



Entwicklerhandbuch

# Amazon Simple Workflow Service



API-Version 2012-01-25

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

# Amazon Simple Workflow Service: Entwicklerhandbuch

Copyright © 2024 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
Entwickeln mit AWS SDKs .....	1
Betrachten Sie die AWS Flow Framework .....	2
Erstellen mit der HTTP-Service-API .....	2
Einführung in Amazon SWF .....	3
Konzepte .....	3
Workflow-Ausführung .....	5
Einrichtung .....	6
Erste Schritte .....	8
Über den Workflow .....	9
Voraussetzungen .....	10
Anleitungsschritte .....	10
Teil 1: Verwenden von Amazon SWF mit dem SDK for Ruby .....	10
Schließen Sie das AWS SDK for Ruby mit ein .....	11
Konfigurieren von AWS-Sitzung .....	11
Registrieren einer Amazon SWF-Domain .....	13
Nächste Schritte .....	14
Teil 2: Implementieren des Workflows .....	14
Entwerfen des Workflows .....	14
Einrichten des Workflow-Codes .....	15
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“ .....	26
Definieren von „SubscribeTopicActivity“ .....	29
Definieren von „WaitForConfirmationActivity“ .....	32
Definieren von „SendResultActivity“ .....	35
Nächste Schritte .....	36
Teil 4: Implementieren des Pollers Aktivitätsaufgaben .....	36
Ausführen des Workflows .....	40
Wie geht es weiter? .....	44

Tutorials .....	45
Registrieren einer Domäne .....	45
Registrieren eines Workflow-Typs .....	48
Registrieren eines Aktivitätstyps .....	53
Starten einer Workflow-Ausführung .....	58
So starten Sie eine Workflow-Ausführung mithilfe der Konsole .....	58
Anzeigen ausstehender Aufgaben .....	64
Verwalten von Workflow-Ausführungen .....	65
Verwalten von Amazon SWF-Domänen .....	73
Funktionsweise von Amazon SWF .....	75
Workflows .....	75
Was ist ein Workflow? .....	76
Ein einfaches Beispiel für einen Workflow: E-Commerce-Anwendung .....	76
Registrierung und Ausführung des Workflows .....	77
Weitere Informationen finden Sie unter: .....	78
Workflow-Verlauf .....	78
Akteure .....	83
Was ist ein Akteur in Amazon SWF? .....	83
Workflow-Starter .....	84
Entscheider .....	84
Aktivitäts-Worker .....	86
Datenaustausch zwischen Akteuren .....	86
Aufgaben .....	87
Domänen .....	88
Objekt-IDs .....	89
Aufgabenlisten .....	89
Entscheidungsaufgabenlisten .....	90
Aktivitätsaufgabenlisten .....	90
Aufgabenweiterleitung .....	91
Beenden einer Workflow-Ausführung .....	91
Lebenszyklus einer -Workflow-Ausführung .....	92
Lebenszyklus einer Workflow-Ausführung .....	93
Abfragen von Aufgaben .....	102
Fortgeschrittene Konzepte .....	102
Versioning .....	103
Signale .....	104

Untergeordnete Workflows .....	106
Marker .....	108
Tags .....	110
Implementieren einer exklusiven Wahl .....	113
Timer .....	116
Abbrechen einer Aktivitätsaufgabe .....	117
Sicherheit .....	120
Datenschutz .....	120
Verschlüsselung .....	121
Identitäts- und Zugriffsverwaltung .....	122
Zielgruppe .....	123
Authentifizierung mit Identitäten .....	124
Verwalten des Zugriffs mit Richtlinien .....	128
Zugriffskontrolle .....	130
Richtlinienaktionen .....	131
Richtlinienressourcen .....	132
Bedingungsschlüssel für die Richtlinie .....	132
ACLs .....	133
ABAC .....	134
Temporäre Anmeldeinformationen .....	134
Prinzipal-Berechtigungen .....	135
Servicerollen .....	135
Serviceverknüpfte Rollen .....	136
Identitätsbasierte Richtlinien .....	136
Ressourcenbasierte Richtlinien .....	137
So funktioniert Amazon Simple Workflow Service mit IAM .....	137
Beispiele für identitätsbasierte Richtlinien .....	138
Grundlegende Prinzipien .....	142
Amazon SWF IAM-Richtlinien .....	143
Übersicht über API-Befehle .....	152
Tagbasierte Richtlinien .....	161
Amazon VPC-Endpunkte für Amazon SWF .....	162
Fehlerbehebung .....	165
Protokollieren und Überwachen .....	167
Amazon SWF-Metriken für CloudWatch .....	167
Anzeigen von Amazon SWF SWF-Metriken .....	178

Aufnahme in CloudTrail .....	182
EventBridge für Amazon SWF .....	189
Verwendung AWS-Benutzerbenachrichtigungen mit Amazon SWF .....	198
Compliance-Validierung .....	198
Ausfallsicherheit .....	199
Sicherheit der Infrastruktur .....	199
Konfigurations- und Schwachstellenanalyse .....	200
Verwendung von AWS CLI .....	201
Arbeiten mit APIs .....	203
Senden von HTTP-Anfragen .....	203
Inhalt des HTTP-Headers .....	204
HTTP-Textinhalt .....	206
Beispiel , JSON-Anforderung und -Antwort .....	206
Berechnen der HMAC-SHA-Signatur .....	208
Liste der Amazon-SWF-Aktionen .....	210
Aktionen im Zusammenhang mit Aktivitäten .....	211
Aktionen im Zusammenhang mit Entscheidern .....	211
Aktionen im Zusammenhang mit Workflow-Ausführungen .....	211
Aktionen im Zusammenhang mit der Administration .....	212
Sichtbarkeitsaktionen .....	212
Erstellen eines einfachen Workflows .....	213
Modellieren des Workflows und der entsprechenden Aktivitäten .....	214
Registrieren einer Domäne .....	215
Weitere Informationen finden Sie unter: .....	215
Einrichten von Timeout-Werten .....	215
Kontingente für Timeout-Werte .....	216
Timeouts für Workflow-Ausführungen und Entscheidungsaufgaben .....	216
Timeouts für Aktivitätsaufgaben .....	216
Weitere Informationen finden Sie unter: .....	217
Registrieren eines Workflow-Typs .....	217
Weitere Informationen finden Sie unter: .....	218
Registrieren eines Aktivitätstyps .....	218
Weitere Informationen finden Sie unter: .....	219
Lambda-Aufgaben .....	219
Informationen zu AWS Lambda .....	219
Vorteile und Einschränkungen der Verwendung von Lambda-Aufgaben .....	220

Verwenden von Lambda-Aufgaben in Ihren -Workflows .....	220
Entwickeln eines Aktivitäts-Workers .....	226
Abrufen von Aktivitätsaufgaben .....	227
Verarbeiten von Aktivitätsaufgaben .....	227
Senden von Heartbeats für Aktivitätsaufgaben .....	228
Abschließen oder fehlschlagen einer Aktivitätsaufgabe .....	229
Starten von Aktivitäts-Workern .....	230
Entwickeln von Entscheidern .....	231
Definieren einer Koordinationslogik .....	232
Abrufen von Entscheidungsaufgaben .....	233
Anwenden der Koordinationslogik .....	235
Reagieren auf Entscheidungen .....	236
Beenden der Workflow-Ausführung .....	237
Starten von Entscheidern .....	238
Starten von Workflow-Ausführungen .....	239
Einrichten der Aufgabenprioritäten .....	240
Einrichten der Aufgabenpriorität für Workflows .....	241
Einrichten der Aufgabenpriorität für Aktivitäten .....	243
Aktionen, die Informationen zur Aufgabenpriorität zurückgeben .....	245
Fehlerbehandlung .....	245
Validierungsfehler .....	245
Fehler beim Umsetzen von Aktionen und Entscheidungen .....	246
Zeitüberschreitungen .....	246
Fehler durch Benutzercode .....	247
Fehler beim Schließen einer Workflow-Ausführung .....	247
Kontingente .....	248
Allgemeine Kontokontingente für Amazon SWF .....	248
Kontingente für Workflow-Ausführungen .....	249
Kontingente für Aufgabenausführungen .....	250
Drosselungskontingente für Amazon SWF .....	252
Drosselungsquoten für alle Regionen .....	252
Entscheidungsquoten für alle Regionen .....	254
Kontingente auf Workflow-Ebene .....	254
Beantragen einer Kontingenterhöhung .....	255
Ressourcen .....	256
Zeitüberschreitungstypen .....	256

Zeitüberschreitungen in Workflow- und Entscheidungsaufgaben .....	257
Zeitüberschreitungen in Aktivitätsaufgaben .....	258
Endpunkte .....	259
Ergänzende -Dokumentation .....	260
API-Referenz für Amazon Simple Workflow Service .....	260
AWS Flow Framework Dokumentation .....	260
AWS SDK-Dokumentation .....	260
AWS CLI Dokumentation .....	262
Webressourcen .....	263
Amazon SWF SWF-Forum .....	263
Amazon SWF SWF-Häufig gestellte Fragen .....	263
Amazon SWF SWF-Videos .....	263
Optionen für Ruby Flow .....	264
Weitere Nutzung des Ruby Flow Frameworks .....	264
Migration auf das Java Flow Framework .....	265
Migration auf Step Functions .....	265
Verwenden Sie die Amazon SWF SWF-API direkt .....	266
Dokumentverlauf .....	267
AWS-Glossar .....	272
.....	cclxxiii



# Was ist Amazon Simple Workflow Service?

Der Amazon Simple Workflow Service (Amazon SWF) bietet eine Möglichkeit, Hintergrundaufträge mit parallelen oder sequenziellen Schritten zu erstellen, auszuführen und zu skalieren. Mit Amazon SWF können Sie 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 Koordination von Aufgaben in der gesamten Anwendung umfasst die Verwaltung von Abhängigkeiten zwischen Aufgaben, die Planung und die Gleichzeitigkeit im logischen Ablauf Ihrer Anwendung. Amazon SWF gibt Ihnen die Kontrolle über die Implementierung von Aufgaben und deren Koordination, ohne sich über die zugrunde liegenden Komplexitäten Gedanken zu machen, wie z. B. die Verfolgung ihres Fortschritts und die Aufrechterhaltung ihres Status.

Wenn Sie Amazon SWF verwenden, implementieren Sie Worker zur Ausführung von Aufgaben. Worker können entweder in einer Cloud-Infrastruktur wie Amazon Elastic Compute Cloud (Amazon EC2) oder in Ihrem eigenen Standort ausgeführt werden. Sie können Aufgaben erstellen, die lange laufen oder ausfallen, eine Zeitüberschreitung aufweisen oder Neustarts erfordern oder die mit unterschiedlichem Durchsatz und unterschiedlicher Latenz abgeschlossen werden können. Amazon SWF speichert Aufgaben und weist sie Auftragnehmern zu, wenn sie bereit sind, verfolgt ihren Fortschritt und behält ihren Status bei, einschließlich Details zu ihrem Abschluss.

Um Aufgaben zu koordinieren, schreiben Sie ein Programm, das den neuesten Aufgabenstatus von Amazon SWF erhält und diesen Status verwendet, um nachfolgende Aufgaben zu initiieren. Amazon SWF behält den Ausführungsstatus einer Anwendung dauerhaft bei, sodass Ihre Anwendung widerstandsfähig gegenüber Ausfällen einzelner Komponenten ist. Mit Amazon SWF können Sie Anwendungskomponenten unabhängig voneinander erstellen, bereitstellen, skalieren und ändern.

Amazon SWF eignet sich für Anwendungsfälle, die eine Koordination von Aufgaben erfordern, einschließlich Medienverarbeitung, Webanwendungs-Backends, Workflows für Geschäftsprozesse und Analysepipelines.

## Entwickeln mit AWS SDKs

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

Sie können Entscheider, Aktivitäts-Worker oder Workflow-Starter mithilfe der API entwickeln, die von diesen Bibliotheken bereitgestellt wird. Und Sie können Sichtbarkeitsvorgänge über diese Bibliotheken verwenden, damit Sie Ihre eigenen Tools für die Überwachung und Berichterstattung von Amazon SWF entwickeln können.

Um Tools für die Entwicklung und Verwaltung von Anwendungen in AWS, einschließlich SDKs, herunterzuladen, gehen Sie zum [Entwicklerzentrum](#).

Ausführliche Informationen zu den Amazon SWF-Methoden in jedem SDK finden Sie in der sprachspezifischen Referenzdokumentation für das SDK.

## Betrachten Sie die AWS Flow Framework

Ist AWS Flow Framework 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 der verwenden Sie vorkonfigurierte Typen AWS Flow Framework, um die Definition Ihres Workflows direkt Methoden in Ihrem Programm zuzuordnen. Die AWS Flow Framework unterstützt objektorientierte Standardkonzepte, wie z. B. die Ausnahmebehandlung, wodurch die Implementierung komplexer Workflows vereinfacht wird. Programme, die mit der geschrieben wurden, AWS Flow Framework können vollständig in Ihrem bevorzugten Editor oder Ihrer bevorzugten IDE erstellt, ausgeführt und debuggt werden. Weitere Informationen finden Sie auf der [-AWS Flow Framework](#) Website und im [AWS Flow Framework für Java-Entwicklerhandbuch](#).

## Erstellen mit der HTTP-Service-API

Amazon SWF bietet Service-Operationen, auf die über HTTP-Anfragen zugegriffen werden kann. Sie können diese Operationen verwenden, um direkt mit Amazon SWF zu kommunizieren, und Sie können sie verwenden, um Ihre eigenen Bibliotheken in jeder Sprache zu entwickeln, die über HTTP mit Amazon SWF kommunizieren kann.

Sie können Entscheider, Aktivitätsarbeitsnehmer oder Workflow-Starter mithilfe der Service-API entwickeln. Sie können Sichtbarkeitsvorgänge auch über die API verwenden, um Ihre eigenen Überwachungs- und Berichtstools zu entwickeln.

Informationen zur Verwendung der API finden Sie unter [Erstellen von HTTP-Anforderungen an Amazon SWF](#).

# Einführung in Amazon SWF

Eine zunehmende Zahl von Anwendungen hängt von asynchroner und verteilter Verarbeitung ab. Die Skalierbarkeit solcher Anwendungen ist die Hauptmotivation für die Verwendung dieses Ansatzes. Durch das Konzipieren unabhängiger verteilter Komponenten haben Entwickler die Flexibilität, Teile der Anwendung unabhängig voneinander bereitzustellen und zu erweitern, wenn die Auslastung der Anwendung zunimmt. Ein weiterer Antrieb ist die Verfügbarkeit von Cloud-Services. Nun, da Anwendungsentwickler anfangen, die Vorteile von Cloud Computing zu nutzen, müssen sie ihre bestehenden Komponenten vor Ort mit zusätzlichen Cloud-basierten Komponenten kombinieren. Ein weiterer Vorteil des asynchronen und verteilten Ansatzes ist die damit verbundene verteilte Struktur des Prozesses, die von der Anwendung modelliert wird. So kann beispielsweise die Automatisierung eines Geschäftsprozesses zur Auftragsbearbeitung mehrere Systeme und von Menschen zu erledigende Aufgaben umfassen.

Das Entwickeln solcher Anwendungen kann kompliziert sein. Es erfordert die Koordination der Ausführung mehrerer verteilter Komponenten sowie den richtigen Umgang mit den höheren Latenzen und mit der Unzuverlässigkeit von Remote-Kommunikation. Um dies zu bewerkstelligen, müssten Sie normalerweise eine komplizierte Infrastruktur schreiben, einschließlich Nachrichtenwarteschlangen und Datenbanken, zusammen mit der komplexen Logik, um diese zu synchronisieren.

Der Amazon Simple Workflow Service (Amazon SWF) erleichtert die Entwicklung asynchroner und verteilter Anwendungen, indem ein Programmiermodell und eine Infrastruktur für die Koordination verteilter Komponenten und die zuverlässige Aufrechterhaltung ihres Ausführungsstatus bereitgestellt werden. Wenn Sie sich auf Amazon SWF verlassen, können Sie sich darauf konzentrieren, die Aspekte Ihrer Anwendung zu erstellen, die sie unterscheiden.

## Simple Workflow-Konzepte

Die grundlegenden Konzepte, die für das Verständnis von Amazon SWF-Workflows erforderlich sind, werden unten vorgestellt und in den nachfolgenden Abschnitten dieses Handbuchs näher erläutert. Nachfolgend erhalten Sie eine allgemeine Übersicht über die Struktur und die Komponenten eines Workflows.

Das grundlegende Konzept in 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 erhalten und die erforderlichen Aktionen

durchführen, um die Bestellung abzuwickeln. Jeder Workflow wird in einer - AWS Ressource namens Domain ausgeführt, die den Umfang des Workflows steuert. Ein AWS Konto kann mehrere Domains haben, von denen jede mehrere Workflows enthalten kann, aber Workflows in verschiedenen Domains können nicht interagieren.

Beim Entwerfen eines Amazon SWF-Workflows definieren Sie genau jede der erforderlichen Aktivitäten. Anschließend registrieren Sie jede Aktivität bei Amazon SWF als Aktivitätstyp. Wenn Sie die Aktivität registrieren, stellen Sie Informationen wie einen Namen und eine Version bereit sowie einige Timeout-Werte, basierend darauf, wie lange die Aktivität Ihrer Schätzung nach dauert. Beispielsweise kann es sein, dass der Kunde erwartet, dass eine Bestellung innerhalb von 24 Stunden geliefert wird. Solche Erwartungen beeinflussen die Timeout-Werte, die Sie beim Registrieren Ihrer Aktivitäten festlegen.

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, muss 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 Aktivitäts-Worker ist ein Programm, das Aktivitätsaufgaben empfängt, sie durchführt und Ergebnisse zurückgibt. Beachten Sie, dass die Aufgabe selbst unter Umständen von einer Person durchgeführt wird. In diesem Fall würde die Person die Aktivitäts-Worker-Software für den Empfang und die Planung der Aufgabe verwenden. Ein Beispiel hierfür ist etwa ein Analytiker von Statistiken, der Gruppen von Daten erhält, sie analysiert und dann die Analyse zurücksendet.

Aktivitätsaufgaben – und die Aktivitäts-Worker, die sie ausführen – können synchron oder asynchron ausgeführt werden. Sie können auf mehrere Computer verteilt sein, vielleicht in verschiedenen geographischen Regionen, oder sie können alle auf demselben Computer ausgeführt werden. Unterschiedliche Aktivitäts-Worker können in verschiedenen Programmiersprachen geschrieben und auf unterschiedlichen Betriebssystemen ausgeführt werden. So kann zum Beispiel ein Aktivitäts-Worker auf einem Desktopcomputer in Asien ausgeführt werden, während ein anderer Aktivitäts-Worker auf einem tragbaren Gerät in Nordamerika ausgeführt wird.

Die Koordinationslogik in einem Workflow ist in einem Softwareprogramm enthalten, das als Entscheider bezeichnet wird. Der Entscheider plant Aktivitätsaufgaben, stellt Eingabedaten für die Aktivitäts-Worker bereit, verarbeitet Ereignisse, die eintreffen, während der Workflow aktuell läuft, und beendet (oder schließt) letztendlich den Workflow, wenn das Ziel erreicht ist.

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

Der Entscheider leitet den Workflow weiter, indem er Entscheidungsaufgaben von Amazon SWF erhält und mit Entscheidungen auf Amazon SWF reagiert. Eine Entscheidung stellt eine Aktion oder eine Gruppe von Aktionen dar, die die nächsten Schritte im Workflow sind. Eine typische Entscheidung ist das Planen einer Aktivitätsaufgabe. Entscheidungen können auch verwendet werden, um Timer festzulegen, die Ausführung einer Aktivitätsaufgabe zu verzögern, den Abbruch einer Aktivitätsaufgabe anzufordern, die bereits ausgeführt wird, und den Workflow abzuschließen oder zu beenden.

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

Amazon SWF informiert den Entscheider über den Status des Workflows, indem es bei jeder Entscheidungsaufgabe eine Kopie des aktuellen Workflow-Ausführungsverlaufs einfügt. Der Workflow-Ausführungsverlauf setzt sich zusammen aus Ereignissen, wobei ein Ereignis eine bedeutende Änderung des Status der Workflow-Ausführung darstellt. Beispiele von Ereignissen sind etwa der Abschluss einer Aufgabe, die Benachrichtigung, dass das Zeitlimit für eine Aufgabe überschritten wurde, oder der Ablauf eines Timers, der vorher in der Workflow-Ausführung festgelegt wurde. Der Verlauf ist eine umfassende, konsistente und autoritative Aufzeichnung des Fortschritts des Workflows.

Die Amazon SWF-Zugriffskontrolle verwendet AWS Identity and Access Management (IAM), mit der Sie den Zugriff auf AWS Ressourcen kontrolliert und eingeschränkt ermöglichen können. Sie können einem Benutzer beispielsweise Zugriff auf Ihr Konto gewähren, aber nur, um bestimmte Workflows in einer speziellen Domäne auszuführen.

## Workflow-Ausführung

Zusammenführung der in den vorherigen Abschnitten erörterten Ideen, hier ein Überblick über die Schritte zur Entwicklung und Ausführung eines Workflows in Amazon SWF:

1. Schreiben Sie Aktivitäts-Worker, die die Verarbeitungsschritte in Ihrem Workflow implementieren.
2. Schreiben Sie einen Entscheider zur Implementierung der Koordinationslogik Ihres Workflows.

### 3. Registrieren Sie Ihre Aktivitäten und Ihren Workflow bei Amazon SWF .

Sie können diesen Schritt programmgesteuert oder mithilfe der ausführen AWS Management Console.

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

Diese Akteure können auf jedem Computer ausgeführt werden, der auf einen Amazon SWF-Endpunkt zugreifen kann. Sie können beispielsweise Datenverarbeitungs-Instances in der Cloud verwenden, z. B. Amazon Elastic Compute Cloud (Amazon EC2), Server in Ihrem Rechenzentrum oder sogar ein Mobilgerät, um einen Entscheider oder einen Aktivitätsarbeitsnehmer zu hosten. Nach dem Start sollten der Entscheider und die Aktivitätsarbeiter damit beginnen, Amazon SWF nach Aufgaben abzufragen.

### 5. Starten Sie eine oder mehrere Ausführungen Ihres Workflows.

Ausführungen können entweder programmgesteuert oder über die AWS Management Console initiiert werden.

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, Entscheidungen zu erzeugen, die Aktivitätsaufgaben initiieren. Die Ausführung dauert an, bis Ihr Entscheider die Entscheidung trifft, die Ausführung abzuschließen.

### 6. Zeigen Sie Workflow-Ausführungen mit der an AWS Management Console.

Sie können komplette Ausführungsdetails sowie abgeschlossene Ausführungen filtern und anzeigen. Sie können beispielsweise eine offene Ausführung auswählen, um zu sehen, welche Aufgaben abgeschlossen wurden und was ihre Ergebnisse waren.

## Einrichtung

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

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

Jeder Endpunkt in Amazon SWF ist völlig unabhängig. Alle Domains, Workflows und Aktivitäten, die Sie in einer Region registriert haben, geben keine Daten oder Attribute für diejenigen in einer anderen Region frei.

Wenn Sie eine Amazon SWF-Domain, einen Workflow oder eine Aktivität registrieren, existiert sie nur innerhalb der Region, in der Sie sie registriert haben. Sie können beispielsweise eine Domäne mit dem Namen SWF-Flows-1 in zwei verschiedenen Regionen registrieren, aber sie teilen sich keine Daten oder Attribute – jede dieser Regionen fungiert als vollständig unabhängige Domäne.

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

# Abonnement-Workflow-Tutorial mit Amazon SWF und Amazon SNS

Dieser Abschnitt enthält ein Tutorial, in dem beschrieben wird, wie Sie eine Amazon SWF-Workflow-Anwendung erstellen, die aus einer Reihe von vier Aktivitäten besteht, die sequenziell ausgeführt werden. Außerdem wird Folgendes behandelt:

- Einrichten von Standard- und Laufzeit-Workflow- und Aktivitätsoptionen
- Abfragen von Amazon SWF für Entscheidungs- und Aktivitätsaufgaben.
- Übergeben von Daten zwischen den Aktivitäten und dem Workflow mit Amazon SWF .
- Warten auf menschliche Aufgaben und Melden von Heartbeats von einer Aktivitätsaufgabe an Amazon SWF.
- Verwenden von Amazon SNS zum Erstellen eines Themas, zum Abonnieren eines Benutzers und zum Veröffentlichen von Nachrichten an abonnierten Endpunkten.

Sie können [Amazon Simple Workflow Service \(Amazon SWF\)](#) und [Amazon Simple Notification Service \(Amazon SNS\)](#) zusammen verwenden, um einen Workflow „menschliche Aufgabe“ zu emulieren – einen, in 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 entweder per E-Mail, SMS oder beides mit dem Benutzer zu kommunizieren.

In diesem Tutorial wird die verwendet, [AWS SDK for Ruby](#) um auf Amazon SWF und Amazon SNS zuzugreifen. Es sind jedoch viele Entwicklungsoptionen verfügbar, darunter AWS Flow Framework für Ruby, was eine einfachere Koordination und Kommunikation mit Amazon SWF ermöglicht.

## Note

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

Themen



- [Über den Workflow](#)
- [Voraussetzungen](#)
- [Anleitungsschritte](#)
- [Abonnement-Workflow-Tutorial Teil 1: Verwenden von Amazon SWF mit dem AWS SDK for Ruby](#)
- [Abonnement-Workflow, Anleitung Teil 2: Implementieren des Workflows](#)
- [Abonnement-Workflow, Tutorial Teil 3: Implementieren der Aktivitäten](#)
- [Abonnement-Workflow, Anleitung Teil 4: Implementieren des Pollers Aktivitätsaufgaben](#)
- [Abonnement-Workflow Tutorial: 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 wird auch beschrieben, wie Sie nach Abschluss Aktivitätsergebnisse bereitstellen und wie Sie Eingaben an eine geplante Aufgabe übergeben. Amazon SWF übernimmt die Koordination und Bereitstellung von Informationen zwischen den Aktivitäten und dem Workflow und umgekehrt.

Wir verwenden außerdem sowohl Tastatureingabe als auch Amazon SNS, um die Kommunikation zwischen Amazon SWF und dem Menschen zu übernehmen, der Daten für den Workflow bereitstellt. In der Praxis können Sie viele verschiedene Techniken verwenden, um mit menschlichen Benutzern zu kommunizieren, aber Amazon SNS bietet eine sehr einfache Möglichkeit, E-Mail- oder Textnachrichten zu verwenden, um den Benutzer über Ereignisse im Workflow zu informieren.

# Voraussetzungen

Für diese Anleitung benötigen Sie Folgendes:

- [Amazon Web Services-Konto](#)
- [Einen Ruby-Interpreter](#)
- [AWS SDK for 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 gilt unabhängig von der gewählten Entwicklungsoption für die Verwendung von Amazon SWF und Amazon SNS.

## Anleitungsschritte

Diese Anleitung ist in die folgenden Schritte unterteilt:

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

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

Themen

- [Schließen Sie das AWS SDK for Ruby mit ein](#)
- [Konfigurieren von AWS-Sitzung](#)
- [Registrieren einer Amazon SWF-Domain](#)
- [Nächste Schritte](#)

## Schließen Sie das AWS SDK for Ruby mit ein

Erstellen Sie zunächst eine Datei namens `utils.rb`. Mit dem Code in dieser Datei wird die Amazon SWF-Domäne abgerufen bzw. bei Bedarf erstellt, die sowohl vom Workflow als auch vom Aktivitätscode verwendet wird. Die Datei bietet außerdem einen Ort, um für alle Klassen gemeinsamen Code zu speichern.

Zunächst müssen wir die `aws-sdk-v1`-Bibliothek in unserem Code, sodass wir die vom SDK bereitgestellten Funktionen für Ruby nutzen können.

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

Damit erhalten wir Zugriff auf die `AWS::Namespace`, der die Möglichkeit bietet, globale Sitzungsrelevante Werte, wie Ihre `AWS`-Anmeldeinformationen und Region sowie Zugriff auf `AWS`-Service-APIs.

## Konfigurieren von `AWS`-Sitzung

Im nächsten Schritt konfigurieren wir `AWS`-Sitzung durch Einstellen unserer `AWS`-Anmeldeinformationen (die für den Zugriff benötigt werden `AWS`-Dienstleistungen) und das `AWS`-Region, die verwendet werden soll

Es gibt eine Reihe von Möglichkeiten [einstellen `AWS`-Anmeldeinformationen im `AWS`-`SDK` for Ruby](#): Sie können in den Umgebungsvariablen (`AWS_ACCESS_KEY_ID` und `AWS_SECRET_ACCESS_KEY`) oder mit `AWS.config` aus. 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 diese Datei jetzt und ersetzen Sie die Strings beginnend mit `REPLACE_MIT_` mit Ihrem `AWS`-Zugriffsschlüssel-ID und geheimen Zugriffsschlüssel. Informationen über Ihre `AWS`-Zugriffsschlüssel, siehe [Was sind die Sicherheitsanmeldeinformationen?](#) im Allgemeine Amazon Web Services Services-Referenz aus.

Außerdem müssen wir die `AWS`-Region, die verwendet werden soll Weil wir das benutzen werden [Short Message Service \(SMS\)](#) Um mit Amazon SNS SMS an das Telefon des Benutzers

zu senden, müssen wir die von Amazon SNS unterstützte Region verwenden. Siehe [Unterstützte Regionen und Länder](#) im Amazon Simple Notification Service-Entwicklerhandbuch.

#### 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-Funktion aus dem Beispiel entfernen und E-Mail als alleinigen Endpunkt zum Abonnieren des Amazon SNS SNS-Themas verwenden.

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

Wir fügen jetzt etwas Code hinzu `utils.rb` Um die Konfigurationsdatei zu laden, rufen Sie die Anmeldeinformationen des Benutzers ab und geben Sie die Anmeldeinformationen und die Region an `AWS.config` aus.

```
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
```

## Registrieren einer Amazon SWF-Domain

Um Amazon SWF nutzen zu können, müssen Sie eine Domäne: Eine benannte Entity, die Ihre Workflows und Aktivitäten spannt. Sie können viele Amazon SWF-Domains registrieren lassen, aber sie müssen alle eindeutige Namen in Ihrem AWS-Konto und Workflows können nicht domänenübergreifend interagieren: Alle Workflows und Aktivitäten für Ihre Anwendung müssen sich in derselben Domäne befinden, um interagieren zu können.

Da wir in unserer gesamten Anwendung die gleiche Domäne verwenden, erstellen wir eine Funktion in `utils.rb` benannt `init_domain`, die die Amazon SWF-Domain mit dem Namen `swfSampleDomain` abrufen lässt.

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-Domains im SDK for Ruby zu arbeiten, verwenden Sie [AWS::SimpleWorkflow.Domains](#), das eine zurückgibt [DomainCollection](#) das kann sowohl zum Aufzählen als auch zum Registrieren von Domains verwendet werden:

- Um festzustellen, ob eine Domäne bereits registriert ist, sehen Sie in der von [AWS::Simpleworkflow.domains.registered](#) bereitgestellten Liste nach.
- Verwenden Sie zum Registrieren einer neuen Domäne [AWS::Simpleworkflow.domains.register](#).

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
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 entwerfen, dass der gesamte Code in einer Quelldatei enthalten ist. Das würde aber nicht der Konstruktion von Amazon SWF entsprechen. 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 der Workflows und Aktivitäten verfolgt, kann Ihr Workflow die Planung und den Datentransfer Ihrer Aktivitäten unabhängig vom Ausführungsort koordinieren.

## Einrichten des Workflow-Codes

Als Erstes erstellen wir eine Datei namens `swf_sns_workflow.rb`. In dieser deklarieren wir eine Klasse mit dem Namen `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 Domänennamen, der Aktivitätsname und die Aktivitätsversion reichen aus, um Amazon SWF einen Aktivitätstyp zu identifizieren. Somit stehen alle Daten zur Verfügung, die für das Einplanen der Aktivitäten erforderlich sind.



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 }

    puts "registering workflow: #{workflow_name}, #{workflow_version},
    #{options.inspect}"
  end
end
```

```
@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.

Wird keine Übereinstimmung gefunden, wird ein neuer Workflow-Typ registriert (durch Aufruf von [register](#) in derselben `workflow_types`-Sammlung, in der wir den Workflow gesucht haben). Diesem werden der Name „swf-sns-workflow“, Version „1“ und die folgenden Optionen zugewiesen.

```
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 SWF-Zeitüberschreitungstypen](#).

## 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](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. Den Ereignistyp erhalten Sie über das zurückgegebene `event_type`-Mitglied des Ereignisses. Eine Liste sowie eine Beschreibung der Verlaufereignistypen finden Sie unter `historyEvent` im Amazon Simple Workflow Service API Referenz aus.

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`. Zunächst erhalten wir vermutlich `WorkflowExecutionStarted`. Wurde das Ereignis empfangen, signalisiert Amazon SWF Entscheider, dass er 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 lediglich Maps und bestehen aus einem Namen und einer Versionsnummer. Das ist jedoch alles, was Amazon SWF benötigt, um die Aktivität für die Planung zu identifizieren, sofern die Aktivität bereits registriert wurde.

