



Guide du développeur

# Amazon Simple Workflow Service



Version de l'API 2012-01-25

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

# Amazon Simple Workflow Service: Guide du développeur

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

Les marques commerciales et la présentation commerciale d'Amazon ne peuvent pas être utilisées en relation avec un produit ou un service extérieur à Amazon, d'une manière susceptible d'entraîner une confusion chez les clients, ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

---

# Table of Contents

Qu'est-ce qu'Amazon SWF ? .....	1
Composants du flux de travail .....	2
Composants du flux de travail .....	2
Exécution de votre flux de travail .....	4
Configuration de votre environnement de développement .....	4
Développez avec AWS SDKs .....	5
Tenez compte du AWS Flow Framework .....	5
Premiers pas .....	7
A propos du flux de travail .....	8
Prérequis .....	9
Étapes du didacticiel .....	9
Partie 1 : Utilisation d'Amazon SWF avec le SDK pour Ruby .....	9
Incluez le AWS SDK pour Ruby .....	10
Configuration de la AWS session .....	10
Enregistrement d'un domaine Amazon SWF .....	12
Étapes suivantes .....	13
Deuxième partie : mise en œuvre du flux de travail .....	13
Conception du flux de travail .....	13
Configuration du code de flux de travail .....	14
Enregistrement du flux de travail .....	16
Recherche de décisions .....	17
Lancement de l'exécution du flux de travail .....	20
Étapes suivantes .....	23
Troisième partie : mise en œuvre des activités .....	23
Définition d'un type d'activité de base .....	23
Définition GetContactActivity .....	25
Définition SubscribeTopicActivity .....	28
Définition WaitForConfirmationActivity .....	31
Définition SendResultActivity .....	34
Étapes suivantes .....	35
Quatrième partie : mise en œuvre de l'observateur de tâches d'activité .....	35
Exécution du flux de travail .....	39
Comment procéder ensuite ? .....	43
Travailler dans la console .....	44

Enregistrement d'un domaine .....	44
Enregistrement des types de flux de travail .....	45
Enregistrement des types d'activités .....	46
Démarrage d'un flux de travail .....	47
Pour lancer une exécution de flux de travail à l'aide de la console .....	47
Gestion des exécutions de flux de travail .....	47
Concepts de base .....	51
Création d'un flux de travail .....	53
Modélisation de votre flux de travail et de ses activités .....	53
Flux de travail en cours .....	54
Historique du flux de travail .....	55
Identifiants d'objets .....	60
Domaines .....	60
Acteurs .....	61
Qu'est-ce qu'un acteur dans Amazon SWF ? .....	61
Démarrateurs de flux de travail .....	62
Décideurs .....	62
Outils de traitement d'activité .....	64
Echange de données entre les acteurs .....	64
Tâches .....	65
Listes de tâches .....	66
Listes de tâches de décision .....	67
Listes de tâches d'activité .....	67
Routage des tâches .....	68
Fermeture de l'exécution du workflow .....	68
Cycle de vie d'exécution du flux de travail .....	69
Cycle de vie d'exécution du flux de travail .....	70
Sondage de tâches .....	76
Concepts avancés .....	78
Gestion des versions .....	78
Signaux .....	79
Flux de travail pour enfants .....	81
Marqueurs .....	83
Balises .....	85
Gérer les balises .....	85
Étiqueter les exécutions de workflows .....	86

Contrôlez l'accès aux domaines à l'aide de tags .....	87
Choix exclusif .....	88
Temporisateurs .....	91
Annulation de tâches liées à une activité .....	91
Sécurité .....	95
Protection des données .....	95
Chiffrement .....	96
Gestion de l'identité et des accès .....	97
Public ciblé .....	98
Authentification par des identités .....	98
Gestion de l'accès à l'aide de politiques .....	100
Contrôle d'accès .....	102
Actions de politique .....	102
Ressources de politique .....	103
Clés de condition de politique .....	103
ACLs .....	104
ABAC .....	104
Informations d'identification temporaires .....	105
Autorisations de principal .....	105
Rôles de service .....	105
Rôles liés à un service .....	105
Politiques basées sur l'identité .....	106
Politiques basées sur les ressources .....	106
Comment Amazon Simple Workflow Service fonctionne avec IAM .....	107
Exemples de politiques basées sur l'identité .....	108
Principes de base .....	111
Politiques IAM d'Amazon SWF .....	112
Récapitulatif API .....	119
Stratégies basées sur des balises .....	127
Points de terminaison Amazon VPC .....	128
Résolution des problèmes .....	130
Journalisation et surveillance .....	132
Métriques Amazon SWF pour CloudWatch .....	132
Affichage des métriques Amazon SWF .....	143
Enregistrement vers CloudTrail .....	147
EventBridge pour Amazon SWF .....	154

Utilisation Notifications des utilisateurs AWS avec Amazon SWF .....	163
Validation de la conformité .....	163
Résilience .....	164
Sécurité de l'infrastructure .....	165
Configuration et analyse des vulnérabilités .....	165
À l'aide du AWS CLI .....	166
Travailler avec APIs .....	168
Envoi de demandes d'HTTP .....	168
Contenu de l'en-tête HTTP .....	169
Contenu du corps HTTP .....	171
Exemple de demande et de réponse JSON .....	171
Calcul de la signature HMAC-SHA .....	173
Liste des actions Amazon SWF .....	175
Actions liées aux activités .....	175
Actions liées aux décideurs .....	176
Actions liées aux exécutions de flux de travail .....	176
Actions liées à l'administration .....	177
Actions de visibilité .....	178
Inscription d'un domaine .....	179
consultez aussi .....	179
Définition des valeurs de délai d'expiration .....	179
Quotas sur les valeurs de délai d'expiration .....	180
Délais des exécutions de flux de travail et des tâches de décision .....	180
Délais des tâche d'activité .....	180
consultez aussi .....	181
Enregistrement d'un type de flux de travail .....	181
consultez aussi .....	182
Enregistrement d'un type d'activité .....	182
consultez aussi .....	183
Tâches Lambda .....	183
À propos AWS Lambda .....	183
Avantages et limites de l'utilisation des tâches Lambda .....	183
Utilisation de tâches Lambda dans vos flux de travail .....	184
Développement d'un outil de traitement d'activité .....	189
Recherche de tâches d'activité .....	190
Réalisation de la tâche d'activité .....	190

Création de rapports sur les pulsations des tâches d'activité .....	191
Réussite ou échec d'une tâche d'activité .....	191
Lancement des outils de traitement d'activité .....	193
Élaboration de décideurs .....	194
Définition de la logique de coordination .....	195
Recherche des tâches de décision .....	196
Application de la logique de coordination .....	198
Renvoi des décisions .....	198
Fermeture d'une exécution de flux de travail .....	199
Lancement des décideurs .....	201
Démarrer des flux de travail .....	202
Définition de la priorité des tâches .....	203
Définition d'une priorité de tâche pour les flux de travail .....	204
Définition d'une priorité de tâche pour les activités .....	206
Actions renvoyant les informations relatives aux priorités de tâche .....	207
Gestion des erreurs .....	208
Erreurs de validation .....	208
Erreur lors de l'application d'actions ou de décisions .....	208
Délais .....	209
Erreurs déclenchées par le code utilisateur .....	209
Erreurs liées à la fermeture d'une exécution de flux de travail .....	209
Quotas .....	211
Quotas de compte généraux pour Amazon SWF .....	211
Quotas d'exécution des flux de travail .....	212
Quotas relatifs à l'exécution des tâches .....	213
Limitation des quotas Amazon SWF .....	214
Limiter les quotas pour toutes les régions .....	214
Quotas décisionnels pour toutes les régions .....	217
Quotas au niveau du flux de travail .....	217
Demande d'augmentation de quota .....	218
Ressources supplémentaires .....	219
Types de délai .....	219
Délais liés au flux de travail et aux tâches de décision .....	220
Délais des tâches d'activité .....	221
Points de terminaison .....	222
Documentation supplémentaire .....	223

---

Référence de l'API Amazon Simple Workflow Service .....	223
AWS Flow Framework Documentation .....	223
AWS Documentation du SDK .....	223
AWS CLI Documentation .....	225
Ressources Web .....	226
Forum Amazon SWF .....	226
FAQ sur Amazon SWF .....	226
Vidéos Amazon SWF .....	226
Options Ruby Flow .....	226
Continuer à utiliser Flow Framework pour Ruby .....	227
Migrer vers Flow Framework pour Java .....	227
Migrer vers Step Functions .....	228
Utilisez directement l'API Amazon SWF .....	229
Historique de la documentation .....	230
.....	ccxxxiv

# Qu'est-ce qu'Amazon Simple Workflow Service ?

Avec Amazon Simple Workflow Service (Amazon SWF), vous pouvez créer, exécuter et dimensionner des tâches d'arrière-plan comportant des étapes parallèles ou séquentielles. Vous pouvez coordonner le travail entre les composants distribués et suivre l'état des tâches.

Dans Amazon SWF, une tâche représente une unité logique de travail exécutée par un composant de votre application. La coordination des tâches inclut la gestion des dépendances entre les tâches, la planification et la simultanéité dans le flux de votre application. Avec Amazon SWF, vous pouvez contrôler et coordonner les tâches sans vous soucier des complexités sous-jacentes, telles que le suivi de la progression et le maintien de l'état des tâches.

Lorsque vous utilisez Amazon SWF, vous implémentez des travailleurs pour effectuer des tâches. Les employés peuvent exécuter leurs tâches soit sur une infrastructure cloud, telle qu'Amazon Elastic Compute Cloud (Amazon EC2), soit dans vos propres locaux. Vous pouvez créer des tâches qui s'exécutent longtemps, qui peuvent échouer, qui expirent ou qui nécessitent un redémarrage, ou qui peuvent se terminer avec un débit et une latence variables. Amazon SWF stocke les tâches et les attribue aux employés lorsqu'ils sont prêts, suit leur progression et tient à jour leur état, y compris les détails relatifs à l'achèvement des tâches.

Pour coordonner les tâches, vous écrivez un programme qui obtient le dernier état des tâches depuis Amazon SWF et utilise cet état pour lancer les tâches suivantes. Amazon SWF maintient l'état d'exécution d'une application de manière durable, afin que celle-ci résiste aux défaillances de composants individuels. Avec Amazon SWF, vous pouvez créer, déployer, dimensionner et modifier des composants d'application de manière indépendante.

## Autres services AWS de flux de travail

Dans la plupart des cas d'utilisation, nous vous recommandons de prendre en compte vos besoins en AWS Step Functions matière de flux de travail et d'orchestration.

Avec Step Functions, vous pouvez créer des flux de travail, également appelés machines d'état, pour créer des applications distribuées, automatiser des processus, orchestrer des microservices et créer des pipelines de données et d'apprentissage automatique. Dans la console ou le AWS kit d'outils Step Functions de VS Code, vous pouvez utiliser le Workflow Studio graphique pour visualiser, modifier, tester et déboguer le flux de travail de votre application.

Pour plus d'informations techniques, consultez le [guide du AWS Step Functions développeur](#).

# Développement de composants de flux de travail avec Amazon SWF

Le développement d'applications distribuées nécessite la coordination de nombreux composants et la gestion de la latence et du manque de fiabilité inhérents à la communication à distance.

Avec Amazon Simple Workflow Service (Amazon SWF), vous pouvez développer des applications asynchrones et distribuées en fournissant un modèle de programmation et une infrastructure permettant de coordonner les composants distribués et de maintenir leur état d'exécution de manière fiable. En vous appuyant sur Amazon SWF, vous pouvez vous concentrer sur la création des aspects de votre application qui la différencient.

## Composantes d'un flux de travail

[Composantes d'un flux de travail](#) Le concept fondamental d'Amazon SWF est le flux de travail. Un flux de travail est un ensemble d'activités qui concourent à atteindre un objectif spécifique, avec une logique qui permet de les coordonner. Par exemple, un flux de travail peut recevoir une commande client et prendre les mesures nécessaires pour exécuter la commande.

Chaque flux de travail s'exécute dans une ressource appelée domaine, qui contrôle l'étendue du flux de travail. Un AWS compte peut comporter plusieurs domaines, chacun pouvant contenir plusieurs flux de travail, mais les flux de travail des différents domaines ne peuvent pas interagir.

Lorsque vous concevez un flux de travail Amazon SWF, vous définissez chacune des activités requises. Vous enregistrez ensuite chaque activité auprès d'Amazon SWF en tant que type d'activité. Vous allez fournir un nom, une version et des valeurs de délai d'expiration. Par exemple, si un client espère recevoir une commande en 24 heures,

Lorsque vous exécutez le flux de travail, il se peut que certaines activités nécessitent d'être effectuées plusieurs fois, peut-être avec entrées variables. Par exemple, dans un flux de travail de commande client, vous pouvez avoir une activité qui gère les articles achetés. Si le client achète plusieurs articles, cette activité devra être exécutée plusieurs fois. Amazon SWF utilise le concept d'une tâche d'activité qui représente l'invocation d'une activité. Dans notre exemple, le traitement de chaque article serait représenté par une tâche d'activité unique.

Un agent d'activité est un programme qui reçoit des tâches d'activité, les exécute et fournit des résultats. La tâche peut en fait être exécutée par une personne. Par exemple, un analyste statistique peut recevoir des ensembles de données, les analyser, puis renvoyer son analyse.

Les tâches d'activité et les agents d'activité qui les exécutent peuvent s'exécuter de manière synchrone ou asynchrone. Les travailleurs peuvent travailler sur un seul site ou être répartis sur plusieurs ordinateurs, potentiellement dans différentes régions géographiques. Des outils de traitement d'activité séparés peuvent être écrits dans des langages de programmation différents et peuvent utiliser des systèmes d'exploitation distincts. Par exemple, un agent d'activité peut fonctionner sur un serveur en Asie, tandis qu'un autre peut fonctionner sur un appareil mobile en Amérique du Nord.

La logique de coordination d'un flux de travail se trouve dans un logiciel appelé décideur. Un décideur planifie les tâches d'activité, fournit des informations aux agents d'activité, traite les événements qui arrivent pendant que le flux de travail est en cours et met fin (ou ferme) le flux de travail une fois l'objectif atteint.

Le rôle du service Amazon SWF est de fonctionner comme un hub central fiable via lequel les données sont échangées entre le décideur, les responsables de l'activité et d'autres entités pertinentes telles que la personne administrant le flux de travail. Amazon SWF conserve également l'état de chaque exécution de flux de travail, ce qui évite à votre application d'avoir à stocker l'état de manière durable.

Le décideur dirige le flux de travail en recevant les tâches de décision d'Amazon SWF et en répondant à Amazon SWF avec des décisions. Une décision représente une action ou un ensemble d'actions, qui constituent les étapes suivantes du flux de travail. Une décision classique consiste à planifier une tâche d'activité. Les décisions peuvent également être utilisées pour retarder les tâches à l'aide de minuteries, demander l'annulation de tâches en cours et pour terminer les flux de travail.

Le mécanisme par lequel les travailleurs actifs et le décideur reçoivent leurs tâches (tâches d'activité et tâches de décision respectivement) consiste à interroger le service Amazon SWF.

Amazon SWF informe le décideur de l'état du flux de travail en incluant, à chaque tâche de décision, une copie de l'historique d'exécution du flux de travail actuel. L'historique d'exécution du flux de travail se compose d'événements. Chacun d'eux représente un changement important de l'état d'exécution du flux de travail. Les exemples d'événements incluent l'achèvement d'une tâche, l'expiration d'une tâche ou l'expiration d'un délai. L'historique est un enregistrement complet, cohérent et fiable de la progression du flux de travail.

Le contrôle d'accès Amazon SWF utilise Gestion des identités et des accès AWS (IAM) pour vous permettre de contrôler l'accès aux ressources. AWS Par exemple, vous pouvez autoriser un utilisateur à accéder à votre compte, mais uniquement pour exécuter certains flux de travail dans un domaine particulier.

## Exécution de votre flux de travail

Vous trouverez ci-dessous un aperçu des étapes nécessaires pour développer et exécuter un flux de travail dans Amazon SWF :

1. Écrivez des agents d'activité pour effectuer les étapes de traitement de votre flux de travail.
2. Rédigez un décideur pour gérer la logique de coordination de votre flux de travail.
3. Enregistrez vos activités et votre flux de travail auprès d'Amazon SWF.

Vous pouvez effectuer cette étape par programmation ou à l'aide du AWS Management Console

4. Lancez les outils de traitement d'activité et le décideur.

Ces acteurs peuvent s'exécuter sur n'importe quel appareil informatique pouvant accéder à un point de terminaison Amazon SWF. Par exemple, vous pouvez utiliser des instances de calcul dans le cloud, telles qu'Amazon Elastic Compute Cloud (Amazon EC2), des serveurs dans votre centre de données ou même un appareil mobile pour héberger un décideur ou un agent d'activité. Une fois le processus lancé, le décideur et les responsables de l'activité doivent commencer à interroger Amazon SWF pour identifier les tâches.

5. Démarrez une ou plusieurs exécutions de flux de travail.

Vous pouvez démarrer les flux de travail par programmation ou via le AWS Management Console

Chaque exécution est indépendante et vous pouvez fournir à chacune son propre ensemble de données d'entrée. Lorsqu'une exécution est lancée, Amazon SWF planifie la tâche de décision initiale. En réponse, votre décideur commence à prendre des décisions qui initient des tâches d'activité. L'exécution continue jusqu'à ce que le décideur prenne la décision de fermer l'exécution.

6. Affichez les exécutions de flux de travail à l'aide du AWS Management Console.

Vous pouvez filtrer et afficher les détails complets des exécutions en cours et terminées. Par exemple, vous pouvez sélectionner une exécution ouverte pour voir quelles tâches ont été achevées et quels en ont été les résultats.

## Configuration de votre environnement de développement

Vous avez la possibilité de développer pour Amazon SWF dans l'un des langages de programmation pris en charge par AWS. Pour les développeurs Java, AWS Flow Framework il est également

disponible. Pour plus d'informations, consultez le [AWS Flow Framework](#) site Web et consultez [AWS Flow Framework le guide du développeur Java](#).

Pour réduire la latence et stocker les données dans un emplacement répondant à vos besoins, Amazon SWF fournit des points de terminaison dans différentes régions.

Chaque point de terminaison d'Amazon SWF est totalement indépendant. Les domaines, les flux de travail et les activités que vous avez enregistrés dans une région ne partageront pas de données ou d'attributs avec ceux d'une autre région.

Lorsque vous enregistrez un domaine, un flux de travail ou une activité Amazon SWF, celui-ci n'existe que dans la région dans laquelle vous l'avez enregistré. Par exemple, vous pouvez enregistrer un domaine nommé SWF-Flows-1 dans deux régions différentes, mais elles ne partageront aucune donnée ni aucun attribut, chacune agissant comme un domaine totalement indépendant.

Pour obtenir la liste des points de terminaison Amazon SWF, consultez [Régions](#) et points de terminaison.

## Développez avec AWS SDKs

Amazon SWF est compatible avec Java, .NET, Node.js, PHP, Python et Ruby, ce qui permet d'utiliser facilement l'API HTTP Amazon SWF dans le langage de programmation de votre choix. AWS SDKs

Vous pouvez développer des décideurs, des agents d'activité ou des initiateurs de flux de travail à l'aide de l'API présentée par ces bibliothèques. Vous pouvez également utiliser des opérations de visibilité via ces bibliothèques afin de développer vos propres outils de surveillance et de reporting Amazon SWF.

Pour télécharger des outils permettant de développer et de gérer des applications sur AWS, notamment SDKs, rendez-vous dans le [Developer Center](#).

Pour obtenir des informations détaillées sur les opérations Amazon SWF dans chaque SDK, consultez la documentation de référence spécifique à la langue du SDK.

## Tenez compte du AWS Flow Framework

AWS Flow Framework Il s'agit d'un SDK amélioré permettant d'écrire des programmes asynchrones distribués qui s'exécutent en tant que flux de travail sur Amazon SWF. Le framework est disponible

pour le langage de programmation Java et fournit des classes pour écrire des programmes distribués complexes.

Avec le AWS Flow Framework, vous utilisez des types préconfigurés pour mapper la définition de votre flux de travail directement aux méthodes de votre programme. Il AWS Flow Framework prend en charge les concepts standard orientés objet, tels que la gestion des erreurs basée sur les exceptions. Les programmes écrits avec le AWS Flow Framework peuvent être créés, exécutés et débogués entièrement dans votre éditeur ou IDE préféré. Pour plus d'informations, consultez le [AWS Flow Framework](#) site Web et consultez [AWS Flow Framework le guide du développeur Java](#).

# Commencer à utiliser Amazon SWF

Vous pouvez commencer avec l'application de flux de travail Amazon Simple Workflow Service suivante, qui consiste en un ensemble de quatre activités qui fonctionnent de manière séquentielle. Le didacticiel couvre également les sujets suivants :

- Définition des options de flux de travail et d'activité default et execution-time
- Interrogation d'Amazon SWF pour des tâches de décision et d'activité.
- Transmission de données entre les activités et le flux de travail avec Amazon SWF.
- En attente de tâches humaines et signalement des pulsations cardiaques à Amazon SWF à la suite d'une tâche d'activité.
- Utiliser Amazon SNS pour créer un sujet, y abonner un utilisateur et publier des messages sur les points de terminaison abonnés.

Vous pouvez utiliser Amazon SWF et Amazon Simple Notification Service (Amazon SNS) conjointement pour émuler un flux de travail « tâche humaine », dans lequel un travailleur humain doit effectuer une action, puis communiquer avec Amazon SWF pour lancer l'activité suivante du flux de travail.

Amazon SWF étant un service Web basé sur le cloud, la communication avec Amazon SWF peut provenir de n'importe quel endroit où une connexion à Internet est disponible. Dans ce cas, nous utiliserons Amazon SNS pour communiquer avec l'utilisateur par e-mail, par SMS ou par les deux.

Ce didacticiel utilise le [AWS SDK pour Ruby](#) pour accéder à Amazon SWF et Amazon SNS, mais de nombreuses options de développement sont disponibles, notamment celle pour Ruby, qui facilite AWS Flow Framework la coordination et la communication avec Amazon SWF.

## Note

Ce didacticiel utilise le AWS SDK pour Ruby, mais nous vous recommandons d'utiliser le [AWS Flow Framework pour Java](#).

## Rubriques

- [A propos du flux de travail](#)

- [Prérequis](#)
- [Étapes du didacticiel](#)
- [Tutoriel sur le flux de travail d'abonnement, partie 1 : Utilisation d'Amazon SWF avec le AWS SDK pour Ruby](#)
- [Deuxième partie du didacticiel sur le flux de travail d'abonnement : mise en œuvre du flux de travail](#)
- [Troisième partie du didacticiel sur le flux de travail d'abonnement : mise en œuvre des activités](#)
- [Quatrième partie du didacticiel sur le flux de travail d'abonnement : mise en œuvre de l'observateur de tâches d'activité](#)
- [Didacticiel sur le flux de travail d'abonnement : exécution du flux de travail](#)

## A propos du flux de travail

Le flux de travail que nous allons développer se compose de quatre grandes étapes :

1. Obtenez une adresse d'abonnement (e-mail ou SMS) de la part de l'utilisateur.
2. Créez une rubrique SNS et abonnez-y les points de terminaison fournis.
3. Attendez que l'utilisateur confirme l'abonnement.
4. Si l'utilisateur le confirme, publiez un message de félicitations dans la rubrique.

Ces étapes comprennent des activités qui s'effectuent automatiquement (étapes 2 et 4), et d'autres qui nécessitent l'intervention d'une personne pour fournir des données à l'activité pour que le flux de travail puisse poursuivre (étapes 1 et 3).

Chaque étape s'appuie sur les données qui sont générées par l'étape précédente (vous devez avoir un point de terminaison avant de l'abonner à une rubrique et vous devez avoir un abonnement à la rubrique avant de pouvoir attendre la confirmation, etc.). Ce didacticiel explique également comment fournir des résultats d'activité une fois celle-ci terminée et comment transmettre des données à une tâche en cours de planification. Amazon SWF gère la coordination et la diffusion des informations entre les activités et le flux de travail, et vice-versa.

Nous utilisons également la saisie au clavier et Amazon SNS pour gérer la communication entre Amazon SWF et l'humain qui fournit des données au flux de travail. Dans la pratique, vous pouvez utiliser de nombreuses techniques différentes pour communiquer avec des utilisateurs humains, mais Amazon SNS fournit un moyen très simple d'utiliser des e-mails ou des SMS pour informer l'utilisateur des événements du flux de travail.

# Prérequis

Pour suivre ce didacticiel, vous avez besoin des éléments suivants :

- [Compte Amazon Web Services](#)
- [Un interpréteur Ruby](#)
- [AWS SDK pour Ruby](#)

Si vous remplissez ces conditions, vous êtes prêt à continuer. Si vous ne souhaitez pas exécuter cet exemple, vous pouvez toujours suivre le didacticiel. La majeure partie du contenu de ce didacticiel concerne l'utilisation d'Amazon SWF et d'Amazon SNS, quelle que soit l'option de développement choisie.

## Étapes du didacticiel

Ce didacticiel est divisé selon les étapes suivantes :

1. [Tutoriel sur le flux de travail d'abonnement, partie 1 : Utilisation d'Amazon SWF avec le AWS SDK pour Ruby](#)
2. [Deuxième partie du didacticiel sur le flux de travail d'abonnement : mise en œuvre du flux de travail](#)
3. [Troisième partie du didacticiel sur le flux de travail d'abonnement : mise en œuvre des activités](#)
4. [Quatrième partie du didacticiel sur le flux de travail d'abonnement : mise en œuvre de l'observateur de tâches d'activité](#)
5. [Didacticiel sur le flux de travail d'abonnement : exécution du flux de travail](#)

## Tutoriel sur le flux de travail d'abonnement, partie 1 : Utilisation d'Amazon SWF avec le AWS SDK pour Ruby

Rubriques

- [Incluez le AWS SDK pour Ruby](#)
- [Configuration de la AWS session](#)
- [Enregistrement d'un domaine Amazon SWF](#)
- [Étapes suivantes](#)

## Incluez le AWS SDK pour Ruby

Commencez par créer un fichier nommé `utils.rb`. Le code contenu dans ce fichier permettra d'obtenir, ou de créer si nécessaire, le domaine Amazon SWF utilisé à la fois par le code du flux de travail et des activités et fournira un emplacement pour insérer le code commun à toutes nos classes.

Tout d'abord, nous devons inclure la `aws-sdk-v1` bibliothèque dans notre code afin de pouvoir utiliser les fonctionnalités fournies par le SDK pour Ruby.

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

Cela nous donne accès à l'espace de noms AWS, qui permet de définir des valeurs globales liées à la session, telles que vos informations d'identification AWS et votre région, et donne également accès au service AWS APIs.

## Configuration de la AWS session

Nous allons configurer la AWS session en définissant nos informations d'identification AWS (nécessaires pour accéder aux services AWS) et la région AWS à utiliser.

Il existe plusieurs manières de [définir les informations d'identification AWS dans le SDK pour Ruby](#) : en les définissant dans des variables d'environnement `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY` et `AWS_REGION` ou en les définissant avec `AWS.config`. Nous allons utiliser cette deuxième méthode de ce dernier, en les chargeant à partir d'un fichier de configuration YAML, appelé `aws-config.txt`, qui ressemble à cela.

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

Créez ce fichier maintenant en remplaçant les chaînes commençant par `REPLACE_WITH_` par votre identifiant de clé d'accès et votre clé AWS d'accès secrète. Pour plus d'informations sur vos clés AWS d'accès, voir [Comment obtenir des informations d'identification de sécurité ?](#) dans le manuel Amazon Web Services General Reference.

Nous devons également définir la région AWS à utiliser. Comme nous utiliserons le [service de messages courts \(SMS\)](#) pour envoyer des SMS au téléphone de l'utilisateur via Amazon SNS, nous devons nous assurer que nous utilisons une région prise en charge par Amazon SNS. Consultez la

section [Régions et pays pris en charge](#) dans le guide du développeur Amazon Simple Notification Service.

### Note

Si vous n'avez pas accès à us-east-1 ou si cette fonctionnalité n'a pas d'intérêt pour vous dans le cadre de cette démonstration, vous pouvez utiliser la région que vous souhaitez. Vous pouvez supprimer la fonctionnalité SMS de l'exemple et utiliser le courrier électronique comme seul point de terminaison pour vous abonner à la rubrique Amazon SNS. Pour plus d'informations sur l'envoi de SMS, consultez la section [Envoyer et recevoir des notifications par SMS à l'aide d'Amazon SNS](#) dans le manuel du développeur Amazon Simple Notification Service.

