.

Sie können die Aufgabenpriorität sowohl für Workflows als auch Aktivitäten einrichten. Die Aufgabenpriorität eines Workflows wirkt sich weder auf die Priorität von durch den Workflow geplanten Aktivitätsaufgaben noch auf vom Workflow gestartete untergeordnete Workflows aus. Die Standardpriorität für eine Aktivität oder einen Workflow wird bei der Registrierung festgelegt (entweder von Ihnen oder von Amazon SWF), und die registrierte Aufgabenpriorität wird immer verwendet, sofern sie nicht beim Planen der Aktivität oder beim Starten einer Workflow-Ausführung außer Kraft gesetzt wird.

Die Werte für die Aufgabenpriorität müssen im Bereich von "-2147483648" und "2147483647" liegen. Höhere Zahlen geben dabei eine höhere Priorität an. Wenn Sie für eine Aktivität oder einen Workflow keine Aufgabenpriorität festlegen, wird eine Priorität von Null ("0") zugewiesen.

Themen

- [Einrichten der Aufgabenpriorität für Workflows](#)
- [Einrichten der Aufgabenpriorität für Aktivitäten](#)
- [Aktionen, die Informationen zur Aufgabenpriorität zurückgeben](#)

Einrichten der Aufgabenpriorität für Workflows

Sie können die Aufgabenpriorität für einen Workflow beim Registrieren oder Starten des Workflows einrichten. Die beim Registrieren eines Workflowtyps festgelegte Aufgabenpriorität wird standardmäßig für alle Workflow-Ausführungen dieses Typs verwendet, sofern sie beim Starten der Workflow-Ausführung nicht überschrieben wird.

Um einen Workflow-Typ mit einer standardmäßigen Aufgabenpriorität zu registrieren, legen Sie bei der Verwendung der `defaultTaskPriority`-Aktion die folgende Option fest: [RegisterWorkflowType](#)

```
{
  "domain": "867530901",
  "name": "expeditedOrderWorkflow",
  "version": "1.0",
  "description": "Expedited customer orders workflow",
  "defaultTaskStartToCloseTimeout": "600",
  "defaultExecutionStartToCloseTimeout": "3600",
  "defaultTaskList": {"name": "mainTaskList"},
  "defaultTaskPriority": "10",
  "defaultChildPolicy": "TERMINATE"
}
```

Sie können die registrierte Aufgabenpriorität eines Workflowtyps überschreiben, wenn Sie eine Workflow-Ausführung starten mit [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"},
}
```

```
}

```

Sie können die Priorität der registrierten Aufgabe auch überschreiben, wenn Sie einen untergeordneten Workflow starten oder wenn Sie einen Workflow wie neu fortsetzen, z. B. wenn Sie auf eine Entscheidung mit antworten [RespondDecisionTaskCompleted](#).

Um die Aufgabenpriorität eines untergeordneten Workflows einzurichten, geben Sie den Wert in `startChildWorkflowExecutionDecisionAttributes` an:

```
{
  "taskToken": "AAAAKgAAAAEAAAAAAAAAAAA...",
  "decisions": [
    {
      "decisionType": "StartChildWorkflowExecution",
      "startChildWorkflowExecutionDecisionAttributes": {
        "childPolicy": "TERMINATE",
        "control": "digital music",
        "executionStartToCloseTimeout": "900",
        "input": "201412-Smith-011x",
        "taskList": {"name": "specialTaskList"},
        "taskPriority": "5",
        "taskStartToCloseTimeout": "600",
        "workflowId": "verification-workflow",
        "workflowType": {
          "name": "MyChildWorkflow",
          "version": "1.0"
        }
      }
    }
  ]
}
```

Wenn Sie einen Workflow als neuen Workflow fortsetzen, legen Sie die Aufgabenpriorität in `continueAsNewWorkflowExecutionDecisionAttributes` fest:

```
{
  "taskToken": "AAAAKgAAAAEAAAAAAAAAAAA...",
  "decisions": [
    {
      "decisionType": "ContinueAsNewWorkflowExecution",
      "continueAsNewWorkflowExecutionDecisionAttributes": {
        "childPolicy": "TERMINATE",

```

```

        "executionStartToCloseTimeout": "1800",
        "input": "5634-0056-4367-0923,12/12,437",
        "taskList": {"name": "specialTaskList"},
        "taskStartToCloseTimeout": "600",
        "taskPriority": "100",
        "workflowTypeVersion": "1.0"
    }
}
]
}

```

Einrichten der Aufgabenpriorität für Aktivitäten

Sie können die Aufgabenpriorität für eine Aktivität entweder beim Registrieren oder Planen der Aufgabe einrichten. Die beim Registrieren eines Aktivitätstyps festgelegte Aufgabenpriorität wird standardmäßig beim Ausführen der Aktivität verwendet, sofern sie beim Planen der Aktivität nicht überschrieben wird.

Um die Aufgabenpriorität bei der Registrierung eines Aktivitätstyps festzulegen, legen Sie die `defaultTaskPriorityOption` fest, wenn Sie die [RegisterActivityType](#)Aktion verwenden:

```

{
  "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"
}

```

Um eine Aufgabe mit einer Aufgabenpriorität zu planen, verwenden Sie die Option `TaskPriority`, wenn Sie die Aktivität mit der folgenden [RespondDecisionTaskCompleted](#)Aktion planen:

```

{
  "taskToken": "AAAAKgAAAAEAAAAAAAAAAAA...",
  "decisions": [
    {
      "decisionType": "ScheduleActivityTask",

```