```
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 Aktivitäts-Aufgabe zur Aktivitätsaufgabenliste, die wir beim Planen übergeben, um den Beginn der Aufgabe zu signalisieren. Wir beschäftigen uns in [Abonnement-Workflow, Tutorial Teil 3: Implementieren der Aktivitäten](#) mit Aktivitätsaufgaben, an dieser Stelle sollte aber erwähnt werden, dass wir hier die Aufgabe nicht ausführen. Wir weisen Amazon SWF nur an, dass es so sein sollte geplant aus.

Die nächste Aktivität, die wir ansprechen müssen, ist die `ActivityTaskCompleted`-Ereignis, das auftritt, wenn Amazon SWF 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 jeweils eine Aktivität zurzeit ausgeführt wird, nutzen wir die Gelegenheit, die gesamte Aufgabe aus dem `activity_list`-Stack abzurufen. Führt dies zu einer leeren Liste, wissen wir, dass unser Workflow abgeschlossen ist. In diesem

Fall signalisieren wir Amazon SWF, dass unser Workflow durch Aufrufen abgeschlossen ist [complete\\_workflow\\_execution](#) über die Aufgabe.

Enthält die Liste noch Einträge, planen wir die nächste Aktivität auf der Liste (auch hier an letzter Stelle). Dieses Mal prüfen wir jedoch, ob die vorherige Aktivität nach dem Abschluss Ergebnisdaten an Amazon SWF zurückgegeben hat, die dem Workflow in den Attributen des Ereignisses zur Verfügung gestellt werden, im optionalen `resultKey`. 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. Dennoch kann eine Aktivität andere Ereignisse als `ActivityTaskCompleted` generieren. Wir werden unseren Entscheidercode abschließen, indem wir den Demo-Handler-Code für die `ActivityTaskTimedOut` und `ActivityTaskFailed` Events und für die `WorkflowExecutionCompleted`-Ereignis, das generiert wird, wenn Amazon SWF `complete_workflow_execution` rufen Sie an, die wir machen, wenn uns die 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.

Zum Starten der Workflow-Ausführung rufen Sie [start\\_execution](#) für Ihren registrierten Workflow-Typ ([AWS::SimpleWorkflow::WorkflowType](#)) auf. 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.

Wir möchten, dass die Workflow-Ausführung startet, wenn wir die Datei ausführen. Fügen Sie deshalb einen Code hinzu, der die Klasse instanziiert und dann die von uns soeben definierte `start_execution`-Methode aufruft.

```
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 Aufgabenlistenname 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, Tutorial Teil 3: Implementieren der Aktivitäten](#).

## Abonnement-Workflow, Tutorial 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](#)
- [Definieren von „GetContactActivity“](#)
- [Definieren von „SubscribeTopicActivity“](#)
- [Definieren von „WaitForConfirmationActivity“](#)
- [Definieren von „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 teilweise die gleichen Funktionen haben, leisten wir zunächst ein wenig Vorarbeit und erstellen Code, den sie gemeinsam nutzen können. Wir nennen diesen Code `BasicActivity` und definieren ihn 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 Instance [AWS::SimpleWorkflow::ActivityType](#), 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 Klasse definiert sind `initialize` Methode, die eine Aktivität annimmt `name` und ein optionales `ausführung` und Karte von `Optionen` die bei der Registrierung der Aktivität bei Amazon SWF verwendet werden soll.

```
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
```

```
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
end
```

Dadurch wird die Klasse `BasicActivity` zusammengefasst. Wir können sie nun verwenden, um unsere Aktivitäten auf einfache und konsistente Weise zu definieren.

## Definieren von „GetContactActivity“

Die erste Aktivität, die bei einer Workflow-Ausführung ausgeführt wird, ist `get_contact_activity`, das die Amazon-SNS-Themenabonnements-Informationen des Benutzers abrufen.

Erstellen Sie eine neue Datei mit dem Namen `get_contact_activity.rb`, und benötigen beides `yaml`, die wir verwenden, um eine Zeichenfolge für die Weitergabe an Amazon SWF vorzubereiten, und `basic_activity.rb`, die wir als Grundlage dafür verwenden werden `GetContactActivity`-Klasse.

```
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 Code für die Registrierung der Aktivität in `BasicActivity` definiert haben, ist die `initialize`-Methode für `GetContactActivity` recht 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 nun die `do_activity`-Methode, die zur Eingabe der E-Mail-Adresse und/oder 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 an Amazon SWF übergeben, wenn Sie eine Aktivität abschließen, müssen nur Zeichenfolgendaten ausgeben. 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.

## Definieren von „SubscribeTopicActivity“

Wir beschäftigen uns nun mit Amazon SNS und erstellen eine Aktivität, 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 erstellt haben, fügen wir etwas Code hinzu, um ein Amazon SNS SNS-Thema zu erstellen. Hierfür benutzen wir die [AWS::SNS::-Client-Objekte](#) `create_topic`-Methode.

Hinzufügen der `create_topic`-Methode zu Ihrer -Klasse, die ein übergebendes 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!
    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
  end
end
```

```
    return nil
  end

  return topic_arn
end
```

Sobald wir den Amazon-Ressourcennamen (ARN) des Themas haben, können wir ihn mit den Amazon SNS Themas verwenden [set\\_topic\\_attributes](#) Methode zum Festlegen des `TopicDisplayName`, das 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.

Davon ausgehend, dass alles okay ist, füllen wir weiter `unseredo_activity`, holen Sie sich einen Amazon SNS SNS-Client mit dem AWS SDK for Ruby, und geben Sie es an `unsererecreate_topic`-Methode zum Erstellen des Amazon-SNS-Themas.

```
# 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 benutzen [AWS.config.withum](#) 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 hat diese Aktivität den Benutzer für das Amazon-SNS-Thema unter Verwendung der übergebenen Endpunkte (E-Mail und SMS) für den Amazon SNS-Thema abonniert. 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
```

[AWS::SNS::Client.subscribe](#) verwendet den ARN des Themas, das Protokoll (das für den entsprechenden Endpunkt als `activity_data`-Zuweisungsschlüssel dargestellt ist).

Schließlich verpacken wir die Informationen erneut im YAML-Format, damit 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`.

## Definieren von „WaitForConfirmationActivity“

Sobald ein Benutzer für ein Amazon SNS SNS-Thema abonniert wurde, muss er die Abonnementanfrage weiterhin 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)
```

Nun, da wir über den ARN des Themas verfügen, können wir das Thema abrufen, indem wir eine neue Instance von [AWS::SNS::Topic](#) erstellen und sie an 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-Thema pflegt eine Liste der [Abonnements](#). Für dieses Thema können wir prüfen, ob der Benutzer ein Abonnement bestätigt hat oder nicht, indem wir prüfen, ob der ARN des Abonnements auf etwas anderes festgelegt ist als `PendingConfirmation`.

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

```
        @results = subscription_data.to_yaml
        return true
      else
        puts "Topic subscription still pending for (#{sub.protocol}:
#{sub.endpoint})"
      end
    end
  end
end
```

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 es eine Weile dauern kann, bis ein Abonnement bestätigt wird, rufen wir gelegentlich `record_heartbeat` für die Aktivitätsaufgabe auf. Dies zeigt Amazon SWF an, dass die Aktivität noch verarbeitet wird. Es kann auch verwendet werden, um Aktualisierungen zum Fortschritt der Aktivität zu vermelden (wenn Sie etwas tun, z. B. Dateien verarbeiten, für das ein Fortschritt angezeigt werden kann).

```
    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`.

## Definieren von „SendResultActivity“

Wenn der Workflow bis hierhin gelangt ist, haben wir den Benutzer erfolgreich für ein Amazon SNS SNS-Thema abonniert und er 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 ähnlich, das heißt, die Eingabedaten werden vom Workflow abgerufen, von YAML konvertiert und dann wird mithilfe des ARN des Themas eine [AWS::SNS::Topic](#)-Instance erstellt.

```
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
```

Durch Veröffentlichung in einem Amazon SNS SNS-Thema wird die Nachricht gesendet, die Sie an alle abonnierten und bestätigten Endpunkte, 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 Aktivitätsaufgaben](#)).

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

In Amazon SWF werden Aktivitätsaufgaben für eine laufende Workflow-Ausführung auf der Aktivitätsaufgabenliste, die bereitgestellt wird, sobald Sie eine Aktivität im Workflow planen.

Wir werden einen einfachen Aktivitäten-Poller implementieren, der diese Aufgaben für unseren Workflow handhabt und mit dem unsere Aktivitäten gestartet werden, sobald Amazon SWF eine Aufgabe auf die Aktivitätsaufgabenliste setzt.

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

Zusätzlich zum Speichern der vergangenen Aktivitäten in der Domäne und Aufgabenliste, instanziiert dieser Code alle der von uns erstellten Aktivitätsklassen. Weil jede Klasse ihre zugehörige Aktivität registriert (siehe `basic_activity.rb` Wenn 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](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
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 vom Poller nach dem Start der letzten Aktivität beendet, da seine Aufgabe abgeschlossen ist und alle Aktivitäten gestartet wurden. Wenn in Ihrem eigenen Amazon SWF SWF-Code Ihre Aktivitäten erneut ausgeführt werden, sollten Sie den Aktivitäten-Poller möglicherweise unbegrenzt laufen lassen.

Damit ist der Code für unsere `ActivitiesPoller`-Klasse fast abgeschlossen. Wir werden noch ein wenig mehr Code am Ende der Datei hinzufügen, um dem Benutzer die Möglichkeit zu bieten, sie von der Befehlszeile aus auszuführen.

```
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 Tutorial: Ausführen des Workflows](#) sehen, wie er funktioniert.

## Abonnement-Workflow Tutorial: 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 AWS Zugriffsschlüssel im `aws-config.txt` Dateifeld, wie in [Konfigurieren von AWS-Sitzung](#) in Teil 1 des Tutorials.

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
```

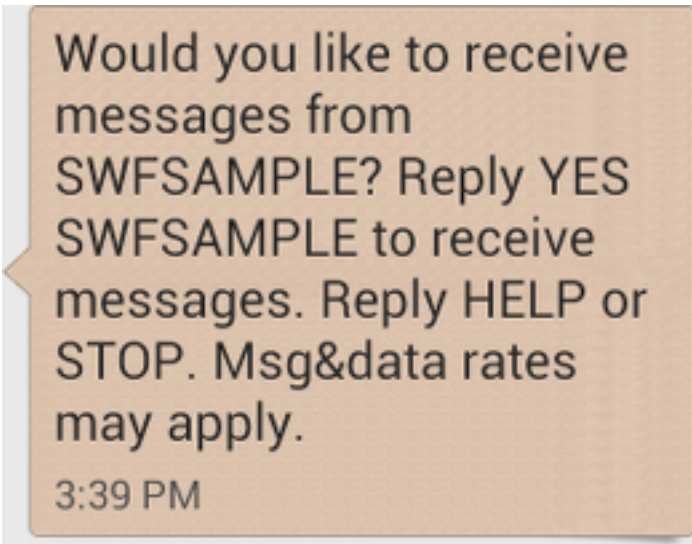
```
  phone: 12065550101
```

```
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!

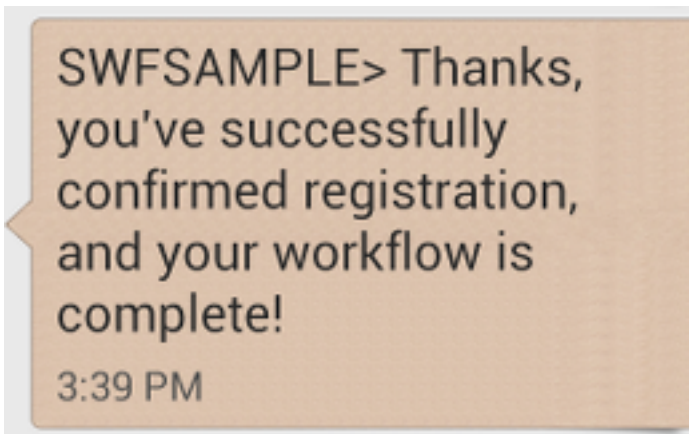
Kurz nach der Eingabe dieser Informationen sollten Sie eine E-Mail oder SMS von Amazon SNS erhalten, in der Sie gebeten werden, Ihr Themenabonnement zu bestätigen. Wenn Sie eine SMS-Telefonnummer eingegeben haben, wird eine Meldung ähnlich der folgenden auf Ihrem Mobiltelefon angezeigt.



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. Das erneute Ausführen des Workflows

vor dem Beenden des Abonnements der Themen wird wahrscheinlich automatisch erfolgreich sein, da `wait_for_confirmation_activity` erkennt, dass Ihr Abonnement bereits bestätigt ist.

So melden Sie sich von dem 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?

Dieses Tutorial hat eine Menge abgedeckt, aber es gibt noch viel mehr, das Sie über AWS SDK for Ruby, Amazon SWF oder Amazon SNS. Weitere Informationen und viele weitere Beispiele finden Sie in der jeweiligen offiziellen Dokumentation:

- [AWS SDK for Ruby-Dokumentation](#):
- [Amazon Simple Notification Service-Dokumentation](#)
- [Amazon Simple Workflow Service-Dokumentation](#)

# Tutorials

Die Amazon-Simple-Workflow-Service-Konsole (Amazon-SWF) bietet eine alternative Möglichkeit, Workflowausführungen zu konfigurieren, zu initiieren und zu verwalten.

Mit der Amazon SWF SWF-Konsole können Sie:

- Workflow-Domänen registrieren.
- Workflow-Typen registrieren.
- Aktivitätstypen registrieren.
- Workflow-Ausführungen initiieren.
- Informationen zu ausstehenden Aufgaben anzeigen.
- Laufende Workflow-Ausführungen anzeigen.
- Signale für laufende Workflow-Ausführungen abbrechen, beenden und senden.
- Geschlossene Workflow-Ausführungen neu starten.

Die Amazon-SWF-Konsole ist Teil der größeren -KonsoleAWS Management ConsoleErfahrung, auf die Sie zugreifen können, indem Sie sich unter<https://aws.amazon.com/>. Sie finden den Anmeldelink in der oberen rechten Ecke der Seite.

## Themen

- [Registrierung einer Amazon SWF-Domain](#)
- [Registrieren eines Workflow-Typs](#)
- [Registrieren eines Aktivitätstyps](#)
- [Starten einer Workflow-Ausführung](#)
- [Anzeigen ausstehender Aufgaben](#)
- [Verwalten von Workflow-Ausführungen](#)
- [Verwalten von Amazon SWF-Domänen](#)

## Registrierung einer Amazon SWF-Domain

Bis mindestens eine Domäne registriert ist, ist die Domänenregistrierung die einzige in der Konsole verfügbare Funktion.

## New console

So registrieren Sie eine Amazon SWF-Domain mit der Konsole

1. Wenn keine Domains registriert wurden, wählen Sie in der Mitte des Hauptfensters die Option Domain registrieren aus.
2. Geben Sie auf der Seite „Domain registrieren“ einen Namen, eine Aufbewahrungsdauer und optional eine Beschreibung für die Domain ein.

Diese Werte entsprechen den ähnlich benannten Parametern in der Aktion Register Domain.

### Domain details

**Name**

Up to 256 characters. Must not contain a : (colon), / (slash), | (vertical bar), or any control characters.


**Retention period**

Duration for which the record of workflow executions should be kept. Executions will not be available after the specified retention period elapses.

 days  
Must be 0-90 days.

**Description - optional**

Up to 1024 characters.

► **Tags - optional** [Learn more](#) 

**Cancel** **Register domain**

3. Wählen Sie Domain registrieren aus.
4. Nachdem die Domain registriert wurde, zeigt die Konsole eine Seite mit dem Titel Ihres Domainnamens an. Hier sind folgende Aktionen möglich:
  - [Registrieren Sie einen Workflow in Ihrer Domain](#)

- [Registrierte eine Aktivität in deiner Domain](#)
- [Starten Sie die Ausführung eines Workflows](#)
- [Managen Sie Ihre Workflow-Ausführungen](#)
- [Die Domain als veraltet oder als veraltet markieren](#)

## Old console

So registrieren Sie eine Amazon SWF-Domain mit der Konsole

1. Wenn keine Domains registriert wurden, wählen Sie in der Mitte des Hauptfensters die Option Domain registrieren aus.

Wenn mindestens eine Domäne registriert wurde, wählen Sie in der Dashboard-Ansicht die Schaltfläche Manage Domains und dann im Dialogfeld Manage Domains die Option Register New.

2. Geben Sie im Dialogfeld Register Domain einen Namen, einen Aufbewahrungszeitraum und eine Beschreibung ein. Diese Werte entsprechen den ähnlich benannten Parametern in der Aktion RegisterDomain.

**Register Domain** Cancel X

Provide the details of your new Domain below, then click Register

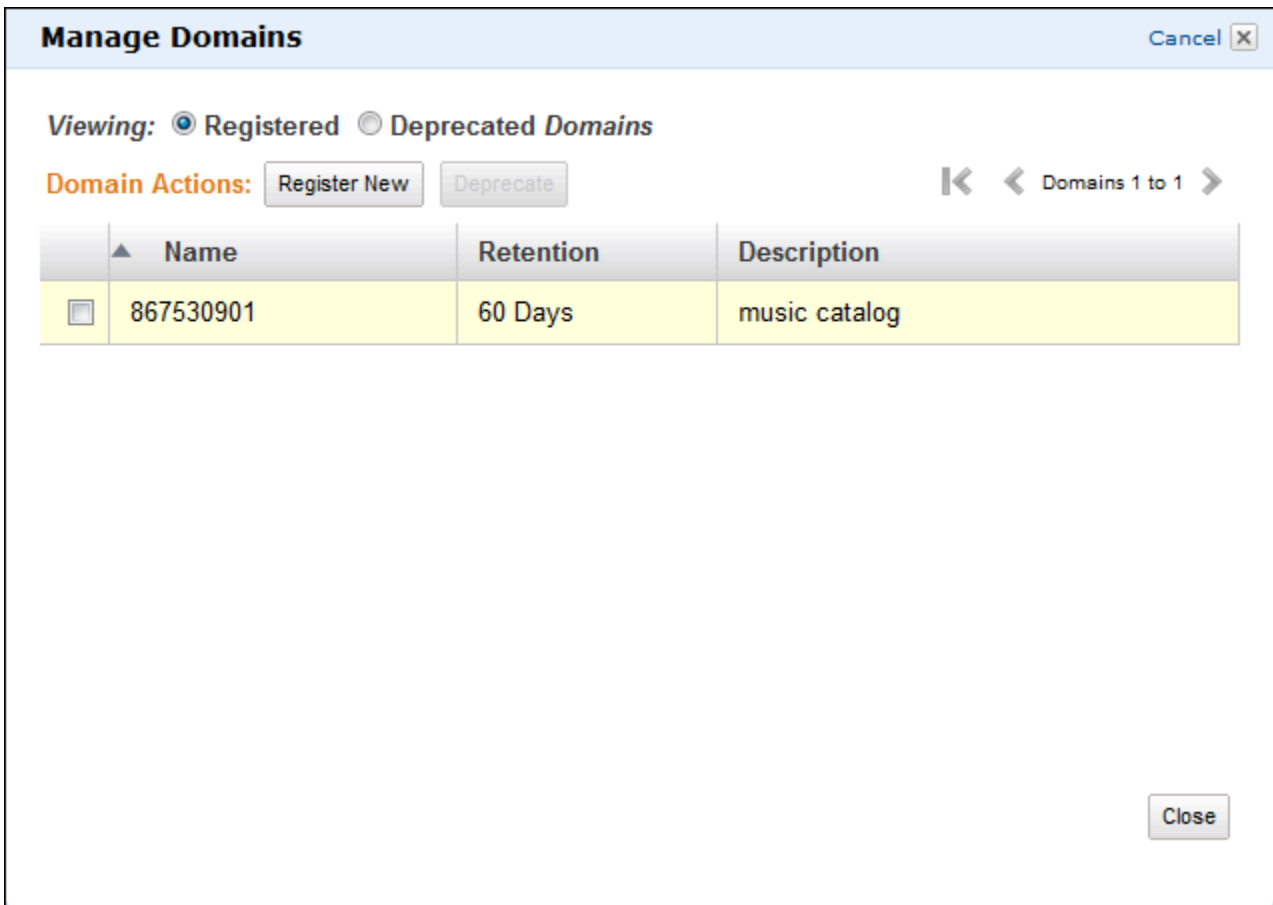
**Name:\***

**Workflow Execution Retention Period:\***  Days

**Description:**

3. Wählen Sie Register aus.

4. Nachdem die Domäne registriert wurde, zeigt die Konsole das Dialogfeld **Manage Domains** an.



## Registrieren eines Workflow-Typs

Sie können Workflowtypen mithilfe der Amazon SWF SWF-Konsole registrieren. Sie können einen Workflow erst registrieren, wenn mindestens eine Domain registriert ist.

### New console

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

1. Öffnen Sie die Domain, in der Sie einen Workflow registrieren möchten.
2. Wählen Sie **Registrieren** und dann **Workflow** zur Registrierung.
3. Auf der **Workflow zur Registrierung** Seite, geben Sie das **Workflow-Name** und **Workflow-Version**. Optional können Sie auch eine [Standard-Aufgabenliste](#) die verwendet wird, um Entscheidungsaufgaben für die Ausführung dieses Workflows zu planen.



## Register workflow [Learn more](#)

A workflow is a set of activities that carry out an objective and contains the logic that coordinates the activities. Workflows coordinate and manage the execution of activities that can be run asynchronously across multiple computing devices and feature both sequential and parallel processing methods.

### Workflow details

#### Domain

#### Workflow name

Up to 256 characters. Must not contain a : (colon), / (slash), | (vertical bar), or any control characters.

#### Workflow version

Up to 64 characters. Must not contain a : (colon), / (slash), | (vertical bar), or any control characters.

#### Description - *optional*

Up to 1024 characters.

#### Default task list - *optional* [Learn more](#)

Used for decision tasks generated for executions of this workflow.


Up to 256 characters. Must not contain a : (colon), / (slash), | (vertical bar), or any control characters.

#### 4. (Optional) Wählen Erweiterte Optionen um die folgenden Details für Ihren Workflow anzugeben:


- [Standardpriorität der Aufgaben](#)— Die standardmäßige Aufgabenpriorität, die dem Workflow zugewiesen werden soll.
- [Standardpriorität der Aufgaben](#)— Die standardmäßige Aufgabenpriorität, die dem Workflow zugewiesen werden soll.
- [Standardzeitlimit für Start bis Schließen der Ausführung](#)— Die standardmäßige Höchstdauer für die Ausführung dieses Workflows.
- [Standardzeitlimit für Start und Schließen der Aufgabe](#)— Die standardmäßige maximale Dauer von Entscheidungsaufgaben für diesen Workflow.

- [Standard-Richtlinie für Kinder](#)— Die Standardrichtlinie, die für die Ausführung des untergeordneten Workflows verwendet wird.
- [Lambda-Standardrolle](#)— Die standardmäßige IAM-Rolle, die diesem Workflow zugeordnet ist.


▼ **Advanced options**

**Default Task priority** [Learn more](#)   
Specify the priority to schedule decision tasks generated for this execution. A higher number indicates higher priority. Default value is 0.


Must be from -2147483648 to 2147483647

**Default Execution start to close timeout** [Learn more](#)   
Executions of this workflow will be timed out if not completed within this duration. Default value is 3600 seconds

  ▼  
Up to 8 characters. Use NONE for unlimited duration.

**Default Task start to close timeout** [Learn more](#)   
Decision task will be timed out if not completed within this duration. Default value is 600 seconds

  ▼  
Up to 8 characters. Use NONE for unlimited duration.

**Default Child policy** [Learn more](#)   
Determines a policy for child workflow executions if this execution is terminated.

Terminate  
 Request cancel  
 Abandon

**Default Lambda role - optional**  
It gives Amazon SWF the ability to invoke any Lambda function associated with your account in place of, or alongside traditional activities.

 ▼

**Cancel** **Register workflow**

## 5. Wählen Workflow zur Registrierung.

## Old console

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

1. In der Amazon Simple Workflow-Service Dashboard, unter Schnelllinks, wählen Neuen Workflow-Typ registrieren.

Geben Sie in das Dialogfeld Workflow Details die folgenden Informationen ein.

- Bereich
- Workflow-Name
- Workflow-Version
- Standardaufgabenliste
- Standard-Laufzeit der Ausführung
- Standard-Laufzeit der Aufgabe

Mit einem Sternchen ((\*)) markierte Felder sind Pflichtangaben.

**Register New Workflow** Cancel

**WORKFLOW DETAILS**    **ADDITIONAL OPTIONS**    **REVIEW**

Provide the details of your new **Workflow** below:

**Domain\***: 867530901

**Workflow Name\***: customerOrderWorkflow

**Workflow Version\***: 1.0

**Default Task List\***: mainTaskList

**Default Execution Run Time\***: 3600 seconds

**Default Task Run Time\***: 600 seconds

Continue

Klicken Sie auf Continue (Weiter).

2. Geben Sie im Dialogfeld Additional Options eine Beschreibung ein und geben Sie eine Standardmäßige untergeordnete Richtlinie an. Wählen Sie Review.

**Register New Workflow** Cancel

WORKFLOW DETAILS    **ADDITIONAL OPTIONS**    REVIEW

Provide additional options for your new **Workflow** below:

**Description:**

**Default Child Policy:**

[Back](#) Review

- Überprüfen Sie im Dialogfeld Review die Daten, die Sie in den vorherigen Dialogfeldern eingegeben haben. Wenn alle Informationen richtig sind, wählen Sie Register Workflow aus. Klicken Sie andernfalls auf Back, um Änderungen an den Daten vorzunehmen.

## Register New Workflow Cancel X

WORKFLOW DETAILS    ADDITIONAL OPTIONS    REVIEW

Please review the information below, then click **Register Workflow**

**Domain:** 867530901

**Workflow Name:** customerOrderWorkflow

**Workflow Version:** 1.0

**Default Task List:** mainTaskList

**Default Child Policy:** TERMINATE

**Default Execution Run Time:** 1 hour

**Default Task Run Time:** 10 minutes

**Description:**

[Back](#) Register Workflow

4. Wählen Workflow zur Registrierung.

## Registrieren eines Aktivitätstyps

Sie können Aktivitätstypen mit der Amazon SWF SWF-Konsole registrieren. Sie können eine Aktivität erst registrieren, wenn mindestens eine Domain registriert ist.

### New console

1. Öffnen Sie die Domain, in der Sie eine Aktivität registrieren möchten.
2. Wählen Registrieren und dann wähle Registrieren Sie Aktivität.
3. Auf der Registrieren Sie Aktivität geben Sie die Seite ein [Name der Aktivität](#) und [Version der Aktivität](#). Optional können Sie auch eine [Liste der Standardaufgaben](#) die verwendet wird, um Aufgaben dieser Aktivität zu planen.

## Register activity [Learn more](#)

Activities are tasks which you want your workflow to coordinate and execute (for example: verify customer's order, charge credit card etc.). The order in which activities are performed is determined by the workflow's coordination logic.

### Activity details

#### Domain

#### Activity name

Up to 256 characters. Must not contain a : (colon), / (slash), | (vertical bar), or any control characters.

#### Activity version

Up to 64 characters. Must not contain a : (colon), / (slash), | (vertical bar), or any control characters.

#### Description - *optional*

*Enter description for this activity*

Up to 1024 characters.

#### Default task list - *optional* [Learn more](#)

Used for tasks of this activity


Up to 256 characters. Must not contain a : (colon), / (slash), | (vertical bar), or any control characters.

4. (Optional) Wählen Sie Erweiterte Optionenum die folgenden Details für Ihre Aktivität anzugeben:

- [Standardpriorität für Aufgaben](#)— Die standardmäßige Aufgabenpriorität, die der Aktivität zugewiesen werden soll.
- [Standard-Task-Zeitplan zum Starten des Timeouts](#)— Die standardmäßige maximale Dauer, die eine Aufgabe dieser Aktivität warten kann, bevor sie einer Arbeitskraft zugewiesen wird.
- [Standardzeitlimit für Start und Schließen der Aufgabe](#)— Die standardmäßige maximale Dauer, die ein Mitarbeiter zur Bearbeitung von Aufgaben dieser Aktivität benötigen kann.
- [Standard-Task-Zeitplan zum Schließen des Timeouts](#)— Die standardmäßige Höchstdauer für eine Aufgabe dieser Aktivität.


- [Standardtimeout für Task-Heartbeat](#)— Die standardmäßige maximale Zeit, vor der ein Worker, der eine Aufgabe dieses Typs verarbeitet, den Fortschritt durch Aufrufen melden muss [RecordActivityTaskHeartbeat](#).

**▼ Advanced options**

**Default Task priority** [Learn more](#) 

Specify the priority to schedule activity tasks generated for this execution. A higher number indicates higher priority. Default value is 0.


Must be from -2147483648 to 2147483647

**Default task schedule to start timeout** [Learn more](#) 

Tasks of this activity will be timed out if not started within this duration. Default value is 1800 seconds.

  ▼
 


Up to 8 characters. Use NONE for unlimited duration.

**Default Task start to close timeout** [Learn more](#) 

Tasks of this activity will be timed out if not completed within this duration. Default value is 600 seconds.

  ▼
 


Up to 8 characters. Use NONE for unlimited duration.

**Default task schedule to close timeout** [Learn more](#) 

Tasks of this activity will be timed out if not started and completed within this duration. Default value is 5400 seconds.

  ▼
 

Up to 8 characters. Use NONE for unlimited duration.

**Default task heartbeat timeout** [Learn more](#) 

Maximum time before which a worker processing a task must report progress. Default value is 1800 seconds.

  ▼
 

Up to 8 characters. Use NONE for unlimited duration.

## 5. Wählen/Registrieren Sie Aktivität.

### Old console

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

1. In der Amazon Simple Workflow Service Dashboard, unter Schnelllinks, wählen Neuen Aktivitätstyp registrieren.

Geben Sie in das Dialogfeld Activity Details die folgenden Informationen ein.

- Bereich
- Name der Aktivität
- Version der Aktivität
- Standardaufgabenliste
- Zeitüberschreitung von der Aufgabenplanung bis zum Starten
- Zeitüberschreitung vom Starten bis zum Schließen

Mit einem Sternchen ((\*)) markierte Felder sind Pflichtangaben.

The screenshot shows a dialog box titled "Register New Activity" with a "Cancel" button in the top right corner. The dialog has three tabs: "ACTIVITY DETAILS" (selected), "ADDITIONAL OPTIONS", and "REVIEW". Below the tabs, there is a progress indicator and the text "Provide the details of your new Activity below:". The form contains the following fields:

- Domain\*:** 867530901
- Activity Name\*:** activityVerify
- Activity Version\*:** 1.0
- Task List:** mainTaskList
- Task Schedule to Start Timeout:** 5 minutes
- Task Start to Close Timeout:** 15 minutes

A "Continue" button is located at the bottom right of the dialog.

Klicken Sie auf Continue (Weiter).

2. Geben Sie im Dialogfeld Additional Options eine Description ein und legen Sie einen Wert für Heartbeat Timeout und Task Schedule to Close Timeout fest. Wählen Sie Review.



## Register New Activity Cancel X

ACTIVITY DETAILS **ADDITIONAL OPTIONS** REVIEW

Provide additional options for your new Activity below:

**Description:**

**Heartbeat Timeout:**

**Task Schedule to Close Timeout:**

[Back](#) Review

- Überprüfen Sie im Dialogfeld Review die Daten, die Sie in den vorherigen Dialogfeldern eingegeben haben. Wenn alle Informationen richtig sind, wählen Sie Register Activity aus. Klicken Sie andernfalls auf Back, um Änderungen an den Daten vorzunehmen.

## Register New Activity Cancel

ACTIVITY DETAILS    ADDITIONAL OPTIONS    **REVIEW**

Please review the information below, and then click **Register**

---

**Domain:** 867530901  
**Activity Name:** activityVerify  
**Activity Version:** 1.0  
**Task List:** mainTaskList  
**Task Schedule to Close Timeout:** 15 minutes  
**Task Schedule to Start Timeout:** 5 minutes  
**Task Start to Close Timeout:** 10 minutes  
**Task Heartbeat Timeout:** 2 minutes  
**Description:**

---

[← Back](#)

4. Wählen Sie die Aktivität registrieren Sie.

## Starten einer Workflow-Ausführung

Sie können eine Workflow-Ausführung über die Amazon SWF Konsole starten. Sie können eine Workflow-Ausführung erst starten, wenn Sie mindestens einen Workflow registriert haben.

### So starten Sie eine Workflow-Ausführung mithilfe der Konsole


New console

1. Öffnen Sie die Amazon SWF-Konsole und wählen Sie im linken Navigationsbereich auf Amazon SWF Console (-Domänen).