Nous allons maintenant ajouter du code `utils.rb` pour charger le fichier de configuration, obtenir les informations d'identification de l'utilisateur, puis fournir à la fois les informations d'identification et la région à [AWS.config](#).

```
require 'yaml'

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

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

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

# Now, set the options
AWS.config = options
```

## Enregistrement d'un domaine Amazon SWF

Pour utiliser Amazon SWF, vous devez configurer un domaine : une entité nommée qui contiendra vos flux de travail et vos activités. De nombreux domaines Amazon SWF peuvent être enregistrés, mais ils doivent tous porter des noms uniques dans votre AWS compte, et les flux de travail ne peuvent pas interagir entre les domaines : tous les flux de travail et activités de votre application doivent se trouver dans le même domaine pour interagir les uns avec les autres.

Comme nous utiliserons le même domaine dans l'ensemble de notre application, nous allons créer une fonction dans `utils.rb` appelée `init_domain`, qui récupérera le domaine Amazon SWF nommé SWFSample Domain.

Une fois que vous avez enregistré un domaine, vous pouvez le réutiliser pour de nombreuses exécutions de flux de travail. Toutefois, il n'est pas correct d'essayer d'enregistrer un domaine qui existe déjà. Dès lors, notre code doit vérifier d'abord si le domaine existe et utilisera le domaine existant, s'il le trouve. S'il ne trouve pas le domaine, nous le créerons.

Pour utiliser les domaines Amazon SWF dans le SDK pour Ruby, utilisez [AWS::SimpleWorkflow.Domains](#), qui renvoie un `DomainCollection` qui peut être utilisé à la fois pour énumérer et enregistrer des domaines :

- Pour vérifier si un domaine est déjà enregistré, vous pouvez consulter la liste fournie par [AWS::Simpleworkflow.domains.registered](#).
- Pour enregistrer un nouveau domaine, utilisez [AWS::Simpleworkflow.domains.register](#).

Voici le code pour `init_domain` dans le fichier `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
```

## Étapes suivantes

Vous allez ensuite créer le code du flux de travail et du démarreur dans [Deuxième partie du didacticiel sur le flux de travail d'abonnement : mise en œuvre du flux de travail](#).

## Deuxième partie du didacticiel sur le flux de travail d'abonnement : mise en œuvre du flux de travail

Jusqu'à présent, notre code est assez générique. Nous entrons à présent dans la partie où l'on commence à définir réellement ce que fait notre flux de travail, ainsi que les activités nécessaires pour le mettre en œuvre.

### Rubriques

- [Conception du flux de travail](#)
- [Configuration du code de flux de travail](#)
- [Enregistrement du flux de travail](#)
- [Recherche de décisions](#)
- [Lancement de l'exécution du flux de travail](#)
- [Étapes suivantes](#)

## Conception du flux de travail

Pour rappel, l'idée initiale de ce flux de travail comprenait les étapes suivantes :

1. Obtenez une adresse d'abonnement (e-mail ou SMS) de la part de l'utilisateur.
2. Créez une rubrique SNS et abonnez-y les points de terminaison fournis.
3. Attendez que l'utilisateur confirme l'abonnement.

4. Si l'utilisateur le confirme, publiez un message de félicitations dans la rubrique.

Nous pouvons considérer chaque étape de notre flux de travail comme activité qu'il doit exécuter. Le flux de travail est responsable de la planification de chaque activité à l'heure appropriée et de la coordination du transfert des données entre les activités.

Pour ce flux de travail, nous créerons une activité distincte pour chacune de ces étapes, en leur donnant le nom descriptif suivant :

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

Ces activités seront exécutées dans l'ordre, et les données de chaque étape seront utilisées à l'étape suivante.

Nous pourrions concevoir notre application de telle sorte que tout le code existe dans un seul fichier source, mais cela va à l'encontre de la façon dont Amazon SWF a été conçu. En effet, il convient particulièrement aux flux de travail qui couvrent l'intégralité d'Internet. Dès lors, nous allons répartir l'application en deux fichiers exécutables distincts :

- `swf_sns_workflow.rb` : contient le flux de travail et le démarreur.
- `swf_sns_activities.rb` : contient les activités et leur démarreur.

Les implémentations du flux de travail et des activités peuvent être exécutées dans des fenêtres distinctes, sur des ordinateurs séparés, voire dans différentes régions du monde. Amazon SWF assurant le suivi des détails de vos flux de travail et de vos activités, ceux-ci peuvent coordonner la planification et le transfert de données de vos activités, quel que soit leur lieu d'exécution.

## Configuration du code de flux de travail

Nous commencerons par créer un fichier nommé `swf_sns_workflow.rb`. Dans ce fichier, déclarez une classe appelée `SampleWorkflow`. Voici la déclaration de classe et son constructeur, la méthode `initialize`.

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

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

  attr_accessor :name

  def initialize(workflowId)

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

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

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

    register_workflow
  end
end
```

Comme vous pouvez le voir, nous conservons les données d'instance de classe suivantes :

- `domain` : nom de domaine extrait d'`init_domain` dans `utils.rb`.
- `workflowId` : liste des tâches transmises à `initialize`.
- `activity_list` : liste d'activités, avec le nom et la version des activités que nous allons exécuter.

Le nom de domaine, le nom de l'activité et la version de l'activité sont suffisants pour qu'Amazon SWF identifie avec certitude un type d'activité. Il s'agit donc de toutes les données que nous devons conserver sur nos activités afin de les planifier.

La liste des tâches est utilisée par le code de décideur du flux de travail pour rechercher les tâches de décision et les activités de planification.

A la fin de cette fonction, nous appelons une méthode que nous n'avons pas encore définie : `register_workflow`. Nous définirons cette méthode ensuite.

## Enregistrement du flux de travail

Pour utiliser un type de flux de travail, nous devons tout d'abord l'enregistrer. Comme un type d'activité, un type de flux de travail est identifié par son domaine, son nom et sa version. En outre, comme les domaines et les types d'activités, vous ne peut pas ré-enregistrer un type de flux de travail existant. Si vous avez besoin de modifier un type de flux de travail, vous devez le faire via une nouvelle version, ce qui crée un autre type.

Voici le code du flux de travail `register_workflow`, qui est utilisé pour récupérer le type de flux de travail existant que nous avons enregistré lors d'une exécution précédente ou pour l'enregistrer si cela n'est pas déjà fait.

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

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

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

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

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

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

Tout d'abord, pour vérifier si le nom et la version du flux de travail sont déjà enregistrés, nous procédons à l'itération via la collection [workflow\\_types](#) du domaine. Si nous constatons une correspondance, nous utiliserons le type de flux de travail qui a déjà été enregistré.

Si aucune correspondance n'est trouvée, un nouveau type de flux de travail est enregistré (en appelant [Register](#) dans la même `workflow_types` collection dans laquelle nous recherchions le flux de travail) avec le nom `swf-sns-workflow` « », la version « 1 » et les options suivantes.

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

Les options transmises lors de l'enregistrement sont utilisées pour définir le comportement par défaut de notre type de flux de travail. Dès lors, nous n'avons pas besoin de définir ces valeurs chaque fois nous que nous débutons une nouvelle exécution de flux de travail.

Ici, nous définissons uniquement certaines valeurs de délai : la durée maximale entre le début d'une tâche et sa fin (une heure), et la durée maximale d'exécution du flux de travail (24 heures). Si un de ces délais est dépassé, la tâche ou le flux de travail expire.

Pour plus d'informations sur les valeurs de ces délais, consultez la section [Types de délai d'expiration Amazon SWF](#).

## Recherche de décisions

Un décideur se trouve au cœur de l'exécution de chaque flux de travail. La responsabilité du décideur consiste à gérer l'exécution du flux de travail lui-même. Il reçoit les tâches de décision et y répond soit en planifiant de nouvelles activités, en annulant des activités et en les redémarrant, soit en définissant l'état de l'exécution du flux de travail comme terminé, comme annulé ou comme ayant échoué.

Le décideur utilise le nom de la liste des tâches de l'exécution du flux de travail pour recevoir des tâches de décision et y répondre. Pour rechercher des tâches de décision, appelez [pool](#) au niveau de la collection [decision\\_tasks](#) du domaine afin de parcourir les tâches de décision disponibles. Vous pouvez ensuite rechercher les nouveaux événements dans la tâche de la décision en procédant à une itération avec sa collection [new\\_events](#).

Les événements renvoyés sont [AWS::SimpleWorkflow::HistoryEvent](#) des objets, et vous pouvez obtenir le type de l'événement en utilisant le membre [event\\_type](#) de l'événement renvoyé. Pour obtenir une liste et une description des types d'événements historiques, consultez [HistoryEvent](#) manuel Amazon Simple Workflow Service API Reference.

Voici le début de la logique de l'observateur de tâches de décision. Une nouvelle méthode dans notre classe de flux de travail a appelé `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
```

Nous allons maintenant connecter l'exécution de notre décideur en fonction de l'`event_type` reçu. Le premier que nous sommes susceptibles de recevoir est `WorkflowExecutionStarted`. Lorsque cet événement est reçu, cela signifie qu'Amazon SWF indique à votre décideur qu'il doit commencer l'exécution du flux de travail. Nous allons commencer par planifier la première activité. Pour ce faire, nous appelons [schedule\\_activity\\_task](#) au niveau de la tâche que nous avons reçu lors de la recherche.

Nous lui transmettons la première activité que nous avons déclarée dans notre liste d'activités, qui occupe la position `last` dans la liste, car nous avons inversé cette dernière pour l'utiliser telle une pile. Les « activités » que nous avons définies ne sont que des cartes composées d'un nom et d'un numéro de version, mais c'est tout ce dont Amazon SWF a besoin pour identifier l'activité à planifier, en supposant que l'activité ait déjà été enregistrée.

```
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" } )
```

Lorsque nous programmons une activité, Amazon SWF envoie une tâche d'activité à la liste des tâches d'activité que nous transmettons lors de la planification, signalant ainsi le début de la tâche. Nous nous intéresserons aux tâches d'activité dans la section [Troisième partie du didacticiel sur le flux de travail d'abonnement : mise en œuvre des activités](#), mais il est important de noter que nous n'exécutons pas la tâche ici. Nous indiquons uniquement à Amazon SWF qu'il doit être planifié.

La prochaine activité que nous devons aborder est l'ActivityTaskCompleted événement, qui se produit lorsqu'Amazon SWF reçoit une réponse indiquant que l'activité est terminée suite à une tâche d'activité.

```
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
```

Comme nous exécutons nos tâches de manière linéaire et qu'une seule activité est exécutée à la fois, nous allons profiter de l'occasion pour extraire la tâche terminée de la `activity_list` pile. Si une liste vide apparaît, nous savons que le flux de travail est terminé. Dans ce cas, nous signalons à Amazon SWF que notre flux de travail est terminé en appelant [complete\\_workflow\\_execution](#) sur la tâche.

Dans le cas où la liste contiendrait encore des entrées, nous allons planifier l'activité suivante qui s'y trouve (une fois encore, en dernière position). Cette fois-ci, toutefois, nous verrons si l'activité précédente a renvoyé des données de résultat à Amazon SWF une fois terminée, lesquelles sont fournies au flux de travail dans les attributs de l'événement, dans la clé facultative `result`. Si l'activité

a généré un résultat, nous le transmettons comme option `input` à la prochaine activité planifiée, avec la liste des tâches d'activité.

En récupérant les valeurs `result` des activités terminées et en définissant les valeurs `input` des activités planifiées, nous pouvons transférer les données d'une activité à l'autre, ou nous pouvons utiliser les données d'une activité pour modifier le comportement du décideur en fonction des résultats d'une activité.

Pour les besoins de ce didacticiel, ces deux types d'événements sont les plus importants pour définir le comportement de notre flux de travail. Cependant, une activité peut générer des événements autres que `ActivityTaskCompleted`. Nous allons terminer notre code de décision en fournissant un code de gestionnaire de démonstration pour les `ActivityTaskFailed` événements `ActivityTaskTimedOut` pour l'`WorkflowExecutionCompleted` événement, qui sera généré lorsque Amazon SWF traitera `complete_workflow_execution` l'appel que nous faisons lorsque nous n'avons plus d'activités à exécuter.

```
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
```

## Lancement de l'exécution du flux de travail

Avant que le flux de travail puisse rechercher des tâches de décision, nous devons lancer l'exécution du flux de travail.

Pour démarrer l'exécution du flux de travail, appelez [start\\_execution](#) sur le type de flux de travail enregistré ( ) [AWS::SimpleWorkflow::WorkflowType](#). Nous allons définir un petit wrapper à ce niveau afin d'exploiter le membre d'instance `workflow_type` que nous avons récupéré dans le constructeur de classe.

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

Une fois que le flux de travail s'exécute, les événements décision commencent à apparaître dans la liste des tâches correspondante, qui est transmise comme une option d'exécution du flux de travail dans [start\\_execution](#).

Contrairement aux options qui sont fournies lorsque le type de flux de travail est enregistré, celles qui sont transmises à `start_execution` ne sont pas considérées comme faisant partie du type de flux de travail. Vous êtes libre de les modifier pour chaque exécution du flux de travail sans avoir à changer de version du flux de travail.

Comme nous aimerions que le flux de travail commence à s'exécuter lorsque nous exécutons le fichier, ajoutez du code qui instancie la classe, puis appelle la `start_execution` méthode que nous venons de définir.

```
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
```

Pour éviter tout conflit de nom dans la liste des tâches, nous utiliserons `SecureRandom.uuid` pour générer un UUID aléatoire que nous utiliserons comme nom de la liste de tâches. Nous garantissons ainsi qu'un nom de liste de tâches différent s'applique à chaque exécution de flux de travail.

#### Note

Les listes de tâches permettent d'enregistrer les événements concernant une exécution de flux de travail. Donc, si vous utilisez la même liste des tâches pour plusieurs exécutions du même type de flux de travail, vous pouvez obtenir des événements qui ont été générés au cours d'une exécution précédente, surtout si vous avez procédé de manière quasi consécutive (ce qui est souvent le cas lorsque vous effectuez des tests, comme celui d'un nouveau code).

Pour éviter d'avoir à gérer les éléments issues des précédentes exécutions, nous pouvons utiliser une nouvelle liste des tâches pour chaque exécution, en le spécifiant lorsque nous commençons l'exécution du flux de travail.

Un peu de code est également nécessaire ici pour fournir des instructions à la personne chargée de l'exécution (vous, dans la plupart des cas) et pour fournir la version d'« activité » de la liste des tâches. Le décideur utilisera le nom de cette liste de tâches pour planifier les activités du flux de travail, tandis que la mise en œuvre des activités écoutera les événements d'activité correspondant

à cette liste pour savoir quand commencer les activités planifiées et pour fournir des informations sur l'exécution de l'activité.

Le code attend également que l'utilisateur commence à exécuter le démarreur d'activités avant de lancer l'exécution du flux de travail. Le démarreur d'activités sera donc en mesure de réagir lorsque les tâches d'activité commenceront à apparaître dans la liste des tâches fournie.

## Étapes suivantes

Vous avez implémenté le flux de travail. Vous définirez ensuite les activités et un démarreur, dans la section [Troisième partie du didacticiel sur le flux de travail d'abonnement : mise en œuvre des activités](#).

## Troisième partie du didacticiel sur le flux de travail d'abonnement : mise en œuvre des activités

Nous allons maintenant mettre en œuvre chacune des activités de notre flux de travail, en commençant par une classe de base qui fournit des fonctions courantes pour le code d'activité.

### Rubriques

- [Définition d'un type d'activité de base](#)
- [Définition GetContactActivity](#)
- [Définition SubscribeTopicActivity](#)
- [Définition WaitForConfirmationActivity](#)
- [Définition SendResultActivity](#)
- [Étapes suivantes](#)

## Définition d'un type d'activité de base

Lorsque nous avons conçu le flux de travail, nous avons identifié les activités suivantes :

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

Nous allons à présent mettre en œuvre chacune de ces activités. Étant donné que nos activités partageront certaines fonctionnalités, faisons un petit travail de base et créons un code commun qu'ils pourront partager. Nous allons l'`BasicActivity` appeler et le définir dans un nouveau fichier appelé `basic_activity.rb`.

Comme avec les autres fichiers sources, nous allons inclure `utils.rb` pour accéder à la fonction `init_domain` permettant de configurer l'exemple de domaine.

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

Ensuite, nous allons déclarer la classe de l'activité de base et certaines données courantes qui nous intéressent pour chaque activité. Nous allons enregistrer l'[AWS::SimpleWorkflow::ActivityType](#) instance, le nom et les résultats de l'activité dans les attributs de la classe.

```
class BasicActivity

  attr_accessor :activity_type
  attr_accessor :name
  attr_accessor :results

end
```

Ces attributs accèdent aux données d'instance définies dans la `initialize` méthode de la classe, qui prend un nom d'activité, ainsi qu'une version facultative et une carte d'options à utiliser lors de l'enregistrement de l'activité auprès d'Amazon SWF.

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

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

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

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

```
if @activity_type.nil?  
  # If no options were specified, use some reasonable defaults.  
  if options.nil?  
    options = {  
      # All timeouts are in seconds.  
      :default_task_heartbeat_timeout => 900,  
      :default_task_schedule_to_start_timeout => 120,  
      :default_task_schedule_to_close_timeout => 3800,  
      :default_task_start_to_close_timeout => 3600 }  
    end  
    @activity_type = @domain.activity_types.register(@name, version, options)  
  end  
end  
end
```

Comme avec l'enregistrement du type de flux de travail, si un type d'activité est déjà enregistré, nous pouvons le récupérer en examinant la collection [activity\\_types](#) du domaine. Si le type d'activité est introuvable, il sera enregistré.

En outre, comme avec les types de flux de travail, vous pouvez définir des options par défaut qui seront stockées avec votre type d'activité lorsque vous l'enregistrerez.

La dernière chose que nécessite notre activité de base est une façon uniforme de l'exécuter. Nous allons définir une méthode `do_activity` qui utilise une tâche d'activité. Comme illustré ici, nous pouvons utiliser la tâche d'activité transmise pour recevoir les données via l'attribut d'instance `input`.

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

Cela met fin au `BasicActivity` cours. Maintenant, nous allons nous en servir pour définir nos activités de manière simple et uniforme.

## Définition `GetContactActivity`

La première activité exécutée pendant l'exécution d'un flux de travail consiste `get_contact_activity` à récupérer les informations d'abonnement à la rubrique Amazon SNS de l'utilisateur.

Créez un nouveau fichier appelé `get_contact_activity.rb`, et exigez les deux `yaml`, que nous utiliserons pour préparer une chaîne à transmettre à Amazon SWF et `basic_activity.rb` que nous utiliserons comme base pour cette `GetContactActivity` classe.

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

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

Comme nous avons saisi le code d'enregistrement de l'activité `BasicActivity`, la `initialize` méthode `GetContactActivity` est assez simple. Il suffit d'appeler le constructeur de la classe de base avec le nom de l'activité, `get_contact_activity`. C'est tout ce qui est nécessaire pour enregistrer notre activité.

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

Nous allons maintenant définir la `do_activity` méthode, qui demande le numéro de and/or téléphone e-mail de l'utilisateur.

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

  input_confirmed = false
  while !input_confirmed
    puts ""
    print "Email: "
```

```
    email = $stdin.gets.strip

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

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

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

A la fin de `do_activity`, nous prenons le numéro de téléphone et l'adresse e-mail de l'utilisateur, plaçons ces informations dans une carte, puis utilisons `to_yaml` pour convertir cette carte entière en chaîne YAML. Il y a une raison importante à cela : tous les résultats que vous transmettez à Amazon SWF lorsque vous terminez une activité doivent être des chaînes de données uniquement. La capacité de Ruby à convertir facilement des objets en chaînes YAML et à les reconverter en objets est particulièrement utile dans ce cas.

Voilà ce qui marque la fin de la mise en œuvre de `get_contact_activity`. Nous utiliserons ensuite ces données dans la mise en œuvre de `subscribe_topic_activity`.

## Définition SubscribeTopicActivity

Nous allons maintenant nous intéresser à Amazon SNS et créer une activité qui utilise les informations générées `get_contact_activity` par pour abonner l'utilisateur à une rubrique Amazon SNS.

Créez un fichier appelé `subscribe_topic_activity.rb`, ajoutez les mêmes exigences que celles que nous avons utilisées pour `get_contact_activity`, déclarez votre classe et fournissez sa méthode `initialize`.

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

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

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

Maintenant que le code est en place pour configurer et enregistrer l'activité, nous allons ajouter du code pour créer une rubrique Amazon SNS. Pour ce faire, nous allons utiliser la méthode [create\\_topic](#) de l'[AWS::SNS::Client](#)objet.

Ajoutez la `create_topic` méthode à votre classe, qui prend un objet client Amazon SNS transmis.

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

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

```
end

return topic_arn
end
```

Une fois que nous avons le nom de ressource Amazon (ARN) du sujet, nous pouvons l'utiliser avec la méthode [set\\_topic\\_attributes](#) du client Amazon SNS pour définir le sujet DisplayName, qui est requis pour envoyer des SMS avec Amazon SNS.

Enfin, nous allons définir la méthode `do_activity`. Nous allons commencer par recueillir les données qui ont été transmises via l'option `input` lorsque l'activité a été planifiée. Comme nous l'avons mentionné précédemment, elles doivent être transmises sous la forme d'une chaîne que nous avons créée avec `to_yaml`. Lors de son extraction, nous utiliserons `YAML.load` pour convertir les données en objets Ruby.

Voici le début de `do_activity`, qui nous permet de récupérer les données d'entrée.

```
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))
```

Si nous n'avons reçu aucune entrée, il n'y aurait pas eu beaucoup à faire. Nous aurions tout simplement arrêté l'activité.

En supposant que tout va bien, nous continuerons à renseigner notre `do_activity` méthode, à obtenir un client Amazon SNS avec le et à la AWS SDK pour Ruby transmettre à notre `create_topic` méthode pour créer la rubrique Amazon SNS.

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

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

Deux éléments sont à ici à prendre en compte :

- Nous l'utilisons [AWS.config.with](#) pour définir la région de notre client Amazon SNS. Comme nous voulons envoyer des messages SMS, nous utilisons la région SMS que nous avons déclarée dans `utils.rb`.
- Nous enregistrons l'ARN de la rubrique dans notre carte `activity_data`. Ces données seront transmises à l'activité suivante dans notre flux de travail.

Enfin, cette activité permet d'abonner l'utilisateur à la rubrique Amazon SNS, en utilisant les points de terminaison transmis (e-mail et SMS). L'utilisateur n'est pas obligé de renseigner les deux points de terminaison, mais au moins un.

```
# 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](#) prend le sujet ARN, le protocole (que nous avons habilement déguisé en clé `activity_data` cartographique pour le point de terminaison correspondant).

Enfin, nous reconditionnons les informations pour l'activité suivante au format YAML, afin de pouvoir les renvoyer à Amazon SWF.

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

Nous avons terminé la mise en œuvre de l'activité `subscribe_topic_activity`. Ensuite, nous allons définir `wait_for_confirmation_activity`.

## Définition `WaitForConfirmationActivity`

Une fois qu'un utilisateur est abonné à une rubrique Amazon SNS, il doit encore confirmer sa demande d'abonnement. Dans ce cas, nous allons attendre que l'utilisateur confirme l'abonnement par e-mail ou par SMS.

L'activité qui attend que l'utilisateur confirme l'abonnement s'appelle `wait_for_confirmation_activity`, et nous allons la définir ici. Pour commencer, créez un fichier appelé `wait_for_confirmation_activity.rb` et configurez-le comme nous l'avons fait dans les activités précédentes.

```
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
```

Ensuite, nous allons commencer par définir la méthode `do_activity` et nous allons extraire les données d'entrée dans une variable locale appelée `subscription_data`.

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

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

Maintenant que nous avons l'ARN du sujet, nous pouvons le récupérer en créant une nouvelle instance de [AWS::SNS::Topic](#) et en lui transmettant l'ARN.

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

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

Nous allons maintenant vérifier la rubrique afin de déterminer si l'utilisateur a confirmé l'abonnement à l'aide de l'un points de terminaison. Pour que l'activité soit considérée comme ayant réussi, seul un point de terminaison doit être confirmé.

Une rubrique Amazon SNS contient une liste des [abonnements](#) à cette rubrique, et nous pouvons vérifier si l'utilisateur a confirmé un abonnement en particulier en vérifiant si l'ARN de l'abonnement est défini sur autre chose que `PendingConfirmation`

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

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

Si nous obtenons un ARN pour l'abonnement, nous l'enregistrerons dans les données de résultat de l'activité, les convertirons au format YAML et renverrons la valeur `true` à partir de `do_activity`, qui indique que l'activité s'est correctement déroulée.

Comme l'attente de la confirmation d'un abonnement peut prendre un certain temps, nous faisons parfois appel `record_heartbeat` à la tâche d'activité. Cela indique à Amazon SWF que l'activité est toujours en cours de traitement et peut également être utilisé pour fournir des mises à jour sur la progression de l'activité (si vous effectuez quelque chose, comme le traitement de fichiers, pour lequel vous pouvez signaler la progression).

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

Cela termine la boucle `while`. Si, pour une raison ou une autre, nous sortons de la boucle `while` sans succès, nous indiquerons un échec et mettrons fin à la méthode `do_activity`.

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

Nous arrivons maintenant au terme de la mise en œuvre de `wait_for_confirmation_activity`. Il ne reste plus qu'une seule activité à définir : `send_result_activity`.

## Définition SendResultActivity

Si le flux de travail a progressé jusqu'ici, nous avons correctement inscrit l'utilisateur à une rubrique Amazon SNS et l'utilisateur a confirmé son abonnement.

Notre dernière activité, `send_result_activity`, envoie à l'utilisateur une confirmation de l'abonnement à la rubrique, à l'aide de la rubrique à laquelle il s'est abonné et du point de terminaison avec lequel il a confirmé l'abonnement.

Créez un fichier appelé `send_result_activity.rb` et configurez-le comme nous l'avons fait pour toutes les autres activités.

```
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
```

Notre `do_activity` méthode commence également de la même manière, en obtenant les données d'entrée du flux de travail, en les convertissant à partir de YAML, puis en utilisant le sujet ARN pour créer une [AWS::SNS::Topic](#) instance.

```
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
```

Une fois que nous aurons la rubrique, nous allons y [publier](#) un message (et le reproduire à l'écran).

```
    @results = "Thanks, you've successfully confirmed registration, and your  
workflow is complete!"  
  
    # send the message via SNS, and also print it on the screen.  
    topic.publish(@results)  
    puts(@results)  
  
    return true  
  end  
end
```

La publication sur une rubrique Amazon SNS envoie le message que vous fournissez à tous les points de terminaison abonnés et confirmés qui existent pour cette rubrique. Donc, si l'utilisateur a confirmé l'abonnement avec à la fois un e-mail et un numéro de SMS, il reçoit deux messages de confirmation, un à chaque point de terminaison.

## Étapes suivantes

Nous avons terminé la mise en œuvre de l'activité `send_result_activity`. A présent, vous allez relier toutes ces activités dans une application d'activité qui gère les tâches d'activité et qui peut lancer les activités en réponse, comme décrit dans la section [Quatrième partie du didacticiel sur le flux de travail d'abonnement : mise en œuvre de l'observateur de tâches d'activité](#).

## Quatrième partie du didacticiel sur le flux de travail d'abonnement : mise en œuvre de l'observateur de tâches d'activité

Dans Amazon SWF, les tâches d'activité relatives à l'exécution d'un flux de travail en cours apparaissent dans la liste des tâches d'activité, qui est fournie lorsque vous planifiez une activité dans le flux de travail.

Nous allons implémenter un outil de sondage d'activité de base pour gérer ces tâches dans le cadre de notre flux de travail, et nous l'utiliserons pour lancer nos activités lorsqu'Amazon SWF place une tâche dans la liste des tâches d'activité pour démarrer l'activité.