```
    "scheduleActivityTaskDecisionAttributes": {
      "activityId": "verify-account",
      "activityType": {
        "name": "activityVerify",
        "version": "1.0"
      },
      "control": "digital music",
      "input": "abab-101",
      "taskList": {"name": "mainTaskList"},
      "taskPriority": "15"
    }
  }
]
```

Aktionen, die Informationen zur Aufgabenpriorität zurückgeben

Sie können Informationen über die festgelegte Aufgabenpriorität (oder die festgelegte Standard-Aufgabenpriorität) über die folgenden Amazon SWF SWF-Aktionen abrufen:

- [DescribeActivityType](#) gibt den Wert `defaultTaskPriority` des Aktivitätstyps im `configuration` Abschnitt der Antwort zurück.
- [DescribeWorkflowExecution](#) gibt die `TaskPriority` der Workflow-Ausführung im `executionConfiguration` Abschnitt der Antwort zurück.
- [DescribeWorkflowType](#) gibt den Wert `defaultTaskPriority` des Workflowntyps im `configuration` Abschnitt der Antwort zurück.
- [GetWorkflowExecutionHistory](#) und [PollForDecisionTask](#) geben Sie in den `workflowExecutionStartedEventAttributes` Abschnitten `activityTaskScheduledEventAttributes`, `decisionTaskScheduledEventAttributes` `workflowExecutionContinuedAsNewEventAttributes` und der Antwort Informationen zur Aufgabenpriorität an.

Behandlung von Fehlern in Amazon SWF

Es gibt verschiedene Fehler, die im Laufe einer Workflow-Ausführung auftreten können.

Themen

- [Validierungsfehler](#)

- [Fehler beim Umsetzen von Aktionen und Entscheidungen](#)
- [Timeouts](#)
- [Fehler durch Benutzercode](#)
- [Fehler beim Schließen einer Workflow-Ausführung](#)

Validierungsfehler

Validierungsfehler treten auf, wenn eine Anfrage an Amazon SWF fehlschlägt, weil sie nicht richtig formatiert ist oder ungültige Daten enthält. In diesem Zusammenhang kann eine Anforderung eine Aktion wie `DescribeDomain` oder eine Entscheidung wie `StartTimer` sein. Wenn es sich bei der Anfrage um eine Aktion handelt, gibt Amazon SWF in der Antwort einen Fehlercode zurück. Überprüfen Sie diesen Fehlercode, da er Informationen darüber enthalten kann, welcher Teil der Anforderung den Fehler verursacht hat. Beispielsweise sind ein oder mehrere Argumente, die mit der Anforderung übergeben wurden, ungültig. Eine Liste der häufigsten Fehlercodes finden Sie unter dem Thema für die Aktion in der Amazon Simple Workflow Service API-Referenz.

Wenn es sich bei der fehlgeschlagenen Anforderung um eine Entscheidung handelt, wird ein entsprechendes Ereignis im Verlauf der Workflow-Ausführungen aufgeführt. Wenn beispielsweise die `StartTimer`-Entscheidung fehlgeschlagen ist, sehen Sie ein `StartTimerFailed`-Ereignis im Verlauf. Der Entscheider sollte den Verlauf auf diese Ereignisse prüfen, wenn er ihn als Antwort auf `PollForDecisionTask` oder `GetWorkflowExecutionHistory` erhält. Die nachfolgende Liste enthält mögliche Entscheidungsfehlerereignisse, die auftreten können, wenn die Entscheidung falsch formatiert ist oder ungültige Daten enthält.

Fehler beim Umsetzen von Aktionen und Entscheidungen

Auch wenn die Anforderung korrekt formuliert ist, können Fehler auftreten, wenn Amazon SWF versucht, die Anforderung auszuführen. In diesen Fällen weist eines der folgenden Ereignisse im Verlauf darauf hin, dass ein Fehler aufgetreten ist. Aufschluss über die Fehlerursache gibt das Feld `reason`.

- [CancelTimerFailed](#)
- [RequestCancelActivityTaskFailed](#)
- [RequestCancelExternalWorkflowExecutionFailed](#)
- [ScheduleActivityTaskFailed](#)
- [SignalExternalWorkflowExecutionFailed](#)

- [StartChildWorkflowExecutionFailed](#)
- [StartTimerFailed](#)

Timeouts

[Entscheider](#), [Aktivitäts-Worker](#) und [Workflow-Ausführungen](#) unterliegen alle den Beschränkungen der Zeiträume für Zeitüberschreitungen. Bei diesem Fehlertyp tritt bei einer Aufgabe oder einem untergeordneten Workflow eine Zeitüberschreitung auf. Im Verlauf wird ein Ereignis zur Beschreibung der Zeitüberschreitung angezeigt. Der Entscheider sollte dieses Ereignis bearbeiten, indem er z. B. die Aufgabe neu plant oder den untergeordneten Workflow erneut startet. Weitere Informationen zu Zeitüberschreitungen finden Sie unter [Amazon SWF-Timeout-Typen](#).

- [ActivityTaskTimedOut](#)
- [ChildWorkflowExecutionTimedOut](#)
- [DecisionTaskTimedOut](#)
- [WorkflowExecutionTimedOut](#)

Fehler durch Benutzercode

Beispiele für diese Art der Fehlerbedingung sind Fehler bei Aktivitätsaufgaben und untergeordneten Workflows. Wie bei Timeoutfehlern fügt Amazon SWF dem Workflow-Ausführungsverlauf ein entsprechendes Ereignis hinzu. Der Entscheider sollte dieses Ereignis bearbeiten, indem er beispielsweise die Aufgabe neu plant oder den untergeordneten Workflow erneut startet.

- [ActivityTaskFailed](#)
- [ChildWorkflowExecutionFailed](#)

Fehler beim Schließen einer Workflow-Ausführung

Entscheider sehen möglicherweise die folgenden Ereignisse, wenn sie versuchen, einen Workflow mit einer ausstehenden Entscheidungsaufgabe zu schließen.

- [FailWorkflowExecutionFailed](#)
- [CompleteWorkFlowExecutionFailed](#)
- [ContinueAsNewWorkflowExecutionFailed](#)

- [CancelWorkflowExecutionFailed](#)

Weitere Informationen zu den oben aufgeführten Ereignissen finden Sie unter [History Event](#) in der Amazon SWF API-Referenz.

Amazon SWF SWF-Kontingente

Amazon SWF legt Kontingente für die Größe bestimmter Workflow-Parameter fest, z. B. für die Anzahl der Domains pro Konto und für die Größe des Workflow-Ausführungsverlaufs. Diese Kontingente sollen verhindern, dass fehlerhafte Workflows alle Ressourcen des Systems verbrauchen, es handelt sich dabei jedoch nicht um feste Grenzwerte. Wenn Sie feststellen, dass Ihre Anwendung diese Kontingente häufig überschreitet, können Sie [eine Erhöhung der Servicekontingenten beantragen](#).

Inhalt

- [Allgemeine Kontokontingente für Amazon SWF](#)
- [Kontingente für Workflow-Ausführungen](#)
- [Kontingente für Aufgabenausführungen](#)