- [Aufgabenpriorität](#)— Die Aufgabenpriorität, die für diese Workflow-Ausführung verwendet werden soll.
- [Timeout von Beginn bis Ende der Ausführung](#)— Die Gesamtdauer für diese Workflow-Ausführung.
- [Zeitlimit für Start und Schließen der Aufgabe](#)— Die maximale Dauer von Entscheidungsaufgaben für diese Workflow-Ausführung.
- [Politik für Kinder](#)— Die Richtlinie, die für die Ausführung des untergeordneten Workflows dieser Workflowausführung verwendet werden soll, wenn sie beendet wird, indem die [TerminateWorkflowExecution](#)Aktion explizit oder aufgrund eines abgelaufenen Timeouts.
- [Lambda-Rolle](#)— Die IAM-Rolle, die an diese Workflow-Ausführung angehängt werden soll.

▼ **Advanced options**

**Task priority** [Learn more](#) 

Specify the priority to schedule decision tasks generated for this execution. A higher number indicates higher priority.


  
Must be from -2147483648 to 2147483647

**Execution start to close timeout** [Learn more](#) 

Executions of this workflow will be timed out if not completed within this duration.


Up to 8 characters. Use NONE for unlimited duration.

**Task start to close timeout** [Learn more](#) 

Decision task will be timed out if not completed within this duration.

Up to 8 characters. Use NONE for unlimited duration.

**Child policy** [Learn more](#) 

Determines a policy for child workflow executions if this execution is terminated.

Terminate  
 Request cancel  
 Abandon

**Lambda role - optional**  
It gives Amazon SWF the ability to invoke any Lambda function associated with your account in place of, or alongside traditional activities.

7. Wählen Sie Start execution (Ausführung starten) aus.

## Old console

1. In der Amazon Simple Workflow Service Dashboard, unter Schnelllinks, wählen Starten einer neuen Workflow-Ausführung.

Geben Sie in das Dialogfeld Execution Details die folgenden Informationen ein.

- Bereich
- Workflow-Name
- Workflow-Version

- Workflow-ID
- Aufgabenliste
- Maximale Ausführungszeit
- Zeitüberschreitung vom Starten bis zum Schließen

Mit einem Sternchen ((\*)) markierte Felder sind Pflichtangaben.

**Start New Execution** Cancel X

EXECUTION DETAILS    ADDITIONAL OPTIONS    REVIEW

Provide the details of your Execution below:

**Domain\*:** 867530901

**Workflow Name\*:** customerOrderWorkflow

**Workflow Version\*:** 1.0

**Workflow ID\*:** 20110927-T-1

**Task List:** specialTaskList

**Max. Execution Run Time:** 1800 seconds

**Task Start to Close Timeout:** 600 seconds

Continue

Klicken Sie auf Continue (Weiter).

2. Geben Sie im Dialogfeld Additional Options Folgendes an:
  - Eine Reihe von Tags für die Zuordnung zur Workflow-Ausführung. Sie können diese Tags verwenden, um Informationen über Ihre Workflow-Ausführungen abzufragen.
  - Unter Input eine Eingabezeichenfolge, die für die Ausführung bedeutsam ist. Diese Zeichenfolge wird von Amazon SWF nicht interpretiert.
  - Unter Child Policy eine untergeordnete Richtlinie.

## Start New Execution Cancel X

EXECUTION DETAILS **ADDITIONAL OPTIONS** REVIEW

Provide additional options for your Execution below:

**Tags:** music purchase, digital, ricoh-the-dog

**Input:** arbitrary-string-that-is-meaningful-to-the-workflow

**Child Policy:** Terminate

[Back](#) Review

- Überprüfen Sie im Dialogfeld Review die Daten, die Sie in den vorherigen Dialogfeldern eingegeben haben. Wenn alle Informationen richtig sind, wählen Sie Start Execution aus. Klicken Sie andernfalls auf Back, um Änderungen an den Daten vorzunehmen.

## Start New Execution Cancel

EXECUTION DETAILS    ADDITIONAL OPTIONS    **REVIEW**

Please review the information below, then click **Start**

**Domain:** 867530901

**Workflow Name:** customerOrderWorkflow

**Workflow Version:** 1.0

**Workflow ID:** 20110927-T-1

**Max. Execution Run Time:** 30 minutes

**Task Start to Close Timeout:** 10 minutes

**Task List:** specialTaskList

**Child Policy:** TERMINATE

**Tags:**

**Input:**

[Back](#)

## Anzeigen ausstehender Aufgaben

### Note

Diese Aufgabe ist nur in der alten Amazon SWF SWF-Konsole verfügbar.

Im Amazon Simple Workflow Service Dashboard können Sie die Anzahl der ausstehenden Aufgaben anzeigen, die mit einer bestimmten Aufgabenliste verknüpft sind.



1. Wählen Sie aus, ob die Aufgabenliste eine Decider Task List oder Activity Task List sein soll.
2. Geben Sie den Namen der Aufgabenliste in das Textfeld ein.
3. Wählen Sie View Backlog aus.

Enter the Task List Name:

Decider Task List  Activity Task List

*Task List "specialTaskList" has a backlog of 3 tasks.*

## Verwalten von Workflow-Ausführungen

In der Amazon SWF SWF-Konsole können Sie die Workflow-Ausführungen verwalten, die derzeit ausgeführt werden und/oder geschlossen sind.

### New console

So verwalten Sie Ihre Workflow-Ausführungen

1. Öffnen Sie eine Domain, um ihre Workflow-Ausführungen zu verwalten.
2. Wählen Sie nach Ausführung.
3. Auf der Ausführung von Workflow-Ausführungen Seite, wählen Sie Filtern aus, um die Ausführungen nach Eigenschaft zu filtern und dann unter Eigenschaften wählen Sie einen der folgenden Filter aus:

Klicken Sie auf	So wenden Sie diesen Filter an
Workflow	<p>Wählen Sie diesen Filter, um die Ausführungen eines bestimmten Workflows aufzulisten. Zum Beispiel, um Ausführungen von <code>derfiction-books-order-workflow</code> , gehen Sie wie folgt vor:</p> <ol style="list-style-type: none"> <li>1. Wählen Sie Workflow.</li> <li>2. Unter Operatoren, wählen Sie Gleich.</li> <li>3. Unter Workflows, wählen Sie <code>fiction-books-order-workflow</code>.</li> </ol>

Klicken Sie auf	So wenden Sie diesen Filter an
	<p>4. (Optional) Wählen Sie Löschen des Filters um den Filter zu entfernen und eine neue Suche nach Ausführungen zu starten.</p>
Status	<p>Wählen Sie diesen Filter, um Ausführungen mit einem bestimmten Status aufzulisten. Zum Beispiel, um Ausführungen mit dem Status anzuzeigen Fehlgeschlagen, gehen Sie wie folgt vor:</p> <ol style="list-style-type: none"> <li>1. Wählen-Status.</li> <li>2. Unter Operatoren, wählen Gleich ch ch.</li> <li>3. Unter Status, wählen Fehlgeschlagen.</li> <li>4. (Optional) Wählen Sie Löschen des Filters um den Filter zu entfernen und eine neue Suche nach Ausführungen zu starten.</li> </ol>
Ausführungsfehler ID ID	<p>Wählen Sie diesen Filter, um eine Workflow-Ausführung basierend auf ihrer ID anzuzeigen. Um beispielsweise die Ausführung mit ID anzuzeigen fiction-books-order-category1 , gehen Sie wie folgt vor:</p> <ol style="list-style-type: none"> <li>1. Wählen Ausführungsfehler ID ID.</li> <li>2. Unter Operatoren, wählen Gleich ch ch.</li> <li>3. Unter Ausführungsoptionen IDs, wählen fiction-books-order-category1.</li> <li>4. (Optional) Wählen Sie Löschen des Filters um den Filter zu entfernen und eine neue Suche nach Ausführungen zu starten.</li> </ol>
Tag	<p>Wählen Sie diesen Filter, um Ausführungen mit einem bestimmten Tag aufzulisten. Zum Beispiel, um Ausführungen mit dem Status anzuzeigen purchaseOrder , gehen Sie wie folgt vor:</p> <ol style="list-style-type: none"> <li>1. Wählen Tag.</li> <li>2. Unter Operatoren, wählen Gleich ch ch.</li> <li>3. Unter Tag, wählen Bestellung.</li> <li>4. (Optional) Wählen Sie Löschen des Filters um den Filter zu entfernen und eine neue Suche nach Ausführungen zu starten.</li> </ol>

4. (Optional) Nachdem Sie den erforderlichen Filter angewendet haben, um Workflow-Ausführungen aufzulisten, können Sie die folgenden Vorgänge an einer Ausführung:
  - **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 **Signal**, und geben Sie dann die Daten im Feld **Ausführung des Signals** ein.
    3. Wählen **Signal**.

### Signal execution [Learn more](#) ✕

**Name**

Up to 256 characters

**Input - optional**  
Value to attach to the `WorkflowExecutionSignaled` event in the target execution's history.

Up to 32768 characters

**Execution ID (1)**

**Cancel** **Send signal**

- **Versuche es abubrechen**— Verwenden Sie diese Option, um zu versuchen, eine Workflow-Ausführung abubrechen. 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 abubrechen möchten.
  2. Wählen **Versuche es abubrechen**.

- **Beenden**— Verwenden Sie diese Option, um eine Workflow-Ausführung zu beenden. Beachten Sie, dass Workflow-Ausführungen besser abgebrochen als beendet werden sollten.
  1. Wählen Sie die Ausführung aus, die Sie beenden möchten.
  2. Für Politik für Kinder, stellen Sie sicher Beenden ist ausgewählt.
  3. (Optional) Geben Sie die Option Grund und Details für die Beendigung der Hinrichtung.
  4. Wählen Sie Beenden.

### Terminate execution [Learn more](#) ✕

**Child policy**  
Determines a policy for child workflow executions if this execution is terminated.

Terminate  
 Request cancel  
 Abandon

**Reason - optional**

Cancellation unsuccessful

Up to 256 characters

**Details - optional**

Attempted cancellation of the workflow execution was unsuccessful.

Up to 32768 characters

**Execution ID (1)**

• trial-fiction-1

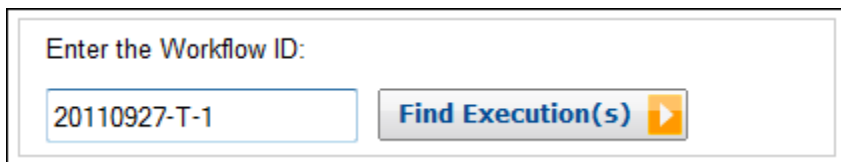
**Cancel** **Terminate**

5. (Optional) Führen Sie erneut starten— Verwenden Sie diese Option, um eine geschlossene 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 Ausführung erneut gestartet. Wählen Sie die Ausführung erneut aus.
2. Auf der Ausführung erneut starten geben Sie die Details für die Workflow-Ausführung an, wie unter [Starten einer Workflow-Ausführung](#).

## Old console

Um auf die zuzugreifen Ausführungen meines Workflows ansehen, wählen Sie die Ausführung (s) im Amazon SWF-Dashboard.



Wenn Sie zunächst eine Workflow-ID eingeben, werden in der Konsole die Ausführungen mit dieser Workflow-ID angezeigt. Wenn Sie stattdessen auf Find Execution(s) klicken, können Sie in der Ansicht My Workflow Executions Workflow-Ausführungen anhand des Startzeitpunkts, des Ausführungsstatus und der zugeordneten Metadaten suchen. Bei der Abfrage können Sie dabei beliebige der folgenden Metadattentypen auswählen:

- Workflow-ID
- Workflow-Typ
- Tags (Markierungen)
- Schließstatus

Wenn die Workflow-Ausführung geschlossen wurde, enthält der Schließstatus einen der folgenden Werte, die Aufschluss darüber geben, unter welchen Umständen die Workflow-Ausführung geschlossen wurde:

- Completed
- Fehlgeschlagen
- Canceled
- Timed Out (Zeitüberschreitung)
- Continued as New

**Note**

Sie müssen in der Dropdown-Liste Domain zunächst eine Domäne auswählen, bevor Sie Workflow-Ausführungen auflisten können.

**My Workflow Executions**

Domain: 867530901 Manage Domains Refresh Help

Workflow Execution List Parameters

Filter by:    
 Workflow ID\*:    
 Workflow Type  
 Close Status  
 Tag  
 No Filter

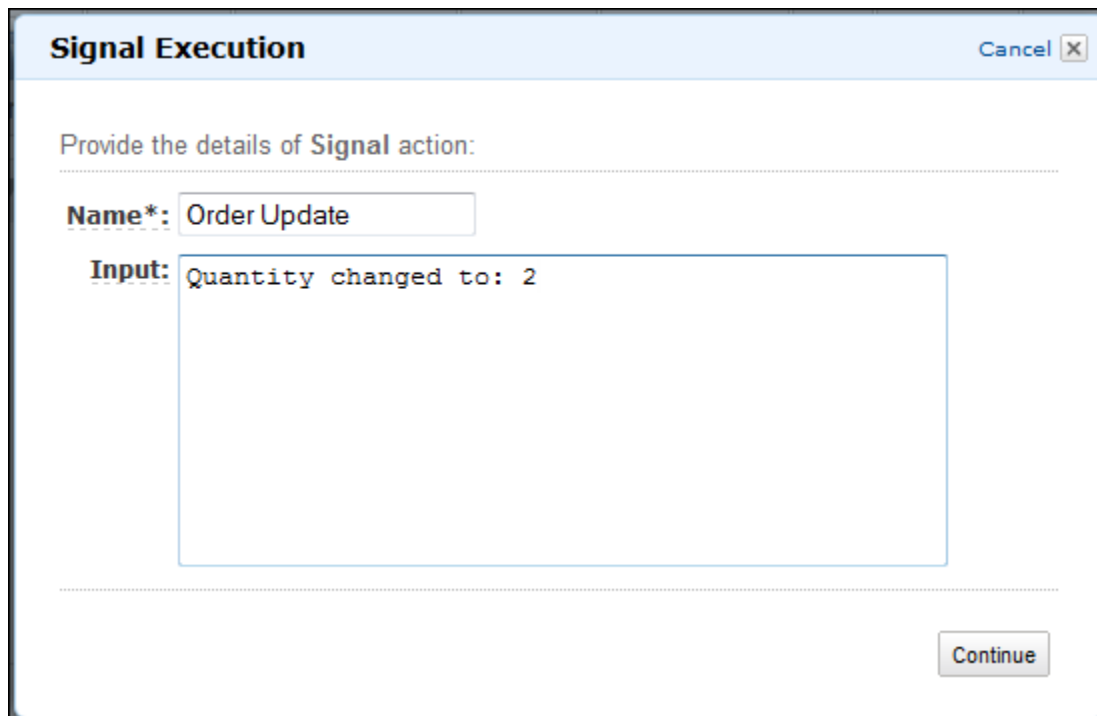
Execution Status: *List Active and Closed*  
 Started between  and

Execution Actions:     1 to 3 of 3 items

<input type="checkbox"/>	Workflow ID	Run ID	Name (Version)	Tags	Execution Status	Start Time	Close Time
<input checked="" type="checkbox"/>	20110927-T-1	817e8eb0-353b-47de-90b0-5f6754bf278a	customerOrderWorkflow (1.0)	ricoh-the-dog	Active	Sat Jan 21 21:25:20 GMT-800 2012	
<input type="checkbox"/>	20110927-T-1	c4f8b600-f3a7-4c6a-be2b-440f92b7afe0	customerOrderWorkflow (1.0)	music purchase,digital,ricoh-the-dog	Timed Out	Fri Dec 23 14:25:14 GMT-800 2011	Fri Dec 23 14:55:14 GMT-800 2011
<input type="checkbox"/>	20110927-T-1	6c585d49-82ca-4b3e-adcb-852768dabfcd	customerOrderWorkflow (1.0)	music purchase,digital,ricoh-the-dog	Timed Out	Tue Dec 20 22:13:21 GMT-800 2011	Tue Dec 20 22:43:21 GMT-800 2011

Nachdem Sie eine Liste der Workflow-Ausführungen angezeigt haben, können Sie folgende Operationen ausführen.

- Signalisieren Sie eine Workflow-Ausführung, d. h. senden Sie einer laufenden Workflow-Ausführung zusätzliche Daten.



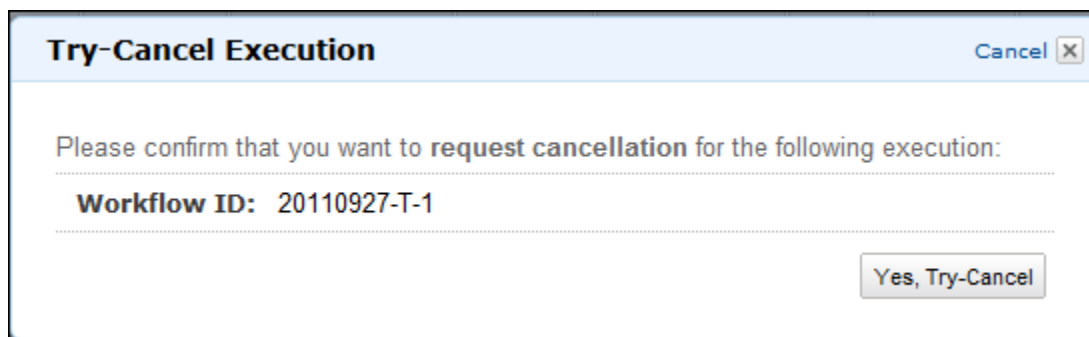
**Signal Execution** Cancel

Provide the details of Signal action:

**Name\*:**

**Input:**

- Versuchen, eine Workflow-Ausführung abubrechen. 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.



**Try-Cancel Execution** Cancel

Please confirm that you want to **request cancellation** for the following execution:

**Workflow ID:**

- Beenden einer Workflow-Ausführung. Beachten Sie, dass Workflow-Ausführungen besser abgebrochen als beendet werden sollten.

### Terminate Execution Cancel

Provide the details of Terminate action:

---

**Child Policy:**

**Reason:**

**Details:**

---

- Eine geschlossene Workflow-Ausführung neu starten

**Navigation**

- > Dashboard
- > Workflow Executions
- > Workflow Types
- > Activity Types
- > Workflow Executions
- > Workflow Types
- > Activity Types

**My Workflow Executions**

Domain:  Manage Domains Refresh Help

**Workflow Execution List Parameters**

Filter by:

Tag\*:

Execution Status:  Active  Closed

Started between  and

**Execution Actions:**     1 to 1 of 1 items

<input checked="" type="checkbox"/>	Workflow ID	Run ID	Name (Version)	Tags	Execution Sta	Start Time	Close Time
<input checked="" type="checkbox"/>	20110927-T-1	6c585d49-82ca-4b3e-adcb-852768dabfcd	customerOrderWorkflow (1.0)	music purchase,digital,ricoh-the-dog	Timed Out	Tue Dec 20 22:13:21 GMT-800 2011	Tue Dec 20 22:43:21 GMT-800 2011

So starten Sie eine geschlossene Workflow-Ausführung neu

1. Wählen Sie in der Liste der Workflow-Ausführungen die gewünschte Ausführung aus. Wenn Sie eine geschlossene Ausführung auswählen wird die Schaltfläche Re-Run angezeigt. Klicken Sie auf Re-Run.



Es wird eine Sequenz von Dialogfenstern für Re-Run Execution angezeigt.

2. Geben Sie im Dialogfeld Execution Details die folgenden Informationen ein. Im Dialogfeld sind die Daten der ursprünglichen Ausführung bereits eingetragen.
  - Bereich
  - Workflow-Name
  - Workflow-Version
  - Workflow-ID

Wenn Sie auf den Link Advanced Options klicken, können Sie die folgenden zusätzlichen Optionen festlegen.

- Aufgabenliste
- Maximale Ausführungszeit
- Zeitüberschreitung vom Starten bis zum Schließen

Klicken Sie auf Continue

3. Geben Sie im Dialogfeld Additional Options eine Eingabezeichenfolge für die Ausführung ein. Wenn Sie auf den Link Advanced Options klicken, können Sie Tags festlegen, die dieser Ausführung oder der Workflow-Ausführung zugeordnet werden. Außerdem können Sie die Child Policy der Ausführung ändern. Wie im vorherigen Dialogfeld sind die Daten der ursprünglichen Ausführung bereits eingetragen.

Wählen Sie Review.

4. Überprüfen Sie im Dialogfeld Review die angezeigten Informationen. Wenn alle Informationen richtig sind, klicken Sie auf Re-Run Execution. Klicken Sie andernfalls auf Back, um Änderungen an den Daten vorzunehmen.


## Verwalten von Amazon SWF-Domänen

Mithilfe der Amazon SWF SWF-Konsole können Sie Aktionen zur Domänenverwaltung ausführen, z. B. veraltete und nicht veraltete Domains. Nachdem Sie eine Domäne nicht mehr unterstützen, können Sie sie nicht mehr verwenden, um neue Workflow-Ausführungen zu erstellen oder neue Workflow-Ausführungen zu registrieren. Durch das Verwerfen einer Domain werden auch alle in der

Domain registrierten Aktivitäten und Workflows verworfen. Ausführungen, die gestartet wurden, bevor die Domain veraltet war, werden weiterhin ausgeführt.

Sie können auch eine zuvor veraltete Domäne nicht ab. Nachdem Sie eine Domain aufgehoben haben, können Sie sie verwenden, um [registriere neue Workflows in deiner Domain](#) oder [starte neue Workflow-Ausführungen](#).

Weitere Hinweise zu diesen Domänenverwaltungsaktionen finden Sie unter [DeprecateDomain](#) und [UndeprecateDomain](#).

 Note

Das folgende Amazon SWF-Domänenverwaltungsverfahren gilt für die neue Konsolenumgebung.

So lassen Sie sich eine Domäne mithilfe der Konsole nicht ab

1. Öffnen Sie die Amazon SWF-Domain, die Sie nicht mehr unterstützen möchten.
2. Wählen Domäne verwalten und dann wähle Veraltet.
3. Wählen Veraltet in der Veraltete Domäne Dialogfeld, um die Aktion zu bestätigen.

So lassen Sie eine Domäne mithilfe der Konsole nicht ab

1. Öffnen Sie die Amazon SWF-Domain, die Sie nicht mehr unterstützen möchten.
2. Wählen Domäne verwalten und dann wähle Unmissbilligt.

Alternativ können Sie auch wählen Nicht veraltete Domäne in der auf der Seite angezeigten Nachricht.

3. Wählen Unmissbilligt in der Nicht veraltete Domäne Dialogfeld, um die Aktion zu bestätigen.

# Funktionsweise von Amazon SWF

Die Konzepte in diesem Kapitel geben eine Übersicht über den Amazon Simple Workflow Service und beschreiben die wichtigsten Funktionen. Auch wenn in den Themen in diesem Kapitel einige Beispiele zur Verwendung von Amazon SWF beschrieben werden, finden Sie im Abschnitt mit dem Titel [Arbeiten mit Amazon SWF-APIs](#) für konkretere Beispiele für die Implementierung der hier beschriebenen Funktionen.

## Themen

- [Amazon SWF SWF-Workflows](#)
- [Amazon SWF SWF-Workflow-Verlauf](#)
- [Amazon SWF SWF-Akteure](#)
- [Amazon SWF SWF-Aufgaben](#)
- [Amazon SWF-Domäne](#)
- [Amazon SWF SWF-Objekt-IDs](#)
- [Amazon SWF SWF-Aufgabenlisten](#)
- [Beenden einer Beenden einer Workflow-Ausführung](#)
- [Lebenszyklus einer -Workflow-Ausführung](#)
- [Abfragen von Aufgaben in Amazon SWF](#)
- [Fortgeschrittene Konzepte in Amazon SWF](#)

## Amazon SWF SWF-Workflows

### Themen

- [Was ist ein Workflow?](#)
- [Ein einfaches Beispiel für einen Workflow: E-Commerce-Anwendung](#)
- [Registrierung und Ausführung des Workflows](#)
- [Weitere Informationen finden Sie unter:](#)

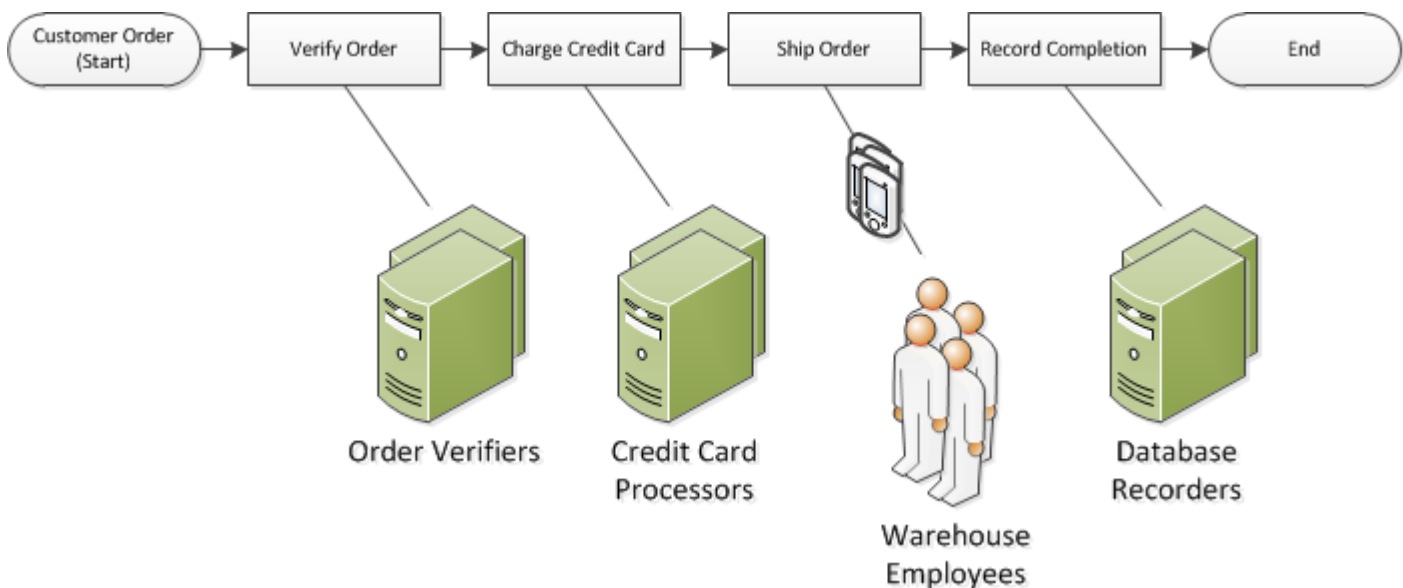
## Was ist ein Workflow?

Mit dem Amazon Simple Workflow Service (Amazon SWF) können Sie verteilte, asynchrone Anwendungen als Workflows ausführen. 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 dargestellt durch Aktivitäten. In welcher Reihenfolge die Aktivitäten ausgeführt werden, wird von der Koordinationslogik des Workflows bestimmt.

## Ein einfaches Beispiel für einen Workflow: E-Commerce-Anwendung

Die folgende Abbildung zeigt einen Beispielworkflow für die Auftragsabwicklung im E-Commerce, der menschliche und automatisierte Prozesse umfasst.



Dieser Workflow wird gestartet, wenn ein Kunde eine Bestellung aufgibt. Er 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 an verschiedenen Orten ausgeführt werden. Wenn es sich um programmgesteuerte Aufgaben handelt, können sie auch in unterschiedlichen Programmiersprachen oder mithilfe unterschiedlicher Tools geschrieben werden.

Amazon SWF unterstützt neben der sequenziellen Verarbeitung von Aufgaben auch Workflows, die Aufgaben parallel verarbeiten. 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.

## Registrierung und Ausführung des Workflows

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

Mit Amazon SWF können nur registrierte Workflow- und Aktivitätstypen verwendet werden. Bei dem E-Commerce-Beispiel würden Sie den Workflowtyp „CustomerOrder“ und die Aktivitätstypen „VerifyOrder“, „ChargeCreditCard“, „ShipOrder“ und „RecordCompletion“ registrieren.

Nachdem Sie den Workflowtyp registriert haben, können Sie ihn beliebig oft ausführen. Eine Workflow-Ausführung ist eine laufende Instance eines Workflows. Bei dem E-Commerce-Beispiel wird bei jeder Kundenbestellung eine neue Workflow-Ausführung gestartet.

Eine Workflow-Ausführung kann von beliebigen Prozessen oder Anwendungen, sogar von einer anderen Workflow-Ausführung, gestartet werden. Bei dem E-Commerce-Beispiel hängt der Anwendungstyp, der den Workflow startet, 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 einen Identifikator zuordnen — genannt `workflowId` — mit Ihren Workflow-Ausführungen, sodass Sie Ihre bestehenden IDs in den Workflow einbinden können. Bei dem E-Commerce-Beispiel lässt sich jede Workflow-Ausführung anhand der Rechnungsnummer identifizieren.

Zusätzlich zu der von Ihnen bereitgestellten Kennung verknüpft Amazon SWF einen eindeutigen vom System generierten Identifikator — `runId`— bei jeder Workflow-Ausführung. Amazon SWF erlaubt nur eine Workflow-Ausführung mit dieser Kennung. Es können aber mehrere Workflow-Ausführungen desselben Workflow-Typs gestartet werden, wobei jede Workflow-Ausführung eine eigene `runId` aus.

Weitere Informationen finden Sie unter:

[Amazon SWF SWF-Workflow-Verlauf](#)

## Amazon SWF SWF-Workflow-Verlauf

Der Fortschritt einer jeden Workflow-Ausführung wird in dem von Amazon SWF verwaltenden Workflow-Verlauf aufgezeichnet. Der Workflow-Verlauf ist ein detaillierter, vollständiger und konsistenter Datensatz eines jeden Ereignisses, das seit dem Start der Workflow-Ausführung aufgetreten ist. Ein Ereignis stellt eine eigenständige Änderung des Status einer Workflow-Ausführung da. Dabei kann es sich beispielsweise um eine neue geplante Aktivität oder das Abschließen der Ausführung einer Aktivität handeln. 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:

- Er ermöglicht eine Zustandslosigkeit von Anwendungen, da alle Informationen zu einer Workflow-Ausführung im 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*

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

- Einen Typ wie [WorkflowExecutionStarted](#) oder [ActivityTaskCompleted](#)
- 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. So verfügt beispielsweise das `ActivityTaskCompleted`-Ereignis über Attribute, die die IDs für die Ereignisse enthalten, die dem Zeitpunkt entsprechen, zu dem die Aktivitätsaufgabe geplant und gestartet wurde, sowie über ein Attribut, indem die Ergebnisdaten gespeichert werden.

Über die [GetWorkflowExecutionHistory](#)-Aktion erhalten Sie eine Kopie des aktuellen Status des Workflow-Ausführungsverlaufs. Im Rahmen der Interaktion zwischen Amazon SWF und dem Entscheider Ihres Workflows empfängt der Entscheider darüber hinaus 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 Workflow-Ausführungsverlauf angezeigt werden können, finden Sie unter dem [historyEvent](#)-Datentyp im Amazon Simple Workflow Service API-Referenz aus.

Amazon SWF speichert den vollständigen Verlauf der Workflow-Ausführungen für eine konfigurierbare Anzahl von Tagen nach Ausführungsabbruch. 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.