Pour commencer, créez un fichier nommé `swf_sns_activities.rb`. Nous allons l'utiliser pour effectuer les actions suivantes :

- Instancier les classes d'activité que nous avons créées
- Enregistrez chaque activité auprès d'Amazon SWF.
- Rechercher les activités et appeler `do_activity` pour chacune d'elles lorsque son nom apparaît dans la liste des tâches d'activité

Dans le fichier `swf_sns_activities.rb`, ajoutez les instructions suivantes afin de demander chacune des classes d'activité que nous avons définies.

```
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'
```

Nous allons maintenant créer la classe et fournir un code d'initialisation.

```
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
```

En plus d'enregistrer le domaine et la liste de tâches, ce code instancie chacune des classes d'activité que nous avons créées. Comme chaque classe enregistre son activité associée (référez-vous à ce code `basic_activity.rb` si vous devez consulter ce code), cela suffit pour informer Amazon SWF de toutes les activités que nous allons exécuter.

Nous stockons chaque activité instanciée dans une carte en utilisant le nom de l'activité (par exemple, `get_contact_activity`) comme clé. Cela nous permettra de rechercher les activités facilement dans le code de l'observateur d'activité, que nous allons définir ensuite.

Créez une méthode dénommée `poll_for_activities` et appelez [poll](#) au niveau des [tâches d'activité](#) détenues par le domaine afin de les récupérer.

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

Nous pouvons obtenir le nom de l'activité à partir du membre [activity\\_type](#) de la tâche. Ensuite, nous allons utiliser le nom d'activité associé à cette tâche pour rechercher la classe dans laquelle exécuter `do_activity`, en le transmettant à la tâche (qui inclut toutes les données d'entrée qui doivent être transférées à l'activité).

```
# 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
  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
```

Le code seulement attend que `do_activity` termine, puis appelle soit [terminée !](#) soit [échouer !](#) sur la tâche en fonction du code de retour.

#### Note

Ce code sort du sondeur une fois que l'activité finale a été lancée, car il a terminé sa mission et a lancé toutes les activités. Dans votre propre code Amazon SWF, si vos activités sont susceptibles d'être réexécutées, vous souhaitez peut-être que le sondeur d'activité fonctionne indéfiniment.

C'est la fin du code de notre `ActivitiesPoller` classe, mais nous allons ajouter un peu plus de code à la fin du fichier pour permettre à l'utilisateur de l'exécuter depuis la ligne de commande.

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

Si l'utilisateur exécute le fichier à partir de la ligne de commande (en lui transmettant une liste de tâches activité comme premier argument), ce codeinstanciera la classe de l'observateur et lancera la recherche d'activités. Une fois que l'observateur a terminé (après avoir lancé l'activité finale), il nous suffit d'imprimer un message et de sortir.

Voilà ce qui marque la fin de l'observateur d'activité. Il ne vous reste plus qu'à exécuter le code et à voir comment il fonctionne, comme décrit dans la section [Didacticiel sur le flux de travail d'abonnement : exécution du flux de travail](#).

## Didacticiel sur le flux de travail d'abonnement : exécution du flux de travail

Maintenant que vous avez terminé la mise en œuvre du flux de travail, des activités et des observateurs de flux de travail et d'activités, vous êtes prêt à exécuter le flux de travail.

Si ce n'est pas déjà fait, vous devrez fournir vos clés AWS d'accès dans le `aws-config.txt` fichier, comme [Configuration de la AWS session](#) dans la partie 1 du didacticiel.

A présent, accédez à votre ligne de commande et activez le répertoire dans lequel se trouvent les fichiers sources du didacticiel. Les fichiers suivants devraient s'y trouver :

```
.
|-- 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
```

Désormais, démarrez le flux de travail avec la commande suivante.

```
ruby swf_sns_workflow.rb
```

Cette opération lance le flux de travail et devrait imprimer un message avec une ligne que vous pouvez copier et coller dans une nouvelle fenêtre de ligne de commande (voire sur un autre ordinateur, si vous y avez copié les fichiers sources du didacticiel).

```
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...
```

Le code du flux de travail attend patiemment que vous démarriez l'observateur d'activités dans une fenêtre séparée.

Ouvrez une nouvelle fenêtre de ligne de commande, activez à nouveau le répertoire dans lequel les fichiers sources se trouvent, puis utilisez la commande fournie par le fichier `swf_sns_workflow.rb` pour démarrer l'observateur d'activités. Par exemple, si vous avez reçu le résultat précédent, vous tapez (ou collez) ce qui suit.

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

Une fois que vous commencez à exécuter l'observateur d'activités, il se met à générer des informations sur l'enregistrement des activités.

```
** 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
```

Vous pouvez maintenant revenir à la fenêtre de ligne de commande d'origine et appuyer sur Entrée pour lancer l'exécution du flux de travail. Cette action enregistre le flux de travail et planifie la première activité.

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

Retournez dans la fenêtre où l'observateur d'activités est exécuté. Le résultat de la première activité qui en cours d'exécution s'affiche, avec une invite vous permettant de saisir votre adresse e-mail ou le numéro de téléphone SMS. Renseignez l'une de ces informations ou les deux, puis confirmez-les.

```
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

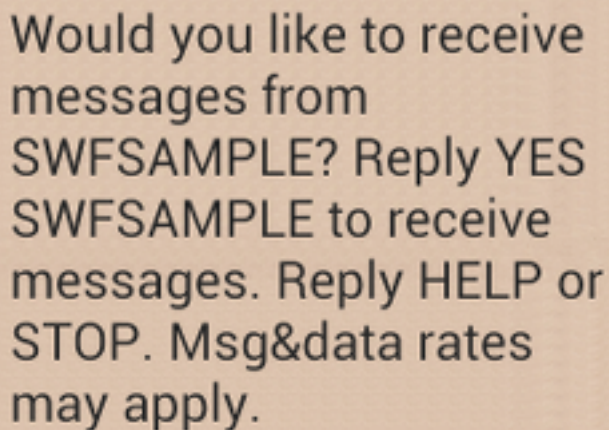
phone: 12065550101

Is this correct? (y/n): y

### Note

Le numéro de téléphone fourni ici est fictif. Il est utilisé uniquement à des fins d'illustration. Indiquez votre propre numéro de téléphone et votre adresse e-mail.

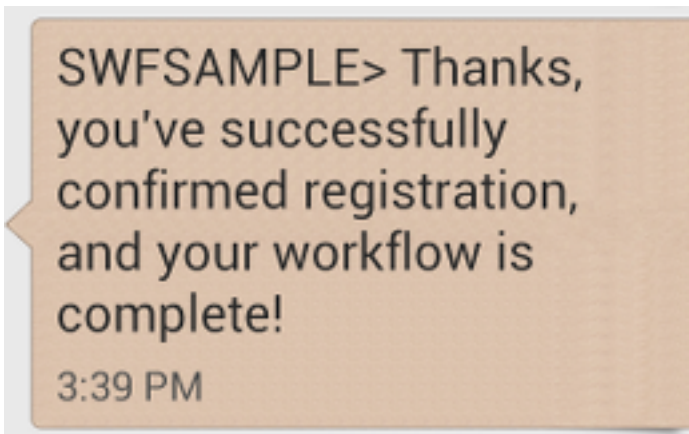
Peu après avoir saisi ces informations, vous devriez recevoir un e-mail ou un SMS d'Amazon SNS vous demandant de confirmer votre inscription à la rubrique. Si vous avez saisi un numéro de SMS, ce qui apparaît sur votre téléphone ressemble à ceci.



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

Si vous répondez à ce message par YES, vous obtiendrez la réponse que nous avons fournie dans `send_result_activity`.



Avez-vous vu ce qui s'est passé dans la fenêtre de ligne de commande pendant ce temps ? L'observateur de flux de travail et l'observateur d'activités étaient affairés.

Voici le résultat de l'observateur de flux de travail.

```
** 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...
```

Voici le résultat de l'observateur d'activités, qui a été généré en même temps dans une autre fenêtre de ligne de commande.

```
++ 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!
```

Félicitations, votre flux de travail est terminé, ce qui marque également la fin de ce didacticiel !

Vous souhaitez peut-être exécuter le flux de travail à nouveau pour voir comment les délais fonctionnent ou pour saisir des données différentes. N'oubliez pas qu'une fois que vous vous abonnez à une rubrique, vous y restez abonné tant que vous n'annulez pas l'abonnement. La

réexécution du flux de travail avant de vous désinscrire aux sujets se traduira probablement par un succès automatique, car votre abonnement `wait_for_confirmation_activity` sera déjà confirmé.

Pour vous désabonner de la rubrique Amazon SNS

- Répondez au SMS par la négative (envoyez STOP).
- Choisissez le lien de désabonnement que vous avez reçu dans votre e-mail.

Vous pouvez maintenant vous ré-abonner à la rubrique.

## Comment procéder ensuite ?

Ce didacticiel a abordé de nombreux sujets, mais vous pouvez encore en apprendre davantage sur Amazon SWF ou Amazon SNS. AWS SDK pour Ruby Pour consulter plus d'informations, ainsi que d'autres exemples, reportez-vous à la documentation officielle de chaque produit :

- [Documentation AWS SDK pour Ruby](#)
- [Documentation relative au service de notification Amazon Simple](#)
- [Documentation du service Amazon Simple Workflow](#)

# Utilisation de la console Amazon SWF

La console Amazon SWF fournit des options pour configurer, lancer et gérer les exécutions de flux de travail.

Avec la console Amazon SWF, vous pouvez :

- Enregistrement des domaines de flux de travail
- Enregistrez les types de flux de travail et les types d'activités.
- Démarrez, visualisez, signalez, annulez, arrêtez et redémarrez les exécutions de flux de travail.

## Enregistrement d'un domaine

Les flux de travail s'exécutent dans une AWS ressource appelée domaine, qui contrôle l'étendue du flux de travail. Un AWS compte peut comporter plusieurs domaines, chacun pouvant contenir plusieurs flux de travail, mais les flux de travail des différents domaines ne peuvent pas interagir.

L'enregistrement de domaine est la seule fonctionnalité initialement disponible dans la console. Une fois qu'au moins un domaine est enregistré, vous pouvez effectuer les actions suivantes pour le domaine :

- Enregistrez les types de flux de travail et d'activités.
- Lancement des exécutions de flux de travail
- Annulation, arrêt et envoi des signaux aux exécutions de flux de travail en cours
- Redémarrage des exécutions de flux de travail fermées

Vous pouvez également effectuer des actions de gestion de domaine, telles que la dépréciation ou la désapprobation de domaines.

Une fois qu'un domaine est obsolète, vous ne pouvez pas l'utiliser pour créer de nouvelles exécutions de flux de travail ou enregistrer de nouveaux flux de travail. La dépréciation d'un domaine entraîne également la désapprobation de toutes les activités et de tous les flux de travail enregistrés dans le domaine. Les exécutions lancées avant que le domaine ne soit devenu obsolète continuent de s'exécuter.

Après avoir désapprouvé un domaine précédemment obsolète, vous pouvez recommencer à l'utiliser pour enregistrer les types de flux de travail et démarrer de nouvelles exécutions de flux de travail.

Pour plus d'informations sur ces actions de gestion de domaine, reportez-vous [DeprecateDomain](#) aux sections et [UndeprecateDomain](#).

## Enregistrement des types de flux de travail

Vous pouvez enregistrer les types de flux de travail dans la console Amazon SWF après avoir enregistré au moins un domaine.

Un type de flux de travail est un ensemble de types d'activités qui répondent à un objectif et contiennent la logique qui coordonne les activités. Les types de flux de travail coordonnent et gèrent l'exécution d'activités qui peuvent être exécutées de manière asynchrone sur plusieurs appareils informatiques et proposent des méthodes de traitement séquentielles et parallèles.

Pour enregistrer un type de flux de travail Amazon SWF à l'aide de la console

1. Ouvrez le domaine dans lequel vous souhaitez enregistrer un flux de travail.
2. Choisissez Enregistrer, puis sélectionnez Enregistrer le flux de travail.
3. Sur la page Enregistrer le flux de travail, entrez le nom du flux de travail et la version du flux de travail. Facultativement, vous pouvez également spécifier une [liste de tâches par défaut](#) qui sera utilisée pour planifier les tâches de décision pour les exécutions de ce flux de travail.
4. (Facultatif) Choisissez les options avancées pour spécifier les détails suivants pour votre flux de travail :
  - [Priorité de tâche](#) par défaut : priorité de tâche par défaut à attribuer au flux de travail.
  - [Délai d'exécution par défaut entre le début et la fin de l'exécution](#) : durée maximale par défaut pour les exécutions de ce flux de travail.
  - [Délai d'expiration par défaut entre le début et la fin des tâches](#) : durée maximale par défaut des tâches de décision pour ce flux de travail.
  - [Politique relative aux enfants par défaut](#) : politique par défaut à utiliser pour les exécutions des flux de travail enfants.
  - Rôle [Lambda par défaut : rôle](#) IAM par défaut associé à ce flux de travail.
5. Choisissez Enregistrer le flux de travail.

# Enregistrement des types d'activités

Les activités sont des tâches que vous souhaitez que votre type de flux de travail coordonne et exécute (par exemple : vérifier la commande du client, débiter une carte de crédit, etc.). L'ordre dans lequel les activités sont effectuées est déterminé par la logique de coordination du type de flux de travail.

Vous pouvez enregistrer des types d'activités après l'enregistrement d'au moins un domaine.

Pour enregistrer un type d'activité Amazon SWF à l'aide de la console

1. Ouvrez le domaine dans lequel vous souhaitez enregistrer une activité.
2. Choisissez Enregistrer, puis sélectionnez Enregistrer l'activité.
3. Sur la page Enregistrer une activité, entrez le [nom de l'activité](#) et la [version de l'activité](#).  
Facultativement, vous pouvez également spécifier une [liste de tâches par défaut](#) qui sera utilisée pour planifier les tâches de cette activité.
4. (Facultatif) Choisissez les options avancées pour spécifier les détails suivants pour votre activité :
  - [Priorité de tâche](#) par défaut : priorité de tâche par défaut à attribuer à l'activité.
  - [Calendrier des tâches par défaut pour le début du délai](#) d'expiration : durée maximale par défaut pendant laquelle une tâche de cette activité peut attendre avant d'être affectée à un travailleur.
  - [Délai d'expiration par défaut entre le début et la fin de la tâche](#) : durée maximale par défaut qu'un collaborateur peut prendre pour traiter les tâches de cette activité.
  - [Calendrier des tâches par défaut pour clôturer le délai](#) d'expiration : durée maximale par défaut pour une tâche de cette activité.
  - [Délai d'expiration du rythme cardiaque d'une tâche](#) par défaut : durée maximale par défaut avant laquelle un collaborateur traitant une tâche de ce type doit signaler la progression en appelant. [RecordActivityTaskHeartbeat](#)
5. Choisissez Enregistrer une activité.

## Démarrage d'un flux de travail

Vous pouvez démarrer l'exécution d'un flux de travail depuis la console Amazon SWF. Vous ne pouvez pas démarrer l'exécution d'un flux de travail tant que vous n'avez pas enregistré au moins un flux de travail.

### Pour lancer une exécution de flux de travail à l'aide de la console

1. Ouvrez la console Amazon SWF, puis dans le volet de navigation de gauche, sélectionnez **Domains**.
2. Sous le nom de domaine, sélectionnez **Workflows**.
3. Sur la page Flux de travail, choisissez le flux de travail que vous souhaitez exécuter.
4. Choisissez **Start execution** (Démarrer l'exécution).
5. Sur la page Démarrer l'exécution, entrez le [nom du flux](#) de travail et l'ID d'exécution pour identifier votre exécution par un nom. Facultativement, vous pouvez également spécifier une [liste de tâches](#) qui sera utilisée pour les tâches de décision générées pour l'exécution de ce flux de travail.
6. (Facultatif) Choisissez les options avancées pour spécifier les détails suivants pour l'exécution de votre flux de travail :
  - [Priorité de tâche](#) : priorité de tâche à utiliser pour l'exécution de ce flux de travail.
  - [Délai d'expiration du début à la fin de l'exécution](#) : durée totale de l'exécution de ce flux de travail.
  - [Délai entre le début et la fin des tâches](#) : durée maximale des tâches de décision pour l'exécution de ce flux de travail.
  - [Politique relative aux enfants](#) : politique à utiliser pour les exécutions de flux de travail enfants de cette exécution de flux de travail si celle-ci est interrompue, en appelant l'[TerminateWorkflowExecution](#) action explicitement ou en raison d'un délai expiré.
  - Rôle [Lambda : rôle](#) IAM à associer à l'exécution de ce flux de travail.
7. Choisissez **Start execution** (Démarrer l'exécution).

## Gestion des exécutions de flux de travail

Vous pouvez filtrer les exécutions de vos flux de travail par nom, statut, ID et balise. Vous pouvez envoyer des signaux avec des entrées dans des exécutions de flux de travail actives. Si vous

devez annuler ou mettre fin à un flux de travail, vous pouvez utiliser l'option Essayer d'annuler. Il est préférable d'annuler plutôt que de mettre fin à l'exécution d'un flux de travail, car l'annulation donne au flux de travail la possibilité d'effectuer toutes les tâches de nettoyage, puis de se fermer correctement.

Dans la console, vous pouvez gérer les exécutions de flux de travail actuellement and/or fermées.

Pour gérer les exécutions de vos flux de travail

1. Ouvrez un domaine pour gérer ses exécutions de flux de travail.
2. Choisissez Find Execution.
3. Sur la page Exécutions du flux de travail, choisissez Filtrer les exécutions par propriété, puis sous Propriétés, choisissez l'un des filtres suivants :

Choix	Pour appliquer ce filtre
Flux de travail	<p>Choisissez ce filtre pour répertorier les exécutions d'un flux de travail spécifique. Par exemple, pour visualiser les exécutions <code>fiction-books-order-workflow</code> , procédez comme suit :</p> <ol style="list-style-type: none"><li>1. Choisissez Workflow.</li><li>2. Sous Opérateurs, sélectionnez Égaux.</li><li>3. Sous Workflows, sélectionnez <code>fiction-books-order-workflow</code>.</li><li>4. (Facultatif) Choisissez Effacer les filtres pour supprimer le filtre et lancer une nouvelle recherche d'exécutions.</li></ol>
Statut	<p>Choisissez ce filtre pour répertorier les exécutions ayant un statut spécifique. Par exemple, pour afficher les exécutions ayant le statut Echec, procédez comme suit :</p> <ol style="list-style-type: none"><li>1. Choisissez Status.</li><li>2. Sous Opérateurs, sélectionnez Égaux.</li><li>3. Sous Statuts, sélectionnez Echec.</li><li>4. (Facultatif) Choisissez Effacer les filtres pour supprimer le filtre et lancer une nouvelle recherche d'exécutions.</li></ol>

Choix	Pour appliquer ce filtre
ID d'exécution	<p>Choisissez ce filtre pour afficher l'exécution d'un flux de travail en fonction de son ID. Par exemple, pour afficher l'exécution avec un ID <code>fiction-books-order-category1</code> , procédez comme suit :</p> <ol style="list-style-type: none"> <li>1. Choisissez Execution ID.</li> <li>2. Sous Opérateurs, sélectionnez Égaux.</li> <li>3. Sous Exécution IDs, choisissez <code>fiction-books-order-category1</code>.</li> <li>4. (Facultatif) Choisissez Effacer les filtres pour supprimer le filtre et lancer une nouvelle recherche d'exécutions.</li> </ol>
Balise	<p>Choisissez ce filtre pour répertorier les exécutions avec une balise spécifique. Par exemple, pour afficher les exécutions avec le statut <code>purchaseOrder</code> , procédez comme suit :</p> <ol style="list-style-type: none"> <li>1. Choisissez Tag.</li> <li>2. Sous Opérateurs, sélectionnez Égaux.</li> <li>3. Sous Tag, choisissez <code>PurchaseOrder</code>.</li> <li>4. (Facultatif) Choisissez Effacer les filtres pour supprimer le filtre et lancer une nouvelle recherche d'exécutions.</li> </ol>

4. (Facultatif) Après avoir appliqué le filtre requis pour répertorier les exécutions de flux de travail, vous pouvez effectuer les opérations suivantes sur une exécution active :
  - Signal — Utilisez cette option pour envoyer des données supplémentaires à un flux de travail en cours d'exécution. Pour cela :
    1. Choisissez l'exécution à laquelle vous souhaitez envoyer des données supplémentaires.
    2. Choisissez Signal, puis spécifiez les données dans la boîte de dialogue d'exécution du signal.
    3. Choisissez Signal.
  - Essayer d'annuler : utilisez cette option pour essayer d'annuler l'exécution d'un flux de travail. Il est préférable d'annuler une exécution de flux de travail plutôt que de la résilier. L'annulation permet à l'exécution de flux de travail d'effectuer des tâches de nettoyage, puis de se fermer correctement.
    1. Choisissez l'exécution que vous souhaitez annuler.

2. Choisissez Essayer d'annuler.
  - Terminer : utilisez cette option pour mettre fin à l'exécution d'un flux de travail. Notez qu'il est préférable d'annuler une exécution de flux de travail plutôt que de la résilier.
    1. Choisissez l'exécution à laquelle vous souhaitez mettre fin.
    2. Pour la politique relative aux enfants, assurez-vous que l'option Terminate est sélectionnée.
    3. (Facultatif) Spécifiez la raison et les détails de la fin de l'exécution.
    4. Sélectionnez Résilier.
5. (Facultatif) Réexécuter : utilisez cette option pour réexécuter une exécution de flux de travail fermée.
  1. Dans la liste des exécutions de flux de travail, sélectionnez celle que vous souhaitez ré-exécuter. Lorsque vous sélectionnez une exécution fermée, le bouton Réexécuter est activé. Choisissez Re-run.
  2. Sur la page Réexécuter l'exécution, spécifiez les détails de l'exécution du flux de travail comme indiqué dans [Démarrage d'un flux de travail](#).

# Concepts de flux de travail de base dans Amazon SWF

## Note

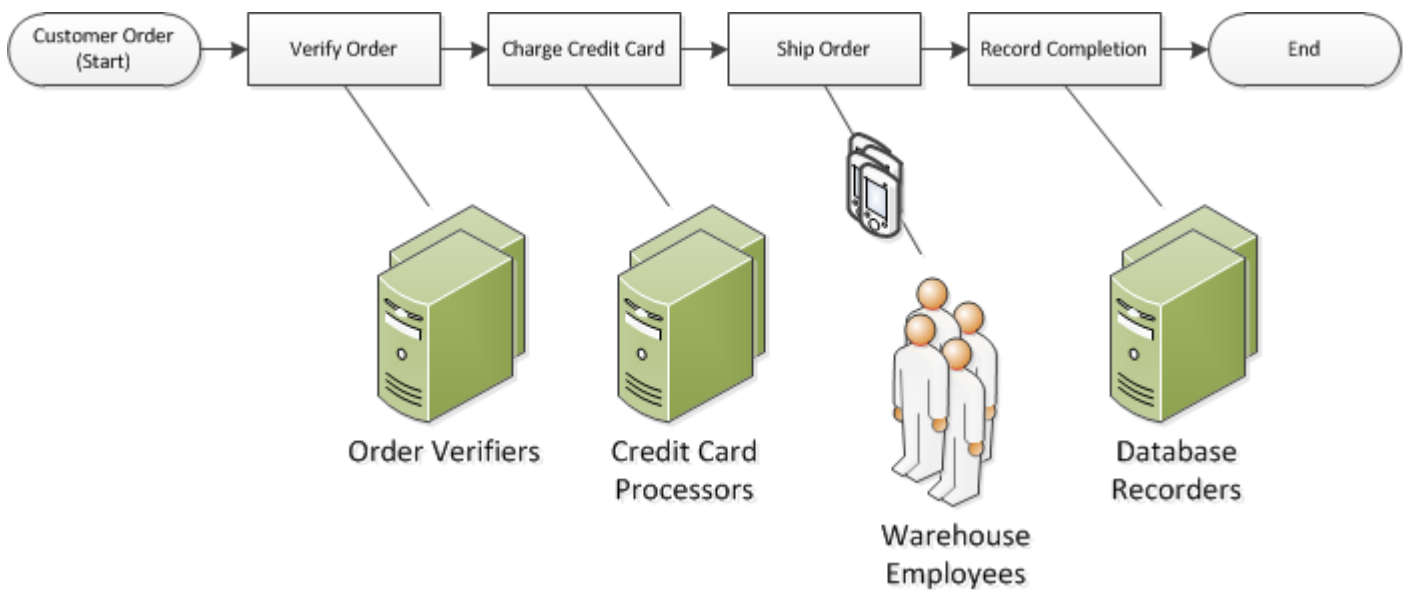
Les concepts présentés dans ce chapitre fournissent une vue d'ensemble du service Amazon Simple Workflow et décrivent ses principales fonctionnalités. Si vous recherchez des exemples, consultez [Utilisation d'Amazon SWF APIs](#).

À l'aide d'Amazon Simple Workflow Service (Amazon SWF), vous pouvez implémenter des applications asynchrones distribuées en tant que flux de travail. Les flux de travail permettent de coordonner et de gérer l'exécution des activités qui peuvent être exécutées de façon asynchrone sur plusieurs équipements informatiques et qui peuvent comporter à la fois le traitement séquentiel et parallèle.

Lorsque vous créez un flux de travail, vous analysez votre application pour identifier son composant tâches. Dans Amazon SWF, ces tâches sont représentées par des activités. L'ordre dans lequel les activités sont exécutées est déterminé par la logique de coordination du flux de travail.

Exemple de flux de travail pour une application de commerce électronique

La figure suivante montre un flux de travail de traitement des commandes de commerce électronique impliquant à la fois des personnes et des processus automatisés :



Le flux de travail des applications de commerce électronique commence lorsqu'un client passe une commande et comprend quatre tâches :

1. Vérifier la commande
2. Si la commande est valide, facturer le client
3. Si le paiement est effectué, expédier la commande
4. Si la commande est livrée, enregistrer les détails de cette dernière

Les tâches de ce flux de travail sont séquentielles : la commande doit être confirmée avant de pouvoir facturer la carte de crédit ; la carte de crédit doit être débitée avant de pouvoir livrer la commande ; et la commande doit être livrée avant de pouvoir être enregistrée. Néanmoins, étant donné qu'Amazon SWF prend en charge les processus distribués, ces tâches peuvent être effectuées à différents endroits. Si les tâches doivent être programmées, elles peuvent également être écrites dans différents langages de programmation ou à l'aide de différents outils.

Outre le traitement séquentiel des tâches, Amazon SWF prend également en charge les flux de travail avec traitement parallèle des tâches. Les tâches parallèles sont effectuées en même temps et peuvent être accomplies indépendamment par différentes applications ou personnes. Le flux de travail prend des décisions sur la procédure à suivre à une fois qu'une ou plusieurs tâches parallèles ont été terminées.

### Concepts supplémentaires

- [Création d'un flux de travail dans Amazon SWF](#)
- [Exécution de flux de travail dans Amazon SWF](#)
- [Historique du flux de travail dans Amazon SWF](#)
- [Identifiants d'objets dans Amazon SWF](#)
- [Domaines dans Amazon SWF](#)
- [Acteurs d'Amazon SWF](#)
- [Tâches dans Amazon SWF](#)
- [Listes de tâches dans Amazon SWF](#)
- [Clôture de l'exécution du flux de travail dans Amazon SWF](#)
- [Cycle de vie d'un flux de travail Amazon SWF](#)
- [Sondage de tâches dans Amazon SWF](#)

# Création d'un flux de travail dans Amazon SWF

La création d'un flux de travail séquentiel base implique les étapes suivantes.

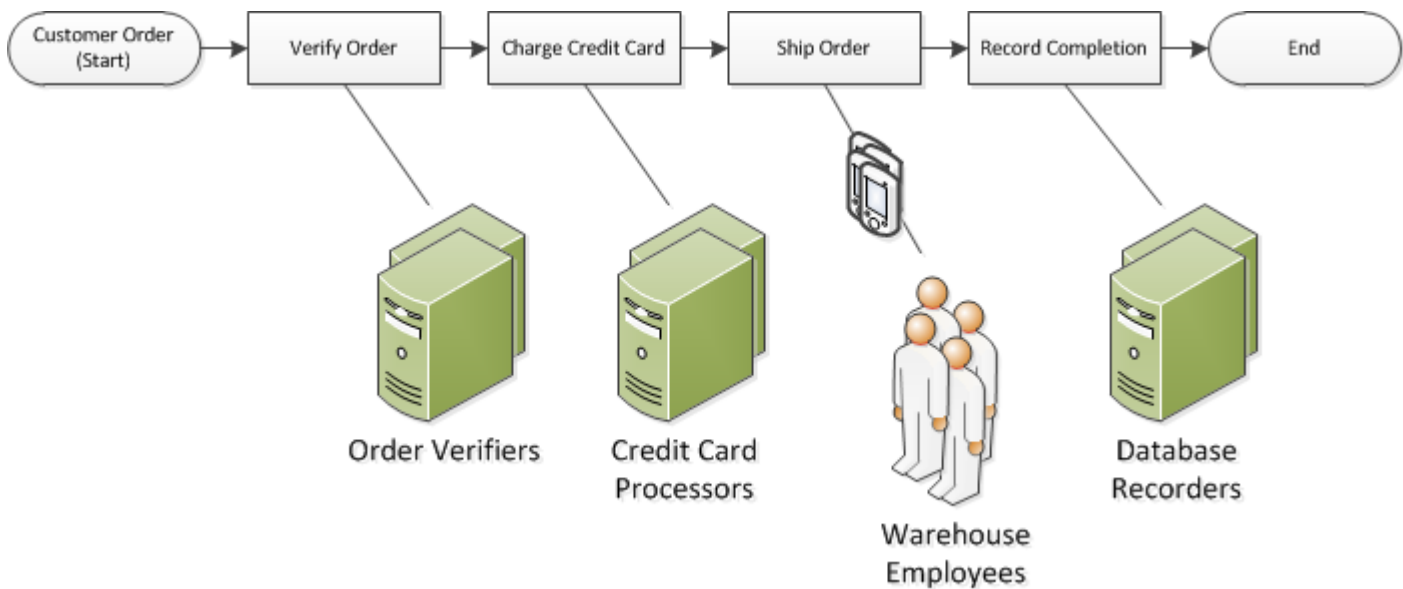
- Modélisation d'un flux de travail et enregistrement de son type et de ses types d'activité
- Développement et lancement des outils de traitement d'activité qui effectuent les tâches d'activité
- Développement et lancement des décideurs qui utilisent l'historique du flux de travail pour déterminer comment procéder ensuite
- Développement et lancement des démarreurs de flux de travail (applications qui lancent les exécutions de flux de travail)

## Modélisation de votre flux de travail et de ses activités

Pour utiliser Amazon SWF, modélisez les étapes logiques de votre application sous forme d'activités. Une activité représente une étape logique unique ou une tâche de votre flux de travail. Par exemple, l'autorisation d'une carte de crédit est une activité qui consiste à fournir un numéro de carte de crédit et d'autres informations, et à recevoir un code d'approbation ou un message indiquant que la carte a été refusée.

Outre les activités, vous devez également définir la logique de coordination qui gère les points de décision. Par exemple, la logique de coordination peut planifier une activité de suivi différente selon que la carte de crédit a été approuvée ou refusée.

La figure suivante illustre un exemple de flux de travail séquentiel pour une commande client avec quatre activités (confirmation de la commande, débit de la carte de crédit, livraison de la commande et consignation de l'achèvement de la tâche).



## Exécution de flux de travail dans Amazon SWF

Une fois que la logique de coordination et les activités ont été conçues, vous enregistrez ces composants en tant que types de flux de travail et d'activités auprès d'Amazon SWF. Lors de l'enregistrement, vous spécifiez un nom, une version et des valeurs de configuration par défaut pour chaque type.

Seuls les types de flux de travail et d'activité enregistrés peuvent être utilisés avec Amazon SWF. Dans l'exemple du commerce électronique, vous devez enregistrer le type `CustomerOrder` de flux de travail ainsi que les types `VerifyOrder`, `ChargeCreditCard`, `ShipOrder`, et `RecordCompletion` d'activité.

Après avoir enregistré votre type de flux de travail, vous pouvez l'exécuter aussi souvent que vous le voulez. Une exécution de flux de travail est une instance en cours d'exécution d'un flux de travail.

Une exécution de flux de travail peut être lancée par n'importe quel processus ou n'importe quelle application, même par une autre exécution de flux de travail. Dans l'exemple de commerce en ligne, une nouvelle exécution de flux de travail se lance à chaque commande client. Le type d'application qui lance le flux de travail dépend de la manière dont le client passe la commande. Le flux de travail peut être initié par un site Web, une application mobile ou un représentant du service client utilisant une application d'entreprise interne.

Avec Amazon SWF, vous pouvez associer un identifiant, appelé un `workflowId` travail, à vos exécutions de flux de travail, afin d'intégrer vos identifiants commerciaux existants dans votre flux de travail. Dans l'exemple du commerce en ligne, l'exécution de chaque flux de travail peut être identifiée à l'aide du numéro de facture du client.

Outre l'identifiant que vous fournissez, Amazon SWF associe un identifiant unique généré par le système, à chaque exécution de flux de travail `runId`. Amazon SWF n'autorise qu'une seule exécution de flux de travail avec cet identifiant à la fois ; bien que vous puissiez avoir plusieurs exécutions de flux de travail du même type de flux de travail, chaque exécution de flux de travail possède une exécution distincte. `runId`

## Historique du flux de travail dans Amazon SWF

Amazon SWF enregistre la progression de chaque exécution du flux de travail dans l'historique du flux de travail, c'est-à-dire un enregistrement détaillé, complet et cohérent de chaque événement survenu depuis le début de l'exécution du flux de travail.

Un événement représente un changement discret de l'état d'exécution de votre flux de travail, tel qu'une nouvelle activité planifiée ou la fin d'une activité en cours. L'historique du flux de travail contient tous les événements qui entraînent la modification de l'état de l'exécution du flux de travail, telle que les activités planifiées et terminées, l'expiration des tâches et les signaux.

Généralement, les opérations qui ne modifient pas l'état de l'exécution du flux de travail n'apparaissent pas dans l'historique de ce dernier. Par exemple, l'historique du flux de travail ne montre pas les tentatives de recherche de tâches ni l'utilisation des opérations de visibilité.

L'historique du flux de travail présente plusieurs avantages clés :

- Les applications peuvent être apatrides, car toutes les informations relatives à l'exécution d'un flux de travail sont stockées dans l'historique du flux de travail.
- Pour chaque exécution de flux de travail, l'historique fournit un enregistrement des activités qui ont été planifiées, de leur état actuel et de leurs résultats. L'exécution du flux de travail utilise ces informations pour déterminer les étapes suivantes.
- L'historique fournit une piste d'audit détaillée que vous pouvez utiliser pour surveiller les exécutions de flux de travail en cours et pour vérifier les exécutions de flux de travail terminées.

Voici une vue conceptuelle de l'historique de flux de travail du commerce en ligne :