- [Drosselungskontingente für Amazon SWF](#)
 - [Drosselungsquoten für alle Regionen](#)
 - [Entscheidungsquoten für alle Regionen](#)
 - [Kontingente auf Workflow-Ebene](#)
- [Beantragen einer Kontingenterhöhung](#)

Allgemeine Kontokontingente für Amazon SWF

- Maximale Anzahl registrierter Domains — 100

Dieses Kontingent umfasst sowohl registrierte als auch veraltete Domains.

- Maximale Workflow- und Aktivitätstypen — jeweils 10.000 pro Domain

Dieses Kontingent umfasst sowohl registrierte als auch veraltete Typen.

- Kontingent für API-Aufrufe — Abgesehen von seltenen Spitzenwerten können Anwendungen gedrosselt werden, wenn sie in sehr kurzer Zeit eine große Anzahl von API-Aufrufen tätigen.
- Maximale Anforderungsgröße — 1 MB pro Anfrage

Dies ist die Gesamtdatengröße pro Amazon SWF SWF-API-Anfrage, einschließlich des Anforderungsheaders und aller anderen zugehörigen Anforderungsdaten.

- Verkürzte Antworten für die Anzahl APIs — Zeigt an, dass ein internes Kontingent erreicht wurde und dass die Antwort nicht die volle Anzahl entspricht.

Bei einigen Abfragen wird intern das oben genannte Kontingent von 1 MB erreicht, bevor eine vollständige Antwort zurückgegeben wird. In folgenden Fällen kann eine gekürzte Antwort anstatt der vollen Anzahl zurückgegeben werden.

- [CountClosedWorkflowExecutions](#)
- [CountOpenWorkflowExecutions](#)
- [CountPendingActivityTasks](#)
- [CountPendingDecisionTasks](#)

Wenn in diesen Fällen die `truncated`-Antwort jeweils auf „true“ gesetzt wurde, beträgt die Anzahl weniger als die vollständige Menge. Dieses interne Kontingent kann nicht erhöht werden.

- Maximale Anzahl von Tags — 50 Tags pro Ressource.

Wenn Sie versuchen, mehr als 50 Tags hinzuzufügen, führt zu einem Fehler 400, `TooManyTagsFault`.

Kontingente für Workflow-Ausführungen

- Maximale Anzahl offener Workflow-Ausführungen — 100.000 pro Domain

In der Anzahl sind untergeordnete Workflow-Ausführungen enthalten.

- Maximale Ausführungszeit für Workflows — 1 Jahr. Dies ist ein festes Kontingent, das nicht geändert werden kann.
- Maximale Größe des Workflow-Ausführungsverlaufs: 25.000 Ereignisse. Dies ist ein festes Kontingent, das nicht geändert werden kann.

Es hat sich bewährt, jeden Workflow so zu strukturieren, dass dessen Verlauf nicht mehr als 10 000 Ereignisse umfasst. Da der Entscheider den Workflow-Verlauf abrufen muss, ermöglicht ein kleinerer Verlauf dem Entscheider, den Workflow-Verlauf schneller abzuschließen. Wenn Sie das [Flow Framework](#) verwenden, können Sie es verwenden, `ContinueAsNew` um einen Workflow mit einem neuen Verlauf fortzusetzen.

- Maximale Anzahl offener untergeordneter Workflow-Ausführungen — 1.000 pro Workflow-Ausführung

Wenn Ihr Anwendungsfall erfordert, dass Sie diese Kontingente überschreiten, können Sie die Funktionen von Amazon SWF verwenden, um Ausführungen fortzusetzen und Ihre Anwendungen mithilfe [untergeordneter Workflow-Ausführungen](#) zu strukturieren. Wenn Sie feststellen, dass Sie immer noch eine Erhöhung des Kontingents benötigen, finden Sie weitere Informationen unter [Beantragen einer Kontingenterhöhung](#)

Kontingente für Aufgabenausführungen

- Maximale Anzahl von Abfragern pro Aufgabenliste — 1.000 pro Aufgabenliste

Es können maximal 1 000 Poller gleichzeitig eine bestimmte Aufgabenliste abfragen. Wenn Sie das Limit überschreiten, erhalten Sie eine `LimitExceededException`.

Note

Das Maximum liegt zwar bei 1.000, es können jedoch `LimitExceededException` Fehler auftreten, wenn dieses Kontingent überschritten wird. Dieser Fehler bedeutet nicht, dass sich Ihre Aufgaben verzögern. Stattdessen bedeutet es, dass Sie die maximale Anzahl an inaktiven Pollern auf einer Aufgabenliste haben. Amazon SWF legt dieses Limit fest, um Ressourcen sowohl auf der Client- als auch auf der Serverseite zu sparen. Durch die Festlegung des Limits wird verhindert, dass zu viele Umfrageteilnehmer unnötig warten. Sie können die Anzahl der `LimitExceededException` Fehler reduzieren, indem Sie mehrere Aufgabenlisten für die Verteilung der Abfragen verwenden.

- Maximale Anzahl der pro Sekunde geplanten Aufgaben — 2.000 pro Aufgabenliste

Sie können maximal 2.000 Aufgaben pro Sekunde für eine bestimmte Aufgabenliste planen. Wenn Sie 2.000 überschreiten, schlagen Ihre `ScheduleActivityTask` Entscheidungen `ACTIVITY_CREATION_RATE_EXCEEDED` fehlerhaft fehl.

Note

Das Maximum liegt zwar bei 2.000, es können jedoch `ACTIVITY_CREATION_RATE_EXCEEDED` Fehler auftreten, die weit vor diesem Kontingent liegen. Um diese Fehler zu reduzieren, verwenden Sie mehrere Aufgabenlisten, um die Last zu verteilen.

- Maximale Ausführungszeit für Aufgaben — 1 Jahr (begrenzt durch die maximale Ausführungszeit des Workflows)

Sie können [Aktivitäts-Timeouts](#) konfigurieren, damit ein Timeout-Ereignis eintritt, wenn eine bestimmte Phase der [Aktivitätsaufgabe](#) zu lange dauert.

- Maximale Zeit, für die SWF eine Aufgabe in der Warteschlange behält — 1 Jahr (begrenzt durch das Zeitkontingent für die Workflow-Ausführung)

Sie können während der Aktivitätsregistrierung Standard-[Aktivitäts-Timeouts](#) konfigurieren, damit ein Timeout-Ereignis eintritt, wenn eine bestimmte Ausführungsphase Ihrer [Aktivitätsaufgabe](#) zu lange dauert. Sie können Standard-Aktivitäts-Timeouts überschreiben, wenn Sie eine Aktivitätsaufgabe im Entscheidercode planen.

- Maximale Anzahl offener Aktivitätsaufgaben: 1.000 pro Workflow-Ausführung.

Dieses Kontingent umfasst sowohl Aktivitäten, die geplant wurden, als auch solche, die von Mitarbeitern bearbeitet werden.

- Maximale Anzahl offener Timer: 1.000 pro Workflow-Ausführung
- Maximale input/result Datengröße — 32.768 Zeichen

Dieses Kontingent wirkt sich auf die Ergebnisdaten der Aktivität oder Workflow-Ausführung, auf Eingabedaten bei der Planung von Aktivitätsaufgaben oder Workflow-Ausführungen und auf Eingaben aus, die mit einem [Workflow-Ausführungssignal](#) gesendet werden.