## Amazon SWF SWF-Akteure

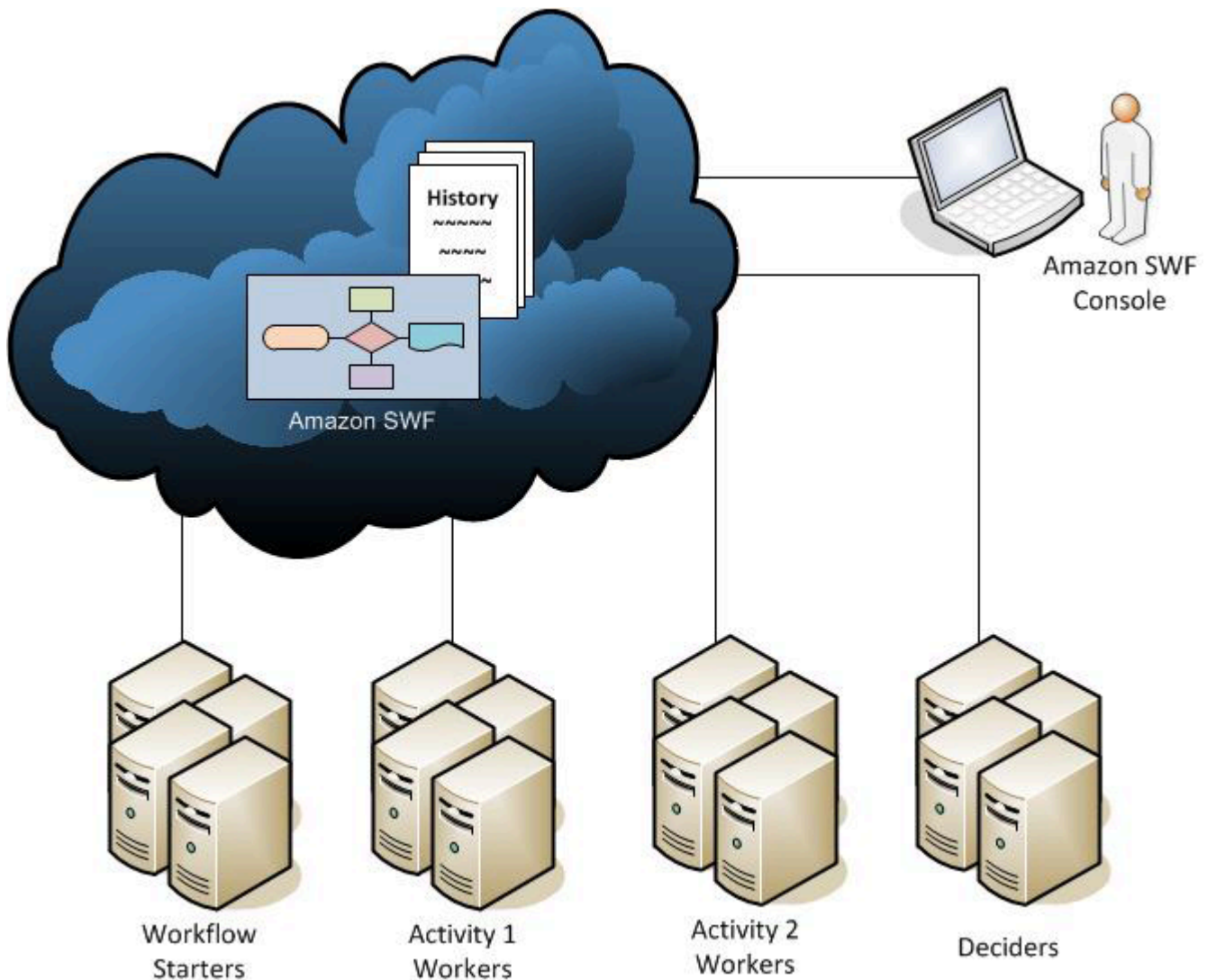
Themen

- [Was ist ein Akteur in Amazon SWF?](#)
- [Workflow-Starter](#)
- [Entscheider](#)
- [Aktivitäts-Worker](#)
- [Datenaustausch zwischen Akteuren](#)

### Was ist ein Akteur in Amazon SWF?

Während seiner Operationen interagiert Amazon SWF mit einer Vielzahl unterschiedlicher Programmtypen Schauspielers. Akteure können [Workflow-Starter](#), [Entscheider](#) oder [Aktivitäts-Worker](#) sein. Diese Akteure kommunizieren mit Amazon SWF über seine API. 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 Workflow-Ausführungsverlauf, um die nächsten Schritte in der Workflow-Ausführung zu bestimmen. Der Entscheider kommuniziert diese Schritte mit Amazon SWF über [Entscheidungen](#). Eine Entscheidung ist ein Amazon SWF SWF-Datentyp, der verschiedene weitere Aktionen darstellen kann. Eine Liste möglicher Entscheidungen finden Sie unter [Entscheidung](#) in Amazon Simple Workflow Service-API-Referenz.

Hier sehen Sie ein Beispiel einer 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 bringen die Workflow-Ausführung weiterhin voran, indem sie Entscheidungsaufgaben empfangen und mit mehr Entscheidungen auf Amazon SWF antworten, bis der Entscheider bestimmt, dass die Workflow-Ausführung abgeschlossen ist. Er antwortet dann mit der Entscheidung, die Workflow-Ausführung abzuschließen. Nachdem die Workflow-Ausführung abgeschlossen ist, 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 registrieren, entweder über die Amazon SWF SWF-Konsole oder über die [RegisterActivityType](#)Aktion

Jeder Aktivitäts-Worker führt Amazon SWF bei Abfragen für neue Aufgaben durch, die für die Ausführung durch diesen Aktivitäts-Worker geeignet sind. Bestimmte Aufgaben können nur von bestimmten Aktivitäts-Workern durchgeführt werden. Nachdem der Aktivitäts-Worker eine Aufgabe erhalten hat, verarbeitet er die Aufgabe bis zur Beendigung. Er 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 Aktivitäts-Worker Ergebnisse an Amazon SWF zurückgeben. Ebenso kann ein Entscheider die Ergebnisse einer Workflow-Ausführung melden, wenn die Ausführung abgeschlossen ist. Über Zeichenfolgen, deren Format benutzerdefiniert ist, kann jeder Akteur Daten an Amazon SWF senden bzw. von empfangen. Abhängig von der Größe und Vertraulichkeit der Daten können Sie Daten direkt übergeben oder Sie übergeben einen Zeiger auf Daten, die in einem anderen System oder Service gespeichert sind (wie Amazon S3 oder DynamoDB). Sowohl die direkt übergebenen Daten als auch die Zeiger auf andere Datenspeicher werden im Workflow-Ausführungsverlauf aufgezeichnet. Amazon SWF kopiert oder zwischengespeichert keinerlei Daten von externen Speichern als Teil des Verlaufs.

Da Amazon SWF den vollständigen Ausführungsstatus jeder Workflow-Ausführung aufzeichnet, einschließlich der Eingaben und der 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.

## Amazon SWF SWF-Aufgaben

Amazon SWF interagiert mit Aktivitäts-Workern und Entscheidern, indem Sie ihnen Arbeitsaufträge, so genannte Aufgaben, bereitstellen. Es gibt drei Arten von Aufgaben in Amazon SWF:

- **Aktivitätsaufgabe**— Eine Aktivitäts-Aufgabe weist den Aktivitäts-Worker an, seine Funktion auszuüben, z. B. den Bestand prüfen oder eine Kreditkarte belasten. Der Aktivitätsaufgabe enthält alle Informationen, die der Aktivitäts-Worker zum Ausüben seiner Funktion benötigt.
- **Lambda-Aufgabe**— Eine Lambda-Aktivitätsaufgabe ähnelt einer Aktivitätsaufgabe, führt jedoch statt einer herkömmlichen Amazon SWF SWF-Aktivität eine Lambda-Funktion aus. Weitere Informationen zum Definieren einer Lambda-Aufgabe finden Sie unter [AWS Lambda-Aufgaben](#).
- **Entscheidungsaufgabe**— Eine Entscheidungsaufgabe informiert einen Entscheider darüber, dass die Workflow-Ausführung geändert wurde, sodass der Entscheider bestimmen kann, welche Aktivität als Nächstes ausgeführt werden muss. Die Entscheidungsaufgabe enthält den aktuellen Workflow-Verlauf.

Amazon SWF plant eine Entscheidungsaufgabe, wenn der Workflow gestartet wird und bei jeder Änderung des Workflow-Zustands, z. B. wenn eine Aktivitätsaufgabe abgeschlossen wird. Jede Entscheidungsaufgabe enthält eine paginierte Ansicht des gesamten Workflow-Ausführungsverlaufs. Der Entscheider analysiert den Workflow-Ausführungsverlauf und gibt eine Reihe von Entscheidungen an Amazon SWF zurück, die angeben, was im nächsten Schritt der Workflow-Ausführung passieren soll. Im Prinzip gibt jede Entscheidungsaufgabe dem Entscheider Gelegenheit, den Workflow auszuwerten und Amazon SWF -Anweisungen zu erteilen.

Um sicherzustellen, dass keine in Konflikt stehenden Entscheidungen verarbeitet werden, weist Amazon SWF jede Entscheidungsaufgabe genau einem Entscheider zu und lässt nur eine aktive Entscheidungsaufgabe in der Workflow-Ausführung zu.

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

Amazon SWF weist jede Aktivitätsaufgabe genau einem Aktivitäts-Worker 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

## Amazon SWF-Domäne

Domänen bieten Ihnen eine Möglichkeit, Amazon-SWF-Ressourcen innerhalb Ihres AWS-Konto. Alle Komponenten eines Workflows, wie beispielsweise der Workflow-Typ und die Aktivitätstypen, müssen einer Domäne zugewiesen sein. Es ist möglich, mehr als einen Workflow in einer Domäne zu haben, jedoch können Workflows in verschiedenen Domänen nicht miteinander 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. Bei diesem Zeitraum handelt es sich um die Dauer, die Amazon SWF Informationen über die Workflow-Ausführung aufbewahrt, nachdem die Workflow-Ausführung abgeschlossen wurde.



## Amazon SWF SWF-Objekt-IDs

In der folgenden Liste wird beschrieben, wie Amazon SWF SWF-Objekte wie Workflow-Ausführungen eindeutig gekennzeichnet 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 Entscheidungs- und Aktivitätsaufgabe wird durch ein eindeutiges Aufgabentoken identifiziert. Das Aufgabentoken wird von Amazon SWF generiert und zusammen mit anderen Informationen zu der Aufgabe in der Antwort von `PollForDecisionTask` oder `PollForActivityTask` aus. 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 Werte sind Parameter, die an [StartWorkflowExecution](#) übergeben werden. Die Ausführungs-ID wird von `StartWorkflowExecution` zurückgegeben.

## Amazon SWF SWF-Aufgabenlisten

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 wird, können Sie eine Warteschlange (Aufgabenliste) angeben, in der diese Aufgabe entnommen werden soll. Wenn Sie Amazon SWF für eine Aufgabe an senden, geben Sie an, aus welcher Warteschlange (Aufgabenliste) die Aufgabe entnommen 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 werden Aufgabenlisten ebenso wie Aktivitäten und Workflows in einem bestimmten Bereich `AWSRegion` und `Amazon SWF-Domain`.

Themen

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

## Entscheidungsaufgabenlisten

Jede Workflow-Ausführung ist einer bestimmten Entscheidungsaufgabenliste zugeordnet. Wenn ein Workflow-Typ (mit der Aktion [RegisterWorkflowType](#)) registriert wird, können Sie eine Standardaufgabenliste für die Ausführung dieses Workflow-Typs festlegen. 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. Aufgaben werden in der Aufgabenliste in der Reihenfolge geplant. Amazon SWF gibt die Aufgaben aus der Liste möglichst in dieser Reihenfolge zurück. Es kann jedoch vorkommen, dass Aufgaben in anderer Reihenfolge zurückgegeben werden.

Wenn ein Aktivitätstyp (mit der Aktion [RegisterActivityType](#)) registriert wird, können Sie eine Standardaufgabenliste für diesen Aktivitätstyp festlegen. Standardmäßig werden Aktivitätsaufgaben dieses Typs für die angegebene Aufgabenliste geplant. Wenn der Entscheider jedoch eine

Aktivitätsaufgabe (mit der Entscheidung [ScheduleActivityTask](#)) plant, kann optional eine andere Aufgabenliste für diese Aufgabe angegeben werden. 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 Aktivitäts-Worker eine neue Aufgabe (mit der Aktion [PollForActivityTask](#)) abrufen, kann er eine Aktivitätsaufgabenliste angeben, aus der die Aufgabe abgerufen werden soll. 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.

## Beenden einer Workflow-Ausführung

Sobald Sie eine Workflow-Ausführung gestartet haben, ist diese offen. 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, dem Workflow-Administrator oder 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 benutzerdefinierte Einstellungen für die Zeitüberschreitung nicht überschritten werden. Wenn bei einer Workflow-Ausführung eine Zeitüberschreitung auftritt, wird sie von Amazon automatisch geschlossen. Weitere Informationen zu den Werten für die Zeitüberschreitung finden Sie im Abschnitt [Amazon SWF SWF-Zeitüberschreitungstypen](#).

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.

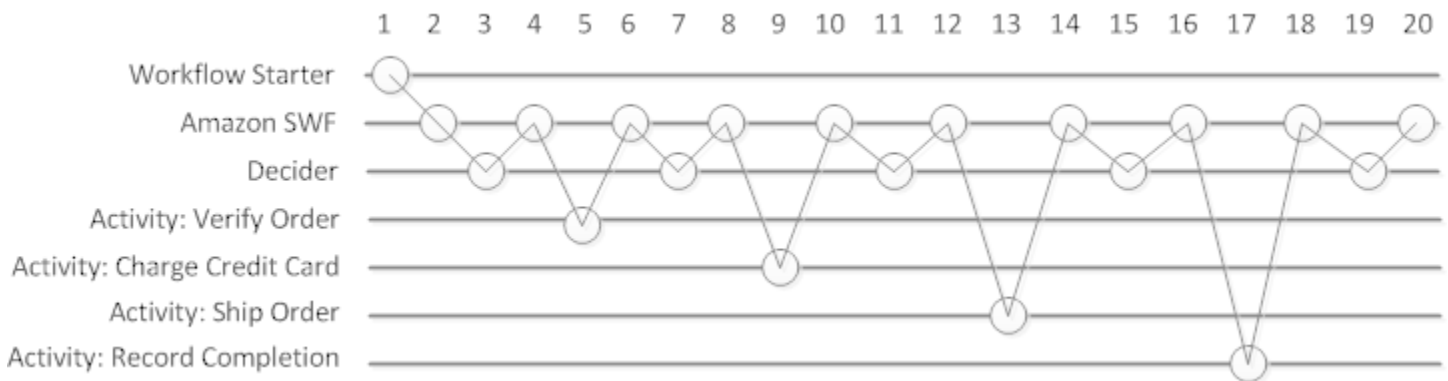
Schließlich können Sie Workflow-Ausführungen direkt in der Amazon SWF SWF-Konsole oder programmgesteuert mit dem [TerminateWorkflowExecution](#) API. Durch das Beenden wird ein Schließen der Workflow-Ausführung erzwungen. Ein Abbruch ist dem Beenden vorzuziehen, da Ihre Entscheider das Schließen der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

Amazon SWF beendet eine Workflow-Ausführung, wenn die Ausführung bestimmte servicedefinierte Grenzwerte überschreitet. Amazon SWF beendet untergeordnete Workflows, wenn der übergeordnete Workflow beendet wurde und die angewandte untergeordnete Richtlinie vorsehen, dass der untergeordnete Workflow ebenfalls beendet werden muss.

## Lebenszyklus einer -Workflow-Ausführung

Vom Starten einer Workflow-Ausführung bis zu Ihrem Abschluss weist Amazon SWF passende Aufgaben, also Aktivitäts- oder Entscheidungsaufgaben zu.

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.

Description	Aktion, Entscheidung oder Ereignis
1. Der Workflow-Starter ruft die passende Amazon SWF SWF-Aktion auf, um die Workflow-Ausführung für eine Bestellung zu starten, und stellt die Bestellinformationen bereit.	Aktion <a href="#">StartWorkflowExecution</a>
2. Amazon SWF empfängt die Startanforderung für die Workflow-Ausführung und plant die erste Entscheidungsaufgabe.	Ereignis <a href="#">WorkflowExecutionStarted</a> und Ereignis <a href="#">DecisionTaskScheduled</a>
3. Der Entscheider empfängt die Aufgabe von Amazon SWF, prüft den Verlauf,	<a href="#">PollForDecisionTask</a> -Aktion. <a href="#">RespondDecisionTaskCompleted</a> -Aktion und <a href="#">ScheduleActivityTask</a> -Entscheidung.

Description	Aktion, Entscheidung oder Ereignis
<p>wendet die Koordinationslogik an, um zu bestimmen, ob vorherige Aktivitäten aufgetreten sind, trifft die Entscheidung, die Aktivität „Bestellung prüfen“ mit den Informationen zu planen, die der Aktivitäts-Worker zur Verarbeitung der Aufgabe benötigt, und gibt die Entscheidung an Amazon SWF zurück.</p>	
<p>4. Amazon SWF empfängt die Entscheidung, plant die Aktivitätsaufgabe „Bestellung prüfen“ und wartet, bis die Aktivitätsaufgabe abgeschlossen oder abgeschlossen ist.</p>	<p><a href="#">ActivityTaskScheduled</a> event</p>

Description	Aktion, Entscheidung oder Ereignis
5. Ein Aktivitäts-Worker, der die Aktivität „Bestellung prüfen“ verarbeiten kann, empfängt die Aufgabe, verarbeitet sie und gibt die Ergebnisse an Amazon SWF zurück.	Aktion <a href="#">PollForActivityTask</a> und Aktion <a href="#">RespondActivityTaskCompleted</a>
6. Amazon SWF empfängt die Ergebnisse der Aktivität „Bestellung prüfen“, fügt sie dem Verlauf des Workflows hinzu und plant eine Entscheidungsaufgabe.	Ereignis <a href="#">ActivityTaskCompleted</a> und Ereignis <a href="#">DecisionTaskScheduled</a>

Description	Aktion, Entscheidung oder Ereignis
<p>7. Der Entscheider empfängt die Aufgabe von Amazon SWF, prüft den Verlauf, wendet die Koordinationslogik an, trifft die Entscheidung, die Aktivitätsaufgabe „Kreditkarte belasten“ mit den Informationen zu planen, die der Aktivitäts-Worker zur Verarbeitung der Aufgabe benötigt, und gibt die Entscheidung an Amazon SWF zurück.</p>	<p><a href="#">PollForDecisionTask</a> -Aktion. <a href="#">RespondDecisionTaskCompleted</a> -Aktion mit <a href="#">ScheduleActivityTask</a> -Entscheidung.</p>
<p>8. Amazon SWF empfängt die Entscheidung, plant die Aktivitätsaufgabe „Kreditkarte belasten“ und wartet, bis sie abgeschlossen oder beendet wird.</p>	<p>Ereignis <a href="#">DecisionTaskCompleted</a> und Ereignis <a href="#">ActivityTaskScheduled</a></p>



Description	Aktion, Entscheidung oder Ereignis
9. Ein Aktivitäts-Worker, der die Aktivität „Kreditkarte belasten“ verarbeiten kann, empfängt die Aufgabe, verarbeitet sie und gibt die Ergebnisse an Amazon SWF zurück.	Aktion <a href="#">PollForActivityTask</a> und Aktion <a href="#">RespondActivityTaskCompleted</a>
10. Amazon SWF empfängt die Ergebnisse der Aktivitätsaufgabe „Kreditkarte belasten“, fügt sie dem Verlauf des Workflows hinzu und plant eine Entscheidungsaufgabe.	Ereignis <a href="#">ActivityTaskCompleted</a> und Ereignis <a href="#">DecisionTaskScheduled</a>

Description	Aktion, Entscheidung oder Ereignis
<p>11. Der Entscheider empfängt die Aufgabe von Amazon SWF, prüft den Verlauf, wendet die Koordinationslogik an, trifft die Entscheidung, die Aktivitätsaufgabe „Bestellung versenden“ mit den Informationen zu planen, die der Aktivitäts-Worker zur Verarbeitung der Aufgabe benötigt, und gibt die Entscheidung an Amazon SWF zurück.</p>	<p><a href="#">PollForDecisionTask</a> -Aktion. <a href="#">RespondDecisionTaskCompleted</a> mit <a href="#">ScheduleActivityTask</a> -Entscheidung.</p>
<p>12. Amazon SWF empfängt die Entscheidung, plant die Aktivitätsaufgabe „Bestellung versenden“ und wartet, bis sie abgeschlossen oder beendet wird.</p>	<p>Ereignis <a href="#">DecisionTaskCompleted</a> und Ereignis <a href="#">ActivityTaskScheduled</a></p>

Description	Aktion, Entscheidung oder Ereignis
13. Ein Aktivitäts-Worker, der die Aktivität „Bestellung versenden“ verarbeiten kann, empfängt die Aufgabe, verarbeitet sie und gibt die Ergebnisse an Amazon SWF zurück.	Aktion <a href="#">PollForActivityTaskCompleted</a> und Aktion <a href="#">RespondActivityTaskCompleted</a>
14. Amazon SWF empfängt die Ergebnisse der Aktivitätsaufgabe „Bestellung versenden“, fügt sie dem Verlauf des Workflows hinzu und plant eine Entscheidungsaufgabe.	Ereignis <a href="#">ActivityTaskCompleted</a> und Ereignis <a href="#">DecisionTaskScheduled</a>

Description	Aktion, Entscheidung oder Ereignis
<p>15. Der Entscheider empfängt die Aufgabe von Amazon SWF, prüft den Verlauf, wendet die Koordinationslogik an, trifft die Entscheidung, die Aktivitätsaufgabe „Vorgang schließen“ mit den Informationen zu planen, die der Aktivitäts-Worker zur Verarbeitung der Aufgabe benötigt, und gibt die Entscheidung an Amazon SWF zurück.</p>	<p><a href="#">PollForDecisionTask</a> -Aktion. <a href="#">RespondDecisionTaskCompleted</a> -Aktion mit <a href="#">ScheduleActivityTask</a> -Entscheidung.</p>
<p>16. Amazon SWF empfängt die Entscheidung, plant die Aktivitätsaufgabe „Vorgang schließen“ und wartet, bis sie abgeschlossen oder beendet wird.</p>	<p>Ereignis <a href="#">DecisionTaskCompleted</a> und Ereignis <a href="#">ActivityTaskScheduled</a></p>

Description	Aktion, Entscheidung oder Ereignis
<p>17. Ein Aktivitäts-Worker, der die Aktivität „Vorgang schließen“ verarbeiten kann, empfängt die Aufgabe, verarbeitet sie und gibt die Ergebnisse an Amazon SWF zurück.</p>	<p>Aktion <a href="#">PollForActivityTask</a> und Aktion <a href="#">RespondActivityTaskCompleted</a></p>
<p>18. Amazon SWF empfängt die Ergebnisse der Aktivitätsaufgabe „Vorgang schließen“, fügt sie dem Verlauf des Workflows hinzu und plant eine Entscheidungsaufgabe.</p>	<p>Ereignis <a href="#">ActivityTaskCompleted</a> und Ereignis <a href="#">DecisionTaskScheduled</a></p>
<p>19. Der Entscheider empfängt die Aufgabe von Amazon SWF, prüft den Verlauf, wendet die Koordinationslogik an, trifft die Entscheidung, die Workflow-Ausführung zu schließen, und gibt die Entscheidung zusammen mit den Ergebnissen an Amazon SWF zurück.</p>	<p><a href="#">PollForDecisionTask</a> -Aktion. <a href="#">RespondDecisionTaskCompleted</a> -Aktion mit <a href="#">CompleteWorkflowExecution</a> -Entscheidung.</p>

Description	Aktion, Entscheidung oder Ereignis
20 Amazon SWF schließt die Workflow-Ausführung und archiviert den Verlauf für eine spätere Verwendung.	<a href="#">WorkflowExecutionCompleted</a> event.

## Abfragen von Aufgaben in Amazon SWF

Entscheider und Aktivitäts-Worker kommunizieren mit Amazon SWF über `Langabfrageaus`. Der Entscheider oder Aktivitäts-Worker initiiert die Kommunikation mit Amazon SWF, benachrichtigt Amazon SWF über die Verfügbarkeit zum Akzeptieren einer Aufgabe und gibt anschließend eine Aufgabenliste zum Abfragen von Aufgaben an.

Wenn eine Aufgabe in der angegebenen Aufgabenliste verfügbar ist, gibt Amazon SWF diese sofort in der Antwort zurück. Wenn keine Aufgabe verfügbar ist, hält Amazon SWF die TCP-Verbindung für bis zu 60 Sekunden offen, sodass eine Aufgabe, die in dieser Zeit verfügbar wird, in derselben 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 Konzepte in Amazon SWF

Das E-Commerce-Beispiel im Abschnitt [Funktionsweise von Amazon SWF](#) 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-Funktionen beschrieben, mit denen Sie robuste und komplexe Workflows erstellen können.

## Themen

- [Versioning](#)
- [Signale](#)
- [Untergeordnete Workflows](#)
- [Marker](#)
- [Tags](#)
- [Implementierung einer exklusiven Wahl mit Amazon Simple Workflow Service](#)
- [Amazon Simple Workflow Service Timer](#)
- [Abbrechen einer Aufgabe für Amazon Simple Workflow Service](#)

## Versioning

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. [Amazon SWF SWF-Aufgabenlisten](#) 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 übergeben werden sollen, falls vorhanden. Dann programmieren Sie den Entscheider so, dass er das Signalereignis ([WorkflowExecutionSignaled](#)) in der Historie erkennt und es entsprechend verarbeitet. Wenn ein Prozess eine Workflow-Ausführung signalisieren möchte, ruft er Amazon SWF auf (mithilfe der [SignalWorkflowExecution](#)Aktion oder, im Fall eines Entscheiders, anhand der [SignalExternalWorkflowExecution](#)Entscheidung), das die Kennung für die Ziel-Workflow-Ausführung, den Signalnamen und die Signaldaten angibt. Amazon SWF empfängt dann das Signal, zeichnet es in der Historie der Ziel-Workflow-Ausführung auf und plant dafür eine Entscheidungsaufgabe. 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 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": "f5ebbac6-941c-4342-ad69-dfd2f8be6689",
  "signalName": "CancelOrder",
  "input": "order 3553"}
```

War die Workflow-Ausführung wie folgt aussieht, gibt Amazon SWF eine erfolgreiche Antwort zurück, die etwa wie folgt aussieht. Amazon SWF generiert eine Entscheidungsaufgabe, um den Entscheider über die Verarbeitung des Signals zu informieren.

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

## Untergeordnete Workflows

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 legen auch die Child-Richtlinie fest, 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:

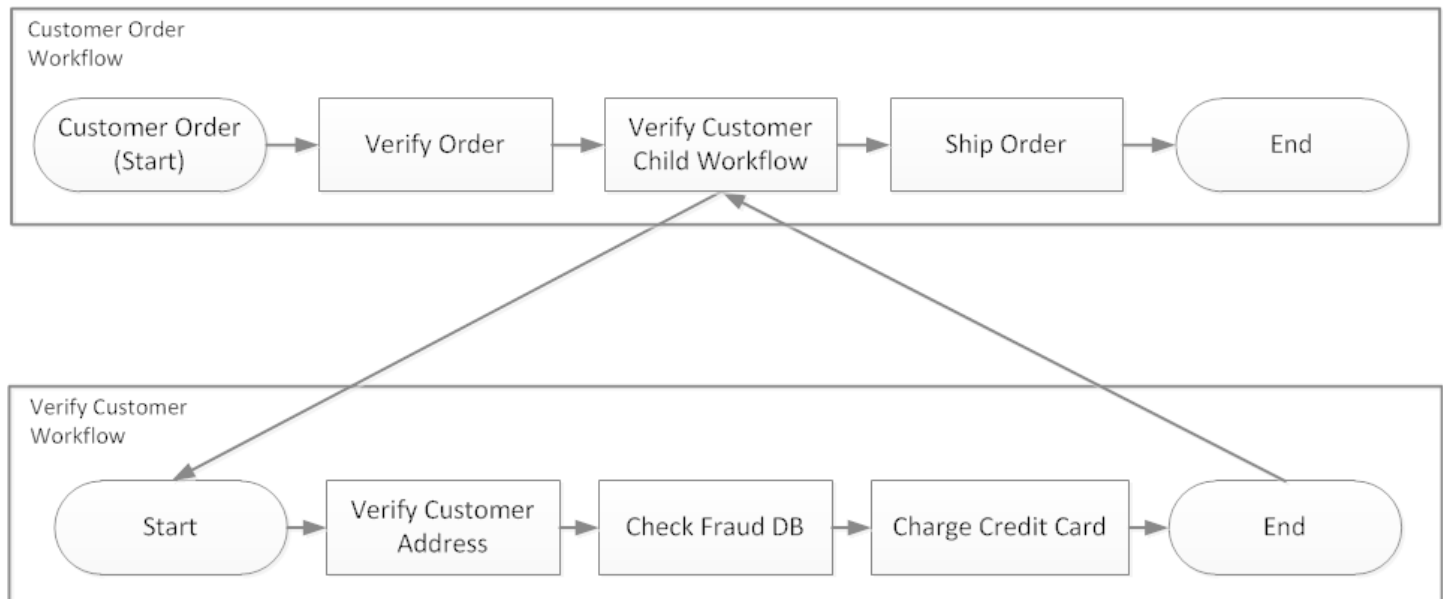
- **BEENDEN:** Amazon SWF beendet die Hinrichtungen von Kindern.
- **REQUEST\_CANCEL:** Amazon SWF versucht, die untergeordnete Ausführung abzubrechen, indem ein `WorkflowExecutionCancelRequested` Ereignis in die Workflow-Ausführungshistorie des untergeordneten Workflows aufgenommen wird.
- **ABBRUCH:** Amazon SWF wird keine Maßnahmen ergreifen; die Hinrichtungen von Kindern werden fortgesetzt.

Nachdem die untergeordnete Ausführung gestartet wurde, wird sie wie eine reguläre Ausführung durchgeführt. Nach Abschluss zeichnet Amazon SWF den Abschluss zusammen mit seinen 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.

## Marker

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 mithilfe der `RespondDecisionTaskCompleted` Aktion. 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 das der gesamte Verlauf überprüft werden muss.

Im folgenden Beispiel schließt der Entscheider eine Entscheidungsaufgabe ab und antwortet mit der Aktion `RespondDecisionTaskCompleted`, die ein Entscheidung `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"
    }
  },
]
}
```

War Amazon SWF Markierung erfolgreich, wird eine erfolgreiche HTTP-Antwort zurückgegeben, die etwa wie folgt aussieht.

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

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

## Tags

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 basierend auf 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 durch direkte Interaktion mit der Amazon SWF-API. 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 unter [Arbeiten mit Amazon SWF-APIs](#) und [Amazon Simple Workflow Service – API-Referenz](#).

## Markieren von Workflow-Ausführungen

Mit Amazon SWF können Sie Tags Workflow-Ausführungen zuordnen und dann basierend auf diesen Tags nach Workflow-Ausführungen abfragen. Sie können die Liste filtern, wenn Sie die Sichtbarkeitsoperationen verwenden. Wenn Sie die Tags, die Sie einer Ausführung zuweisen, sorgfältig auswählen, können Sie sie verwenden, um aussagekräftige Auflistungen bereitzustellen.

Angenommen, Sie betreiben mehrere Fulfillment-Center. Mit Tags können Sie die Prozesse auflisten, die in einem bestimmten -Erfüllungszentrum stattfinden. Oder wenn ein Kunde verschiedene Arten von Mediendateien konvertiert, können Tags beim Konvertieren von Video-, Audio- und Bilddateien auf unterschiedliche Prozesse 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 Ergebnisse basierend auf Ihren 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 des Zugriffs auf Domains mit Tags

Sie können den Zugriff auf Amazon Simple Workflow Service-Domains steuern, indem Sie auf Tags verweisen, die Amazon SWF-Domains in IAM zugeordnet sind. Sie können beispielsweise Domänen einschränken, die ein Tag enthalten, bei dem der Schlüssel `environment` und der Wert `istproduction`:

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



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

Durch diese Richtlinie wird der Befehl Deny (Verweigern) bei Zugriffsversuchen auf Domänen ausgeführt, die als `environment/production` markiert wurden.

Weitere Informationen finden Sie hier:

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

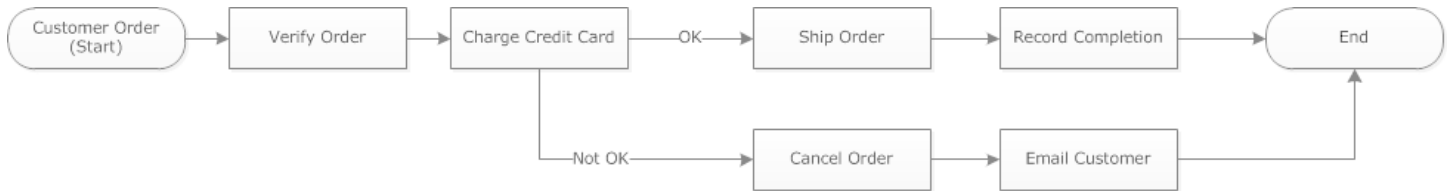
## Implementierung einer exklusiven Wahl mit Amazon Simple Workflow Service

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

Der Amazon Simple Workflow Service (Amazon SWF) bietet nicht ausdrücklich eine Aktion für die exklusive Wahl. Um eine exklusive Wahl zu nutzen, schreiben Sie die Entscheiderlogik so, dass basierend auf den Ergebnissen der vorherigen Aufgabe unterschiedliche Entscheidungen 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 im `RespondDecisionTaskCompleted`-Aktion gibt Amazon SWF eine erfolgreiche HTTP-Antwort ähnlich wie im folgenden Beispiel zurück.