```
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
```

Dans l'exemple précédent, la commande est en attente d'envoi. Dans l'exemple suivant, la commande est terminée. Comme l'historique du flux de travail est cumulé, les événements les plus récents s'ajoutent :

```
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
```

Du point de vue de la programmation, les événements de l'historique d'exécution du flux de travail sont représentés sous forme d'objets JSON ( JavaScript Object Notation). L'historique lui-même est un tableau JSON de ces objets. Chaque événement comporte les éléments suivants :

- Un type, tel que [WorkflowExecutionStarted](#) ou [ActivityTaskCompleted](#)
- Un horodatage au format horaire Unix

- Un ID qui identifie l'événement de manière unique

En outre, chaque type d'événement inclut un ensemble distinct d'attributs descriptifs qui lui sont spécifiques. Par exemple, l'`ActivityTaskCompleted` événement possède des attributs qui contiennent les IDs événements correspondant à l'heure à laquelle la tâche d'activité a été planifiée et à laquelle elle a été démarrée, ainsi qu'un attribut contenant les données de résultat.

Vous pouvez obtenir une copie de l'état actuel de l'historique d'exécution du flux de travail en utilisant cette [GetWorkflowExecutionHistory](#) action. En outre, dans le cadre de l'interaction entre Amazon SWF et le décideur de votre flux de travail, celui-ci reçoit régulièrement des copies de l'historique.

Voici un extrait d'historique d'exécution du flux de travail au format JSON.

```
[ {
  "eventId": 11,
  "eventTimestamp": 1326671603.102,
  "eventType": "WorkflowExecutionTimedOut",
  "workflowExecutionTimedOutEventAttributes": {
    "childPolicy": "TERMINATE",
    "timeoutType": "START_TO_CLOSE"
  }
}, {
  "decisionTaskScheduledEventAttributes": {
    "startToCloseTimeout": "600",
    "taskList": {
      "name": "specialTaskList"
    }
  },
  "eventId": 10,
  "eventTimestamp": 1326670566.124,
  "eventType": "DecisionTaskScheduled"
}, {
  "activityTaskTimedOutEventAttributes": {
    "details": "Waiting for confirmation",
    "scheduledEventId": 8,
    "startedEventId": 0,
    "timeoutType": "SCHEDULE_TO_START"
  },
  "eventId": 9,
  "eventTimestamp": 1326670566.124,
  "eventType": "ActivityTaskTimedOut"
}, {
```

```
"activityTaskScheduledEventAttributes": {
  "activityId": "verification-27",
  "activityType": {
    "name": "activityVerify",
    "version": "1.0"
  },
  "control": "digital music",
  "decisionTaskCompletedEventId": 7,
  "heartbeatTimeout": "120",
  "input": "5634-0056-4367-0923,12/12,437",
  "scheduleToCloseTimeout": "900",
  "scheduleToStartTimeout": "300",
  "startToCloseTimeout": "600",
  "taskList": {
    "name": "specialTaskList"
  }
},
"eventId": 8,
"eventTimestamp": 1326670266.115,
"eventType": "ActivityTaskScheduled"
}, {
  "decisionTaskCompletedEventAttributes": {
    "executionContext": "Black Friday",
    "scheduledEventId": 5,
    "startedEventId": 6
  },
  "eventId": 7,
  "eventTimestamp": 1326670266.103,
  "eventType": "DecisionTaskCompleted"
}, {
  "decisionTaskStartedEventAttributes": {
    "identity": "Decider01",
    "scheduledEventId": 5
  },
  "eventId": 6,
  "eventTimestamp": 1326670161.497,
  "eventType": "DecisionTaskStarted"
}, {
  "decisionTaskScheduledEventAttributes": {
    "startToCloseTimeout": "600",
    "taskList": {
      "name": "specialTaskList"
    }
  }
},
```