- Maximale Anzahl an Entscheidungen in einer Antwort auf eine Entscheidungsaufgabe — variiert

Aufgrund des Kontingents von 1 MB für die [maximale API-Anforderungsgröße](#) ist die Anzahl der Entscheidungen, die in einem einzigen Aufruf an zurückgegeben [RespondDecisionTaskCompleted](#) werden, entsprechend der Größe der für jede Entscheidung verwendeten Daten begrenzt, einschließlich der Größe aller Eingabedaten, die für geplante Aktivitätsaufgaben oder Workflow-Ausführungen bereitgestellt werden.

Drosselungskontingente für Amazon SWF

Zusätzlich zu den zuvor beschriebenen Servicekontingenten werden bestimmte Amazon SWF SWF-API-Aufrufe und Entscheidungsereignisse mithilfe eines [Token-Bucket-Schemas](#) gedrosselt, um die Servicebandbreite aufrechtzuerhalten. Wenn Ihre Anforderungsrate die hier aufgeführten Raten durchweg übersteigt, können Sie [eine Erhöhung des Drosselungskontingents beantragen](#).

Die Drosselungs- und Entscheidungsquoten sind in allen Regionen gleich.

Drosselungsquoten für alle Regionen

Die folgenden Kontingente gelten für einzelne Konten. Sie können auch eine Erhöhung der folgenden Kontingente beantragen. Informationen dazu finden Sie unter [Beantragen einer Kontingenterhöhung](#).

API-Name	Bucket-Größe	Nachfüllrate pro Sekunde
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

API-Name	Bucket-Größe	Nachfüllrate pro Sekunde
ListDomains	100	6
ListOpenWorkflowExecutions	200	48
ListTagsForResource	50	30
ListWorkflowTypes	200	6
PollForActivityTask	2000	200
PollForDecisionTask	2000	200
RecordActivityTaskHeartbeat	2000	160
RegisterActivityType	200	60
RegisterDomain	100	6
RegisterWorkflowType	200	60
RequestCancelWorkflowExecution	2000	30
RespondActivityTaskCanceled	2000	200
RespondActivityTaskCompleted	2000	200
RespondActivityTaskFailed	2000	200
RespondDecisionTaskCompleted	2000	200
SignalWorkflowExecution	2000	30
StartWorkflowExecution	2000	200
TagResource	50	30
TerminateWorkflowExecution	2000	60
UndeprecateActivityType	200	6

API-Name	Bucket-Größe	Nachfüllrate pro Sekunde
UndeprecateDomain	100	6
UndeprecateWorkflowType	200	6
UntagResource	50	30

Entscheidungsquoten für alle Regionen

Die folgenden Kontingente gelten für einzelne Konten. Sie können auch eine Erhöhung der folgenden Kontingente beantragen. Informationen dazu finden Sie unter [Beantragen einer Kontingenterhöhung](#).

API-Name	Bucket-Größe	Nachfüllrate pro Sekunde
RequestCancelExternalWorkflowExecution	1200	120
ScheduleActivityTask	1000	200
SignalExternalWorkflowExecution	1200	120
StartChildWorkflowExecution	500	12
StartTimer	2000	200

Kontingente auf Workflow-Ebene

Die folgenden Kontingente gelten auf Workflow-Ebene und können nicht erhöht werden.

API-Name	Bucket-Größe	Nachfüllrate pro Sekunde
GetWorkflowExecutionHistory	400	200

API-Name	Bucket-Größe	Nachfüllrate pro Sekunde
SignalWorkflowExecution	1000	1000
RecordActivityTaskHeartbeat	1000	1000
RequestCancelWorkflowExecution	200	200

Beantragen einer Kontingenterhöhung

Weitere Informationen finden Sie unter [AWS Servicekontingente](#) im Allgemeine AWS-Referenz.

Zusätzliche Ressourcen und Referenzinformationen für Amazon SWF

Dieses Kapitel enthält zusätzliche Ressourcen und Referenzinformationen, die bei der Entwicklung von Workflows mit Amazon SWF nützlich sind.

Themen

- [Amazon SWF-Timeout-Typen](#)
- [Amazon Simple Workflow Service-Endpunkte](#)
- [Zusätzliche Dokumentation für den Amazon Simple Workflow Service](#)
- [Webressourcen für den Amazon Simple Workflow Service](#)
- [Migrationsoptionen für Ruby Flow](#)

Amazon SWF-Timeout-Typen

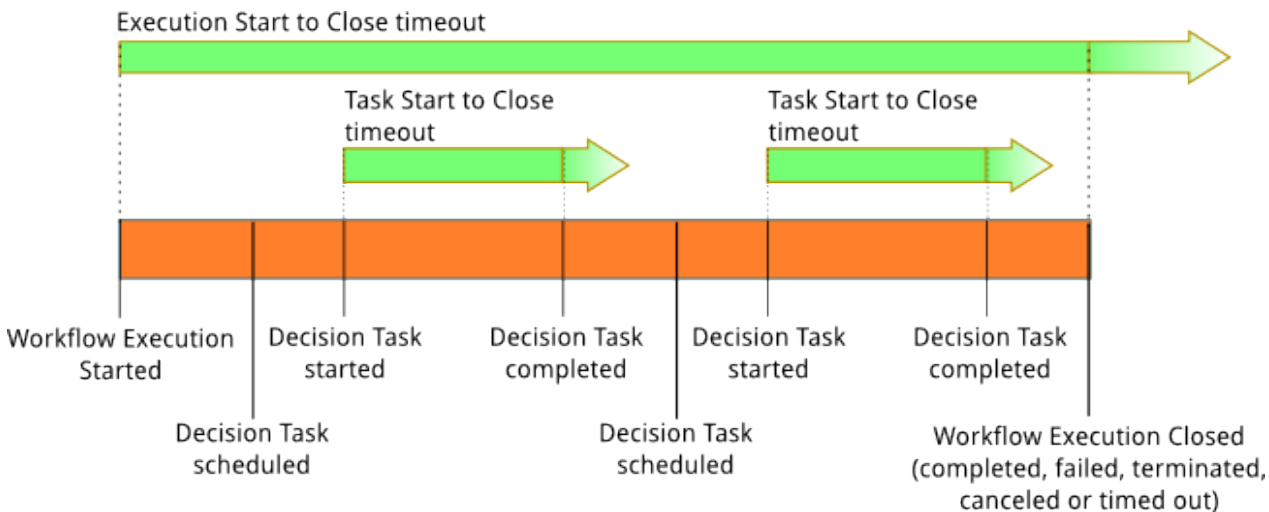
Um sicherzustellen, dass Workflow-Ausführungen korrekt ausgeführt werden, können Sie mit Amazon SWF verschiedene Arten von Timeouts festlegen. Einige Zeitüberschreitungen legen fest, wie lange der Workflow insgesamt ausgeführt werden kann. Andere Zeitüberschreitungen legen fest, wie lange es dauern darf, bis Aktivitätsaufgaben einem Worker zugewiesen werden, und wie lange die Ausführung einer Aufgabe ab der Planung dauern darf. Alle Timeouts in der Amazon SWF SWF-API sind in Sekunden angegeben. Amazon SWF unterstützt die Zeichenfolge auch NONE als Timeout-Wert, was bedeutet, dass es kein Timeout gibt.

Für Zeitüberschreitungen im Zusammenhang mit Entscheidungs- und Aktivitätsaufgaben fügt Amazon SWF dem Workflow-Ausführungsverlauf ein Ereignis hinzu. Die Attribute des Ereignisses geben Auskunft darüber, welche Art von Timeout eingetreten ist und welche Entscheidungs- oder Aktivitätsaufgabe betroffen war. Amazon SWF plant auch eine Entscheidungsaufgabe. Wenn der Entscheider die neue Entscheidungsaufgabe erhält, sieht er das Timeout-Ereignis in der Historie und ergreift die entsprechende Aktion, indem er die [RespondDecisionTaskCompleted](#)Aktion aufruft.

Eine Aufgabe gilt vom Zeitpunkt der Planung bis zum Schließen der Aufgabe als offen. Für Aufgaben, die gerade von einem Worker verarbeitet werden, wird daher der Status "offen" gesendet. Eine Aufgabe ist geschlossen, wenn ein Worker sie als [abgeschlossen](#), [abgebrochen](#) oder [fehlgeschlagen](#) meldet. Eine Aufgabe kann auch von Amazon SWF aufgrund eines Timeouts geschlossen werden.

Zeitüberschreitungen in Workflow- und Entscheidungsaufgaben

Die folgende Abbildung zeigt, wie Zeitüberschreitungen für Workflow- und Entscheidungsaufgaben sich auf die Lebensdauer eines Workflows auswirken:



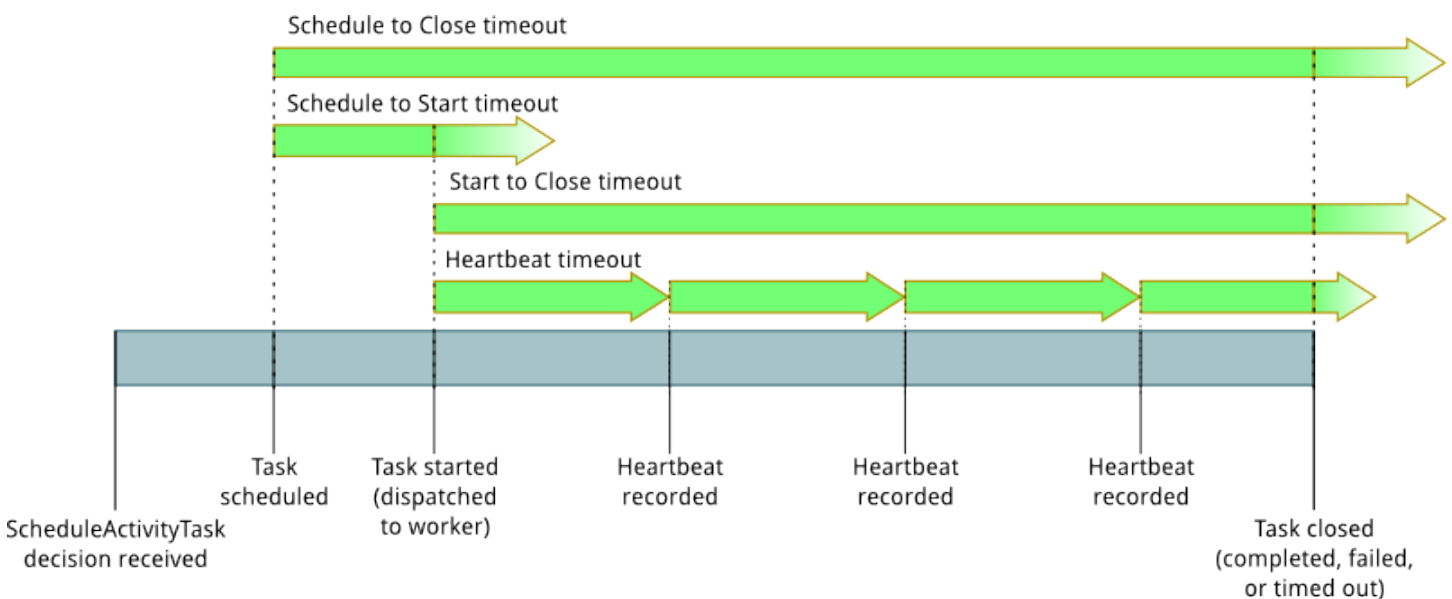
Für Workflow- und Entscheidungsaufgaben gibt es zwei relevante Zeitüberschreitungstypen:

- **Workflow-Start to Close (`timeoutType: START_TO_CLOSE`)** — Dieses Timeout gibt die maximale Zeit an, die bis zum Abschluss einer Workflow-Ausführung in Anspruch nehmen kann. Bei der Registrierung eines Workflows wird ein Standardwert festgelegt, der jedoch beim Starten des Workflows durch andere Werte überschrieben werden kann. Wenn dieses Timeout überschritten wird, schließt Amazon SWF die Workflow-Ausführung und fügt dem Workflow-Ausführungsverlauf ein [Ereignis](#) des Typs [WorkflowExecutionTimedOut](#) hinzu. Neben dem `timeoutType` legen die Ereignisattribute auch die `childPolicy` fest, die sich auf die Workflow-Ausführung auswirkt. Die untergeordnete Richtlinie legt fest, wie mit untergeordneten Workflow-Ausführungen verfahren wird, wenn bei der übergeordneten Workflow-Ausführung eine Zeitüberschreitung auftritt oder sie anderweitig beendet wird. Wenn in der `childPolicy` beispielsweise `TERMINATE` festgelegt ist, werden die untergeordneten Workflow-Ausführungen beendet. Nachdem bei einer Workflow-Ausführung eine Zeitüberschreitung aufgetreten ist, können Sie als einzige Aktionen dafür noch Sichtbarkeitsaufrufe ausführen.
- **Entscheidungsaufgabe von Anfang bis Ende (`timeoutType: START_TO_CLOSE`)** — Dieses Timeout gibt die maximale Zeit an, die der entsprechende Entscheider benötigen kann, um eine Entscheidungsaufgabe abzuschließen. Sie wird während der Registrierung des Workflow-Typs festgelegt. Wenn dieses Timeout überschritten wird, wird die Aufgabe im Workflow-Ausführungsverlauf als Timeout markiert, und Amazon SWF fügt dem Workflow-Verlauf ein Ereignis des Typs [DecisionTaskTimedOut](#) hinzu. Zu den Ereignisattributen gehören die IDs

Ereignisse, die dem Zeitpunkt entsprechen, zu dem diese Entscheidungsaufgabe geplant (`scheduledEventId`) und wann sie gestartet wurde (`startedEventId`). Amazon SWF fügt nicht nur das Ereignis hinzu, sondern plant auch eine neue Entscheidungsaufgabe, um den Entscheider darüber zu informieren, dass bei dieser Entscheidungsaufgabe das Timeout überschritten wurde. Nach einer Zeitüberschreitung schlagen Versuche, die abgelaufene Entscheidungsaufgabe mit `RespondDecisionTaskCompleted` abzuschließen, fehl.

Zeitüberschreitungen in Aktivitätsaufgaben

Die folgende Abbildung zeigt, wie Zeitüberschreitungen sich auf die Lebensdauer einer Aktivitätsaufgabe auswirken:



Für Aktivitätsaufgaben gibt es vier relevante Zeitüberschreitungstypen:

- Aktivitätsaufgabe von Anfang bis Ende (**timeoutType: START_TO_CLOSE**) — Dieses Timeout gibt die maximale Zeit an, die ein Mitarbeiter für die Bearbeitung einer Aufgabe benötigen kann, nachdem der Mitarbeiter die Aufgabe erhalten hat. Versuche, eine Aktivitätsaufgabe mit [RespondActivityTaskCanceled](#), und zu schließen [RespondActivityTaskCompleted](#), schlagen [RespondActivityTaskFailed](#) fehl.
- Activity Task Heartbeat (**timeoutType: HEARTBEAT**) — Dieses Timeout gibt die maximale Zeit an, die eine Aufgabe ausgeführt werden kann, bevor ihr Fortschritt durch die Aktion angezeigt wird. `RecordActivityTaskHeartbeat`
- Zeitplan für den Start der Aktivitätsaufgabe (**timeoutType: SCHEDULE_TO_START**) — Dieses Timeout gibt an, wie lange Amazon SWF wartet, bis das Zeitlimit für die Aktivitätsaufgabe

überschritten wird, wenn keine Mitarbeiter für die Ausführung der Aufgabe verfügbar sind. Nach der Zeitüberschreitung wird die abgelaufene Aufgabe keinem anderen Worker zugewiesen.

- Zeitplan für das Schließen der Aktivitätsaufgabe (**timeoutType: SCHEDULE_TO_CLOSE**) — Dieser Timeout gibt an, wie lange die Aufgabe von der geplanten bis zur Fertigstellung dauern kann. Es hat sich bewährt, dass dieser Wert nicht größer als die Summe aus Task-Timeout und schedule-to-start Task-Timeout sein sollte. start-to-close

Note

Jeder Zeitüberschreitungstyp verfügt über einen Standardwert, in der Regel NONE (unendlich). Die Höchstdauer für die Ausführung einer Aktivität ist jedoch auf ein Jahr beschränkt.

Die Standardwerte für diese Zeitüberschreitungen werden während der Registrierung des Aktivitätstyps festgelegt, können jedoch beim [Planen](#) der Aktivitätsaufgabe überschrieben werden. Wenn einer dieser Timeouts eintritt, fügt Amazon SWF dem Workflow-Verlauf ein [Ereignis](#) des Typs [ActivityTaskTimedOut](#) hinzu. Das Wertattribut `timeoutType` dieses Ereignisses gibt an, welche dieser Zeitüberschreitungen aufgetreten ist. Der Wert von `timeoutType` für jede Zeitüberschreitung ist in Klammern angegeben. Zu den Ereignisattributen gehören auch die IDs Ereignisse, die dem Zeitpunkt entsprechen, zu dem die Aktivitätsaufgabe geplant (`scheduledEventId`) und wann sie gestartet wurde (`startedEventId`). Zusätzlich zum Hinzufügen des Ereignisses plant Amazon SWF auch eine neue Entscheidungsaufgabe, um den Entscheider darüber zu informieren, dass das Timeout eingetreten ist.

Amazon Simple Workflow Service-Endpunkte

Eine Liste der aktuellen [Amazon SWF SWF-Regionen und -Endpunkte](#) finden Sie in der Allgemeine Amazon Web Services-Referenz, zusammen mit den Endpunkten für andere Dienste.

Amazon SWF-Domains und alle zugehörigen Workflows und Aktivitäten müssen in derselben Region existieren, um miteinander kommunizieren zu können. Außerdem existieren alle innerhalb einer Region registrierten Domänen, Workflows und Aktivitäten nicht in anderen Regionen. Wenn Sie beispielsweise sowohl in us-east-1 als auch in us-west-2 eine Domain mit dem Namen MySampleDomain "" erstellen, existieren diese als separate Domänen: Keiner der Workflows, Aufgabenlisten, Aktivitäten oder Daten, die mit Ihren Domains verknüpft sind, wird regionsübergreifend gemeinsam genutzt.

Wenn Sie in Ihren Workflows andere AWS Ressourcen verwenden, z. B. EC2 Amazon-Instances, müssen diese ebenfalls in derselben Region wie Ihre Amazon SWF-Ressourcen vorhanden sein. Die einzigen Ausnahmen sind Dienste, die sich über Regionen erstrecken, wie Amazon S3 und IAM. Sie können auf diese Services von Workflows zugreifen, die in jeder beliebigen Region vorhanden sind, die sie unterstützen.

Zusätzliche Dokumentation für den Amazon Simple Workflow Service

Zusätzlich zu diesem Entwicklerhandbuch sind eventuell auch die folgenden Dokumentationen wertvoll.

Amazon Simple Workflow Service API-Referenz

Die [Amazon Simple Workflow Service API-Referenz](#) enthält detaillierte Informationen zur Amazon SWF HTTP-API, einschließlich Aktionen, Anforderungs- und Antwortstrukturen und Fehlercodes.