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

## Amazon Simple Workflow Service Timer

Über einen Timer können Sie Ihren Entscheider benachrichtigen, wenn eine bestimmte Zeitdauer verstrichen 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 eine `TimerFired`-Ereignis zum Verlauf der Workflow-Ausführung und Planen einer Entscheidungsaufgabe ein. 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 einer Aufgabe für Amazon Simple Workflow Service

Über den Abbruch einer Aktivitätsaufgabe kann der Entscheider Aktivitäten beenden, die nicht mehr benötigt werden. Amazon SWF verwendet einen kooperativen Abbruchmechanismus und unterbricht laufende Aktivitätsaufgaben nicht zwangsweise. 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 erhält, gibt Amazon SWF eine erfolgreiche HTTP-Antwort ähnlich wie im folgenden Beispiel zurück. 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` abzubrechen. 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 Abbruchanforderung erfolgreich erhält, gibt eine erfolgreiche HTTP-Antwort ähnlich wie im folgenden Beispiel zurück:

```
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 abgebrochen wurde — wie durch ein `ActivityTaskCanceledEvent` — Programmieren Sie Ihren Entscheider so, dass er die erforderlichen Schritte wie das Schließen der Workflow-Ausführung ausführt.

Wenn die Aktivitätsaufgabe nicht abgebrochen werden konnte - wenn die Aufgabe beispielsweise fehlschlägt oder eine Zeitüberschreitung auftritt, sollte der Entscheider die Ergebnisse der Aktivität akzeptieren oder je nach Anwendungsfall eine Bereinigung oder Vermeidung durchführen.

Wenn die Aktivitätsaufgabe bereits einem Aktivitäts-Worker zugewiesen wurde, wird die Abbruchanforderung über den Task-Heartbeat-Mechanismus übermittelt. Tätigkeit, die Mitarbeiter

regelmäßig nutzen können `RecordActivityTaskHeartbeat` Amazon SWF zu berichten, dass die Aufgabe noch im Gange 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 eine Anforderung zum Abbrechen der Aufgabe sendet, legt Amazon SWF den Wert `cancelRequestObject` auf „true“. Das Objekt `cancelRequest` gehört zum Objekt `ActivityTaskStatus`, das vom Service als Antwort auf `RecordActivityTaskHeartbeat` zurückgegeben wird.

Amazon SWF verhindert nicht, dass eine Aktivitätsaufgabe, für die eine Abbruchanforderung gesendet wurde, erfolgreich abgeschlossen wird. Wie mit Abbruchanforderungen umgegangen wird, muss festgelegt werden. 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` Wenn der Verlauf der Workflow-Ausführung aktualisiert wird, 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.

# Amazon Simple Workflow Service

Dieser Abschnitt enthält Informationen zur Amazon Simple Workflow Service.

## Themen

- [Datenschutz bei Amazon Simple Workflow Service](#)
- [Identity and Access Management in Amazon Simple Workflow Service](#)
- [Protokollieren und Überwachen](#)
- [Compliance-Validierung für Amazon Simple Workflow Service](#)
- [Amazon Simple Workflow Service](#)
- [Infrastruktursicherheit in Amazon Simple Workflow Service](#)
- [Konfigurations- und Schwachstellenanalyse in Amazon Simple Workflow Service](#)

Amazon SWF verwendet IAM, um den Zugriff auf andere AWS -Ressourcen zu steuern. Eine Übersicht über die Funktionsweise von IAM finden Sie im IAM-Benutzerhandbuch unter [Überblick über die Zugriffsverwaltung](#). 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, ist AWS für den Schutz der globalen Infrastruktur verantwortlich, in der die gesamte AWS Cloud ausgeführt wird. 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, AWS-Konto-Anmeldeinformationen zu schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einzurichten. 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).



- Verwenden Sie SSL/TLS für die Kommunikation mit AWS-Ressourcen. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit AWS CloudTrail ein.
- Verwenden Sie AWS-Verschlüsselungslösungen zusammen mit allen Standardsicherheitskontrollen in 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 auf AWS über eine Befehlszeilenschnittstelle oder über eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter [Federal Information Processing Standard \(FIPS\) 140-2](#).

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 Geräten arbeiten und die Konsole, die API oder AWS SDKs AWS-Services verwenden. AWS CLI 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 kann diese Anmeldeinformationen mit Berechtigungen für den Zugriff auf AWS-Ressourcen verwenden, 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) ist ein AWS-Service, mit dem Administratoren den Zugriff auf AWS-Ressourcen sicher steuern können. IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Amazon SWF-Ressourcen zu verwenden. IAM ist ein AWS-Service, den Sie ohne zusätzliche Kosten verwenden 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 in 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 VPEndpunkte für Amazon SWF](#)
- [Beheben von Identitäts- und Zugriffsfehlern bei Amazon Simple Workflow Service](#)

## Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, die Sie in Amazon SWF ausführen.

**Servicebenutzer** — Wenn Sie den Amazon SWF-Service für Ihre Arbeit verwenden, stellt Ihnen Ihr Administrator die Anmeldeinformationen und Berechtigungen zur Verfügung, die Sie benötigen. Da Sie für Ihre Arbeit mehr Amazon SWF SWF-Funktionen verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Featuresweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anfordern müssen. Wenn Sie in Amazon SWF nicht auf eine Funktion zugreifen können, finden Sie weitere Informationen unter [Beheben von Identitäts- und Zugriffsfehlern bei Amazon Simple Workflow Service](#).

**Service-Administrator** — Wenn Sie in Ihrem Unternehmen für die Amazon SWF-Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollen Zugriff auf Amazon SWF. Es ist Ihre Aufgabe, zu bestimmen, auf welche Funktionen und Ressourcen von Amazon SWF Ihre Servicebenutzer zugreifen sollen. Sie müssen dann Anträge an Ihren IAM-Administrator stellen, um die Berechtigungen Ihrer Servicenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen darüber, wie Ihr Unternehmen IAM mit Amazon SWF verwenden kann, finden Sie unter [So funktioniert Amazon Simple Workflow Service mit IAM](#).

**IAM-Administrator** — Wenn Sie ein IAM-Administrator sind, möchten Sie vielleicht mehr darüber erfahren, wie Sie Richtlinien schreiben können, um den Zugriff auf Amazon SWF zu verwalten. Beispiele für identitätsbasierte Amazon SWF SWF-Richtlinien, die Sie in IAM verwenden können, finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon Simple Workflow Service](#)

## Authentifizierung mit Identitäten

Authentifizierung ist die Art, wie Sie sich mit Ihren Anmeldeinformationen bei AWS anmelden. Die Authentifizierung (Anmeldung bei AWS) muss als Root-Benutzer des AWS-Kontos, als IAM-Benutzer oder durch Übernahme einer IAM-Rolle erfolgen.

Sie können sich bei AWS als Verbundidentität mit Anmeldeinformationen anmelden, die über eine Identitätsquelle bereitgestellt werden. Benutzer von AWS IAM Identity Center. (IAM Identity Center), die Single-Sign-on-Authentifizierung Ihres Unternehmens und Anmeldeinformationen für Google oder Facebook sind Beispiele für Verbundidentitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie auf AWS mithilfe des Verbunds zugreifen, übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich bei der AWS Management Console oder beim AWS-Zugriffportal anmelden. Weitere Informationen zum Anmelden bei AWS finden Sie unter [So melden Sie sich bei Ihrem AWS-Konto an](#) im Benutzerhandbuch von AWS-Anmeldung.

Bei programmgesteuerten Zugriff auf AWS bietet AWS ein Software Development Kit (SDK) und eine Command Line Interface (CLI, Befehlszeilenschnittstelle) zum kryptographischen Signieren Ihrer Anforderungen mit Ihren Anmeldeinformationen. Wenn Sie keine AWS-Tools verwenden, müssen Sie Anforderungen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode zum eigenen Signieren von Anforderungen finden Sie unter [Signieren von AWS-API-Anforderungen](#) im IAM-Benutzerhandbuch.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. AWS empfiehlt beispielsweise die Verwendung von Multi-Faktor Authentifizierung (MFA), um die Sicherheit Ihres Kontos zu verbessern. Weitere Informationen finden Sie unter [Multi-Faktor-Authentifizierung](#) im AWS IAM Identity Center-Benutzerhandbuch und [Verwenden der Multi-Faktor-Authentifizierung \(MFA\) in AWS](#) im IAM-Benutzerhandbuch.

### AWS-Konto-Root-Benutzer

Wenn Sie ein AWS-Konto neu erstellen, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services und Ressourcen des Kontos hat. Diese Identität wird als AWS-Konto-Root-Benutzer bezeichnet. Für den Zugriff auf den Root-Benutzer müssen Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, die zur Erstellung des Kontos verwendet wurden. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, um die Aufgaben

auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter [Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

## Verbundidentität

Als bewährte Methode empfiehlt es sich, menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, aufzufordern, den Verbund mit einem Identitätsanbieter zu verwenden, um auf AWS-Services mit temporären Anmeldeinformationen zuzugreifen.

Eine Verbundidentität ist ein Benutzer aus dem Benutzerverzeichnis Ihres Unternehmens, ein Web Identity Provider, AWS Directory Service, das Identity-Center-Verzeichnis oder jeder Benutzer, der mit Anmeldeinformationen, die über eine Identitätsquelle bereitgestellt werden, auf AWS-Services zugreift. Wenn Verbundidentitäten auf AWS-Konten zugreifen, übernehmen sie Rollen und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, AWS IAM Identity Center zu verwenden. Sie können Benutzer und Gruppen im IAM Identity Center erstellen oder Sie können eine Verbindung mit einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und synchronisieren, um sie in allen AWS-Konten und Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter [Was ist IAM Identity Center?](#) im AWS IAM Identity Center-Benutzerhandbuch.

## IAM-Benutzer und -Gruppen

Ein [IAM-Benutzer](#) ist eine Identität in Ihrem AWS-Konto mit bestimmten Berechtigungen für eine einzelne Person oder eine einzelne Anwendung. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter [Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

Eine [IAM-Gruppe](#) ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise einer Gruppe mit dem Namen IAMAdmins Berechtigungen zum Verwalten von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter [Erstellen eines IAM-Benutzers \(anstatt einer Rolle\)](#) im IAM-Benutzerhandbuch.

## IAM-Rollen

Eine [IAM-Rolle](#) ist eine Identität in Ihrem AWS-Konto mit spezifischen Berechtigungen. Sie ist einem IAM-Benutzer vergleichbar, ist aber nicht mit einer bestimmten Person verknüpft. Sie können vorübergehend eine IAM-Rolle in der AWS Management Console übernehmen, indem Sie [Rollen wechseln](#). Sie können eine Rolle annehmen, indem Sie eine AWS CLI oder AWS-API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter [Verwenden von IAM-Rollen](#) im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- **Verbundbenutzerzugriff** – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie unter [Erstellen von Rollen für externe Identitätsanbieter](#) im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in IAM. Informationen zu Berechtigungssätzen finden Sie unter [Berechtigungssätze](#) im AWS IAM Identity Center-Benutzerhandbuch.
- **Temporäre IAM-Benutzerberechtigungen** – Ein IAM-Benutzer oder eine -Rolle kann eine IAM-Rolle übernehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.
- **Kontoübergreifender Zugriff** – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. In einigen AWS-Services können Sie jedoch eine Richtlinie direkt an eine Ressource anfügen (anstatt eine Rolle als Proxy zu verwenden). Informationen zu den Unterschieden zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter [So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien](#) im IAM-Benutzerhandbuch.

- **Serviceübergreifender Zugriff** – Einige AWS-Services verwenden Features in anderen AWS-Services. Wenn Sie beispielsweise einen Aufruf in einem Service tätigen, führt dieser Service häufig Anwendungen in Amazon EC2 aus oder speichert Objekte in Amazon S3. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
- Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle zum Ausführen von Aktionen in Amazon Managed Service for Prometheus verwenden, gelten Sie als Prinzipal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen AWS-Service aufruft, in Kombination mit der Anforderung an den AWS-Service, Anforderungen an nachgelagerte Services zu stellen. FAS-Anforderungen werden nur dann gestellt, wenn ein Dienst eine Anforderung erhält, die Interaktionen mit anderen AWS-Services oder Ressourcen erfordert, um abgeschlossen werden zu können. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).
- **Servicerolle** – Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service übernimmt, 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.
- **Serviceverknüpfte Rolle** – Eine serviceverknüpfte Rolle ist ein Typ von Servicerolle, die mit einem AWS-Service verknüpft ist. Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Serviceverknüpfte Rollen werden in Ihrem AWS-Konto angezeigt und gehören zum Service. Ein IAM-Administrator kann die Berechtigungen für serviceverbundene Rollen anzeigen, aber nicht bearbeiten.
- **Anwendungen in Amazon EC2** – Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden und AWS CLI- oder AWS-API-Anforderungen durchführen. Das ist eher zu empfehlen, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Erstellen Sie ein Instance-Profil, das an die Instance angefügt ist, um eine AWS-Rolle einer EC2-Instance zuzuweisen und die Rolle für sämtliche Anwendungen der Instance bereitzustellen. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie unter [Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden](#) im IAM-Benutzerhandbuch.

Informationen dazu, wann Sie IAM-Rollen oder IAM-Benutzer verwenden sollten, finden Sie unter [Erstellen einer IAM-Rolle \(anstatt eines Benutzers\)](#) im IAM-Benutzerhandbuch.

## Verwalten des Zugriffs mit Richtlinien

Für die Zugriffssteuerung in AWS erstellen Sie Richtlinien und weisen diese den AWS-Identitäten oder -Ressourcen zu. Eine Richtlinie ist ein Objekt in AWS, das, wenn es einer Identität oder Ressource zugeordnet wird, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anforderung stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden in AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter [Übersicht über JSON-Richtlinien](#) im IAM-Benutzerhandbuch.

Administratoren können mithilfe von AWS-JSON-Richtlinien festlegen, wer zum Zugriff auf was berechtigt ist. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Benutzerinformationen über die AWS Management Console, die AWS CLI oder die AWS -API abrufen.

### Identitätsbasierte Richtlinien

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 [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem AWS-Konto anfügen können.



Verwaltete Richtlinien umfassen von AWS verwaltete und von Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer eingebundenen Richtlinie wählen, finden Sie unter [Auswahl zwischen verwalteten und eingebundenen Richtlinien](#) im IAM-Benutzerhandbuch.

## Ressourcenbasierte Richtlinien

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](#). Prinzipale können Konten, Benutzer, Rollen, Verbundbenutzer oder AWS-Services umfassen.

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

## Zugriffssteuerungslisten (ACLs)

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3, AWS WAF und Amazon VPC sind Beispiele für Dienste, die ACLs unterstützen. Weitere Informationen zu ACLs finden Sie unter [Zugriffskontrollliste \(ACL\) – Übersicht](#) (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

## Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger häufig verwendete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- **Berechtigungsgrenzen** – Eine Berechtigungsgrenze ist ein erweitertes Feature, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer IAM-Entität (IAM-Benutzer oder -Rolle) erteilen kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den

Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Ein ausdrückliches Ablehnen in einer dieser Richtlinien setzt das Zulassen außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter [Berechtigungsgrenzen für IAM-Entitäten](#) im IAM-Benutzerhandbuch.

- **Service-Kontrollrichtlinien (SCPs)** – SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OE) in AWS Organizations angeben. AWS Organizations ist ein Dienst für die Gruppierung und zentrale Verwaltung mehrerer AWS-Konten Ihres Unternehmens. Wenn Sie innerhalb einer Organisation alle Features aktivieren, können Sie Service-Kontrollrichtlinien (SCPs) auf alle oder einzelne Ihrer Konten anwenden. SCPs schränken Berechtigungen für Entitäten in Mitgliedskonten einschließlich des jeweiligen Root-Benutzer des AWS-Kontos ein. Weitere Informationen zu Organisationen und SCPs finden Sie unter [Funktionsweise von SCPs](#) im AWS Organizations-Benutzerhandbuch.
- **Sitzungsrichtlinien** – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter [Sitzungsrichtlinien](#) im IAM-Benutzerhandbuch.

## Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen dazu, wie AWS die Zulässigkeit einer Anforderung ermittelt, wenn mehrere Richtlinientypen beteiligt sind, finden Sie unter [Logik für die Richtlinienauswertung](#) 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](#)
- [Beispiele für Amazon SWF SWF-Richtlinien](#)

## Richtlinienmaßnahmen für Amazon SWF

Unterstützt Richtlinienaktionen	Ja
---------------------------------	----

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

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie die zugehörige AWS-API-Operation. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keine passende API-Operation gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation 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 festlegen, wer zum Zugriff auf was berechtigt ist. Das bedeutet die Festlegung, welcher Prinzipal Aktionen für welche Ressourcen unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein `Resource` oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen [Amazon-Ressourcennamen \(ARN\)](#) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (\*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"

```

Eine Liste der Amazon SWF SWF-Ressourcentypen und ihrer 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 festlegen, wer zum Zugriff auf was berechtigt ist. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition block`) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. 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.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen AND-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungschlüssel angeben, wertet AWS die Bedingung mittels einer logischen OR-Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter [IAM-Richtlinienelemente: Variablen und Tags](#) im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungschlüssel und servicespezifische Bedingungschlüssel. Eine Liste aller globalen AWS-Bedingungschlüssel finden Sie unter [Globale AWS-Bedingungskontextschlüssel](#) im IAM-Benutzerhandbuch.

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

Unterstützt ACLs

Nein

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte 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 definiert werden. In AWS werden diese Attribute als Tags bezeichnet. Sie können Tags an IAM-Entitäten (Benutzer oder Rollen) und mehrere AWS-Ressourcen anfügen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen Sie ABAC-Richtlinien, um Operationen zuzulassen, wenn das Tag des Prinzipals mit dem Tag der Ressource übereinstimmt, auf die sie zugreifen möchten.

ABAC ist in Umgebungen hilfreich, die schnell wachsen, und unterstützt Sie in Situationen, in denen die Richtlinienverwaltung mühsam wird.

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 `aws:TagKeysBedingung` 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 [Was ist ABAC?](#) 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

Einige AWS-Services Featureieren nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, darunter welche AWS-Services mit temporären Anmeldeinformationen Featureieren, finden Sie unter [AWS-Services, die mit IAM Featureieren](#) im IAM-Benutzerhandbuch.

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen Methode als einem Benutzernamen und einem Passwort bei der AWS Management Console anmelden. Wenn

Sie beispielsweise über den Single Sign-On (SSO)-Link Ihres Unternehmens auf AWS zugreifen, erstellt dieser Prozess automatisch temporäre Anmeldeinformationen. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Wechseln von Rollen finden Sie unter [Wechseln zu einer Rolle \(Konsole\)](#) im IAM-Benutzerhandbuch.

Sie können mithilfe der AWS CLI- oder AWS-API manuell temporäre Anmeldeinformationen erstellen. Sie können dann diese temporären Anmeldeinformationen verwenden, um auf AWS zuzugreifen. AWS empfiehlt, dass Sie temporäre Anmeldeinformationen dynamisch generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter [Temporäre Sicherheitsanmeldeinformationen in IAM](#).

## Serviceübergreifende Prinzipalberechtigungen für Amazon SWF

Unterstützt Forward Access Sessions (FAS)	Ja
---	----

Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle zum Ausführen von Aktionen in AWS verwenden, gelten Sie als Prinzipal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service auslösen. FAS verwendet die Berechtigungen des Prinzipals, der einen AWS-Service aufruft, in Kombination mit der Anforderung an den AWS-Service, Anforderungen an nachgelagerte Services zu stellen. FAS-Anforderungen werden nur dann gestellt, wenn ein Dienst eine Anforderung erhält, die Interaktionen mit anderen AWS-Services oder Ressourcen erfordert, um abgeschlossen werden zu können. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen 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 Nein

Eine serviceverknüpfte Rolle ist eine Art von Servicerolle, die mit einem AWS-Service verknüpft ist. Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Serviceverknüpfte Rollen werden in Ihrem AWS-Konto angezeigt und gehören zum Service. Ein IAM-Administrator kann die Berechtigungen für serviceverbundene 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 [Erstellen von IAM-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. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet



ist. 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](#). Prinzipale können Konten, Benutzer, Rollen, Verbundbenutzer oder AWS-Services umfassen.

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource in unterschiedlichen AWS-Konten befinden, muss ein IAM-Administrator im vertrauenswürdigen Konto auch der Prinzipalentsität (Benutzer oder Rolle) die Berechtigung zum Zugriff auf die Ressource erteilen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich.

Weitere Informationen finden Sie unter [Wie sich IAM-Rollen von ressourcenbasierten Richtlinien unterscheiden](#) 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-Funktion	Amazon SWF SWF-Unterstützung
<a href="#">Identitätsbasierte Richtlinien</a>	Ja
<a href="#">Ressourcenbasierte Richtlinien</a>	Nein
<a href="#">Richtlinienaktionen</a>	Ja
<a href="#">Richtlinienressourcen</a>	Ja
<a href="#">Richtlinienbedingungsschlüssel (servicespezifisch)</a>	Ja
<a href="#">ACLs</a>	Nein
<a href="#">ABAC (Tags in Richtlinien)</a>	Teilweise
<a href="#">Temporäre Anmeldeinformationen</a>	Ja
<a href="#">Hauptberechtigungen</a>	Ja
<a href="#">Servicerollen</a>	Ja
<a href="#">Service-verknüpfte Rollen</a>	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. Sie können auch keine Aufgaben über die AWS Management Console, die AWS Command Line Interface (AWS CLI) oder die AWS-API ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

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

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

## Themen

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

## Bewährte Methoden 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. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Erste Schritte mit AWS-verwaltete Richtlinien und Umstellung auf Berechtigungen mit den geringsten Berechtigungen – Um Ihren Benutzern und Workloads Berechtigungen zu gewähren, verwenden Sie die AWS-verwaltete Richtlinien die Berechtigungen für viele allgemeine Anwendungsfälle gewähren. Sie sind in Ihrem AWS-Konto verfügbar. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie AWS-kundenverwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter [AWS-verwaltete Richtlinien](#) oder [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 Service-Aktionen zu gewähren, wenn diese durch ein bestimmtes AWS-Service, wie beispielsweise AWS CloudFormation, verwendet werden. 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 zum IAM Access Analyzer](#) im IAM-Benutzerhandbuch.
- Bedarf einer Multi-Faktor-Authentifizierung (MFA) – Wenn Sie ein Szenario haben, das IAM-Benutzer oder Root-Benutzer in Ihrem AWS-Konto erfordert, 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 [Konfigurieren eines MFA-geschützten API-Zugriffs](#) im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter [Bewährte Methoden 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.

Für Benutzer, die nur Aufrufe an die AWS CLI oder AWS-API durchführen, müssen Sie keine Mindestberechtigungen in der Konsole erteilen. 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 enthält Berechtigungen für die Ausführung dieser Aktion auf der Konsole oder für die programmgesteuerte Ausführung über die AWS CLI oder die AWS-API.

```
{
  "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 komplett 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 *DomainName* ist der Domainname.

## Action

(Erforderlich) Die Aktion, auf die sich die Anweisung bezieht. Sie verweisen darauf durch Verwendung des folgenden Formats: *serviceId:action*. Setzen Sie *ServiceID* für Amazon SWF 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](#).

## Beispiele für Amazon SWF SWF-Richtlinien

Ein Workflow besteht aus mehreren Akteuren — Aktivitäten, Entscheidern usw. Sie können den Zugriff für jeden Akteur steuern, indem Sie eine entsprechende IAM-Richtlinie anhängen. In diesem Abschnitt finden Sie einige Beispiele. Im folgenden Beispiel wird der einfachste Fall dargestellt:

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

```
} ]  
}
```

Wenn Sie diese Richtlinie einem Akteur anfügen, hat er vollständigen Zugriff auf alle Regionen des Kontos. Sie können Platzhalter verwenden, damit ein einzelner Wert mehrere Ressourcen, Aktionen oder Regionen repräsentiert.

- Der erste Platzhalter (\*) im Resource-Wert bedeutet, dass die Ressourcenberechtigungen auf alle Regionen zutreffen. 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](#).

In den folgenden Abschnitten finden Sie Beispielrichtlinien, die die Berechtigungen in einer differenzierteren Art und Weise erteilen.

### Domänenberechtigungen

Wenn Sie die Workflows einer Abteilung auf eine bestimmte Domäne beschränken möchten, können Sie folgendermaßen vorgehen:

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

Wenn Sie diese Richtlinie einem Akteur anfügen, kann er alle Aktionen aufrufen, aber nur für die Domäne department1.

Wenn Sie möchten, dass ein Akteur Zugriff auf mehr als eine Domäne hat, können Sie die Berechtigungen folgendermaßen separat für die einzelnen Domänen erteilen:

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

Wenn Sie diese Richtlinie an einen Akteur anhängen, kann dieser jede Amazon SWF SWF-Aktion in den `department2` Domänen `department1` und verwenden. Manchmal können Sie auch Platzhalter verwenden, um mehrere Domänen zu repräsentieren.

### API-Berechtigungen und -Einschränkungen

Mit dem `Action`-Element können Sie kontrollieren, welche Aktionen ein Akteur verwenden kann. Optional können Sie die zulässigen Parameterwerte einer Aktion mithilfe des `Condition`-Elements beschränken.

Wenn Sie den Akteur auf bestimmte Aktionen beschränken möchten, können Sie folgendermaßen vorgehen:

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

Wenn Sie diese Richtlinie einem Akteur anfügen, kann er `StartWorkflowExecution` aufrufen, um in der `department2`-Domäne Workflows zu starten. Er kann keine anderen Aktionen verwenden oder Workflows in anderen Domänen starten.

Indem Sie folgendermaßen einen oder mehrere `StartWorkflowExecution`-Parameterwerte beschränken, können Sie zusätzlich einschränken, welche Workflows ein Akteur starten kann:

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

Diese Richtlinie beschränkt die Parameter für die `StartWorkflowExecution`-Aktionen `name` und `version`. Wenn Sie diese Richtlinie einem Akteur anfügen, kann er nur `version2` des `workflow1` in der Domäne `department1` ausführen und beide Parameter müssen in der Anfrage 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:

```
{
  "Version": "2012-10-17",
  "Statement" : [ {
    "Effect" : "Allow",
    "Action" : "swf:StartWorkflowExecution",
    "Resource" : "arn:aws:swf:*:123456789012:/domain/some_domain",
    "Condition" : {
      "StringEqualsIfExists" : { "swf:taskList.name" : "task_list_name" }
    }
  } ]
}
```

Diese Richtlinie ermöglicht es einem Akteur, beim Start einer Workflow-Ausführung optional eine Aufgabenliste festzulegen.

Sie können für manche Aktionen eine Reihe von Tags beschränken. In diesem Fall besitzt jedes Tag einen eigenen Schlüssel. Sie verwenden `swf:tagList.member.0`, um das erste Tag in der Liste einzuschränken, `swf:tagList.member.1`, um das zweite Tag in der Liste einzuschränken usw., bis maximal 5 Tags. Allerdings müssen Sie bei der Beschränkung von Taglisten äußerst vorsichtig vorgehen. Das folgende Beispiel zeigt eine Richtlinie, die nicht empfohlen wird:

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

Diese Richtlinie ermöglicht es Ihnen, wahlweise `some_ok_tag` oder `another_ok_tag` festzulegen. Jedoch beschränkt diese Richtlinie nur das erste Element der Tag-Liste. Diese Liste könnte zusätzliche Elemente mit beliebigen Werten umfassen, die alle zugelassen würden, da diese Richtlinie keine Bedingungen auf `swf:tagList.member.1`, `swf:tagList.member.2` usw. anwendet.

Eine Möglichkeit, dieses Problem zu beheben, besteht darin, den Einsatz von Tag-Listen zu untersagen. Die folgende Richtlinie stellt sicher, dass nur `some_ok_tag` oder `another_ok_tag` zulässig sind, da die Liste nur ein Element besitzen darf.

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

```

    "Null" : { "swf:tagList.member.1" : "true" }
  }
} ]
}

```

## Pseudo-API-Berechtigungen und -Einschränkungen

Wenn Sie die für `RespondDecisionTaskCompleted` verfügbaren Entscheidungen beschränken möchten, müssen Sie dem Akteur zunächst die Berechtigung erteilen, `RespondDecisionTaskCompleted` aufzurufen. Anschließend können Sie Berechtigungen für die entsprechenden Pseudo-API-Mitglieder erteilen. Verwenden Sie dazu die gleiche Syntax wie für die reguläre API, wie nachfolgend beschrieben:

```

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

```

Wenn Sie diese Richtlinie einem Akteur anfügen, ermöglicht das erste Statement-Element dem Akteur, `RespondDecisionTaskCompleted` aufzurufen. Das zweite Element 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 in dieser `Condition` ermöglicht `RespondDecisionTaskCompleted` eine Aktivitätsaufgabe für die `SomeActivityType`-Aktivität zu planen und er muss diese Aufgabe auch 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.

## AWSverwaltete Richtlinie: `SimpleWorkflowFullAccess`

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

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

### Berechtigungsdetails

Diese Richtlinie umfasst die folgenden Berechtigungen.

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

## 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.

Beispielsweise wird die folgende Richtlinie für [ListOpenWorkflowExecutions](#) nicht empfohlen:

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

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

Das Amazon SWF-Servicemodell erlaubt nicht, dass die `tagFilter` Parameter `typeFilter` und in derselben `ListOpenWorkflowExecutions` Anfrage verwendet werden. Die Richtlinie erlaubt daher Aufrufe, die der Service ablehnt — indem er sie ausgibt `ValidationException` — als ungültige Anfrage.

## 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`.

### [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.

### [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 einschränken, die ein Tag mit dem Schlüssel `environment` und dem Wert `production` enthalten:

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

```
    "Effect": "Deny",
    "Action": "swf:*",
    "Resource": "arn:aws:swf:*:123456789012:/domain/*",
    "Condition": {
      "StringEquals": {"aws:ResourceTag/environment": "production"}
    }
  ]
}
```

Durch diese Richtlinie wird der Befehl Deny (Verweigern) bei Zugriffsversuchen auf Domänen ausgeführt, die als `environment/production` markiert wurden.

Weitere Informationen zum Markieren finden Sie unter:

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

## Amazon VPEndpunkte für Amazon SWF

### Note

AWS PrivateLinkSupport ist derzeit nur in den Regionen „AWSStreng geheim — Ost“, „AWSGeheim“ 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 Amazon Simple Workflow Service herstellen. Sie können diese Verbindung mit Ihren Amazon SWF-Workflows verwenden, ohne das öffentliche Internet zu nutzen.

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 zu VPCs finden Sie im [Amazon VPC-Benutzerhandbuch](#).

Um Ihre Amazon VPC mit Amazon SWF zu verbinden, müssen Sie zunächst einen Schnittstellen-VPC-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-Endpunkt in Ihrer VPC mithilfe von AWS Management Console, der AWS Command Line Interface (AWS CLI), einem AWS SDK, der Amazon SWF API oder erstellen AWS CloudFormation.

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 an, dass Amazon SWF der Service ist, mit dem Ihre VPC eine Verbindung einrichten soll. In der Amazon VPC-Konsole variieren die Servicenamen abhängig von der Amazon VPC-Konsole die Servicenamen abhängig von der Amazon AWS VPC-Console abhängig In der Region AWS Streng geheim — Ost lautet der Servicenamen für Amazon SWF beispielsweise `com.amazonaws.us-iso-east-1.swf`.

Informationen zum Erstellen und Konfigurieren eines Endpunkts mit AWS CloudFormation finden Sie in der [AWS::EC2::VPCEndpoint](#)-Ressource im AWS CloudFormation-Benutzerhandbuch.

## Amazon-VPC-Endpunktrichtlinie für Amazon

Um den Konnektivitätszugriff auf Amazon SWF zu kontrollieren, können Sie eine AWS Identity and Access Management (IAM) -Endpunktrichtlinie anhängen, während Sie einen Amazon VPC-Endpunkt erstellen. Sie können komplexe IAM-Regeln erstellen, indem Sie mehrere Endpunktrichtlinien anhängen. Weitere Informationen finden Sie unter:

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

## Amazon Virtual Private Cloud Cloud-Endpunktrichtlinie für Amazon SWF- Endpunktrichtlinie für Amazon

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

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

Das folgende Beispiel zeigt eine Amazon VPC-Endpunktrichtlinie, die alle Amazon SWF-Operationen auf einer einzelnen Domain für eine bestimmte IAM-Rolle zulässt.

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

- Weitere Informationen zum Erstellen von Endpunktrichtlinien finden Sie unter [Steuerung des Zugriffs auf Services mit VPC-Endpunkten für das Erstellen von Endpunktrichtlinien für Endpunktrichtlinien](#)
- Informationen dazu, wie Sie IAM zum Steuern des Zugriffs auf Ihre AWS Amazon SWF-Ressourcen verwenden können, finden Sie unter [Identity and Access Management in Amazon Simple Workflow Service](#).

# Beheben von Identitäts- und Zugriffsfehlern bei Amazon Simple Workflow Service

Verwenden Sie die folgenden Informationen, um häufige Probleme zu diagnostizieren und zu beheben, die beim Arbeiten mit Amazon SWF und IAM auftreten könnten.

## Themen

- [Ich bin nicht autorisiert, eine Aktion in Amazon SWF auszuführen](#)
- [Ich bin nicht zur Ausführung von iam:PassRole](#)
- [Ich möchte Personen außerhalb meines ZugriffAWS-Konto auf meine Amazon SWF SWF-Ressourcen erteilen](#)

## Ich bin nicht autorisiert, eine Aktion in Amazon SWF auszuführen

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.

Wenden Sie sich an Ihren AWS-Administrator, falls Sie weitere Unterstützung benötigen. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen odzur Verfügung gestellt.

## Ich bin nicht zur Ausführung von iam:PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zur Ausführung der iam:PassRole Aktion autorisiert sind, müssen Ihre Richtlinien aktualisiert werden, um eine Rolle an Amazon SWF übergeben zu können.

Einige AWS-Services erlauben die Übergabe einer vorhandenen Rolle an diesen Service, sodass keine neue Servicerolle oder serviceverknüpfte Rolle erstellt werden muss. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Service.

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 Service.