```
    "eventId": 5,
    "eventTimestamp": 1326668752.66,
    "eventType": "DecisionTaskScheduled"
  }, {
    "decisionTaskTimedOutEventAttributes": {
      "scheduledEventId": 2,
      "startedEventId": 3,
      "timeoutType": "START_TO_CLOSE"
    },
    "eventId": 4,
    "eventTimestamp": 1326668752.66,
    "eventType": "DecisionTaskTimedOut"
  }, {
    "decisionTaskStartedEventAttributes": {
      "identity": "Decider01",
      "scheduledEventId": 2
    },
    "eventId": 3,
    "eventTimestamp": 1326668152.648,
    "eventType": "DecisionTaskStarted"
  }, {
    "decisionTaskScheduledEventAttributes": {
      "startToCloseTimeout": "600",
      "taskList": {
        "name": "specialTaskList"
      }
    },
    "eventId": 2,
    "eventTimestamp": 1326668003.094,
    "eventType": "DecisionTaskScheduled"
  }
]
```

Pour obtenir une liste détaillée des différents types d'événements pouvant apparaître dans l'historique d'exécution du flux de travail, consultez le type de [HistoryEvent](#) données dans le manuel Amazon Simple Workflow Service API Reference.

Amazon SWF stocke l'historique complet de toutes les exécutions de flux de travail pendant un nombre de jours configurable après la fin de l'exécution. Cette période, qui est appelée la période de conservation de l'historique du flux de travail, est spécifiée lorsque vous enregistrez un domaine pour votre flux de travail. Nous aborderons les domaines plus en détail ultérieurement dans cette section.

## Identifiants d'objets dans Amazon SWF

La liste suivante décrit comment les objets Amazon SWF, tels que les exécutions de flux de travail, sont identifiés de manière unique.

- **Type de flux de travail** : un type de flux de travail enregistré est identifié par son domaine, son nom et sa version. Les types de flux de travail sont spécifiés dans l'appel à `RegisterWorkflowType`.
- **Type d'activité** : un type d'activité enregistré est identifié par son domaine, son nom et sa version. Les types d'activité sont spécifiés dans l'appel à `RegisterActivityType`.
- **Tâches de décision et tâches d'activité** — Chaque tâche de décision et tâche d'activité est identifiée par un jeton de tâche unique. Le jeton de tâche est généré par Amazon SWF et est renvoyé avec d'autres informations sur la tâche dans la réponse de `PollForDecisionTask` ou `PollForActivityTask`. Même si le jeton est plus couramment utilisé par le processus qui a reçu la tâche, ce dernier peut transmettre le jeton à un autre processus, qui peut ensuite signaler la fin ou l'échec de la tâche.
- **Exécution du flux de travail** : une seule exécution d'un flux de travail est identifiée par le domaine, l'ID du flux de travail et l'ID d'exécution. Les deux premiers sont des paramètres transmis à [StartWorkflowExecution](#). L'ID de série est renvoyé par `StartWorkflowExecution`.

## Domaines dans Amazon SWF

Les flux de travail s'exécutent dans une AWS ressource appelée domaine qui permet de délimiter les ressources Amazon SWF au sein AWS de votre compte. Tous les composants d'un flux de travail, tels que le type de flux de travail et les types de l'activité, doivent être spécifiés dans un domaine.

Un AWS compte peut comporter plusieurs domaines, chacun pouvant contenir plusieurs flux de travail, mais les flux de travail des différents domaines ne peuvent pas interagir.

Lorsque vous configurez un nouveau flux de travail, avant de configurer les autres composants du flux de travail, vous devez enregistrer un domaine, si vous ne l'avez pas déjà fait.

Lorsque vous enregistrez un domaine, vous spécifiez une période de conservation de l'historique des flux de travail. La période de conservation est la durée pendant laquelle Amazon SWF continuera à conserver les informations relatives à l'exécution du flux de travail une fois celui-ci terminé.

L'enregistrement de domaine est la seule fonctionnalité initialement disponible dans la console. Une fois qu'au moins un domaine est enregistré, vous pouvez effectuer les actions suivantes pour le domaine :

- Enregistrez les types de flux de travail et d'activités.
- Lancement des exécutions de flux de travail
- Annulation, arrêt et envoi des signaux aux exécutions de flux de travail en cours
- Redémarrage des exécutions de flux de travail fermées

Vous pouvez également effectuer des actions de gestion de domaine, telles que la dépréciation ou la désapprobation de domaines.

Une fois qu'un domaine est obsolète, vous ne pouvez pas l'utiliser pour créer de nouvelles exécutions de flux de travail ou enregistrer de nouveaux flux de travail. La dépréciation d'un domaine entraîne également la désapprobation de toutes les activités et de tous les flux de travail enregistrés dans le domaine. Les exécutions lancées avant que le domaine ne soit devenu obsolète continuent de s'exécuter.

Après avoir désapprouvé un domaine précédemment obsolète, vous pouvez recommencer à l'utiliser pour enregistrer les types de flux de travail et démarrer de nouvelles exécutions de flux de travail.

Pour plus d'informations sur ces actions de gestion de domaine, reportez-vous [DeprecateDomain](#) aux sections et [UndeprecateDomain](#).

## Acteurs d'Amazon SWF

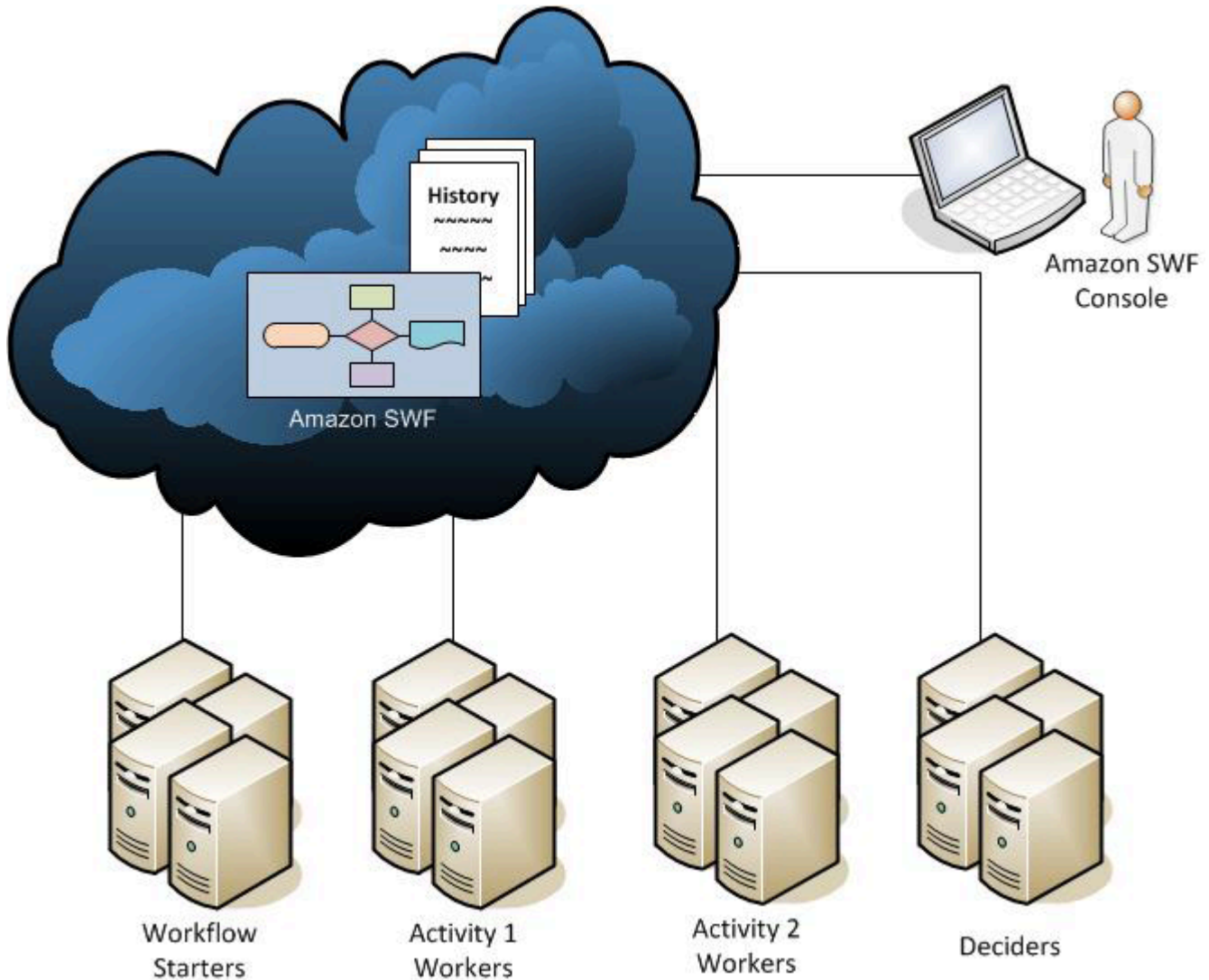
### Rubriques

- [Qu'est-ce qu'un acteur dans Amazon SWF ?](#)
- [Démarrateurs de flux de travail](#)
- [Décideurs](#)
- [Outils de traitement d'activité](#)
- [Echange de données entre les acteurs](#)

## Qu'est-ce qu'un acteur dans Amazon SWF ?

Dans le cadre de ses opérations, Amazon SWF interagit avec différents types d'acteurs programmatiques. Il peut s'agir de [démarrateurs de flux de travail](#), de [décideurs](#) ou d'[outils de traitement d'activité](#). Ces acteurs communiquent avec Amazon SWF par le biais de son API. Vous pouvez développer ces acteurs dans n'importe quel langage de programmation.

Le schéma suivant montre l'architecture Amazon SWF, y compris Amazon SWF et ses acteurs.



## Démarreurs de flux de travail

Un démarreur de flux de travail est une application qui permet de lancer des exécutions de flux de travail. Dans l'exemple de commerce en ligne, il peut s'agir du site Web sur lequel le client passe une commande. Un autre démarreur de flux de travail peut être une application mobile ou un système utilisé par le service client pour passer commande au nom du client.

## Décideurs

Un décideur est l'implémentation de la logique de coordination d'un flux de travail. Les décideurs contrôlent le flux des tâches d'activité d'une exécution de flux de travail. Dès qu'un changement se produit pendant une exécution de flux de travail, tel que l'achèvement d'une tâche, une tâche de

décision qui contient la totalité de l'historique de flux de travail est transmise à un décideur. Lorsque le décideur reçoit la tâche de décision d'Amazon SWF, il analyse l'historique d'exécution du flux de travail afin de déterminer les prochaines étapes appropriées de l'exécution du flux de travail. Le décideur communique ces étapes à Amazon SWF à l'aide de décisions. Une décision est un type de données Amazon SWF qui peut représenter les différentes actions suivantes. Pour obtenir la liste des décisions possibles, consultez la section [Décision](#) du manuel Amazon Simple Workflow Service API Reference.

Voici un exemple de décision au format JSON, le format dans lequel elle est transmise à Amazon SWF. Cette décision planifie une nouvelle tâche d'activité.

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

Un décideur reçoit une tâche de décision lorsque l'exécution du flux de travail commence et qu'elle change d'état. Les décideurs continuent de faire avancer l'exécution du flux de travail en recevant des tâches de décision et en répondant à Amazon SWF avec d'autres décisions jusqu'à ce que le décideur détermine que l'exécution du flux de travail est terminée. Il renvoie alors une décision de fermeture de l'exécution de flux de travail. Une fois l'exécution du flux de travail terminée, Amazon SWF ne planifiera pas de tâches supplémentaires pour cette exécution.

Dans l'exemple de commerce en ligne, le décideur détermine si chaque étape a été effectuée correctement, puis planifie l'étape suivante ou gère les conditions d'erreur.

Un décideur représente un thread ou un processus informatique unique. Plusieurs décideurs peuvent traiter les tâches du même type de flux de travail.

## Outils de traitement d'activité

Un outil de traitement d'activité est un processus ou un thread qui exécute les tâches d'activité faisant partie de votre flux de travail. La tâche d'activité représente une des tâches que vous avez identifiées dans votre application.

Pour utiliser une tâche d'activité dans votre flux de travail, vous devez l'enregistrer à l'aide de la console Amazon SWF ou de l'[RegisterActivityType](#) action.

Chaque travailleur d'activité interroge Amazon SWF pour de nouvelles tâches qu'il doit effectuer ; certaines tâches ne peuvent être effectuées que par certains travailleurs d'activité. Après avoir reçu une tâche, le responsable de l'activité la traite jusqu'à son terme, puis indique à Amazon SWF que la tâche est terminée et fournit le résultat. Il recherche ensuite une nouvelle tâche. Les outils de traitement d'activité associés à une exécution de flux de travail continuent de cette façon : ils traitent les tâches jusqu'à ce que l'exécution de flux de travail soit terminée. Dans l'exemple de commerce en ligne, les outils de traitement d'activité sont des applications et des processus indépendants utilisés par des personnes, telles que des opérateurs de traitement de carte de crédit et des employés d'entrepôt, qui effectuent les différentes étapes du processus.

Un outil de traitement d'activité représente un processus (ou un thread) informatique unique. Plusieurs outils de traitement d'activité peuvent traiter les tâches du même type d'activité.

## Echange de données entre les acteurs

Les données d'entrée peuvent être transmises à une exécution de flux de travail lorsqu'il démarre. De même, elles peuvent être transmises aux outils de traitement d'activité lorsqu'ils planifient des tâches d'activité. Lorsqu'une tâche d'activité est terminée, le responsable de l'activité peut renvoyer les résultats à Amazon SWF. De même, un décideur peut signaler les résultats d'une exécution de flux de travail lorsque celle-ci est terminée. Chaque acteur peut envoyer des données à Amazon SWF et en recevoir des données par le biais de chaînes dont la forme est définie par l'utilisateur. En fonction de la taille et de la sensibilité des données, vous pouvez transmettre les données directement ou transmettre un pointeur vers des données stockées sur un autre système ou service (tel qu'Amazon S3 ou DynamoDB). Les données transmises directement et les pointeurs vers d'autres magasins de données sont enregistrés dans l'historique d'exécution du flux de travail ; cependant, Amazon SWF ne copie ni ne met en cache aucune des données provenant de magasins externes dans le cadre de l'historique.

Dans la mesure où Amazon SWF conserve l'état d'exécution complet de chaque exécution de flux de travail, y compris les entrées et les résultats des tâches, tous les acteurs peuvent être apatrides. Par conséquent, le traitement des flux de travail est hautement évolutif. Lorsque la charge système augmente, il vous suffit d'ajouter des acteurs pour accroître la capacité.

## Tâches dans Amazon SWF

Amazon SWF interagit avec les personnes chargées de l'activité et les décideurs en leur fournissant des tâches appelées tâches. Il existe trois types de tâches dans Amazon SWF :

- Tâche d'activité — Une tâche d'activité indique à un travailleur d'activité d'exécuter sa fonction, par exemple pour vérifier l'inventaire ou débiter une carte de crédit. La tâche d'activité contient toutes les informations dont l'outil de traitement d'activité a besoin pour effectuer l'action requise.
- Tâche Lambda : une tâche Lambda est similaire à une tâche d'activité, mais elle exécute une fonction Lambda au lieu d'une activité Amazon SWF traditionnelle. Pour plus d'informations sur la façon de définir une tâche Lambda, consultez la section [AWS Lambda tâches dans Amazon SWF](#).
- Tâche de décision — Une tâche de décision indique à un décideur que l'état d'exécution du flux de travail a changé afin qu'il puisse déterminer la prochaine activité à effectuer. La tâche de décision contient l'historique du flux de travail en cours.

Amazon SWF planifie une tâche de décision au démarrage du flux de travail et à chaque fois que l'état du flux de travail change, par exemple lorsqu'une tâche d'activité se termine. Chaque tâche de décision contient une vue paginée de l'historique complet d'exécution du flux de travail. Le décideur analyse l'historique d'exécution du flux de travail et répond à Amazon SWF avec un ensemble de décisions qui spécifient la prochaine étape de l'exécution du flux de travail. Essentiellement, chaque tâche décisionnelle donne au décideur l'opportunité d'évaluer le flux de travail et de fournir des directives à Amazon SWF.

Pour garantir qu'aucune décision contradictoire n'est traitée, Amazon SWF affecte chaque tâche de décision à un seul décideur et n'autorise qu'une seule tâche de décision à la fois à être active dans l'exécution d'un flux de travail.

Le tableau suivant illustre la relation entre les différents éléments associés aux flux de travail et aux décideurs.

Conception logique	Enregistré comme	Effectué par	Reçoit et effectue	Génère
Flux de travail	Type de flux de travail	Décideur	Tâches de décision	Décisions

Lorsqu'un travailleur d'activité a terminé la tâche d'activité, il indique à Amazon SWF que la tâche est terminée et inclut tous les résultats pertinents qui ont été générés. Amazon SWF met à jour l'historique d'exécution du flux de travail avec un événement qui indique que la tâche est terminée, puis planifie une tâche de décision afin de transmettre l'historique mis à jour au décideur.

Amazon SWF attribue chaque tâche d'activité à un seul agent d'activité. Une fois que la tâche est attribuée, aucun autre outil de traitement d'activité ne peut se l'approprier ou l'accomplir.

Le tableau suivant illustre la relation entre les différents éléments liés aux activités.

Conception logique	Enregistré comme	Effectué par	Reçoit et effectue	Génère
Activité	Type d'activité	Outil de traitement d'activité	Tâches d'activité	Données de résultat

## Listes de tâches dans Amazon SWF

Les listes de tâches permettent d'organiser les différentes tâches associées à un flux de travail. D'une certaine manière, elles sont similaires aux files d'attente dynamiques. Lorsqu'une tâche est planifiée dans Amazon SWF, vous pouvez spécifier une file d'attente (liste de tâches) dans laquelle la placer. De même, lorsque vous interrogez Amazon SWF pour une tâche, vous indiquez dans quelle file d'attente (liste de tâches) vous souhaitez obtenir la tâche.

Les listes de tâches constituent un mécanisme flexible pour acheminer les tâches aux outils de traitement d'activité en fonction de vos besoins. Comme les listes de tâches sont dynamiques, vous n'avez pas besoin de les enregistrer ni de les créer explicitement via une action : la planification d'une tâche suffit à créer la liste de tâches, si elle n'existe déjà.

Il existe des listes distinctes pour les tâches d'activité et les tâches de décision. Une tâche est toujours planifiée dans une seule liste de tâches. Elle n'est partagée avec aucune autre liste. En outre, tout comme les activités et les flux de travail, les listes de tâches sont limitées à une AWS région et à un domaine Amazon SWF spécifiques.

## Rubriques

- [Listes de tâches de décision](#)
- [Listes de tâches d'activité](#)
- [Routage des tâches](#)

## Listes de tâches de décision

Chaque exécution de flux de travail est associée à une liste de tâches de décision spécifique. Lorsqu'un type de flux de travail est enregistré ([RegisterWorkflowType](#) action), vous pouvez spécifier une liste de tâches par défaut pour les exécutions de ce type de flux de travail. Lorsque le démarreur du flux de travail lance l'exécution de ce dernier (action `StartWorkflowExecution`), il a la possibilité de définir une liste de tâches différentes pour l'exécution de ce flux de travail.

Quand un décideur recherche une nouvelle tâche de décision (action `PollForDecisionTask`), il spécifie une liste de tâches de décision dans laquelle effectuer la recherche. Un seul décideur peut servir plusieurs exécutions de flux de travail en appelant `PollForDecisionTask` plusieurs fois, à l'aide d'une liste de tâches spécifique à chaque exécution de flux de travail dans chaque appel. Le décideur peut également interroger une seule liste de tâches de décision fournissant les tâches de décision de plusieurs exécutions de flux de travail. De même, plusieurs décideurs peuvent interroger la liste de tâches d'une seule exécution de flux de travail.

## Listes de tâches d'activité

Une seule liste de tâches d'activité peut contenir des tâches avec des types d'activités différents. Les tâches sont planifiées dans l'ordre dans la liste des tâches. Amazon SWF renvoie les tâches de la liste dans l'ordre dans la mesure du possible. Dans certains cas, l'ordre de la liste de tâches n'est pas respecté.

Lorsqu'un type d'activité est enregistré ([RegisterActivityType](#) action), vous pouvez spécifier une liste de tâches par défaut pour ce type d'activité. Par défaut, les tâches d'activité de ce type seront planifiées dans la liste de tâches spécifiée ; toutefois, lorsque le décideur planifie une tâche d'activité ([ScheduleActivityTask](#) décision), il peut éventuellement spécifier une liste de tâches différente sur

laquelle planifier la tâche. Si le décideur ne spécifie pas une liste de tâches, c'est la liste de tâches par défaut qui est utilisée. Dès lors, vous pouvez ajouter des tâches d'activité dans des listes de tâche spécifiques en fonction de leurs attributs. Par exemple, vous pouvez placer toutes les instances d'une tâche d'activité pour un type de carte de crédit donné dans une liste de tâches particulière.

## Routage des tâches

Lorsqu'un agent d'activité interroge pour une nouvelle tâche ([PollForActivityTask](#)action), il peut spécifier une liste de tâches d'activité dans laquelle s'appuyer. Dans ce cas, l'outil de traitement d'activité acceptera uniquement les tâches provenant de cette liste. De cette manière, vous pouvez vous assurer que certaines tâches sont uniquement affectées à des outils de traitement d'activité spécifiques. Par exemple, vous pouvez créer une liste pour les tâches qui nécessitent l'utilisation d'un ordinateur hautes performances. Seuls les outils de traitement d'activité qui utilisent le matériel approprié pourront interroger cette liste de tâches. Vous pouvez également créer une liste de tâches pour une région géographique donnée et vous assurer que seuls les outils de traitement d'activité déployés dans cette région sélectionnent ces tâches. De même, vous pouvez créer une liste de tâches pour les commandes prioritaires afin de toujours vérifier cette liste en premier.

L'attribution de tâches particulières à des outils de traitement d'activité spécifiques s'appelle le routage de tâches. Le routage des tâches est facultatif. Si vous ne spécifiez pas une liste des tâches lors de la planification d'une tâche d'activité, celle-ci est automatiquement ajoutée à la liste des tâches par défaut.

## Clôture de l'exécution du flux de travail dans Amazon SWF

Une fois que vous avez démarré l'exécution d'un flux de travail, celui-ci est ouvert. Vous pouvez la fermer comme étant terminée, comme annulée, comme ayant échoué ou comme ayant expiré. Vous pouvez également la traiter comme nouvelle exécution ou comme étant arrêtée. L'exécution d'un flux de travail peut être clôturée par le décideur, par la personne administrant le flux de travail ou par Amazon SWF.

Si le décideur détermine que les activités du flux de travail sont terminées, il doit fermer l'exécution de flux de travail comme étant terminée via l'action [RespondDecisionTaskCompleted](#) et transmettre la décision [CompleteWorkflowExecution](#).

Un décideur peut également fermer l'exécution de flux de travail comme étant annulée ou comme ayant échoué. Pour annuler l'exécution, il doit utiliser l'action [RespondDecisionTaskCompleted](#) et transmettre la décision [CancelWorkflowExecution](#).

Le décideur doit mettre en échec l'exécution de flux de travail si son état sort du domaine d'achèvement normal. Pour mettre en échec l'exécution, il doit utiliser l'action `RespondDecisionTaskCompleted` et transmettre la décision [FailWorkflowExecution](#).

Amazon SWF surveille les exécutions des flux de travail pour s'assurer qu'elles ne dépassent aucun délai d'expiration défini par l'utilisateur. Si le délai d'exécution d'un flux de travail expire, Amazon SWF le ferme automatiquement. Pour plus d'informations sur les valeurs de délai, consultez la section [Types de délai d'expiration Amazon SWF](#).

Un décideur peut aussi fermer l'exécution et la traiter de manière logique comme nouvelle exécution en utilisant l'action `RespondDecisionTaskCompleted` et en transmettant la décision [ContinueAsNewWorkflowExecution](#). Cette stratégie est particulièrement utile pour les exécutions de flux de travail de longue durée pour lesquelles l'historique peut devenir trop volumineux au fil du temps.

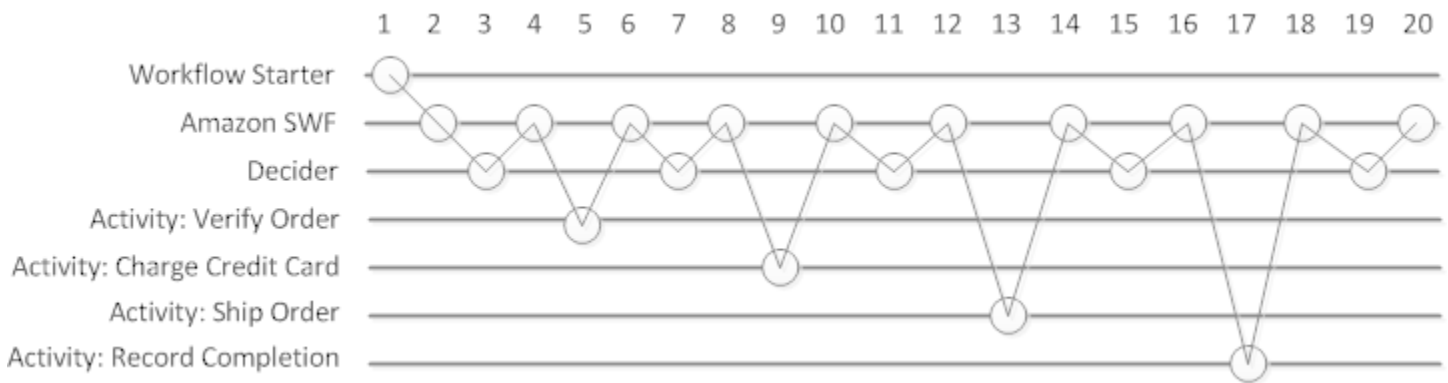
Enfin, vous pouvez mettre fin aux exécutions de flux de travail directement depuis la console Amazon SWF ou par programmation à l'aide de l'API. [TerminateWorkflowExecution](#) Cet arrêt force la fermeture de l'exécution de flux de travail. L'annulation est préférable à l'arrêt, car elle permet aux décideurs de gérer la fermeture de l'exécution de flux de travail.

Amazon SWF met fin à l'exécution d'un flux de travail si l'exécution dépasse certaines limites définies par le service. Amazon SWF met fin à un flux de travail enfant si le flux de travail parent est terminé et si la politique applicable aux enfants indique que le flux de travail enfant doit également être arrêté.

## Cycle de vie d'un flux de travail Amazon SWF

Du début à la fin de l'exécution d'un flux de travail, Amazon SWF interagit avec les acteurs en leur attribuant les tâches appropriées, qu'il s'agisse de tâches d'activité ou de tâches décisionnelles.

Le schéma suivant illustre le cycle de vie d'une exécution de flux de travail de traitement des commandes du point de vue des composants qui y jouent un rôle.



## Cycle de vie d'exécution du flux de travail

Le tableau suivant explique chaque tâche de l'image précédente.

Description	Action, décision ou événement
1. Le démarreur du flux de travail appelle l'action Amazon SWF appropriée pour démarrer l'exécution du flux de travail pour une commande, en fournissant les informations de commande.	<a href="#">StartWorkflowExecution</a> action.
2. Amazon SWF reçoit la demande d'exécution du flux de travail de démarrage, puis planifie la première tâche de décision.	Événement <a href="#">WorkflowExecutionStarted</a> et événement <a href="#">DecisionTaskScheduled</a> .
3. Le décideur reçoit la tâche d'Amazon SWF, passe en	Action <a href="#">PollForDecisionTask</a> . Action <a href="#">RespondDecisionTaskCompleted</a> et décision <a href="#">ScheduleActivityTask</a> .

Description	Action, décision ou événement
<p>revue l'historique, applique la logique de coordination pour déterminer qu'aucune activité précédente n'a eu lieu, prend la décision de planifier l'activité de vérification de la commande avec les informations dont le responsable de l'activité a besoin pour traiter la tâche, et renvoie la décision à Amazon SWF.</p>	
<p>4. Amazon SWF reçoit la décision, planifie la tâche d'activité Verify Order et attend que la tâche soit terminée ou expire.</p>	<p><a href="#">ActivityTaskScheduled</a> event</p>
<p>5. Un travailleur d'activité capable d'effectuer l'activité Verify Order reçoit la tâche, l'exécute et renvoie les résultats à Amazon SWF.</p>	<p>Actions <a href="#">PollForActivityTask</a> et <a href="#">RespondActivityTaskCompleted</a></p>

Description	Action, décision ou événement
6. Amazon SWF reçoit les résultats de l'activité de vérification des commandes, les ajoute à l'historique du flux de travail et planifie une tâche de décision.	Événement <a href="#">ActivityTaskCompleted</a> et événement <a href="#">DecisionTaskScheduled</a> .
7. Le décideur reçoit la tâche d'Amazon SWF, examine l'historique, applique la logique de coordination, prend la décision de planifier ChargeCreditCard une tâche d'activité avec les informations dont le responsable de l'activité a besoin pour traiter la tâche et renvoie la décision à Amazon SWF.	Action <a href="#">PollForDecisionTask</a> . Action <a href="#">RespondDecisionTaskCompleted</a> avec la décision <a href="#">ScheduleActivityTask</a> .
8. Amazon SWF reçoit la décision, planifie la tâche ChargeCreditCard d'activité et attend qu'elle soit terminée ou expire.	Événement <a href="#">DecisionTaskCompleted</a> et événement <a href="#">ActivityTaskScheduled</a> .

Description	Action, décision ou événement
9. Un travailleur d'activité capable d'effectuer l'activité ChargeCreditCard reçoit la tâche, l'exécute et renvoie les résultats à Amazon SWF.	Actions <a href="#">PollForActivityTask</a> et <a href="#">RespondActivityTaskCompleted</a>
10. Amazon SWF reçoit les résultats de la tâche d'activité ChargeCreditCard, les ajoute à l'historique du flux de travail et planifie une tâche de décision.	Événement <a href="#">ActivityTaskCompleted</a> et événement <a href="#">DecisionTaskScheduled</a> .
11. Le décideur reçoit la tâche d'Amazon SWF, examine l'historique, applique la logique de coordination, prend la décision de planifier une tâche d'activité avec les informations dont le responsable de l'activité a besoin pour exécuter la tâche et renvoie la décision à Amazon SWF.	Action <a href="#">PollForDecisionTask</a> . <a href="#">RespondDecisionTaskCompleted</a> avec la décision <a href="#">ScheduleActivityTask</a> .

Description	Action, décision ou événement
12. Amazon SWF reçoit la décision, planifie une tâche d' ShipOrder activité et attend qu'elle soit terminée ou expire.	Événement <a href="#">DecisionTaskCompleted</a> et événement <a href="#">ActivityTaskScheduled</a> .
13. Un travailleur d'activité capable d'effectuer l' ShipOrder activité reçoit la tâche, l'exécute et renvoie les résultats à Amazon SWF.	Actions <a href="#">PollForActivityTask</a> et <a href="#">RespondActivityTaskCompleted</a>
14. Amazon SWF reçoit les résultats de la tâche d' ShipOrder activité, les ajoute à l'historique du flux de travail et planifie une tâche de décision.	Événement <a href="#">ActivityTaskCompleted</a> et événement <a href="#">DecisionTaskScheduled</a> .

Description	Action, décision ou événement
<p>15. Le décideur reçoit la tâche d'Amazon SWF, examine l'historique, applique la logique de coordination, prend la décision de planifier RecordCompletion une tâche d'activité avec les informations dont le responsable de l'activité a besoin pour exécuter la tâche et renvoie la décision à Amazon SWF.</p>	<p>Action <a href="#">PollForDecisionTask</a> . Action <a href="#">RespondDecisionTaskCompleted</a> avec la décision <a href="#">ScheduleActivityTask</a> .</p>
<p>16. Amazon SWF reçoit la décision, planifie une tâche d'RecordCompletion activité et attend qu'elle soit terminée ou expire.</p>	<p>Événement <a href="#">DecisionTaskCompleted</a> et événement <a href="#">ActivityTaskScheduled</a> .</p>
<p>17. Un travailleur d'activité capable d'effectuer l'RecordCompletion activité reçoit la tâche, l'exécute et renvoie les résultats à Amazon SWF.</p>	<p>Actions <a href="#">PollForActivityTask</a> et <a href="#">RespondActivityTaskCompleted</a> .</p>

Description	Action, décision ou événement
18. Amazon SWF reçoit les résultats de la tâche d'RecordCompletion activité, les ajoute à l'historique du flux de travail et planifie une tâche de décision.	Événement <a href="#">ActivityTaskCompleted</a> et événement <a href="#">DecisionTaskScheduled</a> .
19. Le décideur reçoit la tâche d'Amazon SWF, examine l'historique, applique la logique de coordination, prend la décision de clore l'exécution du flux de travail et renvoie la décision ainsi que les résultats éventuels à Amazon SWF.	Action <a href="#">PollForDecisionTask</a> . Action <a href="#">RespondDecisionTaskCompleted</a> avec la décision <a href="#">CompleteWorkflowExecution</a> .
20. Amazon SWF ferme l'exécution du flux de travail et archive l'historique pour référence future.	<a href="#">WorkflowExecutionCompleted</a> event.

## Sondage de tâches dans Amazon SWF

Les décideurs et les animateurs communiquent avec Amazon SWF par le biais de longs sondages. Le décideur ou le responsable de l'activité initie régulièrement une communication avec Amazon SWF, informe Amazon SWF de sa disponibilité pour accepter une tâche, puis spécifie une liste de tâches à partir de laquelle obtenir les tâches.

Si une tâche est disponible dans la liste de tâches spécifiée, Amazon SWF la renvoie immédiatement dans la réponse. Si aucune tâche n'est disponible, Amazon SWF maintient la connexion TCP ouverte pendant 60 secondes au maximum afin que, si une tâche devient disponible pendant cette période, elle puisse être renvoyée via la même connexion. Si aucune tâche n'est disponible dans le délai de 60 secondes, renvoie une réponse vide et ferme la connexion. (Une réponse vide est une structure de tâche dans laquelle la valeur `taskToken` est une chaîne vide.) Dans ce cas, le décideur ou l'outil de traitement d'activité doit relancer l'attente active.

L'attente active de longue durée convient particulièrement au traitement des tâches volumineuses. Elle permet aux décideurs et aux outils de traitement d'activité de gérer leur propre capacité et elle est facile à utiliser lorsque les décideurs et les outils de traitement d'activité se trouvent derrière un pare-feu.

Pour plus d'informations, consultez [Recherche des tâches de décision](#) et [Recherche de tâches d'activité](#).

# Concepts de flux de travail avancés dans Amazon SWF

L'exemple de commerce en ligne de la section [???](#) représente un scénario de flux de travail simplifié. En réalité, vous souhaiterez probablement que votre flux de travail effectue des tâches simultanées (envoi d'un e-mail de confirmation et autorisation d'une carte de crédit), enregistre les événements majeurs (tous les articles ont été rassemblés), mette à jour la commande en cas de modifications (ajout ou suppression d'un article) et prenne d'autres décisions plus avancées dans le cadre de l'exécution de votre flux de travail. Cette section décrit les concepts de flux de travail avancés que vous pouvez utiliser pour créer vos flux de travail.

## Concepts avancés

- [Gestion des versions](#)
- [Signaux](#)
- [Flux de travail pour enfants dans Amazon SWF](#)
- [Marqueurs dans Amazon SWF](#)
- [Balises dans Amazon SWF](#)
- [Mise en œuvre d'un choix exclusif avec Amazon SWF](#)
- [Minuteries dans Amazon SWF](#)
- [Annulation de tâches d'activité dans Amazon SWF](#)

## Gestion des versions

Les besoins de l'entreprise vous obligent parfois à utiliser des implémentations différentes ou des variations du même flux de travail, ou encore l'exécution simultanée de plusieurs activités. Par exemple, il se peut que vous souhaitiez tester une nouvelle implémentation d'un flux de travail pendant qu'un autre est en production. Vous avez peut-être également besoin d'exécuter deux implémentations distinctes avec deux ensembles de fonctions différents, comme une implémentation de base et une implémentation avancée. La gestion des versions vous permet d'exécuter plusieurs implémentations de flux de travail et des activités simultanément, quels que soient vos besoins.

Une version est associée aux types de flux de travail et d'activité au moment de leur enregistrement. La version est une chaîne de forme libre dont vous pouvez choisir le schéma. Pour créer une nouvelle version d'un type enregistré, vous devez l'enregistrer avec le même nom et une autre version. [Listes de tâches dans Amazon SWF](#), décrit précédemment, peut également vous aider à mettre en œuvre la gestion des versions. Imaginez une situation où les exécutions de flux de travail

en cours sont de longue durée pour un type donné et où les circonstances nécessitent une révision du flux de travail, tel que l'ajout d'une nouvelle fonctionnalité. Pour mettre en œuvre cette nouvelle fonctionnalité, vous pourriez créer d'autres versions des types et des outils de traitement d'activité, ainsi qu'un nouveau décideur. Ensuite, vous pourriez lancer des exécutions de la nouvelle version de flux de travail avec un autre ensemble de listes de tâches. De cette façon, différentes versions d'exécution de flux de travail pourraient être exécutées simultanément sans interférer les unes avec les autres.

## Signaux

Les signaux vous permettent d'injecter des informations dans une exécution de flux de travail en cours. Dans certains scénarios, vous souhaitez parfois ajouter des informations à une exécution de flux de travail en cours d'exécution pour l'informer que quelque chose a changé ou pour l'informer d'un événement externe. N'importe quel processus peut envoyer un signal à une exécution de flux de travail ouverte. Par exemple, une exécution de flux de travail peut en signaler une autre.

### Note

Toute tentative d'envoi d'un signal à une exécution de flux de travail qui n'est pas ouverte entraîne l'échec de `SignalWorkflowExecution` avec `UnknownResourceFault`.

Pour utiliser des signaux, définissez le nom du signal et les données à transmettre au signal, le cas échéant. Programmez ensuite le décideur pour qu'il reconnaisse l'événement de signal ([WorkflowExecutionSignaled](#)) dans l'historique et qu'il le traite de manière appropriée. Lorsqu'un processus souhaite signaler l'exécution d'un flux de travail, il appelle Amazon SWF (en utilisant l'[SignalWorkflowExecution](#) action ou, dans le cas d'un décideur, en utilisant la [SignalExternalWorkflowExecution](#) décision) qui spécifie l'identifiant de l'exécution du flux de travail cible, le nom du signal et les données du signal. Amazon SWF reçoit ensuite le signal, l'enregistre dans l'historique de l'exécution du flux de travail cible et planifie une tâche de décision pour celui-ci. Lorsque le décideur reçoit la tâche de la décision, il reçoit également le signal dans l'historique d'exécution du flux de travail. Le décideur peut ensuite effectuer les actions appropriées en fonction du signal et de ses données.

Dans certains cas, vous préférez attendre de recevoir un signal avant de continuer. Par exemple, un utilisateur peut annuler une commande en envoyant un signal, mais uniquement dans l'heure suivant la passation de la commande. Amazon SWF ne possède pas de primitive permettant à un décideur d'attendre un signal du service. Les fonctionnalités d'interruption doivent être mises en œuvre dans

le décideur lui-même. Pour effectuer une interruption, le décideur doit lancer un temporisateur, à l'aide de la décision `StartTimer`, qui spécifie la durée pendant laquelle le décideur attendra le signal tout en continuant à rechercher des tâches de décision. Lorsque le décideur reçoit une tâche de la décision, il doit vérifier l'historique pour voir si le signal a été reçu ou si le temporisateur s'est déclenché. Si le signal a été reçu, il doit annuler le temporisateur. Toutefois, si, au lieu de cela, le temporisateur s'est déclenché, cela signifie que le signal n'est pas arrivé dans le délai spécifié. Pour résumer, si vous souhaitez attendre un signal spécifique, procédez comme suit.

1. Créez un temporisateur correspondant à la quantité de temps que le décideur doit attendre.
2. Lorsqu'une tâche de décision est reçue, vérifiez l'historique pour déterminer si le signal est arrivé ou si le temporisateur s'est déclenché.
3. Si un signal est arrivé, annulez le temporisateur via la décision `CancelTimer` et traitez ce signal. Selon le timing, l'historique peut contenir à la fois les événements `TimerFired` et `WorkflowExecutionSignaled`. Dans ce cas, vous pouvez compter sur l'ordre relatif des événements de l'historique afin de déterminer celui qui s'est produit en premier.
4. Si le temporisateur s'est déclenché avant la réception d'un signal, l'attente du signal a expiré au niveau du décideur. Vous pouvez mettre en échec l'exécution ou concevoir tout autre logique en fonction de vos besoins.

Dans les cas où un flux de travail doit être annulé (par exemple, si la commande elle-même a été annulée par le client), l'action `RequestCancelWorkflowExecution` doit être utilisée plutôt que d'envoyer un signal au flux de travail.

Voici certains scénarios où des signaux peuvent être utilisés :

- Suspension de la progression des exécutions de flux de travail tant qu'un signal n'a pas été reçu (attente d'une expédition d'inventaire, par exemple).
- Envoi d'informations susceptibles d'avoir une incidence sur la logique de décision des décideurs à une exécution de flux de travail. Ce signal est utile pour les flux de travail soumis à des événements externes (tentative de finalisation de la vente d'une action après la fermeture de la Bourse, par exemple).
- Mise à jour d'une exécution de flux de travail lorsque vous anticipez des modifications éventuelles (modification de la quantité d'une commande après l'enregistrement de la commande et avant son envoi).

Dans l'exemple suivant, l'exécution du flux de travail reçoit un signal pour annuler une commande.

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

Si l'exécution du flux de travail reçoit le signal, Amazon SWF renvoie une réponse HTTP réussie similaire à la suivante. Amazon SWF générera une tâche de décision pour demander au décideur de traiter le signal.

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

## Flux de travail pour enfants dans Amazon SWF

Les flux de travail complexes peuvent être divisés en composants plus petits, plus faciles à gérer et éventuellement réutilisables avec les flux de travail enfants. Un flux de travail enfant est une exécution de flux de travail initiée par une autre exécution de flux de travail (parent). Pour lancer un flux de travail enfant, le décideur du flux de travail parent utilise la décision `StartChildWorkflowExecution`. Les données d'entrée spécifiées avec cette décision sont mises à disposition du flux de travail enfant par le biais de son historique.

Les attributs de la `StartChildWorkflowExecution` décision spécifient également la politique de l'enfant, c'est-à-dire la manière dont Amazon SWF doit gérer la situation dans laquelle l'exécution du flux de travail parent prend fin avant celle du flux de travail enfant. Trois valeurs sont possibles :

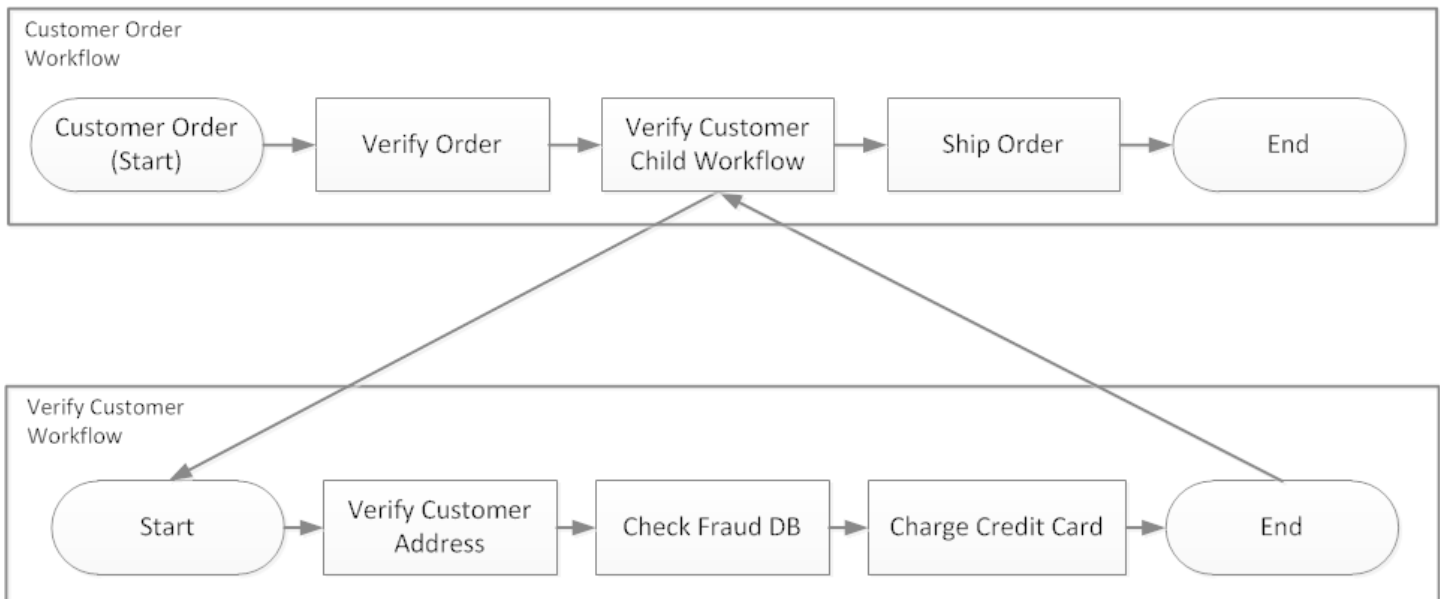
- **TERMINER** : Amazon SWF mettra fin aux exécutions d'enfants.
- **REQUEST\_CANCEL** : Amazon SWF tentera d'annuler l'exécution de l'enfant en plaçant un `WorkflowExecutionCancelRequested` événement dans l'historique d'exécution du flux de travail de l'enfant.
- **ABANDON** : Amazon SWF ne prendra aucune mesure ; les exécutions d'enfants se poursuivront.

Une fois lancée, l'exécution du flux de travail enfant fonctionne comme une exécution habituelle. Une fois l'opération terminée, Amazon SWF enregistre l'achèvement, ainsi que ses résultats, dans l'historique du flux de travail du flux de travail parent. Voici des exemples de flux de travail enfants :

- Flux de travail enfant de traitement de carte de crédit utilisé par les flux de travail de différents sites Web
- Flux de travail enfant qui vérifie l'adresse e-mail du client, qui contrôle que ce dernier ne s'est pas désinscrit de la liste de diffusion, qui envoie l'e-mail et qui s'assure qu'il a été remis ou qu'il n'a pas échoué.
- Flux de travail enfant de récupération et de stockage de base de données, qui combine la connexion, la configuration, la transaction et la vérification.
- Flux de travail enfant de compilation du code source, qui combine le développement, la création de paquets et la vérification.

Dans notre exemple de commerce en ligne, il peut être utile d'inclure l'activité de débit de la carte de crédit dans un flux de travail enfant. Pour ce faire, vous pouvez enregistrer un nouveau flux de travail de vérification du client, enregistrer les activités de vérification de l'adresse du client et de contrôle dans la base de données des fraudes, puis définir la logique de coordination des tâches. Puis, un décideur du flux de travail de commande client peut initier un flux de travail enfant de vérification du client en planifiant la décision `StartChildWorkflowExecution` qui spécifie ce type de flux de travail.

La figure suivante illustre un flux de travail de commande client, qui inclut un nouveau flux de travail enfant de vérification, lequel contrôle l'adresse du client, ainsi que la base de données des fraudes, puis débite la carte de crédit.



Plusieurs flux de travail peuvent créer des exécutions de flux de travail enfants à l'aide du même type de flux de travail. Par exemple, le flux de travail enfant de vérification du client peut également être utilisé dans d'autres sections d'une entreprise. Les événements d'un flux de travail enfant figurent dans son propre historique, et non pas dans celui du flux de travail parent.

Comme les flux de travail enfants ne sont que des exécutions de flux de travail initiées par un décideur, ils peuvent également être lancés comme des exécutions de flux de travail autonomes standard.

## Marqueurs dans Amazon SWF

Il arrive que vous souhaitiez enregistrer des informations spécifiques à votre cas d'utilisation dans l'historique d'exécution d'un flux de travail. Les marqueurs vous permettent d'enregistrer dans l'historique d'exécution du flux de travail des informations que vous pouvez utiliser pour répondre à vos besoins.

Pour utiliser des marqueurs, un décideur utilise la `RecordMarker` décision, nomme le marqueur, associe les données souhaitées à la décision et notifie l'action à Amazon SWF. `RespondDecisionTaskCompleted` Amazon SWF reçoit la demande, enregistre le marqueur dans l'historique du flux de travail et met en œuvre toute autre décision contenue dans la demande. A partir de ce moment, les décideurs peuvent voir le marqueur dans l'historique du flux de travail et l'utiliser comme vous l'avez programmé.

L'enregistrement d'un marqueur n'a pas pour effet de lancer une tâche de la décision. Pour éviter le blocage de l'exécution du flux de travail, un événement qui assure sa continuité doit avoir lieu. Par exemple, le décideur peut planifier une autre tâche d'activité, l'exécution du flux de travail peut recevoir un signal ou une tâche d'activité déjà planifiée prend fin.

Voici quelques exemples de marqueurs :

- Un compteur qui détermine le nombre de boucles dans un flux de travail récursif
- Progression de l'exécution du flux de travail en fonction des résultats des activités
- Informations résumées à partir des événements précédents de l'historique du flux de travail

Dans l'exemple de commerce en ligne, vous pouvez ajouter une activité qui vérifie l'inventaire tous les jours et qui incrémente chaque fois le nombre correspondant dans un marqueur. Ensuite, vous pouvez ajouter une logique de décision qui envoie un e-mail au client ou qui informe un responsable lorsque ce nombre dépasse cinq, sans avoir à passer en revue l'historique complet.

Dans l'exemple suivant, le décideur effectue une tâche de la décision et répond avec une action `RespondDecisionTaskCompleted` qui contient une décision `RecordMarker`.

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

Si Amazon SWF enregistre correctement le marqueur, il renvoie une réponse HTTP réussie similaire à la suivante.

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

# Balises dans Amazon SWF

Amazon SWF prend en charge le balisage de l'exécution d'un flux de travail. Cette approche est utile lorsque vous avez un grand nombre de ressources.

Amazon SWF prend en charge le balisage d'une exécution de flux de travail avec un maximum de cinq balises. Chaque balise est une chaîne de forme libre qui peut contenir jusqu'à 256 caractères. Si vous souhaitez utiliser des balises, vous devez les attribuer lorsque vous lancez une exécution de flux de travail. Vous ne pouvez pas ajouter de balises à une exécution de flux de travail une fois que celle-ci a commencé. Vous ne pouvez pas non plus supprimer ou modifier les balises déjà attribuées.

IAM prend en charge le contrôle de l'accès aux domaines Amazon SWF sur la base de balises. Pour contrôler l'accès basé sur des balises, vous devez fournir des informations sur vos balises dans l'élément de condition d'une stratégie IAM.

## Gérer les balises

Gérez les balises Amazon Simple Workflow Service à l'aide AWS SDKs ou en interagissant directement avec l'API Amazon SWF. L'API vous permet d'ajouter des balises lors de l'enregistrement d'un domaine, de répertorier des balises pour un domaine existant ainsi que d'ajouter ou de supprimer des balises pour un domaine existant.

### Note

La limite est de 50 balises par ressource. Consultez [Quotas de compte généraux pour Amazon SWF](#).

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

Pour plus d'informations [Utilisation d'Amazon SWF APIs](#), consultez et le manuel [Amazon Simple Workflow Service API Reference](#).

## Étiqueter les exécutions de workflows

Avec Amazon SWF, vous pouvez associer des balises à des exécutions de flux de travail, puis demander des exécutions de flux de travail en fonction de ces balises. Vous pouvez filtrer la liste lorsque vous utilisez les opérations de visibilité. En sélectionnant soigneusement les balises que vous attribuez à une exécution, vous pouvez les utiliser pour créer des listes pertinentes.

Par exemple, supposons que vous exécutiez plusieurs centres de traitement. À l'aide de balises, vous pouvez répertorier les processus en cours dans un centre de distribution spécifique. Ou, si un client convertit différents types de fichiers multimédia, les balises peuvent indiquer différents processus lors de la conversion de fichiers vidéo, audio et image.

Vous pouvez associer jusqu'à cinq balises à une exécution de flux de travail à l'aide de l'action `StartWorkflowExecution`, de la décision `StartChildWorkflowExecution` ou de la décision `ContinueAsNewWorkflowExecution`. Lorsque vous utilisez des actions de visibilité pour répertorier ou compter les exécutions de flux de travail, vous pouvez filtrer les résultats en fonction de vos balises.

Pour utiliser le balisage

1. Concevez une stratégie de balisage. Pensez aux besoins de votre entreprise et créez une liste des balises qui ont un intérêt pour vous. Déterminer les exécutions qui recevront des balises. Même si un maximum de cinq balises peuvent être assignés à une exécution, votre bibliothèque peut contenir autant de balises que vous le souhaitez. Comme une balise peut être une valeur de chaîne comprenant jusqu'à 256 caractères, elle permet de décrire presque n'importe quel concept d'entreprise.
2. Identifiez une exécution avec jusqu'à cinq balises lors de sa création.
3. Pour répertorier ou comptabiliser les exécutions qui contiennent une balise spécifique, définissez le paramètre `tagFilter` avec les actions `ListOpenWorkflowExecutions`, `ListClosedWorkflowExecutions`, `CountOpenWorkflowExecutions` et `CountClosedWorkflowExecutions`. L'action filtrera les exécutions en fonction des balises spécifiées.

L'association d'une balise à une exécution de flux de travail est irréversible.

Vous ne pouvez spécifier qu'une seule balise dans le paramètre `tagFilter` avec `ListWorkflowExecutions`. En outre, la mise en correspondance des balises est sensible à la casse. Seules les correspondances exactes renvoient des résultats.

Supposons que vous ayez déjà configuré deux exécutions avec les balises suivantes.

Nom de l'exécution	Balises attribuées
Exécution 1	Consommateur, février 2011
Exécution 2	Vente en gros, mars 2011

Vous pouvez filtrer la liste des exécutions renvoyées par `ListOpenWorkflowExecutions` au niveau de la balise `Consumer`. Les valeurs `oldestDate` et `latestDate` sont définies en tant que valeurs [horaires Unix](#).

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

## Contrôlez l'accès aux domaines à l'aide de tags

Vous pouvez contrôler l'accès aux domaines Amazon Simple Workflow Service en référençant les balises associées aux domaines Amazon SWF dans IAM.

Par exemple, vous pouvez restreindre les domaines Amazon SWF qui incluent une balise avec la clé `environment` et la valeur `production` avec la condition suivante :

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

Pour plus d'informations, consultez :

- [Contrôle de l'accès à l'aide de balises IAM](#)

- [Stratégies basées sur des balises](#)

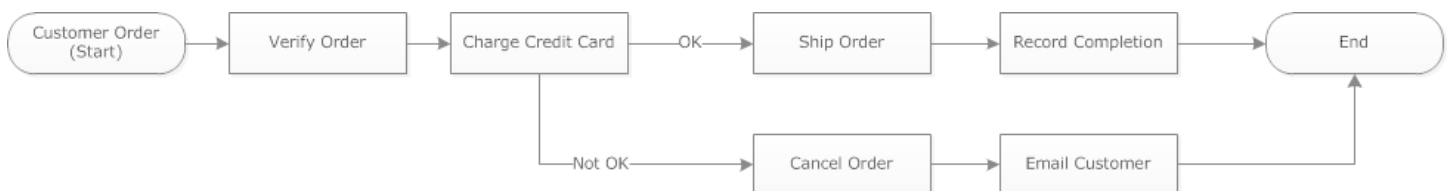
## Mise en œuvre d'un choix exclusif avec Amazon SWF

Dans certains scénarios, vous souhaitez peut-être planifier un autre ensemble d'activités en fonction des résultats d'une activité précédente. Grâce au modèle de choix exclusif, vous pouvez créer des flux de travail flexibles qui répondent aux exigences complexes de votre application.

Amazon SWF ne propose aucune action de choix exclusif spécifique. Pour mettre en œuvre le choix exclusif, vous devez écrire votre logique de décision afin de prendre des décisions en fonction des résultats d'une activité précédente. Voici certaines utilisations possibles avec les choix exclusifs :

- Exécution d'activités de nettoyage, si les résultats d'une activité précédente ont échoué
- Planification de différentes activités selon que le client a acheté un plan de base ou avancé
- Exécution de différentes activités d'authentification en fonction de l'historique de commande du client

Dans l'exemple de commerce en ligne, vous pouvez utiliser le choix exclusif pour expédier ou annuler une commande sur la base des résultats du débit de la carte de crédit. Dans la figure suivante, le décideur planifie les tâches d'activité d'envoi de commande et de consignation de l'achèvement de la tâche si le débit a fonctionné. Sinon, il planifie les tâches d'activité d'annulation de commande et d'envoi d'e-mail au client.



Le décideur planifie l'activité `ShipOrder` si la carte de crédit est débitée avec succès. Sinon, le décideur planifie l'activité `CancelOrder`.

Dans ce cas, programmez le décideur pour qu'il interprète l'historique et pour déterminer si la carte de crédit a été débitée avec succès. Pour ce faire, vous pouvez utiliser une logique similaire à celle-ci :

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