AWS Flow Framework Dokumentation

Das [AWS Flow Framework](#) ist ein Programmierframework, das den Prozess der Implementierung verteilter asynchroner Anwendungen vereinfacht, die Amazon SWF zur Verwaltung ihrer Workflows und Aktivitäten verwenden, sodass Sie sich auf die Implementierung Ihrer Workflow-Logik konzentrieren können.

Jedes AWS Flow Framework ist so konzipiert, dass es idiomatisch in der Sprache funktioniert, für die es entworfen wurde, sodass Sie ganz natürlich mit der Sprache Ihrer Wahl arbeiten können, um Workflows mit allen Vorteilen von Amazon SWF zu implementieren.

Es gibt ein AWS Flow Framework für Java. Das [AWS Flow Framework for Java Developer Guide](#) enthält Informationen darüber, wie Sie das AWS Flow Framework für Java beziehen, einrichten und verwenden können.

AWS SDK-Dokumentation

Die AWS Software Development Kits (SDKs) bieten Zugriff auf Amazon SWF in vielen verschiedenen Programmiersprachen. SDKs orientieren sich eng an der HTTP-API, bieten aber auch sprachspezifische Programmierschnittstellen für einige Amazon SWF SWF-Funktionen. Weitere Informationen über die einzelnen SDKs finden Sie über die folgenden Links.

Note

Nur SDKs diejenigen, die zum Zeitpunkt der Erstellung dieses Artikels Amazon SWF unterstützen, sind hier aufgeführt. Eine vollständige Liste der verfügbaren AWS SDKs Tools finden Sie auf der Seite [Tools für Amazon Web Services](#).

Java

Das AWS SDK für Java bietet eine Java-API für AWS Infrastrukturdienste.

Sie können die verfügbare Dokumentation auf der Seite [AWS SDK für Java -Dokumentation](#) finden. Sie können auch direkt zu den Amazon SWF SWF-Abschnitten in der SDK-Referenz wechseln, indem Sie diesen Links folgen:

- [Class: AmazonSimpleWorkflowClient](#)
- [Class: AmazonSimpleWorkflowAsyncClient](#)
- [Interface: AmazonSimpleWorkflow](#)
- [Interface: AmazonSimpleWorkflowAsync](#)

JavaScript

AWS SDK für JavaScript Damit können Entwickler Bibliotheken oder Anwendungen erstellen, die AWS Dienste mithilfe einer einfachen easy-to-use API nutzen, die sowohl im Browser als auch in den Anwendungen von Node.js auf dem Server verfügbar ist.

Sie können die verfügbare Dokumentation auf der Seite [AWS SDK für JavaScript -Dokumentation](#) finden. Sie können auch direkt zum Amazon SWF SWF-Bereich in der SDK-Referenz wechseln, indem Sie diesem Link folgen:

- [Class: AWS.SimpleWorkflow](#)

.NET

AWS SDK für .NET Es handelt sich um ein einzelnes herunterladbares Paket, das Visual Studio-Projektvorlagen, die AWS .NET-Bibliothek, C#-Codebeispiele und Dokumentation enthält. Das AWS SDK für .NET macht es für Windows-Entwickler einfacher, .NET-Anwendungen für Amazon SWF und andere Dienste zu erstellen.

Sie können die verfügbare Dokumentation auf der Seite [AWS SDK für .NET -Dokumentation](#) finden. Sie können auch direkt zu den Amazon SWF SWF-Abschnitten in der SDK-Referenz wechseln, indem Sie diesen Links folgen:

- [Namespace: Amazon.SimpleWorkflow](#)
- [Namespace: Amazon.SimpleWorkflow.Model](#)

PHP

Das AWS SDK for PHP bietet eine PHP-Programmierschnittstelle für Amazon SWF.

Sie können die verfügbare Dokumentation auf der Seite [AWS SDK for PHP -Dokumentation](#) finden. Sie können auch direkt zum Amazon SWF SWF-Bereich in der SDK-Referenz wechseln, indem Sie diesem Link folgen:

- [Class: SwfClient](#)

Python

Das AWS SDK für Python (Boto) bietet eine Python-Programmierschnittstelle für Amazon SWF.

Die verfügbare Dokumentation finden Sie auf der Seite [boto: Eine Python-Schnittstelle zu Amazon Web Services](#). Sie können auch direkt zu den Amazon SWF SWF-Abschnitten in der Dokumentation wechseln, indem Sie diesen Links folgen:

- [Amazon SWF SWF-Anleitung](#)
- [Amazon SWF SWF-Referenz](#)

Ruby

Das AWS SDK für Ruby bietet eine Ruby-Programmierschnittstelle für Amazon SWF.

Sie können die verfügbare Dokumentation auf der Seite [AWS SDK für Ruby -Dokumentation](#) finden. Sie können auch direkt zum Amazon SWF SWF-Bereich in der SDK-Referenz wechseln, indem Sie diesem Link folgen:

- [Klasse: AWS::Simple Workflow](#)

AWS CLI Dokumentation

Das AWS Command Line Interface (AWS CLI) ist ein einheitliches Tool zur Verwaltung Ihrer AWS Dienste. Mit nur einem Tool zum Herunterladen und Konfigurieren können Sie mehrere AWS Dienste von der Befehlszeile aus steuern und mithilfe von Skripten automatisieren.

Weitere Informationen zu den AWS CLI finden Sie [AWS Command Line Interface](#) auf der Seite.

Einen Überblick über die verfügbaren Befehle für Amazon SWF finden Sie unter [swf](#) in der AWS CLI Befehlsreferenz.

Webressourcen für den Amazon Simple Workflow Service

Es gibt eine Reihe von Webressourcen, mit denen Sie mehr über Amazon SWF erfahren oder Hilfe bei der Nutzung des Service und der Entwicklung von Workflows erhalten können.

Amazon SWF SWF-Forum

Das Amazon SWF SWF-Forum bietet Ihnen die Möglichkeit, mit anderen Amazon SWF SWF-Entwicklern und Mitgliedern des Amazon SWF SWF-Entwicklungsteams bei Amazon zu kommunizieren, Fragen zu stellen und Antworten zu erhalten.

Sie finden das Forum unter: [Forum: Amazon Simple Workflow Service](#).

Häufig gestellte Fragen zu Amazon SWF

Die häufig gestellten Fragen zu Amazon SWF bieten Antworten auf häufig gestellte Fragen zu Amazon SWF, darunter einen Überblick über gängige Anwendungsfälle, Unterschiede zwischen Amazon SWF und anderen Diensten und mehr.

Sie können die häufig gestellten Fragen hier aufrufen: Häufig gestellte [Fragen zu Amazon SWF](#).

Amazon SWF SWF-Videos

Der [Amazon Web Services-Kanal](#) YouTube bietet Videoschulungen für alle Amazon Web Services, einschließlich Amazon SWF. Eine vollständige Liste der Videos zu Amazon SWF finden Sie mit der folgenden Abfrage: [Simple Workflow in Amazon Web Services](#)

Migrationsoptionen für Ruby Flow