```
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.

Wenden Sie sich an Ihren AWS-Administrator, falls Sie weitere Unterstützung benötigen. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen odzur Verfügung gestellt.

## Ich möchte Personen außerhalb meines ZugriffAWS-Konto auf meine Amazon SWF SWF-Ressourcen erteilen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Services, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen dazu, ob Amazon SWF diese Funktionen unterstützt, finden Sie unter [So funktioniert Amazon Simple Workflow Service mit IAM](#).
- Informationen zum Gewähren des Zugriffs auf Ihre Ressourcen für alle Ihre AWS-Konten finden Sie unter [Gewähren des Zugriffs für einen IAM-Benutzer in einem anderen Ihrer AWS-Konto](#) im IAM-Benutzerhandbuch.
- Informationen dazu, wie Sie AWS-Konten-Drittanbieter Zugriff auf Ihre Ressourcen bereitstellen, finden Sie unter [Gewähren des Zugriffs auf AWS-Konten von externen Benutzern](#) im IAM-Benutzerhandbuch.
- 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 kontenübergreifenden Zugriff finden Sie unter [So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien](#) im IAM-Benutzerhandbuch.

## Protokollieren und Überwachen

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

### Themen

- [Amazon SWF-Metriken für CloudWatch](#)
- [Anzeigen von Amazon SWF SWF-Metriken für CloudWatch mithilfe derAWS Management Console](#)
- [Aufzeichnen von API-Aufrufen mit AWS CloudTrail](#)
- [Änderungen des Ausführungsstatus von EventBridge für Amazon SWF](#)
- [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 Management Console. Weitere Informationen finden Sie unter [Anzeigen von Amazon SWF SWF-Metriken für CloudWatch mithilfe derAWS Management Console](#).

### 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 SWF-Zeitüberschreitungstypen](#).

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 Funktionsweise von 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 Alarmer 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 eingeht, 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	Beschreibung
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>

Metrik	Beschreibung
	<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>
StartedDecisionTasksTimedOutOnClose	<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>
WorkflowStartToCloseTime	<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>

Metrik	Beschreibung
<code>WorkflowsCanceled</code>	Anzahl der abgebrochenen Workflows.  CloudWatch Einheiten: Count  Maße: Domain, WorkflowTypeName, WorkflowTypeVersion  Gültige Statistiken: Sum
<code>WorkflowsCompleted</code>	Anzahl der abgeschlossenen Workflows.  CloudWatch Einheiten: Count  Maße: Domain, WorkflowTypeName, WorkflowTypeVersion  Gültige Statistiken: Sum
<code>WorkflowsContinuedAsNew</code>	Anzahl der Workflows, die als neue Workflows fortgesetzt worden sind.  CloudWatch Einheiten: Count  Maße: Domain, WorkflowTypeName, WorkflowTypeVersion  Gültige Statistiken: Sum
<code>WorkflowsFailed</code>	Anzahl der fehlgeschlagenen Workflows  CloudWatch Einheiten: Count  Maße: Domain, WorkflowTypeName, WorkflowTypeVersion  Gültige Statistiken: Sum

Metrik	Beschreibung
<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>
<code>ActivityTaskScheduleToCloseTime</code>	<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>
<code>ActivityTaskScheduleToStartTime</code>	<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>

Metrik	Beschreibung
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>
ActivityTasksCompleted	<p>Anzahl der abgeschlossenen Aktivitäts-Tasks.</p> <p>CloudWatch Einheiten: Count</p> <p>Maße: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Gültige Statistiken: Sum</p>
ActivityTasksFailed	<p>Anzahl der fehlgeschlagenen Aktivitäts-Tasks.</p> <p>CloudWatch Einheiten: Count</p> <p>Maße: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Gültige Statistiken: Sum</p>

Metrik	Beschreibung
<code>ScheduledActivityTasksTimedOutOnClose</code>	<p>Anzahl der Aktivitäts-Tasks, die geplant 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>
<code>ScheduledActivityTasksTimedOutOnStart</code>	<p>Anzahl der Aktivitäts-Tasks, die geplant wurden, aber das Zeitlimit beim Starten überschritten haben.</p> <p>CloudWatch Einheiten: Count</p> <p>Maße: Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Gültige Statistiken: Sum</p>
<code>StartedActivityTasksTimedOutOnClose</code>	<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>

Metrik	Beschreibung
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>
ConsumedCapacity	<p>Anzahl der Anforderungen pro Sekunde</p> <p>CloudWatch Einheiten: Count</p> <p>Maße: APIName, DecisionName</p> <p>Gültige Statistiken: Sum</p>
ConsumedLimit	<p>Die Menge des generellen Grenzwerts, der verbraucht wurde.</p> <p>Maße: GeneralLimitType</p>

Metrik	Beschreibung
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	Beschreibung
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
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.



Dimension	Beschreibung
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
```

## Anzeigen von Amazon SWF SWF-Metriken für CloudWatch mithilfe der AWS Management Console

Amazon CloudWatch bietet eine Vielzahl anzeigbarer Metriken für Amazon SWF SWF-Workflows und -Aktivitäten. Sie können die Metriken für Ihre Amazon SWF SWF-Workflow-Ausführungen mithilfe der [AWS Management Console](#) aus. Sie müssen bei der Konsole angemeldet sein, um fortzufahren.

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

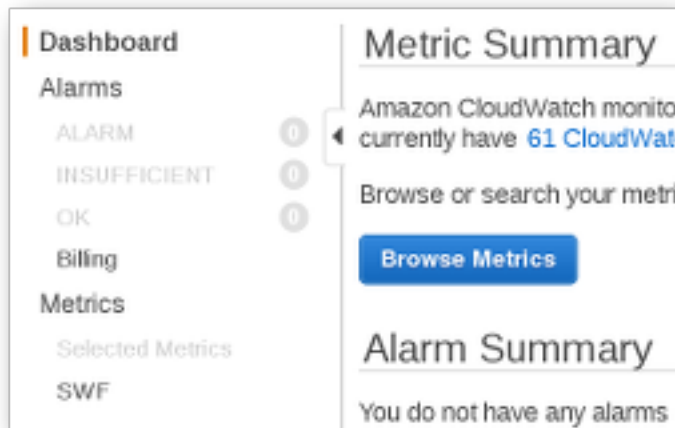
### Themen

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

## Anzeigen von -Metriken

So zeigen Sie Ihre Metriken für Amazon SWF an

1. Melden Sie sich bei AWS Management Console an 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 aus.

SWF > Workflow Type Metrics			
Domain	WorkflowTypeName	WorkflowTypeVersion	Metric Name
HelloWorld	HelloWorldWorkflow.hello_workflow	1.0	WorkflowStartToCloseTime
HelloWorld	HelloWorldWorkflow.hello_workflow	1.0	WorkflowsCompleted
SWF > Activity Type Metrics			
Domain	ActivityTypeName	ActivityTypeVersion	Metric Name
Booking	BookingActivity.reserve_airline	1.0	ActivityTaskScheduleToStartTime
Booking	BookingActivity.reserve_airline	1.0	ActivityTaskStartToCloseTime

### Note

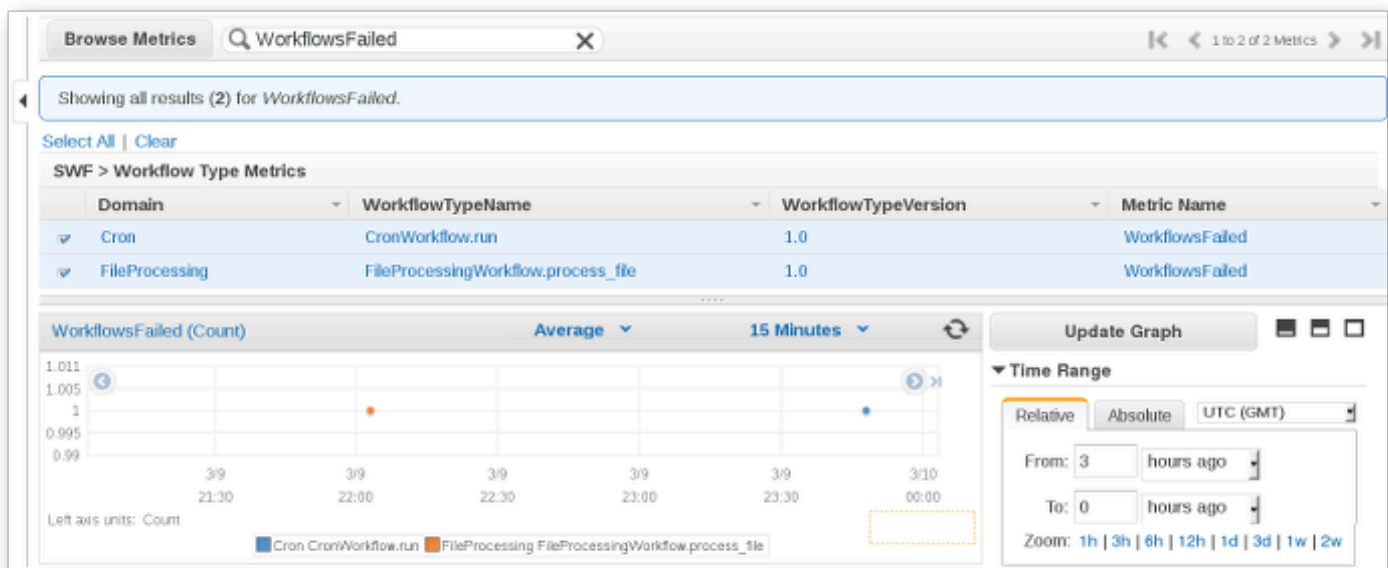
Es kann sein, dass Ihnen zuerst nur die Metriken vom Typ Workflow angezeigt werden. Die Metriken vom Typ Aktivität erscheinen in der gleichen Ansicht, ggf. müssen Sie aber nach unten scrollen, um sie zu sehen.

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

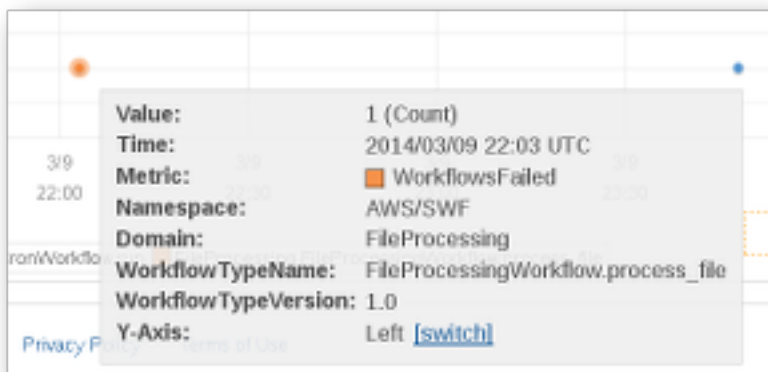
Über die interaktiven Überschriften über jeder Spalte in der Liste können Sie Ihre Metriken anhand einer der bereitgestellten Dimensionen sortieren. Für Workflows lauten die Dimensionen Domain, WorkflowTypeName, WorkflowTypeVersion und Metric Name. Für Aktivitäten lauten die Dimensionen Domain, ActivityTypeName,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

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 Hinweise zur Arbeit mit CloudWatch-Metriken finden Sie unter [Anzeigen, Graphieren und Veröffentlichen von Metriken](#) im Amazon CloudWatch-Benutzerhandbuch.

## Festlegen von Alarmen

Mithilfe von CloudWatch-Alarmen können Sie Aktionen durchführen, z. B. bei Erreichen eines Alarmschwellenwerts Benachrichtigungen versenden. 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:

for:  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 Hinweise zum Einrichten und Verwenden von CloudWatch-Alarmen finden Sie unter [Erstellen von Amazon CloudWatch-Alarmen](#) im Amazon CloudWatch-Benutzerhandbuch.

## Aufzeichnen von API-Aufrufen mit AWS CloudTrail

Amazon Simple Workflow Service ist integriert, einem Service [AWS CloudTrail](#), der die Aktionen eines Benutzers, einer Rolle oder eines aufzeichnet AWS-Service. CloudTrail erfasst alle API-Aufrufe für Amazon SWF als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der Amazon SWF-Konsole und Codeaufrufe der Amazon SWF-API-Operationen. Anhand der von CloudTrail gesammelten Informationen können Sie die an Amazon SWF gestellte Anfrage, die IP-Adresse, von der die Anfrage gestellt wurde, den Zeitpunkt der Anfrage und zusätzliche Details bestimmen.

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.
- Ob die Anfrage gestellt wurde... im Namen eines IAM-Identity-Center-Benutzers.
- 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 automatisch Zugriff auf den CloudTrail Ereignisverlauf haben. Der CloudTrail Ereignisverlauf bietet eine anzeigbare, durchsuchbare, herunterladbare und unveränderliche Aufzeichnung der aufgezeichneten Verwaltungsereignisse der letzten 90 Tage in einem AWS-Region. Weitere Informationen finden Sie unter [Arbeiten mit dem CloudTrail Ereignisverlauf](#) im AWS CloudTrail -Benutzerhandbuch. Für die Anzeige des Ereignisverlaufs fallen keine CloudTrail Gebühren an.

Erstellen Sie für eine fortlaufende Aufzeichnung der Ereignisse in den AWS-Konto letzten 90 Tagen einen Trail oder einen [CloudTrail Lake](#)-Ereignisdatenspeicher.

### CloudTrail Trails

Ein Trail ermöglicht CloudTrail die Bereitstellung von Protokolldateien an einen Amazon S3-Bucket. Alle Trails, die mit der erstellt wurden, AWS Management Console sind multiregional. Sie können einen Trail für eine oder mehrere Regionen erstellen, indem Sie die verwenden AWS CLI. Das Erstellen eines multiregionalen Trails wird empfohlen, da Sie Aktivitäten in allen AWS-

Regionen in Ihrem Konto erfassen. Wenn Sie einen Trail für eine einzelne Region erstellen, können Sie nur die Ereignisse anzeigen, die im des Trails protokolliert wurden AWS-Region. 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 von aus an Ihren Amazon S3-Bucket senden, CloudTrail indem Sie einen Trail erstellen. Es fallen jedoch Amazon S3-Speichergebühren an. Weitere Informationen zu CloudTrail Preisen finden Sie unter [-AWS CloudTrail Preise](#). Informationen zu Amazon-S3-Preisen finden Sie unter [Amazon S3-Preise](#).

## CloudTrail Lake-Ereignisdatenspeicher

Mit CloudTrail Lake können Sie SQL-basierte Abfragen für Ihre Ereignisse ausführen. CloudTrail Lake konvertiert vorhandene 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 Lake-Ereignisdatenspeicher und Abfragen 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 zu CloudTrail Preisen 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. Standardmäßig protokolliert CloudTrail keine Datenereignisse. Der CloudTrail Ereignisverlauf zeichnet keine Datenereignisse auf.

Für Datenereignisse werden zusätzliche Gebühren fällig. Weitere Informationen zu CloudTrail Preisen finden Sie unter [-AWS CloudTrail Preise](#).

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

In der folgenden Tabelle sind die Amazon SWF-Ressourcentypen aufgeführt, für die Sie Datenereignisse protokollieren können. In der Spalte Datenereignistyp wird der Wert angezeigt, den Sie aus der Liste Datenereignistyp in der CloudTrail Konsole auswählen können. Die Spalte `resources.type value` zeigt den `resources.type` Wert an, den Sie bei der Konfiguration erweiterter Ereignisselektoren mit der AWS CLI oder CloudTrail APIs angeben würden. Die Spalte Daten-APIs, die in protokolliert CloudTrail wurden, zeigt die API-Aufrufe an, die CloudTrail für den -Ressourcentyp protokolliert wurden.

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

Datenereignistyp	<code>resources.type</code> -Wert	Daten-APIs, die bei protokolliert wurden CloudTrail
SWF-Domain	<code>AWS::SWF::Domain</code>	Workflow-Ereignisse <ul style="list-style-type: none"> <li>• <a href="#">CountClosedWorkflowExecutions</a></li> <li>• <a href="#">CountOpenWorkflowExecutions</a></li> <li>• <a href="#">DescribeWorkflowExecution</a></li> <li>• <a href="#">ListClosedWorkflowExecutions</a></li> <li>• <a href="#">ListOpenWorkflowExecutions</a></li> <li>• <a href="#">GetWorkflowExecutionHistory</a></li> <li>• <a href="#">RequestCancelWorkflowExecution</a></li> </ul>



Datenereignistyp	resources.type-Wert	Daten-APIs, die bei protokolliert wurden CloudTrail
		<ul style="list-style-type: none"> <li>• <a href="#">SignalWorkflowExecution</a></li> <li>• <a href="#">StartWorkflowExecution</a></li> <li>• <a href="#">TerminateWorkflowExecution</a></li> </ul> <p>Aufgabenereignisse</p> <ul style="list-style-type: none"> <li>• <a href="#">CountPendingActivityTasks</a></li> <li>• <a href="#">PollForDecisionTask</a></li> <li>• <a href="#">PollForActivityTask</a></li> <li>• <a href="#">RecordActivityTaskHeartbeat</a></li> <li>• <a href="#">RespondActivityTaskCanceled</a></li> <li>• <a href="#">RespondActivityTaskCompleted</a></li> <li>• <a href="#">RespondActivityTaskFailed</a></li> <li>• <a href="#">RespondDecisionTaskCompleted</a></li> </ul> <p>Entscheidungsereignisse</p> <ul style="list-style-type: none"> <li>• <a href="#">CancelTimer</a></li> <li>• <a href="#">CancelWorkflowExecution</a></li> <li>• <a href="#">CompleteWorkflowExecution</a></li> <li>• <a href="#">ContinueAsNewWorkflowExecution</a></li> <li>• <a href="#">FailWorkflowExecution</a></li> <li>• <a href="#">RecordMarker</a></li> <li>• <a href="#">RequestCancelActivityTask</a></li> </ul>

Datenereignistyp	resources.type-Wert	Daten-APIs, die bei protokolliert wurden CloudTrail
		<ul style="list-style-type: none"> <li>• <a href="#">RequestCancelExternalWorkflowExecution</a></li> <li>• <a href="#">ScheduleActivityTask</a></li> <li>• <a href="#">ScheduleLambdaFunction</a></li> <li>• <a href="#">SignalExternalWorkflowExecution</a></li> <li>• <a href="#">StartChildWorkflowExecution</a></li> <li>• <a href="#">StartTimer</a></li> </ul>

### CloudTrail -Ereignisse und RespondDecisionTaskCompleted

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

## Verwaltungsereignisse in CloudTrail

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

Amazon Simple Workflow Service protokolliert die folgenden Operationen auf Steuerebene in CloudTrail als Verwaltungsereignisse.

### Domain-Ereignisse

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

- [UndeprecateDomain](#)

### Aktivitätsereignisse

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

### WorkflowType Ereignisse

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

### Tag-Ereignisse

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

### Beispielereignis

Ein Ereignis stellt eine einzelne Anfrage aus einer beliebigen Quelle dar und enthält Informationen über die angeforderte API-Operation, das Datum und die Uhrzeit der Operation, die Anfrageparameter usw. . CloudTrail Protokolldateien sind kein geordnetes Stacktrace der öffentlichen API-Aufrufe, sodass Ereignisse nicht in einer bestimmten Reihenfolge angezeigt werden.

Das folgende Beispiel zeigt ein CloudTrail Ereignis, das die `CountClosedWorkflowExecutions` Operation 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-EXAMPLEbbbbbb",
"readOnly": true,
"resources": [
  {
    "accountId": "111122223333",
    "type": "AWS::SWF::Domain",
    "ARN": "arn:aws:swf:us-east-1:111122223333:/domain/nsg-domain"
  }
]
```

```
],
  "eventType": "AwsApiCall",
  "managementEvent": false,
  "recipientAccountId": "111122223333",
  "eventCategory": "Data",
  "tlsDetails": {
    "clientProvidedHostHeader": "swf.example.amazondomains.com"
  }
}
```

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

## Änderungen des Ausführungsstatus von EventBridge für Amazon SWF

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

Sie können den vollständigen Inhalt eines Ereignisses zur Änderung des Ausführungsstatus mit [DescribeWorkflowExecution](#) aus.

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

### EventBridge-Ereignisse

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

- `eventId`: Die Ereignis-ID, die von `GetWorkflowExecutionHistory` angezeigt wird.
- `workflowExecutionDetail`: Der Status des Workflows, in dem das Ereignis emittiert wurde.
- `eventType`: Der Ereignistyp des Verlaufs, einer der folgenden Optionen:
  - `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~~

## Amazon SWF SWF-Ereignisbeispiele

Im Folgenden finden Sie Beispiele für Amazon SWF Ereignissen an EventBridge:

### Themen

- [Ausführung gestartet](#)
- [Ausführung abgeschlossen](#)
- [Ausführung fehlgeschlagen](#)
- [Zeitlimit für Ausführung überschlagene](#)
- [Ausführung wurde beendet](#)

In jedem Fall bietet der Abschnitt `detail` in den Ereignisdaten die gleichen Informationen wie die [DescribeWorkflowExecution](#) API. Die `executionStatus` gibt den Status der Ausführung zu dem Zeitpunkt an, zu dem das Ereignis gesendet wurde, entweder `OPEN` oder `CLOSED`.

### Ausführung 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": "4444444444444444"
    },
    "taskPriority": null,
    "childPolicy": "ABANDON",
    "lambdaRole": "arn:aws:iam::4444444444444444: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": "4444455556666",
  "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,
    }
  }
}
```

```
}
```

## Ausführung 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,
}
}
}

```

## Zeitlimit für Ausführung überschlagene

```

{
  "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,
}
}
}

```

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, einschließlich E-Mail-, [AWS Chatbot](#)-Chat- 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.

## Compliance-Validierung für Amazon Simple Workflow Service

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

Eine Liste der - AWS Services, die in den Geltungsbereich bestimmter Compliance-Programme fallen, finden Sie unter [AWS -Services im Geltungsbereich nach Compliance-Programm](#) Allgemeine Informationen finden Sie unter [AWS Compliance-Programme](#)

Sie können Auditberichte von Drittanbietern mit herunterladen AWS Artifact. Weitere Informationen finden Sie unter [Herunterladen von Berichten in AWS Artifact](#) .

Ihre Compliance-Verantwortung bei der Verwendung 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 Unterstützung der Compliance bereit:

- [Kurzanleitungen für Sicherheit und Compliance](#) – In diesen Bereitstellungsleitfäden werden architektonische Überlegungen erörtert und Schritte für die Bereitstellung von Sicherheits- und Compliance-orientierten Basisumgebungen in beschrieben AWS.
- [Whitepaper zur Erstellung einer Architektur mit HIPAA-konformer Sicherheit und Compliance](#) – In diesem Whitepaper wird beschrieben, wie Unternehmen mithilfe AWS von HIPAA-konforme Anwendungen erstellen können.
- [AWS Compliance-Ressourcen](#) – Diese Sammlung von Arbeitsmappen und Leitfäden könnte für Ihre Branche und Ihren Standort gelten.

- [Bewertung von Ressourcen mit Regeln](#) im -AWS Config Entwicklerhandbuch – Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- [AWS Security Hub](#) – Dieser AWS Service bietet einen umfassenden Überblick über Ihren Sicherheitsstatus in AWS , mit dem Sie Ihre Compliance mit den Sicherheitsstandards und bewährten Methoden der Branche überprüfen können.

## Amazon Simple Workflow Service

Im Zentrum der globalen AWS Infrastruktur stehen die AWS-Regionen und Availability Zones (Verfügbarkeitszonen, AZs). AWS -Regionen stellen mehrere physisch getrennte und isolierte Availability Zones bereit, die über hoch redundante Netzwerke mit niedriger Latenz und hohen Durchsätzen verbunden sind. With Availability Zones, you can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

Weitere Informationen über AWS Regionen und Availability Zones finden Sie unter [AWS Globale Infrastruktur](#).

Zusätzlich zu demAWSAmazon SWF stellt verschiedene Funktionen bereit, um Ihren Anforderungen an Ausfallsicherheit und Datensicherung gerecht zu werden.

## Infrastruktursicherheit in Amazon Simple Workflow Service

Als verwalteter Service ist Amazon Simple Workflow Service durchAWSglobale Verfahren zur Gewährleistung der Netzwerksicherheit, die in der[Amazon Web Services: Übersicht über Sicherheitsprozesse](#)Whitepaper.

Du benutztAWSveröffentlichte API-Aufrufe, um über das Netzwerk auf Amazon SWF zuzugreifen. Kunden müssen Transport Layer Security (TLS) 1.0 oder neuer unterstützen. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Die meisten modernen Systemen wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit [AWS](#)

[Security Token Service](#) (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Sie können diese API-Vorgänge von jedem Netzwerkstandort aus aufrufen, aber Amazon SWF unterstützt ressourcenbasierte Zugriffsrichtlinien, die Einschränkungen auf Basis der Quell-IP-Adresse enthalten können. Sie können auch Amazon SWF SWF-Richtlinien verwenden, um den Zugriff von bestimmten Amazon Virtual Private Cloud (Amazon VPC) -Endpunkten oder bestimmten VPCs zu steuern. Tatsächlich wird der Netzwerkzugriff hierdurch auf eine bestimmte Amazon SWF-Ressource eingeschränkt, sodass er ausschließlich über eine bestimmte VPC innerhalb der AWS-Netzwerk.

## Konfigurations- und Schwachstellenanalyse in Amazon Simple Workflow Service

Konfiguration und IT-Steuererelemente unterliegen der übergreifenden Verantwortlichkeit von AWS und Ihnen, unserem Kunden. Weitere Informationen finden Sie unter [AWS Modell der übergreifenden Verantwortlichkeit](#).



# Verwendung der AWS CLI mit Amazon Simple Workflow Service

Auf viele der Funktionen von Amazon Simple Workflow Service kann über die AWS CLI zugegriffen werden. Die AWS CLI bietet eine Alternative zur Verwendung von Amazon SWF mit der AWS Management Console oder in manchen Fällen zur Programmierung mit der Amazon SWF SDK-SWF-API und dem Amazon SWF Framework.

Sie können die AWS CLI z. B. verwenden, 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 inAWS CLISie können Workflow-Ausführungen starten und verwalten, Abfragen zu Aktivitätsaufgaben starten, Aufgaben-Heartbeats aufzeichnen usw.! Eine vollständige Liste der Amazon SWF SWF-Befehle mit Beschreibungen der verfügbaren Argumente und Beispiele, die ihre Verwendung zeigen, finden Sie unter [Amazon SWF-Befehle](#) imAWS CLIBefehlsreferenz aus.

DieAWS CLI-Befehle richten sich stark nach der Amazon SWF SWF-API, sodass Sie dieAWS CLlum mehr über die zugrunde liegende Amazon SWF SWF-API zu erfahren. Sie können auch Ihre vorhandenen API-Kenntnisse zum Prototyping von Code oder zur Durchführung von Amazon SWF SWF-Aktionen über die Befehlszeile nutzen.

Weitere Informationen zu finden Sie imAWS CLI, finden Sie im [AWS Command Line Interface-Benutzerhandbuch](#) aus.

# Arbeiten mit Amazon SWF-APIs

Zusätzlich zur Verwendung der AWS SDKs, die in [beschrieben sind](#) [Entwickeln mit AWS SDKs](#), können Sie 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 Senden von HTTP-Anforderungen für Amazon SWF finden Sie unter [Erstellen von HTTP-Anforderungen an Amazon SWF](#).

Dieser Abschnitt enthält grundlegende Informationen zur Verwendung der HTTP-API zur Entwicklung Ihrer Workflows mit Amazon SWF . Erweiterte Funktionen wie die Verwendung von Timern, die Protokollierung mit CloudTrail und das Markieren Ihrer Workflows finden Sie im Abschnitt [Funktionsweise von Amazon SWF](#).

## Themen

- [Erstellen von HTTP-Anforderungen an Amazon SWF](#)
- [Liste der Amazon-SWF-Aktionen nach Kategorie](#)
- [Erstellen eines einfachen Workflows in Amazon SWF](#)
- [Registrieren einer Domäne bei Amazon SWF](#)
- [Einrichten von Timeout-Werten in Amazon SWF](#)
- [Registrieren eines Workflow-Typs bei Amazon SWF](#)
- [Registrieren eines Aktivitätstyps bei Amazon SWF](#)
- [AWS Lambda-Aufgaben](#)
- [Entwickeln eines Aktivitäts-Workers in Amazon SWF](#)
- [Entwickeln von Entscheidern in Amazon SWF](#)
- [Starten von Workflow-Ausführungen mit Amazon SWF](#)
- [Einrichten der Aufgabenprioritäten](#)
- [Fehlerbehandlung in Amazon SWF](#)

## Erstellen von HTTP-Anforderungen an Amazon SWF

Wenn Sie keinen der AWS SDKs mit der POST-Anforderungsmethode verwenden, können Sie Amazon Simple Workflow Service (Amazon SWF-Vorgänge) über HTTP 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-Anforderung:

- `host` Der Amazon SWF SWF-Endpunkt.
- `x-amz-date` Sie müssen den Zeitstempel entweder im HTTP `Date`-Header oder der `AWS x-amz-date header` (Einige HTTP-Client-Bibliotheken lassen den `Date`-Header). 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 Anforderung verwendet wird (derzeit ist dieser Wert für Amazon SWF immer `AWS3`) enthalten.

`AWSAccessKeyId`— Ihre AWS Zugriffsschlüssel-ID.

`Algorithm`— Der Algorithmus, der verwendet wird, um den HMAC-SHA-Wert des Zeichenfolgen-zu-Signs zu erzeugen, z. `HmacSHA256` oder `HmacSHA1` aus.

`Signature`— Base64 (Algorithmus (StringToSign, SigningKey))). Details dazu finden Sie unter [Berechnen der HMAC-SHA-Signatur für Amazon SWF](#).

`SignedHeaders`— (Optional) Falls vorhanden, muss eine Liste aller HTTP-Header enthalten, die in der Berechnung der kanonisierten `HttpHeaders` verwendet werden. Die Listeneinträge müssen durch ein einzelnes Semikolon (;) (ASCII-Zeichen 59) getrennt werden.

- `x-amz-target`— Der Zieldienst der Anforderung und der Vorgang für die Daten, in folgendem 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 mit angeben. `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. Im Folgenden wird beispielsweise eine Anforderung angezeigt, alle Workflow-Ausführungen aufzulisten, die zwischen zwei angegebenen Zeitpunkten begannen — unter Verwendung der Unix-Zeitnotation.

```
{
  "domain": "867530901",
  "startTimeFilter":
  {
    "oldestDate": 1325376070,
    "latestDate": 1356998399
  },
  "tagFilter":
  {
    "tag": "music purchase"
  }
}
```

## Beispiel für eine Amazon SWF SWF-JSON-Anforderung und -Antwort

Das folgende Beispiel zeigt eine Anforderung an Amazon SWF, eine Beschreibung der zuvor erstellten Domäne zu erhalten. Anschließend wird die Amazon SWF SWF-Antwort von veranschaulicht.

## 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"}
```

## Amazon SWF SWF-Antwort

```
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 auf das Serialisieren von Nullwerten für Anforderungen festgelegt ist, werden diese von Amazon SWF von ignoriert.

## Berechnen der HMAC-SHA-Signatur für Amazon SWF

Jede Anforderung an Amazon SWF muss authentifiziert werden. Die AWS SDKs signieren Ihre Anfragen und verwalten Ihre auf Token basierende Authentifizierung automatisch. 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](#). Für den AWS SDK for Java Implementierung von AWS Version 3 signieren, siehe [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 SHA256 verwenden, um Ihre Anforderungen 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 eine zu signierende Zeichenfolge, die die folgenden Elemente enthält:
    - Linie1: Die HTTP-Methode (POST), gefolgt von einem Zeilenumbruch.
    - Linie2: Der Anfrage-URI (/), gefolgt von einem Zeilenumbruch.
    - Linie3: Eine leere Zeichenfolge, gefolgt von einem Zeilenumbruch.

#### Note

Normalerweise erscheint hier die Abfragezeichenfolge, aber Amazon SWF verwendet keine solche Zeichenfolge.

- LINES4–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. Berechnen Sie den SHA256- oder SHA1-Digest der zu signierenden Zeichenfolge. Verwenden Sie die gleiche SHA-Methode für den gesamten Prozess.
  4. Berechnen und nehmen Sie eine Base64-Codierung von HMAC-SHA entweder mit einem SHA256 oder einem SHA1-Digest (abhängig von der verwendeten Methode) des Ergebniswerts

aus dem vorherigen Schritt und dem temporären geheimen Zugriffsschlüssel des AWS Security Token Service mit [GetSessionToken](#) API-Aktion.