```

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

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

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

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

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

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

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

ENDIF

```

Si la carte de crédit a été débitée avec succès, le décideur doit répondre avec `RespondDecisionTaskCompleted` pour planifier l'activité `ShipOrder`.

```

https://swf.us-east-1.amazonaws.com
RespondDecisionTaskCompleted
{
  "taskToken": "12342e17-80f6-FAKE-TASK-TOKEN32f0223",
  "decisions":[
    {
      "decisionType":"ScheduleActivityTask",
      "scheduleActivityTaskDecisionAttributes":{

```

```

    "control": "OPTIONAL_DATA_FOR_DECIDER",
    "activityType": {
      "name": "ShipOrder",
      "version": "2.4"
    },
    "activityId": "3e2e6e55-e7c4-fee-deed-aa815722b7be",
    "scheduleToCloseTimeout": "3600",
    "taskList": {
      "name": "SHIPPING"
    },
    "scheduleToStartTimeout": "600",
    "startToCloseTimeout": "3600",
    "heartbeatTimeout": "300",
    "input": "123 Main Street, Anytown, United States"
  }
}
]
}

```

Si la carte de crédit n'a pas été débitée avec succès, le décideur doit répondre avec `RespondDecisionTaskCompleted` pour planifier l'activité `CancelOrder`.

```

https://swf.us-east-1.amazonaws.com
RespondDecisionTaskCompleted
{
  "taskToken": "12342e17-80f6-FAKE-TASK-TOKEN32f0223",
  "decisions": [
    {
      "decisionType": "ScheduleActivityTask",
      "scheduleActivityTaskDecisionAttributes": {
        "control": "OPTIONAL_DATA_FOR_DECIDER",
        "activityType": {
          "name": "CancelOrder",
          "version": "2.4"
        },
        "activityId": "3e2e6e55-e7c4-fee-deed-aa815722b7be",
        "scheduleToCloseTimeout": "3600",
        "taskList": {
          "name": "CANCELLATIONS"
        },
        "scheduleToStartTimeout": "600",
        "startToCloseTimeout": "3600",
        "heartbeatTimeout": "300",

```

```
        "input": "Out of Stock"
      }
    }
  ]
}
```

Si Amazon SWF parvient à valider les données lors de l'`RespondDecisionTaskCompleted` action, Amazon SWF renvoie une réponse HTTP réussie similaire à la suivante.

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

## Minuteries dans Amazon SWF

Avec un minuteur, vous pouvez informer votre décideur lorsqu'un certain laps de temps s'est écoulé.

Lorsqu'il répond à une tâche de décision, le décideur a la possibilité de répondre avec une décision `StartTimer`. Cette décision spécifie une durée après laquelle le temporisateur doit se déclencher. Une fois le délai spécifié écoulé, Amazon SWF ajoutera `TimerFired` un événement à l'historique d'exécution du flux de travail et planifiera une tâche de décision. Le décideur peut ensuite utiliser ces informations pour documenter les autres décisions. L'un des champs d'application courants d'un temporisateur consiste à retarder l'exécution d'une tâche d'activité. Par exemple, un client peut souhaiter différer la livraison d'un article.

## Annulation de tâches d'activité dans Amazon SWF

L'annulation des tâches d'activité indique au décideur de mettre fin aux activités qui n'ont plus besoin d'être effectuées. Amazon SWF utilise un mécanisme d'annulation coopératif et n'interrompt pas de force les tâches d'activité en cours d'exécution. Vous devez programmer les outils de traitement d'activité pour traiter les demandes d'annulation.

Le décideur peut décider d'annuler une activité tâche alors que c'est une tâche de la décision de traitement. Pour annuler une tâche d'activité, le décideur utilise l'action `RespondDecisionTaskCompleted` avec la décision `RequestCancelActivityTask`.

Si la tâche d'activité n'a pas encore été acquise par un outil de traitement d'activité, le service annule la tâche. Notez qu'il existe une condition de concurrence potentielle dans le sens où un outil de

traitement d'activité peut acquérir la tâche à tout moment. Si la tâche a déjà été attribuée à un outil de traitement d'activité, il sera invité à annuler la tâche.

Dans cet exemple, l'exécution de flux de travail reçoit un signal pour annuler la commande.

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

Si l'exécution du flux de travail reçoit le signal, Amazon SWF renvoie une réponse HTTP réussie similaire à la suivante. Amazon SWF générera une tâche de décision pour demander au décideur de traiter le signal.

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

Lorsque le décideur traite la tâche de décision et voit le signal dans l'historique, il tente d'annuler l'activité en cours qui présente l'ID d'activité ShipOrderActivity0001. L'ID d'activité est fourni dans l'historique du flux de travail à partir de l'événement de la tâche d'activité planifiée.

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

Si Amazon SWF reçoit correctement la demande d'annulation, il renvoie une réponse HTTP réussie similaire à la suivante :

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

La tentative d'annulation est enregistrée dans l'historique en tant qu'événement `ActivityTaskCancelRequested`.

Si la tâche est annulée avec succès, comme indiqué par un `ActivityTaskCanceled` événement, programmez votre décideur pour qu'il prenne les mesures appropriées qui devraient suivre l'annulation de la tâche, telles que la fermeture de l'exécution du flux de travail.

Si la tâche d'activité n'a pas pu être annulée (par exemple, si la tâche se termine, échoue ou expire au lieu de l'annuler), votre décideur doit accepter les résultats de l'activité ou effectuer le nettoyage ou les mesures d'atténuation nécessaires à votre cas d'utilisation.

Si la tâche d'activité a déjà été acquise par un outil de traitement d'activité, la demande d'annulation est transmise via le mécanisme `task-heartbeat`. Les professionnels de l'activité peuvent régulièrement `RecordActivityTaskHeartbeat` l'utiliser pour signaler à Amazon SWF que la tâche est toujours en cours.

Notez que les outils de traitement d'activité ne sont obligés de générer une pulsation, même si cette approche est recommandée pour les tâches de longue durée. L'annulation d'une tâche nécessite l'enregistrement d'une pulsation régulière. Si l'outil de traitement d'activité ne génère pas de pulsation, la tâche ne peut pas être annulée.

Si le décideur demande l'annulation de la tâche, Amazon SWF définit la valeur de `cancelRequest` l'objet sur `true`. L'objet `cancelRequest` fait partie de l'objet `ActivityTaskStatus` qui est renvoyé par le service en réponse à `RecordActivityTaskHeartbeat`.

Amazon SWF n'empêche pas la réussite d'une tâche d'activité dont l'annulation a été demandée ; c'est à l'activité de déterminer comment traiter la demande d'annulation. Selon vos besoins, programmez l'outil de traitement d'activité pour annuler la tâche d'activité ou ignorer la demande d'annulation.

Si vous souhaitez que l'outil de traitement d'activité indique que le travail de la tâche d'activité a été annulé, programmez-le pour répondre avec `RespondActivityTaskCanceled`. Si vous souhaitez que l'outil de traitement d'activité termine la tâche, programmez le pour répondre avec un `RespondActivityTaskCompleted` standard.

Lorsqu'Amazon SWF reçoit la `RespondActivityTaskCanceled` demande `RespondActivityTaskCompleted` ou, il met à jour l'historique d'exécution du flux de travail et planifie une tâche de décision pour informer le décideur.

Programmez le décideur pour traiter la tâche de décision et pour renvoyer toutes décisions supplémentaires. Si la tâche d'activité est annulée avec succès, programmez le décideur pour effectuer les tâches requises pour continuer ou pour fermer l'exécution du flux de travail. Si la tâche d'activité n'est pas annulée avec succès, programmez le décideur pour accepter les résultats, pour les ignorer ou pour planifier tout nettoyage requis.

# Sécurité dans Amazon Simple Workflow Service

Cette section fournit des informations sur la sécurité et l'authentification d'Amazon Simple Workflow Service.

## Rubriques

- [Protection des données dans Amazon Simple Workflow Service](#)
- [Identity and Access Management dans Amazon Simple Workflow Service](#)
- [Journalisation et surveillance](#)
- [Validation de conformité pour Amazon Simple Workflow Service](#)
- [Résilience dans Amazon Simple Workflow Service](#)
- [Sécurité de l'infrastructure dans Amazon Simple Workflow Service](#)
- [Analyse de configuration et de vulnérabilité dans Amazon Simple Workflow Service](#)

Amazon SWF utilise IAM pour contrôler l'accès à d'autres AWS services et ressources. Pour obtenir un vue d'ensemble du fonctionnement d'IAM, consultez [Présentation de la gestion des accès](#) dans le Guide de l'utilisateur IAM. Pour obtenir une présentation des informations d'identification de sécurité, consultez [Informations d'identification de sécuritéAWS](#) dans le Référence générale d'Amazon Web Services.

## Protection des données dans Amazon Simple Workflow Service

Le [modèle de responsabilité AWS partagée](#) s'applique à la protection des données dans Amazon Simple Workflow Service. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez [Questions fréquentes \(FAQ\) sur la confidentialité des données](#). Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog [Modèle de responsabilité partagée AWS et RGPD \(Règlement général sur la protection des données\)](#) sur le Blog de sécuritéAWS .

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou Gestion des identités et des accès AWS (IAM). Ainsi, chaque utilisateur se voit attribuer

uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- SSL/TLS À utiliser pour communiquer avec AWS les ressources. Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail. Pour plus d'informations sur l'utilisation des CloudTrail sentiers pour capturer AWS des activités, consultez la section [Utilisation des CloudTrail sentiers](#) dans le guide de AWS CloudTrail l'utilisateur.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés par la norme FIPS 140-3 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d'informations sur les points de terminaison FIPS disponibles, consultez [Norme FIPS \(Federal Information Processing Standard\) 140-3](#).

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Nom. Cela inclut lorsque vous travaillez avec Amazon SWF ou autre à Services AWS l'aide de la console, de l'API ou. AWS CLI AWS SDKs Toutes les données que vous entrez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

## Chiffrement dans Amazon Simple Workflow Service

### Chiffrement au repos

Amazon SWF chiffre toujours vos données au repos. Les données d'Amazon Simple Workflow Service sont chiffrées au repos à l'aide d'un chiffrement transparent côté serveur. Cela réduit la lourdeur opérationnelle et la complexité induites par la protection des données sensibles. Le chiffrement au repos vous permet de créer des applications sensibles en matière de sécurité qui sont conformes aux exigences réglementaires et de chiffrement

## Chiffrement en transit

Toutes les données transmises entre Amazon SWF et d'autres services sont chiffrées à l'aide du protocole TLS (Transport Layer Security).

## Identity and Access Management dans Amazon Simple Workflow Service

L'accès à Amazon SWF nécessite des informations d'identification qui AWS peuvent être utilisées pour authentifier vos demandes. Ces informations d'identification doivent être autorisées à accéder aux AWS ressources, par exemple pour récupérer des données d'événements à partir d'autres AWS ressources. Les sections suivantes fournissent des informations détaillées sur la manière dont vous pouvez utiliser [Gestion des identités et des accès AWS \(IAM\)](#) et Amazon SWF pour sécuriser vos ressources en contrôlant l'accès à celles-ci.

Gestion des identités et des accès AWS (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser les ressources Amazon SWF. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

### Rubriques

- [Public ciblé](#)
- [Authentification par des identités](#)
- [Gestion de l'accès à l'aide de politiques](#)
- [Contrôle d'accès](#)
- [Actions politiques pour Amazon SWF](#)
- [Ressources relatives aux politiques pour Amazon SWF](#)
- [Clés de conditions de politique pour Amazon SWF](#)
- [ACLs dans Amazon SWF](#)
- [ABAC avec Amazon SWF](#)
- [Utilisation d'informations d'identification temporaires avec Amazon SWF](#)
- [Autorisations principales interservices pour Amazon SWF](#)
- [Rôles de service pour Amazon SWF](#)

- [Rôles liés à un service pour Amazon SWF](#)
- [Politiques basées sur l'identité pour Amazon SWF](#)
- [Politiques basées sur les ressources au sein d'Amazon SWF](#)
- [Comment Amazon Simple Workflow Service fonctionne avec IAM](#)
- [Exemples de politiques basées sur l'identité pour Amazon Simple Workflow Service](#)
- [Principes de base](#)
- [Politiques IAM d'Amazon SWF](#)
- [Récapitulatif API](#)
- [Stratégies basées sur des balises](#)
- [Points de terminaison Amazon VPC pour Amazon SWF](#)
- [Résolution des problèmes d'identité et d'accès à Amazon Simple Workflow Service](#)

## Public ciblé

La façon dont vous utilisez Gestion des identités et des accès AWS (IAM) varie en fonction de votre rôle :

- Utilisateur du service : demandez des autorisations à votre administrateur si vous ne pouvez pas accéder aux fonctionnalités (voir [Résolution des problèmes d'identité et d'accès à Amazon Simple Workflow Service](#))
- Administrateur du service : déterminez l'accès des utilisateurs et soumettez les demandes d'autorisation (voir [Comment Amazon Simple Workflow Service fonctionne avec IAM](#))
- Administrateur IAM : rédigez des politiques pour gérer l'accès (voir [Exemples de politiques basées sur l'identité pour Amazon Simple Workflow Service](#))

## Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS l'aide de vos informations d'identification. Vous devez être authentifié en tant qu'utilisateur IAM ou en assumant un rôle IAM. Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en tant qu'identité fédérée à l'aide d'informations d'identification provenant d'une source d'identité telle que AWS IAM Identity Center (IAM Identity Center), d'une authentification unique ou d'informations d'identification. Google/Facebook Pour plus d'informations

sur la connexion, consultez [Connexion à votre Compte AWS](#) dans le Guide de l'utilisateur Connexion à AWS .

Pour l'accès par programmation, AWS fournit un SDK et une CLI pour signer les demandes de manière cryptographique. Pour plus d'informations, consultez [Signature AWS Version 4 pour les demandes d'API](#) dans le Guide de l'utilisateur IAM.

## Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une seule identité de connexion appelée utilisateur Compte AWS root qui dispose d'un accès complet à toutes Services AWS les ressources. Il est vivement déconseillé d'utiliser l'utilisateur racine pour vos tâches quotidiennes. Pour les tâches qui requièrent des informations d'identification de l'utilisateur racine, consultez [Tâches qui requièrent les informations d'identification de l'utilisateur racine](#) dans le Guide de l'utilisateur IAM.

## Identité fédérée

Il est recommandé d'obliger les utilisateurs humains à utiliser la fédération avec un fournisseur d'identité pour accéder à Services AWS l'aide d'informations d'identification temporaires.

Une identité fédérée est un utilisateur provenant de l'annuaire de votre entreprise, de votre fournisseur d'identité Web ou Directory Service qui y accède à Services AWS l'aide d'informations d'identification provenant d'une source d'identité. Les identités fédérées assument des rôles qui fournissent des informations d'identification temporaires.

Pour une gestion des accès centralisée, nous vous recommandons d'utiliser AWS IAM Identity Center. Pour plus d'informations, consultez [Qu'est-ce que IAM Identity Center ?](#) dans le Guide de l'utilisateur AWS IAM Identity Center .

## Utilisateurs et groupes IAM

Un [utilisateur IAM](#) est une identité qui dispose d'autorisations spécifiques pour une seule personne ou application. Nous vous recommandons d'utiliser ces informations d'identification temporaires au lieu des utilisateurs IAM avec des informations d'identification à long terme. Pour plus d'informations, voir [Exiger des utilisateurs humains qu'ils utilisent la fédération avec un fournisseur d'identité pour accéder à AWS l'aide d'informations d'identification temporaires](#) dans le guide de l'utilisateur IAM.

[Les groupes IAM](#) spécifient une collection d'utilisateurs IAM et permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Pour plus d'informations, consultez [Cas d'utilisation pour les utilisateurs IAM](#) dans le Guide de l'utilisateur IAM.

## Rôles IAM

Un [rôle IAM](#) est une identité dotée d'autorisations spécifiques qui fournit des informations d'identification temporaires. Vous pouvez assumer un rôle en [passant d'un rôle d'utilisateur à un rôle IAM \(console\)](#) ou en appelant une opération d' AWS API AWS CLI ou d'API. Pour plus d'informations, consultez [Méthodes pour endosser un rôle](#) dans le Guide de l'utilisateur IAM.

Les rôles IAM sont utiles pour l'accès des utilisateurs fédérés, les autorisations temporaires des utilisateurs IAM, les accès intercompte, les accès entre services et les applications exécutées sur Amazon EC2. Pour plus d'informations, consultez [Accès intercompte aux ressources dans IAM](#) dans le Guide de l'utilisateur IAM.

## Gestion de l'accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique définit les autorisations lorsqu'elles sont associées à une identité ou à une ressource. AWS évalue ces politiques lorsqu'un directeur fait une demande. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations les documents de politique JSON, consultez [Vue d'ensemble des politiques JSON](#) dans le Guide de l'utilisateur IAM.

À l'aide de politiques, les administrateurs précisent qui a accès à quoi en définissant quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Un administrateur IAM crée des politiques IAM et les ajoute aux rôles, que les utilisateurs peuvent ensuite assumer. Les politiques IAM définissent les autorisations quelle que soit la méthode que vous utilisez pour exécuter l'opération.

### Politiques basées sur l'identité

Les stratégies basées sur l'identité sont des documents de stratégie d'autorisations JSON que vous attachez à une identité (utilisateur, groupe ou rôle). Ces politiques contrôlent les actions que peuvent exécuter ces identités, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez [Définition d'autorisations IAM personnalisées avec des politiques gérées par le client](#) dans le Guide de l'utilisateur IAM.

Les politiques basées sur l'identité peuvent être des politiques intégrées (intégrées directement dans une seule identité) ou des politiques gérées (politiques autonomes associées à plusieurs identités).

Pour découvrir comment choisir entre des politiques gérées et en ligne, consultez [Choix entre les politiques gérées et les politiques en ligne](#) dans le Guide de l'utilisateur IAM.

## Politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Les exemples incluent les politiques de confiance de rôle IAM et les stratégies de compartiment Amazon S3. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Vous devez [spécifier un principal](#) dans une politique basée sur les ressources.

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser les politiques AWS gérées par IAM dans une stratégie basée sur les ressources.

## Autres types de politique

AWS prend en charge des types de politiques supplémentaires qui peuvent définir les autorisations maximales accordées par les types de politiques les plus courants :

- Limites d'autorisations : une limite des autorisations définit le nombre maximum d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM. Pour plus d'informations, consultez [Limites d'autorisations pour des entités IAM](#) dans le Guide de l'utilisateur IAM.
- Politiques de contrôle des services (SCPs) — Spécifiez les autorisations maximales pour une organisation ou une unité organisationnelle dans AWS Organizations. Pour plus d'informations, consultez [Politiques de contrôle de service](#) dans le Guide de l'utilisateur AWS Organizations .
- Politiques de contrôle des ressources (RCPs) : définissez le maximum d'autorisations disponibles pour les ressources de vos comptes. Pour plus d'informations, voir [Politiques de contrôle des ressources \(RCPs\)](#) dans le guide de AWS Organizations l'utilisateur.
- Politiques de session : politiques avancées que vous passez en tant que paramètre lorsque vous créez par programmation une session temporaire pour un rôle ou un utilisateur fédéré. Pour plus d'informations, consultez [Politiques de session](#) dans le Guide de l'utilisateur IAM.

## Plusieurs types de politique

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser

une demande lorsque plusieurs types de politiques sont impliqués, consultez la section [Logique d'évaluation des politiques](#) dans le guide de l'utilisateur IAM.

## Contrôle d'accès

Vous pouvez disposer d'informations d'identification valides pour authentifier vos demandes, mais vous ne pouvez pas créer de ressources Amazon SWF ou y accéder si vous ne disposez pas d'autorisations. Par exemple, vous devez être autorisé à invoquer AWS Lambda les cibles Amazon Simple Notification Service (Amazon SNS) et Amazon Simple Queue Service (Amazon SQS) associées à vos règles Amazon SWF.

Les sections suivantes décrivent comment gérer les autorisations pour Amazon SWF. Nous vous recommandons de lire d'abord la présentation.

- [Principes de base](#)
- [Politiques IAM d'Amazon SWF](#)
- [Règles de rédaction pour Amazon SWF](#)

## Actions politiques pour Amazon SWF

Prend en charge les actions de politique : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Action` d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Intégration d'actions dans une politique afin d'accorder l'autorisation d'exécuter les opérations associées.

Pour consulter la liste des actions Amazon SWF, consultez la section [Ressources définies par Amazon Simple Workflow Service](#) dans la référence d'autorisation du service.

Les actions de politique dans Amazon SWF utilisent le préfixe suivant avant l'action :

```
swf
```

Pour indiquer plusieurs actions dans une seule déclaration, séparez-les par des virgules.