The AWS Flow Framework for Ruby wird nicht mehr aktiv entwickelt. Vorhandener Code funktioniert weiterhin für unbegrenzte Zeit, es gibt jedoch keine neuen Funktionen oder Versionen. In diesem Thema werden Verwendungs- und Migrationsoptionen für die weitere Arbeit mit Amazon SWF sowie Informationen zur Migration zu Step Functions behandelt.

Option	Beschreibung
Weitere Nutzung des Ruby Flow Frameworks	Vorerst wird das Ruby Flow Framework weiterhin funktionieren. Wenn Sie nichts verändern, funktioniert Ihr Code wie bisher. Planen Sie, in naher future von The AWS Flow Framework for Ruby zu migrieren.

Option	Beschreibung
Migration auf das Java Flow Framework	Das Java Flow Framework befindet sich weiterhin in aktiver Entwicklung und wird auch weiterhin neue Funktionen und Updates erhalten.
Migration auf Step Functions	Step Functions bietet eine Möglichkeit, die Komponenten verteilter Anwendungen unter Verwendung visueller Workflows zu koordinieren, gesteuert durch einen Zustandsautomaten.
Direkte Verwendung des SWF API ohne das Flow Framework	Sie können weiterhin in Ruby arbeiten und das SWF API anstelle des Ruby Flow Frameworks verwenden.

Der Vorteil, den das Flow Framework für Ruby oder Java bietet, ist, dass Sie sich ganz auf Ihre Workflow-Logik konzentrieren können. Das Framework übernimmt viele Details im Hinblick auf Kommunikation und Koordination, und ein Teil der Komplexität wird abstrahiert. Sie können die gleiche Abstraktionsebene beibehalten, indem Sie zum Java Flow Framework migrieren, oder Sie können direkt mit dem Amazon SWF SDK interagieren.

Weitere Nutzung des Ruby Flow Frameworks

Das AWS Flow Framework for Ruby wird kurzfristig weiterhin so funktionieren wie bisher. Wenn Sie Workflows AWS Flow Framework für Ruby geschrieben haben, werden diese weiterhin funktionieren. Ohne Updates, Support oder Sicherheitskorrekturen ist es am besten, einen festen Plan zu haben, um in naher future AWS Flow Framework von Ruby zu migrieren.

Migration auf das Java Flow Framework

Das AWS Flow Framework für Java wird weiterhin aktiv weiterentwickelt. Konzeptionell ähnelt das AWS Flow Framework für Java dem AWS Flow Framework für Ruby: Sie können sich immer noch auf Ihre Workflow-Logik konzentrieren, und das Framework hilft Ihnen bei der Verwaltung Ihrer Entscheidungslogik und erleichtert die Verwaltung anderer Aspekte von Amazon SWF.

- [AWS Flow Framework für Java](#)
- [AWS Flow Framework für Java API-Referenz](#)

Migration auf Step Functions

AWS Step Functions bietet einen Service, der Amazon SWF ähnelt, bei dem Ihre Workflow-Logik jedoch von einer Zustandsmaschine gesteuert wird. Mit Step Functions können Sie die Komponenten verteilter Anwendungen und Microservices mithilfe visueller Workflows koordinieren. Sie erstellen Anwendungen aus einzelnen Komponenten, die jeweils eine diskrete Funktion oder Aufgabe ausführen, sodass Sie Anwendungen schnell skalieren und verändern können. Step Functions bietet eine zuverlässige Möglichkeit, Komponenten zu koordinieren und die Funktionen Ihrer Anwendung Schritt für Schritt durchzugehen. Eine grafische Konsole bietet eine Möglichkeit, die Komponenten Ihrer Anwendungen als Abfolge einzelner Schritte zu visualisieren. Es löst jeden Schritt automatisch aus und verfolgt ihn und versucht es erneut, wenn Fehler auftreten, sodass Ihre Anwendung jedes Mal in der richtigen Reihenfolge und wie erwartet ausgeführt wird. Step Functions protokolliert den Status jedes Schritts, sodass Sie Probleme schnell diagnostizieren und debuggen können, wenn etwas schief geht.

In Step Functions verwalten Sie die Koordination Ihrer Aufgaben mithilfe einer Zustandsmaschine, die in deklarativem JSON geschrieben ist und in der [Amazon States Language](#) definiert ist. Durch die Verwendung einer Zustandsmaschine müssen Sie kein Entscheidungsprogramm zur Steuerung Ihrer Anwendungslogik schreiben und verwalten. Step Functions bietet einen intuitiven, produktiven und agilen Ansatz zur Koordination von Anwendungskomponenten mithilfe visueller Workflows. Sie sollten die Verwendung AWS Step Functions für all Ihre neuen Anwendungen in Betracht ziehen, und Step Functions bietet eine hervorragende Plattform, auf die Sie für die Workflows migrieren können, die Sie derzeit in AWS Flow Framework for Ruby implementiert haben.

Um Ihnen bei der Migration Ihrer Aufgaben zu Step Functions zu helfen und gleichzeitig Ihre Ruby-Sprachkenntnisse weiter zu nutzen, bietet Step Functions ein Beispiel für einen Ruby Activity Worker. Dieses Beispiel verwendet bewährte Methoden für die Implementierung eines Activity Workers und kann als Vorlage für die Migration Ihrer Aufgabenlogik zu Step Functions verwendet werden. Weitere Informationen finden Sie im Thema [Example Activity Worker in Ruby](#) im [AWS Step Functions Developer Guide](#).

Note

Für viele Kunden ist die Migration von AWS Flow Framework for Ruby zu Step Functions die beste Option. Wenn Sie jedoch möchten, dass Signale in Ihre Prozesse eingreifen, oder wenn Sie untergeordnete Prozesse starten müssen, die ein Ergebnis an ein übergeordnetes

Objekt zurückgeben, sollten Sie die Amazon SWF SWF-API direkt verwenden oder auf Java migrieren. AWS Flow Framework

Weitere Informationen dazu finden Sie AWS Step Functions unter:

- [AWS Step Functions Entwicklerhandbuch](#)
- [AWS Step Functions API Reference](#)
- [AWS Step Functions Referenz zur Befehlszeile](#)

Verwenden Sie die Amazon SWF SWF-API direkt

Während das AWS Flow Framework für Ruby einen Teil der Komplexität von Amazon SWF bewältigt, können Sie die Amazon SWF SWF-API auch direkt verwenden. Die direkte Verwendung des API gestattet Ihnen, Workflows zu erstellen, in denen Sie vollständige Kontrolle über ihre Implementierung und Koordinierung haben, ohne sich um die zugrunde liegenden Komplexitäten kümmern zu müssen, wie z. B. den Fortschritt nachverfolgen und den Status verwalten zu müssen.

- [Amazon Simple Workflow Service — Entwicklerhandbuch](#)
- [Amazon Simple Workflow Service API-Referenz](#)