 Note

Amazon SWF erwartet ein Gleichheitszeichen (=) am Ende des HMAC-SHA-Werts mit Base64-Codierung. Wenn Ihre Base64-Codierung kein angefügtes Gleichheitszeichen umfasst, fügen Sie ein solches Zeichen am Ende des Werts an.

Weitere Informationen zum Verwenden von temporären Sicherheitsanmeldeinformationen mit Amazon SWF und anderen AWS Dienstleistungen, siehe [AWS Services, die mit IAM funktionieren](#) im IAM User Guide aus.

5. Platzieren Sie den Ergebniswert als Wert für `SignatureName` im `x-amzn-authorizationHeader` der HTTP-Anfrage an Amazon SWF.
6. Amazon SWF überprüft die Anforderung und führt den angegebenen Vorgang durch.

## Liste der Amazon-SWF-Aktionen nach Kategorie

In diesem Abschnitt werden die Referenzthemen für Amazon-SWF-Aktionen in der Amazon-SWF-Anwendungsprogrammierschnittstelle (API) aufgeführt. Diese werden nach der funktionalen Kategorie angegeben.

Für eine alphabetisch sortierte Liste der -Aktionen finden Sie in der [Amazon Simple Workflow Service API Referenz](#) aus.

### 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 zwar administrative Aufgaben in der Amazon-SWF-Konsole ausführen. Mit den Aktionen in diesem Abschnitt können Sie aber Funktionen automatisieren oder eigene Tools zur Verwaltung erstellen.

### Aktivitätsverwaltung

- [RegisterActivityType](#)
- [DeprecateActivityType](#)

### Workflow-Verwaltung

- [RegisterWorkflowType](#)
- [DeprecateWorkflowType](#)

### Domänenverwaltung

Mit diesen Aktionen können Sie Amazon-SWF-Domänen registrieren und als veraltet eingestufen.

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

Weitere Informationen und Beispiele zu diesen Domänenverwaltungsaktionen finden Sie unter [Registrieren einer Domäne bei Amazon SWF](#).

### Workflow-Ausführungsverwaltung

- [RequestCancelWorkflowExecution](#)
- [TerminateWorkflowExecution](#)

### Sichtbarkeitsaktionen

Sie können zwar Sichtbarkeitsaktionen in der Amazon-SWF-Konsole ausführen. Mit den Aktionen in diesem Abschnitt können Sie aber eine eigene Konsole oder Tools zur Verwaltung erstellen.

## Aktivitätssichtbarkeit

- [ListActivityTypes](#)
- [DescribeActivityType](#)

## Workflow-Sichtbarkeit

- [ListWorkflowTypes](#)
- [DescribeWorkflowType](#)

## Workflow-Ausführungssichtbarkeit

- [DescribeWorkflowExecution](#)
- [ListOpenWorkflowExecutions](#)
- [ListClosedWorkflowExecutions](#)
- [CountOpenWorkflowExecutions](#)
- [CountClosedWorkflowExecutions](#)
- [GetWorkflowExecutionHistory](#)

## Domänensichtbarkeit

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

## Task-Listensichtbarkeit

- [CountPendingActivityTasks](#)
- [CountPendingDecisionTasks](#)

## Erstellen eines einfachen Workflows in Amazon SWF

Das Erstellen eines einfachen Workflows umfasst die folgenden Schritte.

- Modellieren eines Workflows, Registrieren des entsprechenden Typs und der Aktivitätstypen

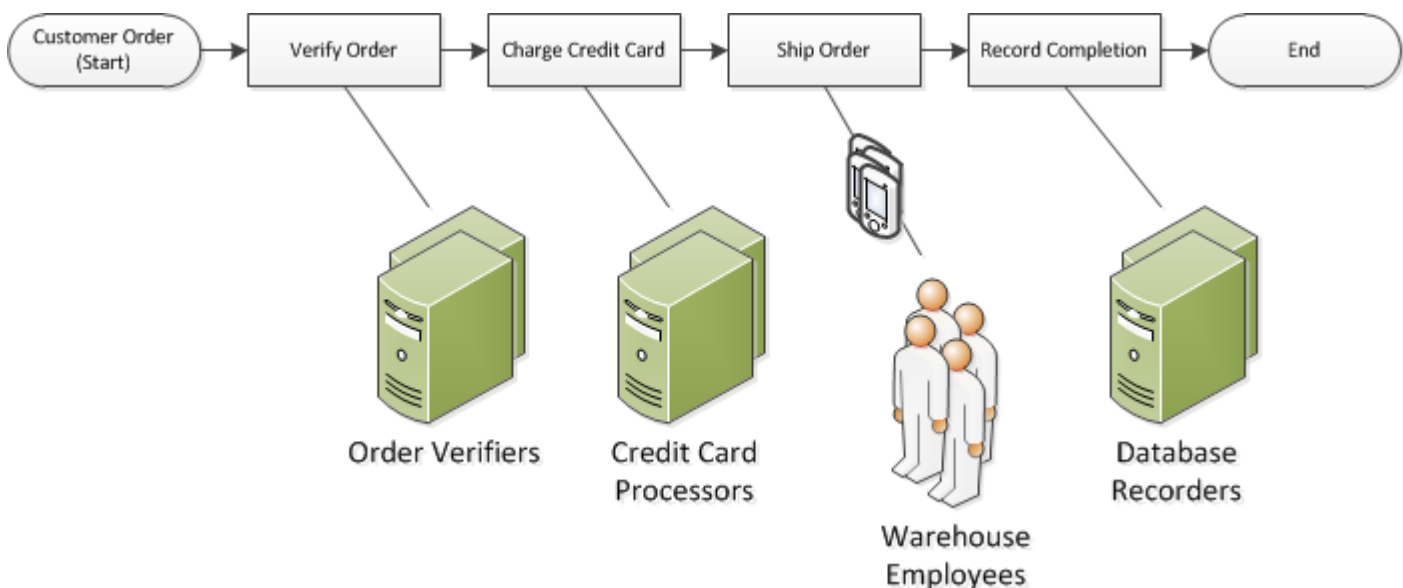
- Entwickeln und Starten von Aktivitäts-Workern, die die Aktivitätsaufgaben ausführen
- Entwickeln und Starten von Entscheidern, die anhand des Workflow-Verlaufs den nächsten Schritt festlegen
- Entwickeln und Starten von Workflow-Startern, d. h. Anwendungen, die Workflow-Ausführungen starten

## Modellieren des Workflows und der entsprechenden Aktivitäten

Modellieren Sie zur Verwendung von Amazon SWF 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).



# Registrieren einer Domäne 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 AWS Management Console oder indem Sie die `RegisterDomain` Aktion in der Amazon SWF SWF-API. Im folgenden Beispiel wird API verwendet.

```
https://swf.us-east-1.amazonaws.com
RegisterDomain
{
  "name" : "867530901",
  "description" : "music",
  "workflowExecutionRetentionPeriodInDays" : "60"
}
```

Die Parameter werden im JavaScript Object Notation (JSON)-Format angegeben. Der Aufbewahrungszeitraum beträgt 60 Tage. Während des Aufbewahrungszeitraums stehen alle Informationen über die Workflow-Ausführung über Sichtbarkeitsoperationen unter AWS Management Console oder die Amazon SWF SWF-API.

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](#) im Amazon Simple Workflow Service

## Einrichten von Timeout-Werten in Amazon SWF

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) eingestellt 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. z. B.:

```
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 über die Registrierung von Workflow-Typen finden Sie unter [Registrieren eines Workflow-Typs bei Amazon SWF](#), und [RegisterWorkflowType](#) im Amazon Simple Workflow Service API-Referenz aus.

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. z. B.:



```
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 über die Aktivität von Workflow-Typen finden Sie unter [Registrieren eines Aktivitätstyps bei Amazon SWF](#), und [RegisterActivityType](#) im Amazon Simple Workflow Service API-Referenz aus.

Sie können den für `defaultTaskStartToCloseTimeout` festgelegten Wert überschreiben, indem Sie bei der Planung der Aktivitätsaufgabe [taskStartToCloseTimeout](#) angeben.

Weitere Informationen finden Sie unter:

[Amazon SWF SWF-Zeitüberschreitungstypen](#)

## Registrieren eines Workflow-Typs bei Amazon SWF

Mit dem in diesem Abschnitt besprochenen Beispiel wird ein Workflow-Typ mithilfe der Amazon SWF SWF-API registriert. 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](#) im Amazon Simple Workflow Service API-Referenz

## Registrieren eines Aktivitätstyps bei Amazon SWF

Im folgenden Beispiel wird ein Aktivitätstyp mit der Amazon SWF SWF-API registriert. 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](#) im Amazon Simple Workflow Service -API-Referenz

## AWS Lambda-Aufgaben

Themen

- [Informationen zu AWS Lambda](#)
- [Vorteile und Einschränkungen der Verwendung von Lambda-Aufgaben](#)
- [Verwenden von Lambda-Aufgaben in Ihren -Workflows](#)

## Informationen zu AWS Lambda

AWS Lambda ist ein vollständig verwalteter Datenverarbeitungsservice, der Ihren Code beim Eintreten bestimmter Ereignisse ausführt, die von einem benutzerdefiniertem Code oder von verschiedenen AWS-Services wie Amazon S3, DynamoDB, Amazon Kinesis, Amazon SNS und Amazon Cognito. Weitere Informationen zu Lambda finden Sie im [AWS Lambda Entwicklerhandbuch](#).

Amazon Simple Workflow Service stellt eine Lambda-Aufgabe bereit, sodass Sie Lambda-Funktionen anstelle von oder neben herkömmlichen Amazon SWF SWF-Aktivitäten ausführen können.

### Important

Ihre AWS-Konto wird für Lambda-Ausführungen (Anfragen) belastet, die von Amazon SWF in Ihrem Namen ausgeführt werden. Weitere Informationen zu Lambda-Preisen finden Sie unter <https://aws.amazon.com/lambda/pricing/>.

## Vorteile und Einschränkungen der Verwendung von Lambda-Aufgaben

Die Verwendung von Lambda-Aufgaben bietet anstelle einer herkömmlichen Amazon SWF SWF-Aktivität 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 Amazon SWF aufgerufen. Im Gegensatz zu herkömmlichen Aktivitäten müssen Sie für ihre Ausführung kein Worker-Programm installieren.
- Lambda stellt Ihnen Metriken und Protokolle zur Verfügung, um Ihre Funktionsausführungen zu verfolgen und zu analysieren.

Bei Lambda-Aufgaben sind jedoch einige Einschränkungen zu beachten:

- Lambda-Aufgaben können nur in ausgeführt werden AWS-Regionen, die Lambda unterstützen. Siehe [.Lambda-Regionen und -Endpunkte](#) im Allgemeine Amazon Web Services Services-Referenz für Einzelheiten über die derzeit unterstützten Regionen für Lambda.
- Lambda-Aufgaben werden derzeit nur von der grundlegenden SWF-HTTP-API und in AWS Flow Framework für Java unterstützt. In AWS Flow Framework für Ruby steht keine Unterstützung für Lambda-Aufgaben zur Verfügung.

## Verwenden von Lambda-Aufgaben in Ihren -Workflows

Um Lambda-Aufgaben in Ihren Amazon SWF SWF-Workflows zu verwenden, müssen Sie:

1. IAM-Rollen einrichten, um Amazon SWF die Berechtigung zum Aufrufen von Lambda-Funktionen zu erteilen.
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 ermöglicht. Führen Sie dazu einen der folgenden Schritte aus:

- wähle eine vordefinierte Rolle, `AWSLambdaRole`, um Ihren Workflows die Berechtigung zum Aufrufen einer Lambda-Funktion zu erteilen, die Ihrem Konto zugeordnet sind.
- Definieren Sie eine eigene Richtlinie und zugewiesene Rolle, um Workflows die Berechtigung zum starten bestimmter Lambda-Funktionen, die durch ihre Amazon-Ressourcennamen (ARNs) angegeben werden, zu erteilen.

### Berechtigung für eine IAM-Rolle einschränken

Sie können Berechtigungen für eine IAM-Rolle einschränken, die Sie Amazon SWF zur Verfügung stellen, indem Sie die `SourceArn` und `SourceAccount` Kontextschlüssel in Ihrer Ressourcenvertrauensrichtlinie. Diese Schlüssel beschränken die Verwendung einer IAM-Richtlinie, sodass sie nur von Amazon Simple Workflow Service-Ausführungen verwendet wird, die in den angegebenen Domänen-ARN gehören. Wenn Sie beide globalen Bedingungskontextschlüssel verwenden, wird `draws:SourceAccountWert` und das Konto, auf das `imaws:SourceArnvalue` muss dieselbe Konto-ID verwenden, wenn sie in derselben Richtlinienanweisung verwendet werden.

Im folgenden Beispiel einer Vertrauensrichtlinie verwenden wir `SourceArn` Kontextschlüssel zur Beschränkung der IAM-Service-Rolle auf die Verwendung nur in Amazon Simple Workflow Service-Ausführungen, die zu `gehörensomedomain` im Konto `123456789012` aus.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "swf.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:swf:*:123456789012:/domain/somedomain"
        }
      }
    }
  ]
}
```

```

    }
  }
]
}

```

Im folgenden Beispiel einer Vertrauensrichtlinie verwenden wir `SourceAccount` Kontextschlüssel zur Beschränkung der IAM-Service-Rolle, die nur in Amazon Simple Workflow Service-Ausführungen im Konto verwendet wird, `123456789012` aus.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "swf.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringLike": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
  ]
}

```

## Amazon SWF den Zugriff zum Aufrufen einer Lambda-Rolle erteilen

Sie können die vordefinierte Rolle verwenden, `AWSLambdaRole`, um Ihren Amazon SWF Workflows die Möglichkeit zu geben, jede Lambda-Funktion aufzurufen, die Ihrem Konto zugeordnet sind.

Um zu verwenden `AWSLambdaRole` um Amazon SWF Zugriff auf Lambda-Funktionen zu erteilen

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. UNTER AWS-Service-Rollen, wählen Amazon SWF, und wählen Nächster Schritt aus.

5. Auf der Richtlinie anfügen bildschirm, AWS Lambda Role aus der Liste.
6. Klicken Sie auf Next Step und auf Create Role, sobald Sie die Rolle überprüft haben.

Definieren einer IAM-Rolle für die Zugriffserteilung für eine bestimmte Lambda-Funktion

Wenn Sie den Zugriff zum Aufrufen einer bestimmten Lambda-Funktion über Ihren Workflow erteilen möchten, müssen Sie Ihre eigene IAM-Richtlinie definieren.

So erstellen Sie eine IAM-Richtlinie für die Zugriffserteilung für eine bestimmte Lambda-Funktion

1. Öffnen Sie die [Amazon IAM-Konsole](#).
2. Wählen Sie Policies und dann Create Policy aus.
3. Klicken Sie auf Kopieren eines AWS-verwaltete Richtlinie und WÄHLEN VON AWS Lambda Role aus der Liste. Es wird eine Richtlinie erstellt. Sie können ihren Namen und die Beschreibung nach Bedarf ändern.
4. In der Resource field des Policy Document, fügen Sie den ARN Ihrer Lambda-Funktion (en) hinzu. Beispiel:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lambda:InvokeFunction"
      ],
      "Resource": [
        "arn:aws:lambda:us-east-1:111111000000:function:hello_lambda_function"
      ]
    }
  ]
}
```

#### Note

Eine vollständige Beschreibung der Vorgehensweisen zum Angeben von Ressourcen in einer IAM-Rolle finden Sie unter [Übersicht über IAM-Richtlinien](#) in Verwenden von IAM aus.

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 den Aufruf Zugriff auf Ihre Amazon SWF SWF-Workflows zu erteilen. Diese Vorgehensweise ist ähnlich wie das Erstellen einer -Rolle mitAWSLambdaRole-Richtlinien. Wählen Sie stattdessen Ihre eigene Richtlinie beim Erstellen der -Rolle aus.

So erstellen Sie eine Amazon SWF SWF-Rolle mit 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. UNDERAW-Servicerollen, wählenAmazon SWF, und wählenNächster Schrittaus.
5. Auf derRichtlinie anfügenWählen Sie Ihre funktionsspezifische 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 anfügen, der die Rolle zum Aufrufen der Lambda-Funktionen, auf die Sie Amazon SWF Zugriff erteilt haben, verwenden wird.

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

- Beim AnrufenRegisterWorkflowGeben Sie ein, stellen Sie diedefaultLambdaRolefield zum ARN der Rolle, die Sie definiert haben.



## So stellen Sie eine Lambda-Rolle während der Workflow-Ausführung bereit

- Beim Anrufen `StartWorkflowLegen` Sie im Feld `LambdaRole` den ARN der Rolle fest, die Sie definiert haben.

### Note

wenn das Konto `anruftRegisterWorkflowTyp` oder `StartWorkflow` Die Ausführung hat keine Berechtigung, die angegebene Rolle zu verwenden, dann schlägt der Aufruf mit einer `OperationNotPermittedFault` aus.

## Rufen Sie Ihre Lambda-Funktion von einem Amazon SWF SWF-Workflow aus auf

Sie können das `ScheduleLambdaFunctionDecisionAttribute` des Datentyps, um die Lambda-Funktion zu identifizieren, die während einer Workflow-Ausführung aufgerufen werden soll.

Während eines Anrufs bei `RespondDecisionTaskCompleted` stellen Sie ein zur Verfügung `ScheduleLambdaFunctionDecisionAttribute` Ihrer Entscheidungsliste. 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 bereitgestellt werden, die den Zugriff zum Aufrufen der Lambda-Funktion gewährt. Der

bereitgestellte Name muss den Einschränkungen für `FunctionNameParameter` wie in der Lambda-Aufrufaktion.

- `input` mit optionalen Eingabedaten für die Funktion. Wenn diese Option festgelegt ist, muss dies den Einschränkungen für `ClientContextParameter` wie in der Lambda-Aufrufaktion.
- `startToCloseTimeout` mit einem optionalen maximalen Zeitraum (in Sekunden), der von der Funktion dazu verwendet werden kann, die Ausführung auszuführen, bevor sie aufgrund einer Ausnahme der Zeitbeschränkung fehlschlägt. Der Wert `NONE` kann dafür verwendet werden, eine unbegrenzte Dauer festzulegen.

Weitere Informationen finden Sie unter [Implementierung AWS Lambda Aufgaben](#)

## Entwickeln eines Aktivitäts-Workers in Amazon SWF

Ein Aktivitäts-Worker implementiert einen oder mehrere Aktivitätstypen. Er kommuniziert mit Amazon SWF, um Aktivitätsaufgaben abzurufen und auszuführen. Sie können mehrere Aktivitäts-Worker für Aktivitätsaufgaben desselben Aktivitätstyps erstellen.

Amazon SWF stellt den Aktivitäts-Workern eine Aktivitätsaufgabe bereit, wenn der Entscheider die Aktivitätsaufgabe plant. Wenn ein Entscheider eine Aktivitätsaufgabe plant, stellt der (von Ihnen festgelegten) Daten bereit, die der Aktivitäts-Worker zum Ausführen der Aktivitätsaufgabe benötigt. Amazon SWF fügt diese Daten in die Aktivitätsaufgabe ein, bevor diese an den Aktivitäts-Worker gesendet wird.

Sie selbst verwalten die Aktivitäts-Worker. Sie können in einer beliebigen Programmiersprache geschrieben werden. Ein Worker kann überall ausgeführt werden, solange er über die API mit Amazon SWF kommunizieren kann. Da Amazon SWF alle für die Ausführung einer Aktivitätsaufgabe erforderlichen Informationen bereitstellt, können alle Aktivitäts-Worker zustandslos 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. Regelmäßig einen Heartbeat an Amazon SWF senden, wenn es sich um eine langfristige Aufgabe handelt.

4. Melden, wenn die Aufgabe abgeschlossen wurde oder fehlgeschlagen ist, und die Ergebnisse an Amazon SWF senden.

## 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 verarbeiten zu können, muss ein Aktivitäts-Worker regelmäßig den `PollForActivityTask` Aktion

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 ist, gibt diese von Amazon SWF an den Aktivitäts-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 Ihren Aktivitäts-Worker so programmieren, dass er bei langfristigen Aktivitäten Statusaktualisierungen an sendet.

## 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 für eine Aktivitätsaufgabe kein Heartbeat in der Zeitüberschreitung festgelegt wurde, tritt die Zeitüberschreitung in Kraft und Amazon SWF schließt die Aufgabe und plant eine neue Entscheidungsaufgabe, um den Entscheider über die Zeitüberschreitung zu informieren. Der Entscheider kann die Aktivitätsaufgabe dann neu planen oder eine andere Aktion ausführen.

Wenn der Aktivitäts-Worker nach einer Zeitüberschreitung versucht, mit Amazon SWF zu kommunizieren, indem er beispielsweise aufruft `RespondActivityTaskCompleted` wird Amazon SWF eine zurückgeben `UnknownResource` Fehlermeldung.

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 sendet der Aktivitäts-Worker einen Heartbeat an Amazon SWF und meldet im Detailfeld den Status der Aktivitätsaufgabe als 40% abgeschlossen. 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"
}
```

Nach Abschluss einer Aktivität plant Amazon SWF eine neue Entscheidungsaufgabe für die Workflow-Ausführung, der die Aktivität zugeordnet 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 `startToCloseTimeout` Zeitraum oder `wennScheduleToCloseTimeout` wurde überschritten, Amazon SWF stellt die Zeitüberschreitung für die Aktivitätsaufgabe bereit und plant eine Entscheidungsaufgabe. So kann ein Entscheider eine angemessene Aktion wie das erneute Planen der Aufgabe ausführen.

Wenn beispielsweise eine Amazon EC2 EC2-Instance eine Aktivitätsaufgabe ausführt und die Instance ausfällt, bevor die Aufgabe abgeschlossen wurde, empfängt der Entscheider ein

Zeitüberschreitungereignis im Verlauf der Workflow-Ausführung. Wenn für die Aktivitätsaufgabe ein Heartbeat verwendet wird, empfängt der Entscheider das Ereignis, wenn die Aufgabe nach dem Ausfall der Amazon EC2 EC2-Instance den nächsten Heartbeat nicht sendet. 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 Aktivitäts-Worker eine Aktivitätsaufgabe aus irgendeinem Grund nicht ausführen, aber dennoch mit Amazon SWF kommunizieren kann, können Sie ihn so programmieren, dass diese 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. Dies sind Freiformfelder. Sie können beliebige Fehlercodes für Ihre Anwendung verwenden. Amazon SWF verarbeitet diese Werte nicht. Allerdings zeigt Amazon SWF diese Werte jedoch in der Konsole an.

Wenn eine Aktivitätsaufgabe fehlgeschlagen ist, plant Amazon SWF eine Entscheidungsaufgabe für die Workflow-Ausführung, der diese Aktivitätsaufgabe zugeordnet ist, um den Entscheider über die fehlgeschlagene Aufgabe 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.

Abrufe sind ausgehende Anforderungen. Deshalb können Aktivitäts-Worker auf 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 abfragenden Aktivitäts-Worker.

## Entwickeln 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 im Workflow-Verlauf gespeichert wird, können Entscheider zustandslos sein. Amazon SWF verwaltet den Workflow-Ausführungsverlauf und stellt diesen einem Entscheider bei jeder Entscheidungsaufgabe 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.

Immer dann, wenn sich der Status einer Workflow-Ausführung ändert, 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.
- Er schließt die Entscheidungsaufgabe ab und übermittelt Amazon SWF eine Entscheidungsliste.

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 Entscheider in einer beliebigen Sprache implementieren und überall ausführen, solange dieser mit Amazon SWF über die entsprechende Service-API kommuniziert.

## 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  
}
```

Ist eine Entscheidungsaufgabe in der angegebenen Aufgabenliste verfügbar, gibt Amazon SWF diese sofort zurück. Ist keine Entscheidungsaufgabe verfügbar, hält Amazon SWF die Verbindung für bis zu 60 Sekunden und gibt eine Aufgabe zurück, sobald diese verfügbar ist. Steht keine Aufgabe zur Verfügung, 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.

Ist eine Entscheidungsaufgabe verfügbar, gibt Amazon SWF eine Antwort zurück, die die Entscheidungsaufgabe enthält sowie eine paginierte Ansicht des Workflow-Ausführungsverlaufs.

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. Zum Abrufen nachfolgender Seiten rufen Sie `PollForDecisionTask` mehrmals mit dem vom ursprünglichen Aufruf zurückgelieferten `nextPageToken` auf. Beachten Sie, dass Sie nicht `GetWorkflowExecutionHistory` mit diesem `nextPageToken` aufrufen. 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 treffenden Entscheidungen ermittelt hat, kann dieser Amazon SWF mit entsprechenden Entscheidungen antworten.

## Reagieren auf Entscheidungen

Nach dem Interpretieren des Workflow-Verlaufs und dem Generieren einer Entscheidungsliste, ist der Entscheider bereit, mit diesen Entscheidungen an Amazon SWF 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 kommuniziert diese Amazon SWF dem `RespondDecisionTaskCompleted` Aktion. Sehen Sie sich die [Amazon Simple Workflow Service API-Referenz](#) für eine Liste der verfügbaren [Entscheidungstypen](#) aus.

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 das Schließen der Ausführung verhinderte.

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, um sicherzustellen, dass die Entscheidung, die Workflow-Ausführung zu schließen oder abubrechen, die letzte vom Entscheider gesendete Entscheidung ist. 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 mit dem Abrufen von Amazon SWF nach Aufgaben beginnen. Bis Sie die Workflow-Ausführung initiiert haben und Amazon SWF Entscheidungsaufgaben plant, kommt es beim Abrufen zu einem Timeout und es werden leere Antworten zurückgegeben. 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 sich, dass immer nur eine Entscheidungsaufgabe für eine Workflow-Ausführung aktiv ist. So werden Probleme, beispielsweise Konflikte zwischen Entscheidungen, verhindert. Darüber hinaus stellt Amazon SWF sicher, dass eine einzelne Entscheidungsaufgabe einem einzelnen Entscheider zugewiesen wird, und zwar unabhängig von der Anzahl der laufenden Entscheider.

Steht etwas vor, 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. Nach Abschluss der aktuellen Aufgabe stellt Amazon SWF die neue Entscheidungsaufgabe bereit. Darüber hinaus kommt es bei Entscheidungsaufgaben zu einer Art Stapelverarbeitung, d. h., wenn mehrere Aktivitäten abgeschlossen werden, während ein Entscheider eine Entscheidungsaufgabe verarbeitet, erstellt Amazon SWF nur eine einzelne neue Entscheidungsaufgabe, um dem Abschließen mehrerer Aufgaben gerecht zu werden. Allerdings wird jedem Abschluss einer Aufgabe ein separates Ereignis im Workflow-Ausführungsverlauf zugewiesen.

Abrufe sind ausgehende Anforderungen. Deshalb können Entscheider auf 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 beliebig viele Entscheider starten. Amazon SWF unterstützt mehrere Entscheider beim Abrufen von derselben Aufgabenliste.

## Starten von Workflow-Ausführungen mit Amazon SWF

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 das `workflowId` einer geschlossenen Hinrichtung. Amazon SWF verknüpft auch eine eindeutige systemgenerierte Kennung, die als `runId` mit jeder Workflow-Ausführung.

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 keinen Wert von `angebenNONE` für diese Zeitüberschreitung besteht ein maximales Limit von einem Jahr für den Zeitraum, in dem eine Workflow-Ausführung ausgeführt werden kann. Ebenso ist `taskStartToCloseTimeout` die Zeitdauer in Sekunden, die eine Entscheidungsaufgabe, die dieser Workflow-Ausführung zugeordnet ist, bis zur Zeitüberschreitung dauern darf.

```
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 das Symbol `StartWorkflowExecutionAktion` ist erfolgreich, Amazon SWF gibt den `runId` für die Workflow-Ausführung. Die `runId` für eine Workflow-Ausführung ist innerhalb einer bestimmten Region eindeutig. `Save` (Speichern `runId` falls Sie diese Workflow-Ausführung später in einem Aufruf an Amazon SWF festlegen müssen. Verwenden Sie die `runId` beispielsweise, wenn Sie später ein Signal an die Workflow-Ausführung senden müssen.

```
{"runId": "9ba33198-4b18-4792-9c15-7181fb3a8852"}
```

## Einrichten der Aufgabenprioritäten

Standardmäßig werden Aufgaben in einer Aufgabenliste basierend auf ihrer Ankunftszeit bereitgestellt: Aufgaben, die zuerst geplant wurden, werden möglichst zuerst ausgeführt. Durch Festlegen eines optionalen Aufgabenpriorität können Sie bestimmten Aufgaben Priorität einräumen:



Amazon SWF wird versuchen, Aufgaben mit einer höheren Priorität in einer Aufgabenliste vor Aufgaben mit niedrigerer Priorität bereitzustellen.

#### 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 (entweder von Ihnen oder von Amazon SWF) bei der Registrierung festgelegt. Die registrierte Aufgabenpriorität wird immer verwendet, sofern sie während der Planung einer Aktivität bzw. dem Starten einer Workflow-Ausführung nicht überschrieben 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 Standardaufgabenpriorität zu registrieren, richten Sie die Option `defaultTaskPriority` mit der Aktion [RegisterWorkflowType](#) ein:

```
{  
  "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 Workflow-Typs beim Starten einer Workflow-Ausführung mit [StartWorkflowExecution](#) überschreiben:

```

{
  "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 registrierte Aufgabenpriorität auch beim Starten eines untergeordneten Workflows überschreiben oder wenn Sie den Workflow als neuen Workflow fortsetzen, beispielsweise als Reaktion auf eine Entscheidung mit [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 beim Registrieren eines Aktivitätstyps einzurichten, richten Sie die Option `defaultTaskPriority` mit der Aktion [RegisterActivityType](#) ein:

```
{
  "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` beim Planen der Aufgabe mit der Aktion [RespondDecisionTaskCompleted](#):

```
{
  "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 zur Aufgabenpriorität (oder zur Standardaufgabenpriorität) über die folgenden Amazon SWF SWF--Aktionen abrufen:

- [DescribeActivityType](#) gibt die `defaultTaskPriority` des Aktivitätstyps im Abschnitt `configuration` der Antwort zurück.
- [DescribeWorkflowExecution](#) gibt die `taskPriority` der Workflow-Ausführung im Abschnitt `executionConfiguration` der Antwort zurück.
- [DescribeWorkflowType](#) gibt die `defaultTaskPriority` des Workflow-Typs im Abschnitt `configuration` der Antwort zurück.
- [GetWorkflowExecutionHistory](#) und [PollForDecisionTask](#) geben Informationen zur Aufgabenpriorität in den Abschnitten `activityTaskScheduledEventAttributes`, `decisionTaskScheduledEventAttributes`, `workflowExecutionContinuedAsNewEventAttributes` und `workflowExecutionStartedEventAttributes` der Antwort zurück.

## Fehlerbehandlung in Amazon SWF

Es gibt verschiedene Fehler, die im Laufe einer Workflow-Ausführung auftreten können.

Themen

- [Validierungsfehler](#)
- [Fehler beim Umsetzen von Aktionen und Entscheidungen](#)
- [Zeitüberschreitungen](#)
- [Fehler durch Benutzercode](#)
- [Fehler beim Schließen einer Workflow-Ausführung](#)

### Validierungsfehler

Validierungsfehler treten auf, wenn eine Anforderung an Amazon SWF fehlschlägt, da sie nicht ordnungsgemäß 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 Anforderung 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 allgemeinen Fehlercodes finden Sie unter dem Thema für die Aktion imReferenz für Amazon Simple Workflow Serviceaus.

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 bei ordnungsgemäßer Formatierung der Anforderung 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](#)

## Zeitüberschreitungen

[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 SWF-Zeitüberschreitungstypen](#) .

- [ActivityTaskTimedOut](#)
- [ChildWorkflowExecutionTimedOut](#)
- [DecisionTaskTimedOut](#)
- [WorkflowExecutionTimedOut](#)

## Fehler durch Benutzercode

Beispiele für diese Art der Fehlerbedingung sind Fehler bei Aktivitätsaufgaben und untergeordneten Workflows. Wie bei Timeout-Fehlern fügt Amazon SWF dem Verlauf der Workflow-Ausführung 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 [Ereignisverlauf](#) in der Amazon SWF 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, sind aber keine festen Grenzen. 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 Count APIs — Zeigt an, dass ein internes Kontingent erreicht wurde und dass die Antwort nicht der vollen 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, kommt es schneller zu einem Abschluss, wenn der Verlauf kleiner ist. Wenn Sie das [Flow Framework](#) verwenden, können Sie `ContinueAsNew` verwenden, um einen Workflow mit einem neuen Verlauf fortzusetzen.

- Maximale Anzahl offener untergeordneter Workflow-Ausführungen — 1.000 pro Workflow-Ausführung
- Quote für Leerlaufzeit bei der Workflow-Ausführung — 1 Jahr (begrenzt durch die maximale Ausführungszeit des Workflows)

Sie können [Workflow-Timeouts](#) konfigurieren, damit ein Timeout-Ereignis eintritt, wenn eine bestimmte Phase des Workflows zu lange dauert.