```
"Action": [
```

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

Pour consulter des exemples de politiques basées sur l'identité Amazon SWF, consultez. [Exemples de politiques basées sur l'identité pour Amazon Simple Workflow Service](#)

## Ressources relatives aux politiques pour Amazon SWF

Prend en charge les ressources de politique : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément de politique JSON `Resource` indique le ou les objets auxquels l'action s'applique. Il est recommandé de définir une ressource à l'aide de son [Amazon Resource Name \(ARN\)](#). Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, utilisez un caractère générique (\*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

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

Pour consulter la liste des types de ressources Amazon SWF et leurs caractéristiques ARNs, consultez la section [Actions définies par Amazon Simple Workflow Service](#) dans la référence d'autorisation du service. Pour savoir avec quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez la section [Ressources définies par Amazon Simple Workflow Service](#).

Pour consulter des exemples de politiques basées sur l'identité Amazon SWF, consultez. [Exemples de politiques basées sur l'identité pour Amazon Simple Workflow Service](#)

## Clés de conditions de politique pour Amazon SWF

Prend en charge les clés de condition de politique spécifiques au service : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Condition` indique à quel moment les instructions s'exécutent en fonction de critères définis. Vous pouvez créer des expressions conditionnelles qui utilisent des [opérateurs de condition](#),

tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique aux valeurs de la demande. Pour voir toutes les clés de condition AWS globales, voir les clés de [contexte de condition AWS globales](#) dans le guide de l'utilisateur IAM.

Pour consulter la liste des clés de condition Amazon SWF, consultez la section Clés de [condition pour Amazon Simple Workflow Service](#) dans la référence d'autorisation du service. Pour savoir avec quelles actions et ressources vous pouvez utiliser une clé de condition, consultez la section [Ressources définies par Amazon Simple Workflow Service](#).

Pour consulter des exemples de politiques basées sur l'identité Amazon SWF, consultez. [Exemples de politiques basées sur l'identité pour Amazon Simple Workflow Service](#)

## ACLs dans Amazon SWF

Supports ACLs : Non

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLs sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

## ABAC avec Amazon SWF

Prend en charge ABAC (identifications dans les politiques) : partiellement

Le contrôle d'accès par attributs (ABAC) est une stratégie d'autorisation qui définit les autorisations en fonction des attributs nommés balise. Vous pouvez associer des balises aux entités et aux AWS ressources IAM, puis concevoir des politiques ABAC pour autoriser les opérations lorsque la balise du principal correspond à la balise de la ressource.

Pour contrôler l'accès basé sur des étiquettes, vous devez fournir les informations d'étiquette dans l'[élément de condition](#) d'une politique utilisant les clés de condition `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`.

Si un service prend en charge les trois clés de condition pour tous les types de ressources, alors la valeur pour ce service est Oui. Si un service prend en charge les trois clés de condition pour certains types de ressources uniquement, la valeur est Partielle.

Pour plus d'informations sur ABAC, consultez [Définition d'autorisations avec l'autorisation ABAC](#) dans le Guide de l'utilisateur IAM. Pour accéder à un didacticiel décrivant les étapes de configuration de l'ABAC, consultez [Utilisation du contrôle d'accès par attributs \(ABAC\)](#) dans le Guide de l'utilisateur IAM.

## Utilisation d'informations d'identification temporaires avec Amazon SWF

Prend en charge les informations d'identification temporaires : oui

Les informations d'identification temporaires fournissent un accès à court terme aux AWS ressources et sont automatiquement créées lorsque vous utilisez la fédération ou que vous changez de rôle. AWS recommande de générer dynamiquement des informations d'identification temporaires au lieu d'utiliser des clés d'accès à long terme. Pour plus d'informations, consultez [Informations d'identification de sécurité temporaires dans IAM](#) et [Services AWS compatibles avec IAM](#) dans le Guide de l'utilisateur IAM.

## Autorisations principales interservices pour Amazon SWF

Prend en charge les sessions d'accès direct (FAS) : oui

Les sessions d'accès direct (FAS) utilisent les autorisations du principal appelant et Service AWS, combinées Service AWS à la demande d'envoi de demandes aux services en aval. Pour plus de détails sur la politique relative à la transmission de demandes FAS, consultez la section [Sessions de transmission d'accès](#).

## Rôles de service pour Amazon SWF

Prend en charge les rôles de service : oui

Un rôle de service est un [rôle IAM](#) qu'un service endosse pour accomplir des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer un rôle de service à partir d'IAM. Pour plus d'informations, consultez [Création d'un rôle pour la délégation d'autorisations à un Service AWS](#) dans le Guide de l'utilisateur IAM.

### Warning

La modification des autorisations associées à un rôle de service peut perturber les fonctionnalités d'Amazon SWF. Modifiez les rôles de service uniquement lorsque Amazon SWF fournit des instructions à cet effet.

## Rôles liés à un service pour Amazon SWF

Prend en charge les rôles liés à un service : non

Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

Pour plus d'informations sur la création ou la gestion des rôles liés à un service, consultez [Services AWS qui fonctionnent avec IAM](#). Recherchez un service dans le tableau qui inclut un Yes dans la colonne Rôle lié à un service. Choisissez le lien Oui pour consulter la documentation du rôle lié à ce service.

## Politiques basées sur l'identité pour Amazon SWF

Prend en charge les politiques basées sur l'identité : oui

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez [Définition d'autorisations IAM personnalisées avec des politiques gérées par le client](#) dans le Guide de l'utilisateur IAM.

Avec les politiques IAM basées sur l'identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Pour découvrir tous les éléments que vous utilisez dans une politique JSON, consultez [Références des éléments de politique JSON IAM](#) dans le Guide de l'utilisateur IAM.

## Exemples de politiques basées sur l'identité pour Amazon SWF

Pour consulter des exemples de politiques basées sur l'identité Amazon SWF, consultez. [Exemples de politiques basées sur l'identité pour Amazon Simple Workflow Service](#)

## Politiques basées sur les ressources au sein d'Amazon SWF

Prend en charge les politiques basées sur les ressources : non

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Par exemple, les politiques de confiance de rôle IAM et les politiques de compartiment Amazon S3 sont des politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve

la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez [spécifier un principal](#) dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Pour permettre un accès intercompte, vous pouvez spécifier un compte entier ou des entités IAM dans un autre compte en tant que principal dans une politique basée sur les ressources. Pour plus d'informations, consultez [Accès intercompte aux ressources dans IAM](#) dans le Guide de l'utilisateur IAM.

## Comment Amazon Simple Workflow Service fonctionne avec IAM

Avant d'utiliser IAM pour gérer l'accès à Amazon SWF, découvrez quelles fonctionnalités IAM peuvent être utilisées avec Amazon SWF.

Fonctionnalités IAM que vous pouvez utiliser avec Amazon Simple Workflow Service

Fonctionnalité IAM	Prise en charge d'Amazon SWF
<a href="#">Politiques basées sur l'identité</a>	Oui
<a href="#">Politiques basées sur les ressources</a>	Non
<a href="#">Actions de politique</a>	Oui
<a href="#">Ressources de politique</a>	Oui
<a href="#">Clés de condition de politique (spécifiques au service)</a>	Oui
<a href="#">ACLs</a>	Non
<a href="#">ABAC (identifications dans les politiques)</a>	Partielle
<a href="#">Informations d'identification temporaires</a>	Oui
<a href="#">Autorisations de principal</a>	Oui
<a href="#">Rôles de service</a>	Oui
<a href="#">Rôles liés à un service</a>	Non

Pour obtenir une vue d'ensemble de la façon dont Amazon SWF et les autres AWS services fonctionnent avec la plupart des fonctionnalités IAM, consultez les [AWS services compatibles avec IAM dans le guide de l'utilisateur IAM](#).

## Exemples de politiques basées sur l'identité pour Amazon Simple Workflow Service

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou à modifier des ressources Amazon SWF. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM.

Pour apprendre à créer une politique basée sur l'identité IAM à l'aide de ces exemples de documents de politique JSON, consultez [Création de politiques IAM \(console\)](#) dans le Guide de l'utilisateur IAM.

Pour plus de détails sur les actions et les types de ressources définis par Amazon SWF, y compris le format de chaque type de ressource, consultez la section [Actions, ressources et clés de condition ARNs pour Amazon Simple Workflow Service](#) dans la référence d'autorisation du service.

### Rubriques

- [Bonnes pratiques en matière de politiques](#)
- [Utilisation de la console Amazon SWF](#)
- [Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations](#)

### Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer des ressources Amazon SWF dans votre compte. Ces actions peuvent entraîner des frais pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire davantage les autorisations en définissant des politiques gérées par les AWS clients spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez [politiques gérées par AWS](#) ou [politiques gérées par AWS pour les activités professionnelles](#) dans le Guide de l'utilisateur IAM.

- Accordez les autorisations de moindre privilège : lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation d'IAM pour appliquer des autorisations, consultez [politiques et autorisations dans IAM](#) dans le Guide de l'utilisateur IAM.
- Utilisez des conditions dans les politiques IAM pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique Service AWS, tel que CloudFormation. Pour plus d'informations, consultez [Conditions pour éléments de politique JSON IAM](#) dans le Guide de l'utilisateur IAM.
- Utilisez l'Analyseur d'accès IAM pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : l'Analyseur d'accès IAM valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez [Validation de politiques avec IAM Access Analyzer](#) dans le Guide de l'utilisateur IAM.
- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root, activez l'authentification MFA pour une sécurité accrue. Compte AWS Pour exiger la MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez [Sécurisation de l'accès aux API avec MFA](#) dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez [Bonnes pratiques de sécurité dans IAM](#) dans le Guide de l'utilisateur IAM.

## Utilisation de la console Amazon SWF

Pour accéder à la console Amazon Simple Workflow Service, vous devez disposer d'un ensemble minimal d'autorisations. Ces autorisations doivent vous permettre de répertorier et de consulter les informations relatives aux ressources Amazon SWF présentes dans votre. Compte AWS Si vous créez une politique basée sur l'identité qui est plus restrictive que l'ensemble minimum d'autorisations

requis, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs ou rôles) tributaires de cette politique.

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement l'API AWS CLI ou l' AWS API. Autorisez plutôt l'accès à uniquement aux actions qui correspondent à l'opération d'API qu'ils tentent d'effectuer.

Pour garantir que les utilisateurs et les rôles peuvent toujours utiliser la console Amazon SWF, associez également Amazon *ConsoleAccess* SWF *ReadOnly* AWS ou la politique gérée aux entités. Pour plus d'informations, consultez [Ajout d'autorisations à un utilisateur](#) dans le Guide de l'utilisateur IAM.

## Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Cet exemple montre comment créer une politique qui permet aux utilisateurs IAM d'afficher les politiques en ligne et gérées attachées à leur identité d'utilisateur. Cette politique inclut les autorisations permettant d'effectuer cette action sur la console ou par programmation à l'aide de l'API AWS CLI or AWS .

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
```

```
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
    ],
    "Resource": "*"
}
]
```

## Principes de base

Le contrôle d'accès Amazon SWF repose principalement sur deux types d'autorisations :

- Autorisations relatives aux ressources : quelles ressources Amazon SWF un utilisateur peut accéder.

Vous pouvez uniquement exprimer des autorisations de ressources pour les domaines.

- Autorisations d'API : quelles actions Amazon SWF un utilisateur peut appeler.

L'approche la plus simple consiste à accorder un accès complet au compte, c'est-à-dire à lancer n'importe quelle action Amazon SWF dans n'importe quel domaine, ou à refuser complètement l'accès. Cependant, IAM soutient une approche plus précise du contrôle d'accès, qui est souvent plus utile. Par exemple, vous pouvez accorder les accès suivants :

- Permettez à un utilisateur d'appeler n'importe quelle action Amazon SWF sans restrictions, mais uniquement dans un domaine spécifique. Vous pouvez utiliser cette stratégie pour permettre aux applications de flux de travail qui sont en cours de développement d'utiliser toutes les actions, mais uniquement dans un domaine « sandbox ».
- Autorisez un utilisateur à accéder à n'importe quel domaine, mais limitez la façon dont il utilise l'API. Vous pouvez tirer parti de cette stratégie pour permettre à une application d'« audit » d'appeler l'API dans n'importe quel domaine, mais uniquement en lecture seule.
- Autorisez un utilisateur à appeler uniquement un ensemble limité d'actions dans certains domaines. Vous pouvez recourir à cette stratégie pour permettre à un démarreur de flux de travail d'appeler uniquement l'action `StartWorkflowExecution` dans un domaine spécifié.

Le contrôle d'accès Amazon SWF repose sur les principes suivants :

- Les décisions de contrôle d'accès sont basées uniquement sur les politiques IAM ; tous les audits et manipulations des politiques sont effectués via IAM.
- Le modèle de contrôle d'accès utilise une deny-by-default politique ; tout accès non explicitement autorisé est refusé.
- Vous contrôlez l'accès aux ressources Amazon SWF en associant les politiques IAM appropriées aux acteurs du flux de travail.
- Vous pouvez uniquement exprimer des autorisations de ressources pour les domaines.
- Pour limiter davantage l'utilisation de certaines actions, vous pouvez appliquer des conditions à un ou plusieurs paramètres.
- Si vous accordez l'autorisation d'utilisation [RespondDecisionTaskCompleted](#), vous pouvez exprimer des autorisations pour la liste des décisions incluses dans cette action.

Chacune des décisions a un ou plusieurs paramètres, tout comme un appel d'API classique. Pour que les stratégies soient aussi lisibles que possible, vous pouvez exprimer des autorisations au niveau des décisions comme s'il s'agissait d'appels d'API réels, y compris en appliquant des conditions à certains paramètres. Ces types d'autorisations sont des autorisations de pseudo-API.

Pour afficher un résumé des paramètres qui peuvent être limités à l'aide de conditions pour l'API classique et la pseudo-API, consultez la section [Récapitulatif API](#).

## Politiques IAM d'Amazon SWF

Une politique IAM contient un ou plusieurs Statement éléments, chacun contenant un ensemble d'éléments qui définissent la stratégie. Pour une liste complète des éléments et une discussion générale sur la façon de créer des politiques, voir [The Access Policy Language](#). Le contrôle d'accès Amazon SWF repose sur les éléments suivants :

### Effet

[Obligatoire] L'effet de l'instruction : deny ou allow.

#### Note

Vous devez autoriser l'accès de manière explicite ; IAM refuse l'accès par défaut.

## Ressource

(Obligatoire) La ressource (entité d'un AWS service avec laquelle un utilisateur peut interagir) à laquelle s'applique l'instruction.

Vous pouvez uniquement exprimer des autorisations de ressources pour les domaines. Par exemple, une stratégie peut autoriser l'accès à seulement certains domaines de votre compte. Pour exprimer des autorisations pour un domaine, définissez `Resource` le nom de ressource Amazon (ARN) du domaine, au format « `arn:aws:swf:::Region/domain/` ». `AccountID` `DomainName` `Region` est la AWS région, `AccountID` l'identifiant du compte sans tiret et `DomainName` le nom de domaine.

## Action

(Obligatoire) L'action à laquelle s'applique l'instruction, à laquelle vous faites référence en utilisant le format suivant : `serviceId:action`. Pour Amazon SWF, définissez sur `serviceID`. `swf` Par exemple, `swf:StartWorkflowExecution` fait référence à l'[StartWorkflowExecution](#) action et est utilisé pour contrôler quels utilisateurs sont autorisés à démarrer des flux de travail.

Si vous accordez l'autorisation d'utilisation [RespondDecisionTaskCompleted](#), vous pouvez également contrôler l'accès à la liste de décisions incluse en utilisant `Action` pour exprimer les autorisations pour la pseudo-API. Comme IAM refuse l'accès par défaut, la décision d'un décideur doit être explicitement autorisée, sinon elle ne sera pas acceptée. Vous pouvez utiliser la valeur `*` pour autoriser toutes les décisions.

## Condition

(Facultatif) Exprime une contrainte au niveau d'un ou de plusieurs paramètres d'action, ce qui limite les valeurs acceptées.

Les actions Amazon SWF ont souvent une portée étendue, que vous pouvez réduire en utilisant les conditions IAM. Par exemple, pour limiter les listes de tâches auxquelles l'[PollForActivityTask](#) action est autorisée à accéder, vous devez inclure un `Condition` et utiliser la `swf:taskList.name` clé pour spécifier les listes autorisées.

Vous pouvez exprimer des contraintes pour les entités suivantes.

- Le type de flux de travail Le nom et la version ont des clés distinctes.
- Le type d'activité. Le nom et la version ont des clés distinctes.
- Les listes de tâches.

- Balises. Vous pouvez spécifier plusieurs balises pour certaines actions. Dans ce cas, chaque balise possède une clé distincte.

#### Note

Pour Amazon SWF, les valeurs sont toutes des chaînes. Vous pouvez donc contraindre un paramètre à l'aide d'un opérateur de chaîne tel que `StringEquals`, qui limite le paramètre à une chaîne spécifiée. Toutefois, avec les opérateurs de comparaison de chaîne habituels comme `StringEquals`, toutes les requêtes doivent inclure le paramètre. Si vous n'incluez pas le paramètre explicitement et s'il n'existe aucune valeur par défaut, telle que la liste de tâches par défaut fournie lors de l'enregistrement du type, l'accès est refusé.

Il est souvent utile de traiter les conditions comme étant facultatives. De la sorte, vous pouvez appeler une action sans être obligé d'inclure le paramètre associé.

Par exemple, vous pouvez autoriser un décideur à spécifier un ensemble de [RespondDecisionTaskCompleted](#) décisions, mais également à n'en spécifier qu'une pour un appel en particulier. Dans ce cas, utilisez un opérateur `StringEqualsIfExists` pour limiter les paramètres appropriés, afin d'autoriser l'accès si le paramètre satisfait à la condition, sans toutefois refuser l'accès si le paramètre est absent.

Pour obtenir une liste complète des paramètres que vous pouvez restreindre et des clés associées, consultez la section [Récapitulatif API](#).

La section suivante fournit des exemples de création de politiques Amazon SWF. Pour plus d'informations, consultez la section [Conditions de chaîne](#).

## Règles de rédaction pour Amazon SWF

Un flux de travail est composé de plusieurs acteurs : activités, décideurs, etc. Vous pouvez contrôler l'accès de chaque acteur en attachant une politique IAM appropriée.

Avec l'action suivante, l'acteur aura un accès complet à son compte dans toutes les régions :

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

Vous pouvez utiliser des caractères génériques de sorte qu'une valeur unique représente plusieurs ressources, actions ou régions.

- Le premier caractère générique (\*) de la `Resource` valeur indique que les autorisations relatives aux ressources s'appliquent à toutes les régions.

Pour limiter les autorisations à une seule région, remplacez le caractère générique par la chaîne de région appropriée, telle que `us-east-1`.

- Le deuxième caractère générique (\*) dans la valeur `Resource` permet à l'acteur d'accéder à tous les domaines du compte dans les régions spécifiées.
- Le caractère générique (\*) contenu dans la `Action` valeur permet à l'acteur d'appeler n'importe quelle action Amazon SWF.

Pour plus d'informations sur la façon d'utiliser les caractères génériques, consultez la page [Descriptions d'éléments](#).

## Autorisations de domaine

Pour restreindre les flux de travail d'un département à un domaine spécifique, vous pouvez accorder une autorisation permettant à un acteur de prendre n'importe quelle action, mais uniquement pour un département spécifique.

Pour accorder à un acteur l'accès à plusieurs domaines, exprimez l'autorisation pour chaque domaine sous forme de liste de déclarations :

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

Vous pouvez autoriser un acteur à utiliser n'importe quelle action Amazon SWF dans les domaines `department1` et `department2`. Vous pouvez aussi parfois utiliser des caractères génériques pour représenter plusieurs domaines.

## Contraintes et autorisations de l'API

Vous pouvez contrôler les actions qu'un acteur peut utiliser en spécifiant l'action dans l'`Action` élément.

Avec l'action suivante, un acteur ne peut appeler que `StartWorkflowExecution` pour démarrer des flux de travail. Il ne peut utiliser aucune autre action.

- Action : `swf:StartWorkflowExecution`

## Conditions

Vous pouvez éventuellement limiter les valeurs de paramètres autorisées de l'action en utilisant un `Condition` élément.

Pour limiter les flux de travail qu'un acteur peut démarrer, limitez une ou plusieurs valeurs de `StartWorkflowExecution` paramètres, comme suit :

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

Un acteur avec les contraintes précédentes ne peut fonctionner que `version2 workflow1` et les deux paramètres doivent être inclus dans la demande.

Pour limiter un paramètre sans nécessiter sa présence dans une requête, utilisez un opérateur `StringEqualsIfExists`, comme suit :

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

Un acteur doté de la politique précédente peut éventuellement spécifier une liste de tâches lorsqu'il lance l'exécution d'un flux de travail.

Vous pouvez limiter la liste des balises pour certaines actions. Chaque balise possède une clé distincte. Vous pouvez donc l'utiliser `swf:tagList.member.0` pour contraindre la première balise de la liste, `swf:tagList.member.1` pour contraindre la deuxième balise de la liste, etc., jusqu'à un maximum de 5.

Vous devez faire attention à la manière dont vous limitez les listes de balises. Par exemple, la condition suivante n'est pas recommandée.

La condition suivante n'est pas recommandée car elle vous permet de spécifier éventuellement l'un `some_ok_tag` ou l'autre `another_ok_tag`. Toutefois, la condition limite uniquement le premier élément de la liste de balises. La liste peut contenir des éléments supplémentaires avec des valeurs arbitraires qui seraient tous autorisés car la condition ne s'applique à aucune condition `swf:tagList.member.1` `swf:tagList.member.2`, etc.

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

L'un des moyens de résoudre le problème précédent consiste à interdire l'utilisation de listes de balises.

La stratégie suivante garantit que seulement `some_ok_tag` ou `another_ok_tag` sont autorisés en imposant un seul élément dans la liste.

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

## Contraintes et autorisations de pseudo-API

Pour limiter les décisions disponibles `RespondDecisionTaskCompleted`, vous devez d'abord autoriser l'acteur à appeler `RespondDecisionTaskCompleted`. Vous exprimez ensuite les autorisations pour les pseudo-membres d'API appropriés en utilisant la même syntaxe que pour l'API normale, comme suit :

- Déclaration 1

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

Action : `swf:RespondDecisionTaskCompleted`

- Déclaration 2

Ressource : \*

Action : `swf:ScheduleActivityTask`

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

Le premier Statement permet à l'acteur d'appeler `RespondDecisionTaskCompleted`. La deuxième déclaration permet à l'acteur d'utiliser la `ScheduleActivityTask` décision de demander à Amazon SWF de planifier une tâche d'activité. Pour autoriser toutes les décisions, remplacez « `swf : ScheduleActivityTask` » par « `swf : *` ».

Vous pouvez utiliser des opérateurs de condition pour limiter les paramètres, tout comme avec l'API classique. Dans `StringEquals` l'exemple précédent, l'opérateur `Condition RespondDecisionTaskCompleted` permet de planifier une tâche d'activité pour cette `SomeActivityType` activité, et il doit planifier cette tâche. Si vous voulez autoriser `RespondDecisionTaskCompleted` à utiliser une valeur de paramètre sans qu'elle ne soit obligatoire, vous pouvez utiliser l'opérateur `StringEqualsIfExists` à la place.

## AWS politique gérée : `SimpleWorkflowFullAccess`

Vous pouvez associer la politique `SimpleWorkflowFullAccess` à vos identités IAM.

Cette politique fournit un accès complet au service de configuration Amazon SWF.

## Limitations du modèle de service sur les politiques IAM

Vous devez tenir compte des contraintes du modèle de service lorsque vous créez des politiques IAM. Il est possible de créer une politique IAM syntaxiquement valide qui représente une demande Amazon SWF non valide ; une demande autorisée en termes de contrôle d'accès peut toujours échouer car elle n'est pas valide.

Par exemple, le modèle de service Amazon SWF n'autorise pas l'utilisation `typeFilter` des `tagFilter` paramètres et dans la même [ListOpenWorkflowExecutions](#) demande. La condition suivante autoriserait les appels que le service rejettera (en lançant) en `ValidationException` tant que demande non valide :

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

```
}  
}
```

## Récapitulatif API

Cette section décrit brièvement comment vous pouvez utiliser les politiques IAM pour contrôler la manière dont un acteur peut utiliser chaque API et pseudo API pour accéder aux ressources Amazon SWF.

- Pour toutes les actions (sauf `RegisterDomain` et `ListDomains`), vous pouvez exprimer des autorisations pour la ressource de domaine afin d'autoriser ou de refuser l'accès à tout ou partie des domaines d'un compte.
- Vous pouvez autoriser ou refuser l'accès pour n'importe quel membre de l'API classique et, si vous accordez l'autorisation d'appeler [RespondDecisionTaskCompleted](#), pour n'importe quel membre de la pseudo-API.
- Vous pouvez utiliser une condition pour limiter les valeurs autorisées de certains paramètres.

Les sections suivantes répertorient les paramètres que vous pouvez limiter pour chaque membre de l'API classique ou de la pseudo-API. Elles fournissent également la clé associée et indiquent les limitations vous permettant de contrôler l'accès au domaine.

### API classique

Cette section répertorie les membres de l'API classique et décrit brièvement les paramètres qui peuvent être limités, ainsi que les clés associées. Elle indique également les limitations liées à la façon dont vous pouvez contrôler l'accès au domaine.

#### [CountClosedWorkflowExecutions](#)

- `tagFilter.tag`— Contrainte de chaîne. La clé est `swf:tagFilter.tag`.
- `typeFilter.name`— Contrainte de chaîne. La clé est `swf:typeFilter.name`.
- `typeFilter.version`— Contrainte de chaîne. La clé est `swf:typeFilter.version`.

#### Note

`CountClosedWorkflowExecutions` exige que `typeFilter` et `tagFilter` soient mutuellement exclusifs.