Dokumentverlauf

In der folgenden Tabelle werden die wichtigen Änderungen an der Dokumentation seit der letzten Version des Amazon Simple Workflow Service Developer Guide beschrieben.

Änderung	Beschreibung	Änderungsdatum
Update nur für die Dokumentation	Amazon SWF enthält jetzt einen Abschnitt über AWS Benutzerbenachrichtigungen, der als zentraler Ort für Ihre AWS Benachrichtigungen im AWS-Managementkonsole. AWS-Service Weitere Informationen finden Sie unter Verwendung AWS-Benutzerbenachrichtigungen mit Amazon Simple Workflow Service .	4. Mai 2023
Aktualisierung	Amazon SWF bietet jetzt ein neues Konsolenerlebnis für die Verwaltung von SWF-Workflows und deren ausführungsbefehlbezogenen Aktionen. Weitere Informationen finden Sie in den Amazon SWF SWF-Konsolen-Tutorials .	12. September 2022
Aktualisierung	Der Kontingente für Aufgabenausführungen Abschnitt und die Amazon SWF-Metriken für CloudWatch Seite wurden um Informationen zur Verwendung von Nicht-ASCII-Ressourcennamen für erweitertMaximum tasks scheduled per second. CloudWatch	12. Mai 2021
Neues Feature	Amazon Simple Workflow Service unterstützt jetzt Amazon EventBridge. Weitere Informationen finden Sie unter: <ul style="list-style-type: none"> EventBridge für Amazon SWF EventBridge Benutzerhandbuch 	18. Dezember 2020
Neues Feature	Amazon Simple Workflow Service unterstützt IAM-Berechtigungen mithilfe von Tags. Weitere Informationen finden Sie unter den folgenden Topics. <ul style="list-style-type: none"> Schlagworte in Amazon SWF 	20. Juni 2019

Änderung	Beschreibung	Änderungsdatum
	<ul style="list-style-type: none"> • Verwalten von Tags • Taggen Sie Workflow-Ausführungen • Steuern Sie den Zugriff auf Domänen mit Tags • TagResource • UntagResource • ListTagsForResource • RegisterDomain 	
Neues Feature	Amazon Simple Workflow Service ist jetzt in der Region Europa (Stockholm) verfügbar.	12. Dezember 2018
Aktualisierung	Das Thema CloudTrail Integration von Amazon Simple Workflow Service wurde verbessert. Siehe API-Aufrufe aufzeichnen mit AWS CloudTrail .	7. August 2018
Aktualisierung	Es wurden Informationen zur neuen PendingTasks Metrik für hinzugefügt CloudWatch. Weitere Informationen finden Sie unter Amazon SWF-Metriken .	18. Juni 2018
Aktualisierung	Verbesserte Syntaxmarkierungen in Code-Beispielen.	29. März 2018
Aktualisierung	Thema hinzugefügt, das Optionen für Benutzer von Ruby Flow für die Migration von dieser Plattform beschreibt. Weitere Informationen finden Sie unter Migrationsoptionen für Ruby Flow .	9. März 2018
Aktualisierung	Verbesserte Navigation im Thema für fortgeschrittene Konzepte. Siehe Fortgeschrittene Workflow-Konzepte in Amazon SWF .	19. Februar 2018
Aktualisierung	Die Dokumentation der CloudWatch Metriken wurde verbessert, indem gültige Statistikinformationen hinzugefügt wurden. Siehe Amazon SWF-Metriken für CloudWatch .	4. Dezember 2017

Änderung	Beschreibung	Änderungsdatum
Aktualisierung	Änderungen am Inhaltsverzeichnis zur Verbesserung der Dokumentstruktur. Neue Informationen zu API- und Entscheidungsereignismetriken hinzugefügt.	9. November 2017
Aktualisierung	Der Abschnitt Amazon SWF SWF-Kontingente wurde mit den Drosselungslimits für alle Regionen aktualisiert.	18. Oktober 2017
Aktualisierung	<code>task_list</code> wurde in <code>workflowId</code> im Erste Schritte mit Amazon SWF geändert, um Verwechslungen mit der <code>activity_list</code> zu vermeiden.	25. Juli 2017
Aktualisierung	Im gesamten Handbuch wurden die Codebeispiele bereinigt.	5. Juni 2017
Aktualisierung	Die Anordnung und die Inhalte des Handbuchs wurden vereinfacht und verbessert.	19. Mai 2017
Aktualisierung	Aktualisierungen und Korrekturen an Links	16. Mai 2017
Aktualisierung	Aktualisierungen und Korrekturen an Links	1. Oktober 2016
Unterstützung für Lambda-Tasks	Sie können Lambda-Aufgaben zusätzlich zu herkömmlichen Aktivitätsaufgaben in Ihren Workflows angeben. Weitere Informationen finden Sie unter AWS Lambda Aufgaben in Amazon SWF .	21. Juli 2015
Unterstützung zur Angabe der Aufgabenpriorität	Amazon SWF unterstützt jetzt das Festlegen der Priorität von Aufgaben auf einer Aufgabenliste und versucht, Aufgaben mit höherer Priorität vor Aufgaben mit niedrigerer Priorität zuzustellen. Informationen zum Festlegen der Aufgabenpriorität für Workflows und Aktivitäten finden Sie unter Aufgabenpriorität in Amazon SWF festlegen .	17. Dezember 2014

Änderung	Beschreibung	Änderungsdatum
Aktualisierung	Es wurde ein neues Thema hinzugefügt, das beschreibt, wie Amazon SWF SWF-API-Aufrufe protokolliert werden mit CloudTrail: API-Aufrufe aufzeichnen mit AWS CloudTrail .	8. Mai 2014
Aktualisierung	Zwei neue Themen im Zusammenhang mit CloudWatch Metriken für Amazon SWF wurden hinzugefügt: Amazon SWF-Metriken für CloudWatch , das eine Liste und Beschreibungen der unterstützten Metriken enthält, und Amazon SWF-Metriken für die CloudWatch Verwendung von anzeigen AWS-Managementkonsole , das Informationen zum Anzeigen von Metriken und zum Einstellen von Alarmen mit dem bietet AWS-Managementkonsole.	28. April 2014
Aktualisierung	Neuer Abschnitt: Zusätzliche Ressourcen und Referenzinformationen für Amazon SWF . Dieser Abschnitt enthält einige Service-Referenzinformationen sowie Informationen zu zusätzlicher Dokumentation, Beispielen, Code und anderen Webressourcen für Amazon SWF SWF-Entwickler.	19. März 2014
Aktualisierung	Es wurde ein Workflow-Tutorial hinzugefügt. Siehe Erste Schritte mit Amazon SWF .	25. Oktober 2013
Aktualisierung	Informationen zu AWS CLI und Beispiele zu wurden hinzugefügt.	26. August 2013
Aktualisierung	Aktualisierungen und Korrekturen.	1. August 2013
Aktualisierung	Das Dokument wurde aktualisiert und beschreibt nun, wie IAM für die Zugriffskontrolle verwendet wird.	22. Februar 2013
Erstversion	Dies ist die erste Version des Amazon Simple Workflow Service Developer Guide.	16. Oktober 2012

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.