- Kontingent für die Aufbewahrungszeit des Workflows — 90 Tage

Nach dieser Zeit kann der Workflow-Verlauf nicht mehr abgerufen oder angezeigt werden. Es gibt keine weitere Beschränkung für die Anzahl geschlossener Workflow-Ausführungen, die von Amazon SWF beibehalten werden.

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.

- Maximal geplante Aufgaben pro Sekunde — 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 geplante Aktivitäten als auch Aufgaben, die von Mitarbeitern bearbeitet werden.

- Maximale Anzahl offener Timer: 1.000 pro Workflow-Ausführung
- Maximale Datengröße für Eingabe-/Ergebnisdaten — 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
DeprecateActivityType	200	6
DeprecateDomain	100	6
UndeprecateDomain	100	6
DeprecateWorkflowType	200	6
UndeprecateWorkflowType	200	6
DescribeActivityType	2000	6
DescribeDomain	200	6
DescribeWorkflowExecution	2000	6

API-Name	Bucket-Größe	Nachfüllrate pro Sekunde
DescribeWorkflowType	2000	6
GetWorkflowExecutionHistory	2000	60
ListActivityTypes	200	6
ListDomains	100	6
ListClosedWorkflowExecutions	200	6
ListOpenWorkflowExecutions	200	48
ListWorkflowTypes	200	6
PollForActivityTask	2000	200
PollForDecisionTask	2000	200
RecordActivityTaskHeartbeat	2000	160
RegisterActivityType	200	60
RegisterDomain	100	6
RegisterWorkflowType	200	60
RequestCancelWorkflowExecution	2000	30
RespondActivityTaskCompleted	2000	200
RespondActivityTaskCanceled	2000	200
RespondActivityTaskFailed	2000	200
RespondDecisionTaskCompleted	2000	200
SignalWorkflowExecution	2000	30
StartWorkflowExecution	2000	200

API-Name	Bucket-Größe	Nachfüllrate pro Sekunde
TerminateWorkflowExecution	2000	60
UndeprecateActivityType	200	6
TagResource	50	30
UntagResource	50	30
ListTagsForResource	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	500	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
SignalWorkflowExecution	1000	1000
RecordActivityTaskHeartbeat	1000	1000
RequestCancelWorkflowExecution	200	200

## Beantragen einer Kontingenterhöhung

Verwenden Sie die Seite Support Center in AWS Management Console, um eine Erhöhung des Kontingents für [Drosselungsquoten für alle Regionen](#) und zu beantragen [Entscheidungsquoten für alle Regionen](#). Weitere Informationen finden Sie unter [So fordern Sie eine Kontingenterhöhung an](#) in der Allgemeine AWS-Referenz.

# Amazon Simple Workflow Service

Dieses Kapitel stellt weitere Ressourcen und Referenzdaten bereit, die für das Entwickeln von Workflows mit Amazon SWF hilfreich sind.

## Themen

- [Amazon SWF SWF-Zeitüberschreitungstypen](#)
- [Amazon Sim](#)
- [Zusätzliche Dokumentation für den Amazon Simple Workflow Service](#)
- [Webressourcen für den Amazon Simple Workflow Service](#)

## Amazon SWF SWF-Zeitüberschreitungstypen

Um die korrekte Ausführung von Workflow-Ausführungen sicherzustellen, können Sie mit Amazon SWF in unterschiedliche Zeitüberschreitungen einstellen. 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 Zeitüberschreitungen in der Amazon SWF SWF-API werden in Sekunden angegeben. Amazon SWF unterstützt auch den StringNONE als Zeitüberschreitungswert, der keine Zeitüberschreitung angibt.

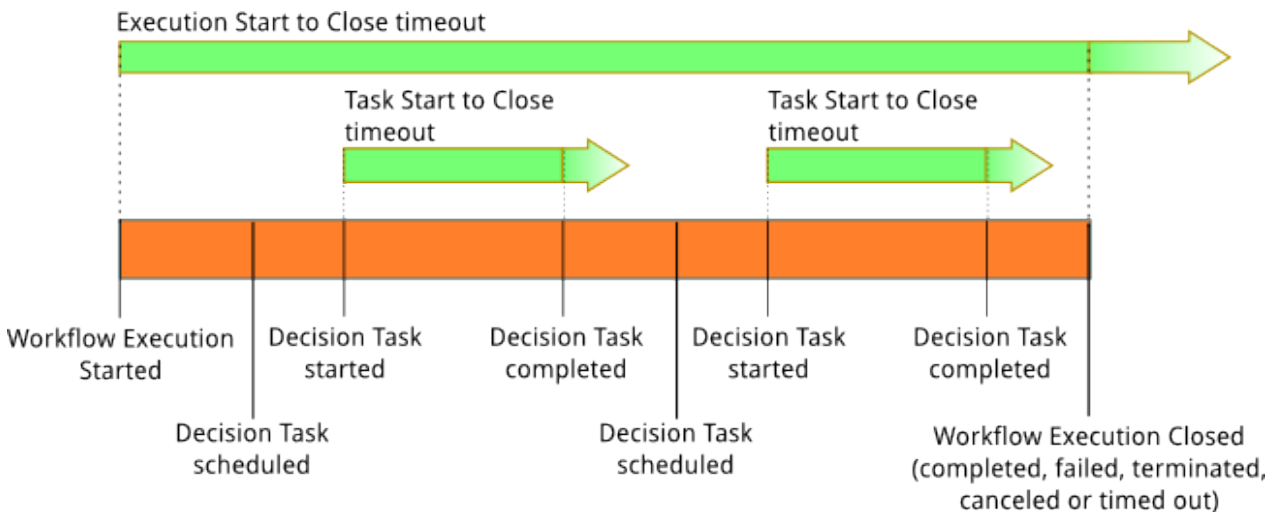
Für Zeitüberschreitungen im Zusammenhang mit Entscheidungsaufgaben und Aktivitätsaufgaben fügt Amazon SWF ein Ereignis zum Verlauf der Workflow-Ausführung hinzu. Die Attribute des Ereignisses enthalten Informationen zum aufgetretenen Zeitüberschreitung und der betroffenen Entscheidungs- oder Aktivitätsaufgabe. Amazon SWF plant auch eine Entscheidungsaufgabe. Wenn der Entscheider die neue Entscheidungsaufgabe empfängt, liest er das Zeitüberschreitungsereignis im Verlauf und ergreift entsprechende Maßnahmen, indem er die Aktion [RespondDecisionTaskCompleted](#) 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 einer Zeitüberschreitung geschlossen werden.



## Zeitüberschreitungen in Workflow- und Entscheidungsaufgaben

Die folgende Abbildung zeigt, wie Zeitüberschreitungen für Workflow- und Entscheidungsaufgaben sich auf die Lebensdauer eines Workflows auswirken:



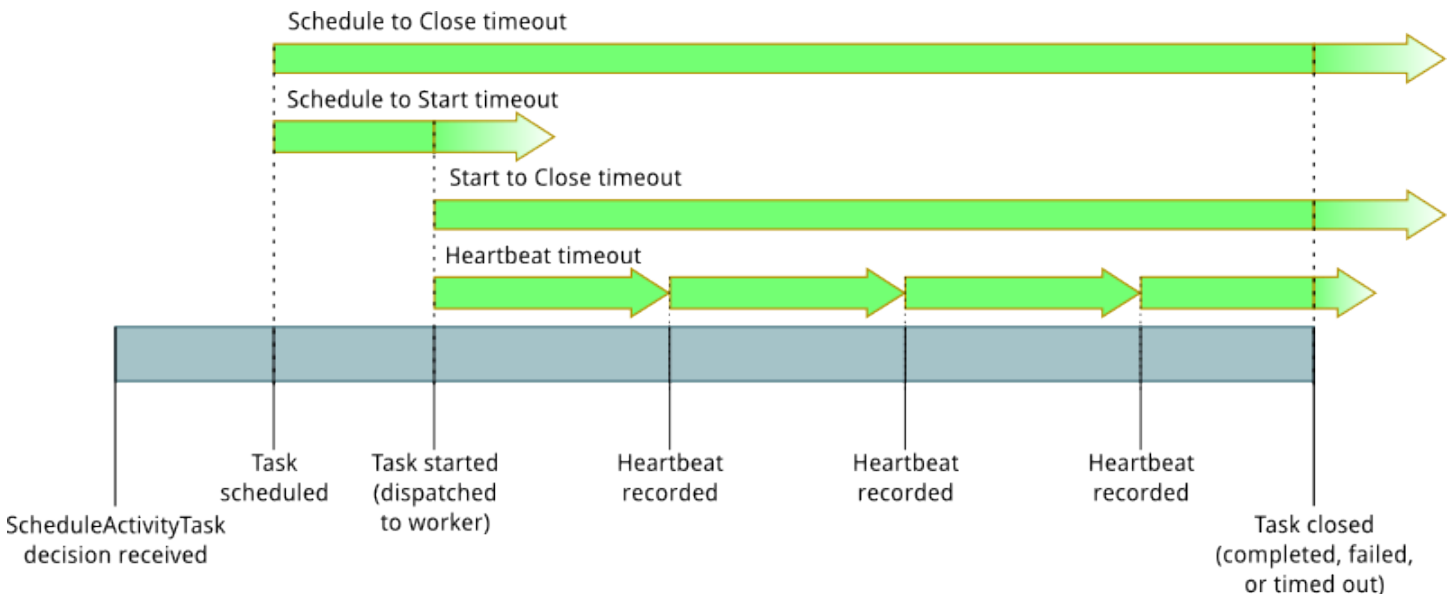
Für Workflow- und Entscheidungsaufgaben gibt es zwei relevante Zeitüberschreitungstypen:

- Workflow vom Starten bis zum Schließen (**timeoutType: START\_TO\_CLOSE**)— Diese Zeitüberschreitung gibt die Höchstdauer an, die eine 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 diese Zeitüberschreitung auftritt, schließt Amazon SWF die Workflow-Ausführung und fügt eine [Veranstaltung](#) des Typs [WorkflowExecutionTimedOut](#) zum Verlauf der Workflow-Ausführung. 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 vom Starten bis zum Schließen (**timeoutType: START\_TO\_CLOSE**)— Diese Zeitüberschreitung gibt die Höchstdauer an, in der der zugehörige Entscheider eine Entscheidungsaufgabe ausführen kann. Sie wird während der Registrierung des Workflow-Typs festgelegt. Wenn diese Zeitüberschreitung auftritt, wird für die Aufgabe im Verlauf der Workflow-Ausführung eine Zeitüberschreitung eingetragen und Amazon SWF fügt ein Ereignis vom Typ hinzu [DecisionTaskTimedOut](#) zum Workflow-Verlauf. Die Ereignisattribute enthalten die IDs für

die Ereignisse, die dem Zeitpunkt entsprechen, zu dem die Entscheidungsaufgabe geplant (`scheduledEventId`) und gestartet (`startedEventId`) wurde. Amazon SWF plant nicht nur das Ereignis hinzu, sondern plant auch eine neue Entscheidungsaufgabe, um den Entscheider über die Zeitüberschreitung zu informieren. 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 — Anfang bis Ende (**timeoutType: START\_TO\_CLOSE**)— Diese Zeitüberschreitung gibt die Höchstdauer an, die ein Aktivitäts-Worker eine Aufgabe nach Erhalt einer Aufgabe benötigen kann. Versuche, eine abgelaufene Aktivitätsaufgabe mit [RespondActivityTaskCanceled](#), [RespondActivityTaskCompleted](#) und [RespondActivityTaskFailed](#) zu schließen, schlagen fehl.
- Heartbeat für Aktivitätsaufgabe (**timeoutType: HEARTBEAT**)— Diese Zeitüberschreitung gibt die Höchstdauer an, die eine Aufgabe ausgeführt werden kann, bevor deren Fortschritt durch die `RecordActivityTaskHeartbeat`-Aktion
- Aktivitätsaufgabe — Planung bis Anfang (**timeoutType: SCHEDULE\_TO\_START**)— Diese Zeitüberschreitung gibt an, wie lange Amazon SWF wartet, bevor die Aktivitätsaufgabe abläuft,

wenn keine Worker zur Ausführung der Aufgabe verfügbar sind. Nach der Zeitüberschreitung wird die abgelaufene Aufgabe keinem anderen Worker zugewiesen.

- Zu schließender Aktivitätsaufgabenplan (**timeoutType: SCHEDULE\_TO\_CLOSE**)— Diese Zeitüberschreitung gibt an, wie lange die Aufgabe vom Zeitpunkt der Planung bis zum Abschluss dauern kann. Dieser Wert sollte nicht größer sein als die Summe der Werte "task schedule-to-start timeout" und "task start-to-close timeout".

### 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 eine Zeitüberschreitung auftritt, fügt Amazon SWF eine [Veranstaltung](#) des Typs [ActivityTaskTimedOut](#) zum Workflow-Verlauf. Das Wertattribut `timeoutType` dieses Ereignisses gibt an, welche dieser Zeitüberschreitungen aufgetreten ist. Der Wert von `timeoutType` für jede Zeitüberschreitung ist in Klammern angegeben. Die Ereignisattribute enthalten auch die IDs für die Ereignisse, die dem Zeitpunkt entsprechen, zu dem die Aktivitätsaufgabe geplant (`scheduledEventId`) und gestartet (`startedEventId`) wurde. Amazon SWF plant nicht nur das Ereignis hinzu, sondern plant auch eine neue Entscheidungsaufgabe, um den Entscheider über die Zeitüberschreitung zu informieren.

## Amazon Sim

Eine Liste der aktuellen [Amazon 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 eine Domäne mit dem Namen "MySampleDomain" sowohl in us-east-1 als auch in us-west-2 erstellen, existieren diese als separate Domänen: Keine der Workflows, Aufgabenlisten, Aktivitäten oder Daten, die mit Ihren Domains verknüpft sind, werden regionsübergreifend gemeinsam genutzt.

Wenn Sie in Ihren Workflows andere AWS Ressourcen verwenden, z. B. Amazon EC2 EC2-Instances, müssen diese ebenfalls in derselben Region wie Ihre Amazon SWF-Ressourcen existieren. Die einzigen Ausnahmen bilden 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.

### API-Referenz für Amazon Simple Workflow Service

Die [API-Referenz zum Amazon Simple Workflow Service](#) enthält detaillierte Informationen zur Amazon SWF-HTTP-API, einschließlich Aktionen, Anforderungs- und Antwortstrukturen und Fehlercodes.

### AWS Flow Framework Dokumentation

[AWS Flow Framework](#) ist ein Programmier-Framework, das die 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.

Jede AWS Flow Framework ist so konzipiert, dass sie in der Sprache, für die sie konzipiert ist, idiomatisch funktioniert, sodass Sie natürlich mit Ihrer 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 für Java-Entwicklerhandbuch](#) enthält Informationen zum Abrufen, Einrichten und Verwenden des AWS Flow Framework für Java.

### AWS SDK-Dokumentation

Die AWS Software Development Kits (SDKs ) bieten Zugriff auf Amazon SWF in vielen verschiedenen Programmiersprachen. Die SDKs folgen genau der HTTP-API, bieten aber auch sprachspezifische Programmierschnittstellen für einige Amazon SWF-Funktionen. Weitere Informationen über die einzelnen SDKs finden Sie über die folgenden Links.

**Note**

Hier sind nur SDKs aufgeführt, die zum Zeitpunkt des Schreibens Unterstützung für Amazon SWF haben. Eine vollständige Liste der verfügbaren AWS SDKs finden Sie auf der Seite [Tools for Amazon Web Services](#).

## Java

stellt eine Java-API für AWS Infrastruktur-Services AWS SDK for Java bereit.

Sie können die verfügbare Dokumentation auf der Seite [AWS SDK for Java -Dokumentation](#) finden. Sie können auch direkt zu den Amazon SWF-Abschnitten in der SDK-Referenz wechseln, indem Sie diesen Links folgen:

- [Class: AmazonSimpleWorkflowClient](#)
- [Class: AmazonSimpleWorkflowAsyncClient](#)
- [Interface: AmazonSimpleWorkflow](#)
- [Interface: AmazonSimpleWorkflowAsync](#)

## JavaScript

Mit dem AWS SDK for JavaScript können Entwickler Bibliotheken oder Anwendungen erstellen, die Services mithilfe einer einfachen und easy-to-use API nutzen AWS , die sowohl im Browser als auch innerhalb von Node.js-Anwendungen auf dem Server verfügbar ist.

Sie können die verfügbare Dokumentation auf der Seite [AWS SDK for JavaScript -Dokumentation](#) finden. Sie können auch direkt zum Abschnitt Amazon SWF in der SDK-Referenz wechseln, indem Sie diesem Link folgen:

- [Class: AWS.SimpleWorkflow](#)

## .NET

Das AWS SDK for .NET ist ein einzelnes, herunterladbares Paket, das Visual Studio-Projektvorlagen, die AWS .NET-Bibliothek, C#-Codebeispiele und Dokumentation enthält. Die AWS SDK for .NET erleichtert Windows-Entwicklern das Erstellen von .NET-Anwendungen für Amazon SWF und andere -Services.

Sie können die verfügbare Dokumentation auf der Seite [AWS SDK for .NET -Dokumentation](#) finden. Sie können auch direkt zu den Amazon SWF-Abschnitten in der SDK-Referenz wechseln, indem Sie diesen Links folgen:

- [Namespace: Amazon.SimpleWorkflow](#)
- [Namespace: Amazon.SimpleWorkflow.Model](#)

## PHP

stellt eine PHP-Programmierschnittstelle zu Amazon SWF AWS SDK for PHP bereit.

Sie können die verfügbare Dokumentation auf der Seite [AWS SDK for PHP -Dokumentation](#) finden. Sie können auch direkt zum Abschnitt Amazon SWF in der SDK-Referenz wechseln, indem Sie diesem Link folgen:

- [Class: SwfClient](#)

## Python

stellt eine Python-Programmierschnittstelle zu Amazon SWF AWS SDK for Python (Boto) bereit.

Die verfügbare Dokumentation finden Sie auf der Seite [boto: Eine Python-Schnittstelle zu Amazon Web Services](#). Sie können auch direkt zu den Amazon SWF-Abschnitten in der -Dokumentation wechseln, indem Sie diesen Links folgen:

- [Amazon SWF-Tutorial](#)
- [Amazon SWF-Referenz](#)

## Ruby

stellt eine Ruby-Programmierschnittstelle zu Amazon SWF AWS SDK for Ruby bereit.

Sie können die verfügbare Dokumentation auf der Seite [AWS SDK for Ruby -Dokumentation](#) finden. Sie können auch direkt zum Abschnitt Amazon SWF in der SDK-Referenz wechseln, indem Sie diesem Link folgen:

- [Klasse: AWS::SimpleWorkflow](#)

## AWS CLI Dokumentation

Die AWS Command Line Interface (AWS CLI) ist ein einheitliches Tool zur Verwaltung Ihrer - AWS Services. Mit nur einem Tool zum Herunterladen und Konfigurieren können Sie mehrere AWS Services über die Befehlszeile steuern und sie über Skripts automatisieren.

Weitere Informationen zur finden Sie AWS CLI auf der Seite [AWS Command Line Interface](#).

Eine Übersicht 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 verschiedene Webressourcen mit weiteren Informationen zu Amazon SWF-Ressourcen oder Hilfeleistung zur Verwendung des Services und zur Entwicklung von Workflows.

Themen

- [Amazon SWF SWF-Forum](#)
- [Amazon SWF SWF-Häufig gestellte Fragen](#)
- [Amazon SWF SWF-Videos](#)

## Amazon SWF SWF-Forum

Im Amazon SWF SWF-Forum können Sie sich mit anderen Amazon SWF SWF-Entwicklern und Mitgliedern des Amazon SWF SWF-Entwicklerteams bei Amazon austauschen und Fragen stellen und Antworten erhalten.

Sie finden das Forum unter: [Forum: Amazon Simple Workflow Service](#) aus. Sie müssen bei Ihrer angemeldet sein AWS-Konto zur Anzeige des Forums aus.

## Amazon SWF SWF-Häufig gestellte Fragen

Die Amazon SWF-FAQ liefern Antworten auf häufig gestellte Fragen zu Amazon SWF, einschließlich einer Übersicht über häufige Anwendungsfälle, Unterschiede zwischen Amazon SWF und anderen Diensten und mehr.

Die häufig gestellten Fragen finden Sie unter: [Amazon SWF SWF-Häufig gestellte Fragen](#) aus.

## Amazon SWF SWF-Videos

Die [Amazon Web Services](#)-Channel auf YouTube enthält Videoschulungen zu allen Webservices von Amazon einschließlich Amazon SWF.

Die Videos werden regelmäßig aktualisiert. Eine vollständige Liste aller Videos zu Amazon SWF-bezogenen Videos zu finden Sie über folgende Suchanfrage: [Simple Workflow in Amazon Web Services](#)

# Optionen für Benutzer von Ruby Flow

Das AWS Flow Framework für Ruby befindet sich nicht mehr aktiv weiterentwickelt. Vorhandener Code funktioniert weiterhin für unbegrenzte Zeit, es gibt jedoch keine neuen Funktionen oder Versionen. Dieses Thema behandelt Nutzungs- und Migrationsoptionen für die weitere Arbeit mit Amazon SWF sowie Informationen zur Migration zu Step Functions.

Option	Description
<a href="#">Weitere Nutzung des Ruby Flow Frameworks</a>	Vorerst wird das Ruby Flow Framework weiterhin funktionieren. Wenn Sie nichts verändern, funktioniert Ihr Code wie bisher. Planen Sie, von dem AWS Flow Framework für Ruby in naher Zukunft.
<a href="#">Migration auf das Java Flow Framework</a>	Das Java Flow Framework befindet sich weiterhin in aktiver Entwicklung und wird auch weiterhin neue Funktionen und Updates erhalten.
<a href="#">Migration auf Step Functions</a>	Step Functions bietet eine Möglichkeit, die Komponenten verteilter Anwendungen unter Verwendung visueller Workflows zu koordinieren, gesteuert durch einen Zustandsautomaten.
<a href="#">Direkte Verwendung des SWF API ohne das Flow Framework</a>	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 dieselbe Abstraktionsebene beibehalten, indem Sie auf das Java Flow Framework migrieren, oder Sie können direkt mit dem Amazon SWF -SDK interagieren.

## Weitere Nutzung des Ruby Flow Frameworks

Das AWS Flow Framework für Ruby wird kurzfristig weiterhin wie gewohnt funktionieren. Wenn Sie Workflows in geschrieben haben, das AWS Flow Framework für Ruby werden diese weiterhin funktionieren. Ohne Updates, Support oder Sicherheitskorrekturen ist es jedoch am besten, von zu migrieren, das AWS Flow Framework für Ruby in naher Zukunft.



## Migration auf das Java Flow Framework

Das AWS Flow Framework für Java wird aktiv weiterentwickelt. Konzeptionell ist das AWS Flow Framework für Java ähnlich wie das AWS Flow Framework für Ruby: Sie können sich weiterhin auf Ihre Workflow-Logik konzentrieren, und das Framework wird Ihnen helfen, Ihre Entscheidungslogik zu verwalten und andere Aspekte von Amazon SWF zu vereinfachen.

- [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 einem Zustandsautomaten gesteuert wird. Mit Step Functions können Sie die Komponenten verteilter Anwendungen und Microservices mithilfe von visuellen 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 Methode zur Koordination von Komponenten und zum Durchlaufen der Funktionen Ihrer Anwendung. 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 führt bei Fehlern erneute Versuche aus, sodass Ihre Anwendung in der richtigen Reihenfolge und wie erwartet ausgeführt wird, jedes Mal. Step Functions protokolliert den Zustand von jedem Schritt. Wenn also Probleme auftreten, können Sie diese schnell diagnostizieren und beheben.

In Step Functions verwalten Sie die Koordination Ihrer Aufgaben unter Verwendung eines Zustandsautomaten, geschrieben in deklarativem JSON-Gerät, das unter Verwendung von [Amazon States Language](#) aus. Dank der Verwendung eines Zustandsautomaten müssen Sie kein Entscheiderprogramm schreiben und pflegen, um Ihre Anwendungslogik zu steuern. Step Functions bietet einen intuitiven, produktiven und flexiblen Ansatz für die Koordination von Anwendungskomponenten unter Verwendung visueller Workflows. Sie sollten in Betracht ziehen, AWS Step Functions für alle Ihre neuen Anwendungen bietet Step Functions eine hervorragende Plattform für die Migration auf die Workflows, auf die Sie derzeit implementiert haben AWS Flow Framework für Ruby.

Um Ihnen bei der Migration Ihrer Aufgaben auf Step Functions zu helfen und gleichzeitig Ihre Ruby-Sprachkenntnisse zu nutzen, bietet Step Functions ein Beispiel für Ruby-Aktivitäts-Worker. Dieses

Beispiel verwendet bewährte Methoden für die Implementierung eines Aktivitäts-Workers und kann als Vorlage für die Migration Ihrer Aufgabenlogik auf Step Functions verwendet werden. Weitere Informationen finden Sie im [Beispiel für einen Aktivitäts-Worker in Ruby](#)-Thema im [AWS Step Functions Entwicklerhandbuch](#).

#### Note

Migrieren Sie für viele Kunden von zu Step Functions AWS Flow Framework für Ruby ist die beste Option. Wenn es für Sie jedoch erforderlich ist, dass Signale in Ihre Prozesse eingreifen, oder wenn Sie untergeordnete Prozesse starten müssen, die ein Ergebnis an einen übergeordneten Prozess zurückgeben, sollten Sie direkt das Amazon SWF SWF-API verwenden oder auf migrieren AWS Flow Framework für Java.

Weitere Informationen zu AWS Step Functions finden Sie unter:

- [AWS Step Functions-Entwicklerhandbuch](#)
- [AWS Step Functions API Referenz](#)
- [AWS Step Functions Befehlszeilenreferenz](#)

## Verwenden Sie die Amazon SWF SWF-API direkt

Während der AWS Flow Framework für Ruby einen Teil der Komplexität von Amazon SWF verwaltet, können Sie auch die Amazon SWF SWF-API 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](#)

# Verlauf des Amazon Simple Workflow Service-Entwicklerhandbuchs — Verlauf

Die folgende Tabelle beschreibt die wichtigsten Änderungen in der Dokumentation seit der letzten Version des Amazon Simple Workflow Service-Entwicklerhandbuchs.

- API-Version: 2012-01-25
- Letzte Aktualisierung Aktualisierung der Dokumentation: Mai 2021

Änderung	Beschreibung	Änderungsdatum
Reines Dokumentationsupdate	Amazon SWF enthält jetzt einen Abschnitt über AWS Benutzerbenachrichtigungen, der als zentrale Stelle für Ihre AWS Benachrichtigungen in der AWS Management Console dient. AWS-Service Weitere Informationen finden Sie unter <a href="#">Verwendung AWS-Benutzerbenachrichtigungen mit Amazon Simple Workflow Service</a> .	4. Mai 2023
Aktualisierung	Amazon SWF bietet jetzt eine neue Konsolenoberfläche zur Verwaltung von SWF-Workflows und deren ausführungsbefehlbezogenen Aktionen. Weitere Informationen finden Sie in den <a href="#">Amazon SWF SWF-Konsolen-Tutorials</a> .	12. September 2022
Aktualisierung	Der <a href="#">Kontingente für Aufgabenausführungen</a> Abschnitt wurde aktualisiert und die <a href="#">Amazon SWF-Metriken für CloudWatch</a> Seite enthält Maximum tasks scheduled per second nun Informationen zur <a href="#">Verwendung von Nicht-ASCII-Ressourcennamen</a> mit CloudWatch	12. Mai 2021
Neue Funktion	Amazon Simple Workflow Service unterstützt jetzt Amazon Simple Workflow Service jetzt AmazonEventBridge. Weitere Informationen finden Sie unter: <ul style="list-style-type: none"> <li>• <a href="#">EventBridge für Amazon SWF</a></li> </ul>	18. Dezember 2020

Änderung	Beschreibung	Änderungsdatum
	<ul style="list-style-type: none"> <li>• <a href="#">EventBridge-Benutzerhandbuch</a></li> </ul>	
Neue Funktion	<p>Amazon Simple Workflow Service unterstützt IAM-Berechtigungen mithilfe von Tags. Weitere Informationen finden Sie unter den folgenden Topics.</p> <ul style="list-style-type: none"> <li>• <a href="#">Tags</a> <ul style="list-style-type: none"> <li>• <a href="#">Verwalten von Tags</a></li> <li>• <a href="#">Markieren von Workflow-Ausführungen</a></li> <li>• <a href="#">Steuern des Zugriffs auf Domains mit Tags</a></li> </ul> </li> <li>• <a href="#">TagResource</a></li> <li>• <a href="#">UntagResource</a></li> <li>• <a href="#">ListTagsForResource</a></li> <li>• <a href="#">RegisterDomain</a></li> </ul>	20. Juni 2019
Neue Funktion	Amazon Simple Workflow Service ist jetzt in der Region Europa (Stockholm) erhältlich.	12. Dezember 2018
Aktualisierung	Das Thema Amazon Simple Workflow Service zur CloudTrail Integration wurde verbessert. Siehe <a href="#">Aufzeichnen von API-Aufrufen mit AWS CloudTrail</a> .	7. August 2018
Aktualisierung	Informationen zu der neuen PendingTasks -Metrik für CloudWatch wurden hinzugefügt. Weitere Informationen finden Sie unter <a href="#">Amazon SWF-Metriken</a> .	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 <a href="#">Optionen für Benutzer von Ruby Flow</a> .	9. März 2018

Änderung	Beschreibung	Änderungsdatum
Aktualisierung	Verbesserte Navigation im Thema für fortgeschrittene Konzepte. Siehe <a href="#">Fortgeschrittene Konzepte in Amazon SWF</a> .	19. Februar 2018
Aktualisierung	Verbesserte Dokumentation der CloudWatch-Metriken durch Aufnahme von Informationen zu gültigen Statistiken. Siehe <a href="#">Amazon SWF-Metriken für CloudWatch</a> .	4. Dezember 2017
Aktualisierung	Änderungen am Inhaltsverzeichnis zur Verbesserung der Dokumentstruktur. Neue Informationen zu <a href="#">API- und Entscheidungsereignismetriken</a> hinzugefügt.	9. November 2017
Aktualisierung	Der Abschnitt <a href="#">Amazon SWF SWF-Kontingente</a> wurde mit den Drosselungslimits für alle Regionen aktualisiert.	18. Oktober 2017
Aktualisierung	<code>task_list</code> wurde in <code>workflowId</code> im <a href="#">Abonnement-Workflow-Tutorial mit Amazon SWF und Amazon SNS</a> 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-Aufgaben	Sie können Lambda-Aufgaben zusätzlich zu den herkömmlichen Aktivitätsaufgaben in Ihren Workflows angeben. Weitere Informationen finden Sie unter <a href="#">AWS Lambda-Aufgaben</a> .	21. Juli 2015

Änderung	Beschreibung	Änderungsdatum
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 bereitzustellen. Informationen zum Festlegen der Aufgabenpriorität für Workflows und Aktivitäten finden Sie unter <a href="#">Einrichten der Aufgabenprioritäten</a> .	17. Dezember 2014
Aktualisierung	Es wurde ein neues Thema hinzugefügt, das beschreibt, wie Amazon SWF SWF-API-Aufrufe protokolliert werden mit CloudTrail: <a href="#">Aufzeichnen von API-Aufrufen mit AWS CloudTrail</a> .	8. Mai 2014
Aktualisierung	Zwei neue Themen im Zusammenhang mit CloudWatch Metriken für Amazon SWF wurden hinzugefügt: <a href="#">Amazon SWF-Metriken für CloudWatch</a> , enthält eine Liste und Beschreibungen der unterstützten Metriken und enthält Informationen darüber <a href="#">Anzeigen von Amazon SWF SWF-Metriken für CloudWatch mithilfe der AWS Management Console</a> , wie Metriken angezeigt und Alarme mit dem gesetzt werden können. AWS Management Console	28. April 2014
Aktualisierung	Neuer Abschnitt: <a href="#">Amazon Simple Workflow Service</a> . 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 <a href="#">Abonnement-Workflow-Tutorial mit Amazon SWF und Amazon SNS</a> .	25. Oktober 2013
Aktualisierung	Informationen zu <a href="#">AWS CLI und Beispiele zu</a> wurden hinzugefügt.	26. August 2013

Änderung	Beschreibung	Änderungsdatum
Aktualisierung	Aktualisierungen und Korrekturen.	1. August 2013
Aktualisierung	Das Dokument wurde aktualisiert, um zu beschreiben, wie IAM für die Zugriffskontrolle verwendet wird.	22. Februar 2013
Erstversion	Dies ist die erste Veröffentlichung des Amazon Simple Workflow Service-Entwicklerhandbuchs.	16. Oktober 2012

# AWS-Glossar

Die neueste AWS-Terminologie finden Sie im [AWS-Glossar](#) in der AWS-Glossar-Referenz.



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.