## [CountOpenWorkflowExecutions](#)

- `tagFilter.tag`— Contrainte de chaîne. La clé est `swf:tagFilter.tag`.
- `typeFilter.name`— Contrainte de chaîne. La clé est `swf:typeFilter.name`.
- `typeFilter.version`— Contrainte de chaîne. La clé est `swf:typeFilter.version`.

### Note

`CountOpenWorkflowExecutions` exige que `typeFilter` et `tagFilter` soient mutuellement exclusifs.

## [CountPendingActivityTasks](#)

- `taskList.name`— Contrainte de chaîne. La clé est `swf:taskList.name`.

## [CountPendingDecisionTasks](#)

- `taskList.name`— Contrainte de chaîne. La clé est `swf:taskList.name`.

## [DeleteActivityType](#)

- `activityType.name`— Contrainte de chaîne. La clé est `swf:activityType.name`.
- `activityType.version`— Contrainte de chaîne. La clé est `swf:activityType.version`.

## [DeprecateActivityType](#)

- `activityType.name`— Contrainte de chaîne. La clé est `swf:activityType.name`.
- `activityType.version`— Contrainte de chaîne. La clé est `swf:activityType.version`.

## [DeprecateDomain](#)

- Vous ne pouvez pas limiter les paramètres de cette action.

## [DeleteWorkflowType](#)

- `workflowType.name`— Contrainte de chaîne. La clé est `swf:workflowType.name`.
- `workflowType.version`— Contrainte de chaîne. La clé est `swf:workflowType.version`.

### [DeprecateWorkflowType](#)

- `workflowType.name`— Contrainte de chaîne. La clé est `swf:workflowType.name`.
- `workflowType.version`— Contrainte de chaîne. La clé est `swf:workflowType.version`.

### [DescribeActivityType](#)

- `activityType.name`— Contrainte de chaîne. La clé est `swf:activityType.name`.
- `activityType.version`— Contrainte de chaîne. La clé est `swf:activityType.version`.

### [DescribeDomain](#)

- Vous ne pouvez pas limiter les paramètres de cette action.

### [DescribeWorkflowExecution](#)

- Vous ne pouvez pas limiter les paramètres de cette action.

### [DescribeWorkflowType](#)

- `workflowType.name`— Contrainte de chaîne. La clé est `swf:workflowType.name`.
- `workflowType.version`— Contrainte de chaîne. La clé est `swf:workflowType.version`.

### [GetWorkflowExecutionHistory](#)


- Vous ne pouvez pas limiter les paramètres de cette action.

### [ListActivityTypes](#)

- Vous ne pouvez pas limiter les paramètres de cette action.

### [ListClosedWorkflowExecutions](#)

- `tagFilter.tag`— Contrainte de chaîne. La clé est `swf:tagFilter.tag`.
- `typeFilter.name`— Contrainte de chaîne. La clé est `swf:typeFilter.name`.
- `typeFilter.version`— Contrainte de chaîne. La clé est `swf:typeFilter.version`.

 Note


`ListClosedWorkflowExecutions` exige que `typeFilter` et `tagFilter` soient mutuellement exclusifs.

### [ListDomains](#)

- Vous ne pouvez pas limiter les paramètres de cette action.

### [ListOpenWorkflowExecutions](#)

- `tagFilter.tag`— Contrainte de chaîne. La clé est `swf:tagFilter.tag`.
- `typeFilter.name`— Contrainte de chaîne. La clé est `swf:typeFilter.name`.
- `typeFilter.version`— Contrainte de chaîne. La clé est `swf:typeFilter.version`.

 Note

`ListOpenWorkflowExecutions` exige que `typeFilter` et `tagFilter` soient mutuellement exclusifs.

### [ListWorkflowTypes](#)

- Vous ne pouvez pas limiter les paramètres de cette action.

### [PollForActivityTask](#)

- `taskList.name`— Contrainte de chaîne. La clé est `swf:taskList.name`.

### [PollForDecisionTask](#)

- `taskList.name`— Contrainte de chaîne. La clé est `swf:taskList.name`.

### [RecordActivityTaskHeartbeat](#)

- Vous ne pouvez pas limiter les paramètres de cette action.

### [RegisterActivityType](#)

- `defaultTaskList.name`— Contrainte de chaîne. La clé est `swf:defaultTaskList.name`.
- `name`— Contrainte de chaîne. La clé est `swf:name`.
- `version`— Contrainte de chaîne. La clé est `swf:version`.

### [RegisterDomain](#)

- `name`— Le nom du domaine en cours d'enregistrement est disponible en tant que ressource de cette action.

### [RegisterWorkflowType](#)

- `defaultTaskList.name`— Contrainte de chaîne. La clé est `swf:defaultTaskList.name`.
- `name`— Contrainte de chaîne. La clé est `swf:name`.
- `version`— Contrainte de chaîne. La clé est `swf:version`.

### [RequestCancelWorkflowExecution](#)

- Vous ne pouvez pas limiter les paramètres de cette action.

### [RespondActivityTaskCanceled](#)

- Vous ne pouvez pas limiter les paramètres de cette action.

### [RespondActivityTaskCompleted](#)

- Vous ne pouvez pas limiter les paramètres de cette action.

## [RespondActivityTaskFailed](#)

- Vous ne pouvez pas limiter les paramètres de cette action.

## [RespondDecisionTaskCompleted](#)

- `decisions.member.N`— Limité indirectement par le biais de pseudo-autorisations d'API. Pour en savoir plus, consultez [Pseudo-API](#).

## [SignalWorkflowExecution](#)

- Vous ne pouvez pas limiter les paramètres de cette action.

## [StartWorkflowExecution](#)

- `tagList.member.0`— Contrainte de chaîne. La clé est `swf:tagList.member.0`.
- `tagList.member.1`— Contrainte de chaîne. La clé est `swf:tagList.member.1`.
- `tagList.member.2`— Contrainte de chaîne. La clé est `swf:tagList.member.2`.
- `tagList.member.3`— Contrainte de chaîne. La clé est `swf:tagList.member.3`.
- `tagList.member.4`— Contrainte de chaîne. La clé est `swf:tagList.member.4`.
- `taskList.name`— Contrainte de chaîne. La clé est `swf:taskList.name`.
- `workflowType.name`— Contrainte de chaîne. La clé est `swf:workflowType.name`.
- `workflowType.version`— Contrainte de chaîne. La clé est `swf:workflowType.version`.

### Note

Vous ne pouvez pas limiter plus de cinq balises.

## [TerminateWorkflowExecution](#)

- Vous ne pouvez pas limiter les paramètres de cette action.

## Pseudo-API

Cette section répertorie les membres de la pseudo-API, qui représentent les décisions incluses dans [RespondDecisionTaskCompleted](#). Si vous avez accordé l'autorisation d'utiliser RespondDecisionTaskCompleted, la stratégie peut exprimer des autorisations pour les membres de cette API de la même manière que l'API classique. Pour restreindre davantage certains membres de la pseudo-API, vous pouvez définir des conditions au niveau d'un ou de plusieurs paramètres. Cette section répertorie les membres de la pseudo-API et décrit brièvement les paramètres qui peuvent être limités, ainsi que les clés associées.

### Note

Les clés `aws:SourceIP`, `aws:UserAgent` et `aws:SecureTransport` ne sont pas disponibles pour la pseudo-API. Si la stratégie de sécurité prévue nécessite que ces clés contrôlent l'accès à la pseudo-API, vous pouvez les utiliser avec l'action RespondDecisionTaskCompleted.

### CancelTimer

- Vous ne pouvez pas limiter les paramètres de cette action.

### CancelWorkflowExecution

- Vous ne pouvez pas limiter les paramètres de cette action.


### CompleteWorkflowExecution

- Vous ne pouvez pas limiter les paramètres de cette action.

### ContinueAsNewWorkflowExecution

- `tagList.member.0`— Contrainte de chaîne. La clé est `swf:tagList.member.0`.
- `tagList.member.1`— Contrainte de chaîne. La clé est `swf:tagList.member.1`.
- `tagList.member.2`— Contrainte de chaîne. La clé est `swf:tagList.member.2`.
- `tagList.member.3`— Contrainte de chaîne. La clé est `swf:tagList.member.3`.
- `tagList.member.4`— Contrainte de chaîne. La clé est `swf:tagList.member.4`.

- `taskList.name`— Contrainte de chaîne. La clé est `swf:taskList.name`.
- `workflowTypeVersion`— Contrainte de chaîne. La clé est `swf:workflowTypeVersion`.

 Note

Vous ne pouvez pas limiter plus de cinq balises.

### FailWorkflowExecution

- Vous ne pouvez pas limiter les paramètres de cette action.

### RecordMarker

- Vous ne pouvez pas limiter les paramètres de cette action.

### RequestCancelActivityTask

- Vous ne pouvez pas limiter les paramètres de cette action.

### RequestCancelExternalWorkflowExecution

- Vous ne pouvez pas limiter les paramètres de cette action.

### ScheduleActivityTask

- `activityType.name`— Contrainte de chaîne. La clé est `swf:activityType.name`.
- `activityType.version`— Contrainte de chaîne. La clé est `swf:activityType.version`.
- `taskList.name`— Contrainte de chaîne. La clé est `swf:taskList.name`.

### SignalExternalWorkflowExecution

- Vous ne pouvez pas limiter les paramètres de cette action.

### StartChildWorkflowExecution

- `tagList.member.0`— Contrainte de chaîne. La clé est `swf:tagList.member.0`.
- `tagList.member.1`— Contrainte de chaîne. La clé est `swf:tagList.member.1`.
- `tagList.member.2`— Contrainte de chaîne. La clé est `swf:tagList.member.2`.
- `tagList.member.3`— Contrainte de chaîne. La clé est `swf:tagList.member.3`.
- `tagList.member.4`— Contrainte de chaîne. La clé est `swf:tagList.member.4`.
- `taskList.name`— Contrainte de chaîne. La clé est `swf:taskList.name`.
- `workflowType.name`— Contrainte de chaîne. La clé est `swf:workflowType.name`.
- `workflowType.version`— Contrainte de chaîne. La clé est `swf:workflowType.version`.

### Note

Vous ne pouvez pas limiter plus de cinq balises.

## StartTimer

- Vous ne pouvez pas limiter les paramètres de cette action.

## Stratégies basées sur des balises

Amazon SWF prend en charge les politiques basées sur les balises. Par exemple, vous pouvez restreindre les domaines Amazon SWF qui incluent une balise avec la clé `environment` et la valeur `production` avec la condition suivante :

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

Pour plus d'informations sur le balisage, consultez :

- [Balises dans Amazon SWF](#)
- [Contrôle de l'accès à l'aide de balises IAM](#)

## Points de terminaison Amazon VPC pour Amazon SWF

### Note

AWS PrivateLink le support n'est actuellement disponible que dans les régions AWS Top AWS Secret - Est, Secret Region et Chine.

Si vous utilisez Amazon Virtual Private Cloud (Amazon VPC) pour héberger vos AWS ressources, vous pouvez établir une connexion entre vos flux de travail Amazon VPC et Amazon Simple Workflow Service. Vous pouvez utiliser cette connexion avec vos flux de travail Amazon SWF sans passer par l'Internet public.

Amazon VPC vous permet de lancer AWS des ressources dans un réseau virtuel personnalisé. Vous pouvez utiliser un VPC pour contrôler vos paramètres réseau, tels que la plage d'adresses IP, les sous-réseaux, les tables de routage et les passerelles réseau. Pour plus d'informations VPCs, consultez le guide de l'[utilisateur Amazon VPC](#).

Pour connecter votre Amazon VPC à Amazon SWF, vous devez d'abord définir un point de terminaison VPC d'interface, qui vous permet de connecter votre VPC à un autre. Services AWS Le point de terminaison assure une connectivité évolutive et fiable, sans qu'une passerelle Internet, une instance NAT (Network Address Translation) ou une connexion VPN ne soit nécessaire. Pour de plus amples informations, consultez [Points de terminaison VPC \(AWS PrivateLink\)](#) dans le Guide de l'utilisateur Amazon VPC.

### Création du point de terminaison

Vous pouvez créer un point de terminaison Amazon SWF dans votre VPC à l'aide du AWS Management Console, the AWS Command Line Interface (AWS CLI), d'un AWS SDK, de l'API Amazon SWF ou. CloudFormation

Pour plus d'informations sur la création et la configuration d'un point de terminaison à l'aide de la console Amazon VPC ou de la AWS CLI, consultez la section [Création d'un point de terminaison d'interface](#) dans le Guide de l'utilisateur Amazon VPC.

### Note

Lorsque vous créez un point de terminaison, spécifiez Amazon SWF comme service auquel vous souhaitez que votre VPC se connecte. Dans la console Amazon VPC, les noms des

services varient en fonction de la AWS région. Par exemple, dans la région AWS Top Secret - Est, le nom du service pour Amazon SWF est `com.amazonaws.us-iso-east-1.swf`.

Pour plus d'informations sur la création et la configuration d'un point de terminaison à l'aide CloudFormation de la VPC Endpoint ressource [AWS EC2 ::::](#) dans le guide de CloudFormation l'utilisateur.

## Politiques relatives aux terminaux Amazon VPC

Pour contrôler l'accès à la connectivité à Amazon SWF, vous pouvez associer une politique de point de terminaison Gestion des identités et des accès AWS (IAM) lors de la création d'un point de terminaison Amazon VPC. Vous pouvez créer des règles IAM complexes en associant plusieurs politiques de point de terminaison. Pour plus d'informations, consultez :

- [Politiques relatives aux terminaux Amazon Virtual Private Cloud pour Amazon SWF](#)
- [Contrôle de l'accès aux services avec les points de terminaison d'un VPC](#)

## Politiques relatives aux terminaux Amazon Virtual Private Cloud pour Amazon SWF

Vous pouvez créer une politique de point de terminaison Amazon VPC pour Amazon SWF dans laquelle vous spécifiez les éléments suivants :

- Le principal qui peut effectuer des actions.
- Les actions qui peuvent être effectuées.
- La ressource sur laquelle les actions peuvent être effectuées.

L'exemple suivant ajoute un rôle IAM spécifique à une politique :

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

- Pour plus d'informations sur la création de politiques de point de terminaison, consultez la section [Contrôle de l'accès aux services avec des points de terminaison VPC](#).

- Pour plus d'informations sur la manière dont vous pouvez utiliser IAM pour contrôler l'accès à vos ressources AWS et à celles d'Amazon SWF, consultez. [Identity and Access Management dans Amazon Simple Workflow Service](#)

## Résolution des problèmes d'identité et d'accès à Amazon Simple Workflow Service

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec Amazon SWF et IAM.

### Rubriques

- [Je ne suis pas autorisé à effectuer une action dans Amazon SWF](#)
- [Je ne suis pas autorisé à effectuer iam : PassRole](#)
- [Je souhaite autoriser des personnes extérieures à moi Compte AWS à accéder à mes ressources Amazon SWF](#)

### Je ne suis pas autorisé à effectuer une action dans Amazon SWF

Si vous recevez une erreur selon laquelle vous n'êtes pas autorisé à effectuer une action, vos stratégies doivent être mises à jour afin de vous permettre d'effectuer l'action.

L'exemple d'erreur suivant se produit quand l'utilisateur mateojackson tente d'utiliser la console pour afficher des informations détaillées sur une ressource *my-example-widget* fictive, mais ne dispose pas des autorisations swf : *GetWidget* fictives.

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

Dans ce cas, la stratégie de Mateo doit être mise à jour pour l'autoriser à accéder à la ressource *my-example-widget* à l'aide de l'action swf : *GetWidget*.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

## Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez un message d'erreur indiquant que vous n'êtes pas autorisé à effectuer l'`iam:PassRole`action, vos politiques doivent être mises à jour pour vous permettre de transmettre un rôle à Amazon SWF.

Certains vos Services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, vous devez disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour effectuer une action dans Amazon SWF. Toutefois, l'action nécessite que le service ait des autorisations accordées par un rôle de service. Mary n'est pas autorisée à transmettre le rôle au service.

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

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

## Je souhaite autoriser des personnes extérieures à moi Compte AWS à accéder à mes ressources Amazon SWF

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACLs), vous pouvez utiliser ces politiques pour autoriser les utilisateurs à accéder à vos ressources.

Pour plus d'informations, consultez les éléments suivants :

- Pour savoir si Amazon SWF prend en charge ces fonctionnalités, consultez [Comment Amazon Simple Workflow Service fonctionne avec IAM](#)
- Pour savoir comment fournir l'accès à vos ressources sur celles Comptes AWS que vous possédez, consultez la section [Fournir l'accès à un utilisateur IAM dans un autre utilisateur Compte AWS que vous possédez](#) dans le Guide de l'utilisateur IAM.

- Pour savoir comment fournir l'accès à vos ressources à des tiers Comptes AWS, consultez la section [Fournir un accès à des ressources Comptes AWS détenues par des tiers](#) dans le guide de l'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d'identité, consultez [Fournir un accès à des utilisateurs authentifiés en externe \(fédération d'identité\)](#) dans le Guide de l'utilisateur IAM.
- Pour en savoir plus sur la différence entre l'utilisation des rôles et des politiques basées sur les ressources pour l'accès intercompte, consultez [Accès intercompte aux ressources dans IAM](#) dans le Guide de l'utilisateur IAM.

## Journalisation et surveillance

Cette section fournit des informations sur la journalisation et la surveillance d'Amazon SWF.

### Rubriques

- [Métriques Amazon SWF pour CloudWatch](#)
- [Affichage des métriques Amazon SWF pour CloudWatch l'utilisation du AWS Management Console](#)
- [Enregistrement des appels d'API avec AWS CloudTrail](#)
- [EventBridge pour les modifications du statut d'exécution d'Amazon SWF](#)
- [Utilisation Notifications des utilisateurs AWS avec Amazon Simple Workflow Service](#)

## Métriques Amazon SWF pour CloudWatch

Amazon SWF fournit désormais des métriques CloudWatch que vous pouvez utiliser pour suivre vos flux de travail et vos activités et définir des alarmes sur les valeurs de seuil que vous choisissez. Vous pouvez consulter les statistiques à l'aide du AWS Management Console. Pour de plus amples informations, veuillez consulter [Affichage des métriques Amazon SWF pour CloudWatch l'utilisation du AWS Management Console](#).

### Rubriques

- [Unités de reporting pour les métriques Amazon SWF](#)
- [Métriques des événements d'API et de décision](#)
- [Métriques Amazon SWF](#)

- [Noms et dimensions des ressources non ASCII Amazon SWF CloudWatch](#)

## Unités de reporting pour les métriques Amazon SWF

### Métriques qui indiquent un intervalle de temps

Certaines mesures Amazon SWF concernent les intervalles de temps, toujours mesurés en millisecondes. CloudWatch L' CloudWatch unité est signalée comme `Time`. Ces métriques correspondent généralement aux étapes d'exécution de flux de travail pour lesquelles vous pouvez définir des délais d'activité et de flux de travail, et utiliser des noms similaires.

Par exemple, la métrique `DecisionTaskStartToCloseTime` mesure le temps nécessaire pour que la tâche de décision soit menée à bien après le début de l'exécution, ce qui correspond à la période pour laquelle vous pouvez définir une valeur `DecisionTaskStartToCloseTimeout`.

Pour afficher le schéma de chacune de ces étapes et pour découvrir quand elles ont lieu dans le cycle de vie des flux de travail et des activités, consultez la section [Types de délai d'expiration Amazon SWF](#).

### Métriques qui indiquent un nombre

Certaines mesures Amazon SWF relatives aux résultats des CloudWatch rapports sont dénombrées. Par exemple, `WorkflowsCanceled` enregistre le résultat sous la forme du numéro un ou zéro pour indiquer si le flux de travail a été annulé ou non. Une valeur de zéro n'indique pas que cette métrique n'a pas été fournie, mais que la condition décrite par la métrique ne s'est pas produite.

Certaines des mesures Amazon SWF associées à CloudWatch ce rapport CloudWatch sont exprimées `Count` en nombre par seconde. Par exemple `ProvisionedRefillRate`, ce qui est indiqué sous forme CloudWatch d'`Count`entrée représente le taux `Count` de demandes par seconde.

Pour les métriques chiffrées, le minimum et le maximum seront toujours zéro ou un, avec une valeur allant de zéro à un comme moyenne.

### Métriques des événements d'API et de décision

Vous pouvez surveiller à la fois les événements liés à l'API et CloudWatch aux décisions pour obtenir un aperçu de votre utilisation et de vos capacités. Consultez les [décideurs](#) dans la [Concepts de flux de travail de base dans Amazon SWF](#) section et la rubrique [Décision](#) dans le manuel [Amazon Simple Workflow Service API Reference](#).

Vous pouvez également surveiller ces limites pour déclencher une alarme lorsque vous approchez de vos limites de régulation Amazon SWF. Consultez [Limitation des quotas Amazon SWF](#) pour obtenir une description de ces limites, ainsi que leurs paramètres par défaut. Ces limites sont conçues pour empêcher que des flux de travail incorrects utilisent des ressources système excessives. Pour demander une augmentation de vos limites, consultez [???](#).

Il est recommandé de configurer les CloudWatch alarmes à environ 60 % de la capacité de votre API ou de vos événements décisionnels. Cela vous permettra soit d'ajuster votre flux de travail, soit de demander une augmentation de la limite de service, avant d'activer la régulation Amazon SWF. En fonction des [variations](#) de vos appels, vous pouvez configurer différentes alarmes afin d'être averti lorsque vous approchez les limites du service :

- Si votre trafic présente des pics importants, définissez une alarme à 60 % de vos limites `ProvisionedBucketSize`.
- Si vos appels sont relativement réguliers, définissez une alarme à 60 % de votre limite `ProvisionedRefillRate` pour les événements d'API et de décision associés.

## Métriques Amazon SWF

Les mesures suivantes sont disponibles pour Amazon SWF :

Métrique	Description
<code>DecisionTaskScheduleToStartTime</code>	<p>Intervalle de temps, en millisecondes, entre le moment où la tâche de décision a été planifiée et le moment où elle a été traitée par un employé et démarrée.</p> <p>CloudWatch Unités : Time</p> <p>Dimensions : Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Statistiques valides : Average, Minimum, Maximum</p>
<code>DecisionTaskStartToCloseTime</code>	<p>Intervalle de temps, en millisecondes, entre le moment où la tâche de décision a été démarrée et le moment où elle a été clôturée.</p> <p>CloudWatch Unités : Time</p>

Métrique	Description
	<p>Dimensions : Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Statistiques valides : Average, Minimum, Maximum</p>
DecisionTasksCompleted	<p>Nombre de tâches de décision qui ont été effectuées.</p> <p>CloudWatch Unités : Count</p> <p>Dimensions : Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Statistiques valides : Sum</p>
PendingTasks	<p>Nombre de tâches en attente dans un intervalle de 1 minute pour une liste des tâches spécifique.</p> <p>CloudWatch Unités : Count</p> <p>Dimensions : Domain, TaskListName</p> <p>Statistiques valides : Sum</p>
StartedDecisionTasksTimedOutOnClose	<p>Nombre de tâches de décision qui ont été démarrées mais ayant expiré à la clôture.</p> <p>CloudWatch Unités : Count</p> <p>Dimensions : Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Statistiques valides : Sum</p>

Métrique	Description
<code>WorkflowStartToCloseTime</code>	<p>Durée, en millisecondes, entre le moment où le flux de travail a démarré et le moment où il a été clôturé.</p> <p>CloudWatch Unités : Time</p> <p>Dimensions : Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Statistiques valides : Average, Minimum, Maximum</p>
<code>WorkflowsCanceled</code>	<p>Nombre de flux de travail ayant été annulés.</p> <p>CloudWatch Unités : Count</p> <p>Dimensions : Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Statistiques valides : Sum</p>
<code>WorkflowsCompleted</code>	<p>Nombre de flux de travail terminés.</p> <p>CloudWatch Unités : Count</p> <p>Dimensions : Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Statistiques valides : Sum</p>
<code>WorkflowsContinuedAsNew</code>	<p>Nombre de flux de travail toujours considérés comme nouveaux.</p> <p>CloudWatch Unités : Count</p> <p>Dimensions : Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Statistiques valides : Sum</p>

Métrique	Description
<code>WorkflowsFailed</code>	<p>Nombre de flux de travail ayant échoué.</p> <p>CloudWatch Unités : Count</p> <p>Dimensions : Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Statistiques valides : Sum</p>
<code>WorkflowsTerminated</code>	<p>Nombre de flux de travail ayant été achevés.</p> <p>CloudWatch Unités : Count</p> <p>Dimensions : Cause, Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Statistiques valides : Sum</p>
<code>WorkflowsTimedOut</code>	<p>Nombre de flux de travail ayant expiré, pour une raison quelconque.</p> <p>CloudWatch Unités : Count</p> <p>Dimensions : Domain, WorkflowTypeName, WorkflowTypeVersion</p> <p>Statistiques valides : Sum</p>
<code>ActivityTaskScheduleToCloseTime</code>	<p>Intervalle de temps, en millisecondes, entre le moment où l'activité a été planifiée et le moment où elle est clôturée.</p> <p>CloudWatch Unités : Time</p> <p>Dimensions : Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Statistiques valides : Average, Minimum, Maximum</p>

Métrique	Description
ActivityTaskScheduleStartTime	<p>Intervalle de temps, en millisecondes, entre le moment où la tâche d'activité a été planifiée et le moment où elle a été démarrée.</p> <p>CloudWatch Unités : Time</p> <p>Dimensions : Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Statistiques valides : Average, Minimum, Maximum</p>
ActivityTaskStartToCloseTime	<p>Intervalle de temps, en millisecondes, entre le moment où la tâche d'activité a été démarrée et le moment où elle a été clôturée.</p> <p>CloudWatch Unités : Time</p> <p>Dimensions : Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Statistiques valides : Average, Minimum, Maximum</p>
ActivityTasksCancelled	<p>Nombre de tâches d'activité ayant été annulées.</p> <p>CloudWatch Unités : Count</p> <p>Dimensions : Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Statistiques valides : Sum</p>
ActivityTasksCompleted	<p>Nombre de tâches d'activité ayant été effectuées.</p> <p>CloudWatch Unités : Count</p> <p>Dimensions : Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Statistiques valides : Sum</p>

Métrique	Description
ActivityTasksFailed	<p>Nombre de tâches d'activité ayant échoué.</p> <p>CloudWatch Unités : Count</p> <p>Dimensions : Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Statistiques valides : Sum</p>
ScheduledActivityTasksTimedOutOnClose	<p>Nombre de tâches d'activité qui ont été planifiées mais ayant expiré lors de la clôture.</p> <p>CloudWatch Unités : Count</p> <p>Dimensions : Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Statistiques valides : Sum</p>
ScheduledActivityTasksTimedOutOnStart	<p>Nombre de tâches d'activité qui ont été planifiées mais ayant expiré lors du démarrage.</p> <p>CloudWatch Unités : Count</p> <p>Dimensions : Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Statistiques valides : Sum</p>
StartedActivityTasksTimedOutOnClose	<p>Nombre de tâches d'activité qui ont été démarrées mais ayant expiré lors de la clôture.</p> <p>CloudWatch Unités : Count</p> <p>Dimensions : Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Statistiques valides : Sum</p>

Métrique	Description
StartedActivityTasksTimedOutOnHeartbeat	<p>Nombre de tâches d'activité qui ont été démarrées mais ayant expiré en raison d'un délai d'attente des pulsations.</p> <p>CloudWatch Unités : Count</p> <p>Dimensions : Domain, ActivityTypeName, ActivityTypeVersion</p> <p>Statistiques valides : Sum</p>
ThrottledEvents	<p>Nombre de demandes ayant été limitées.</p> <p>CloudWatch Unités : Count</p> <p>Dimensions : APIName, DecisionName, ThrottlingScope</p> <p>Statistiques valides : Sum</p>
ProvisionedBucketSize	<p>Nombre de demandes disponibles par seconde.</p> <p>Dimensions : APIName, DecisionName</p> <p>Statistiques valides : Minimum</p>
ConsumedCapacity	<p>Nombre de demandes par seconde.</p> <p>CloudWatch Unités : Count</p> <p>Dimensions : APIName, DecisionName</p> <p>Statistiques valides : Sum</p>
ConsumedLimit	<p>Le montant de la limite générale qui a été consommé.</p> <p>Dimensions : GeneralLimitType</p>

Métrique	Description
ProvisionedRefillRate	<p>Nombre de demandes par seconde autorisées dans le compartiment.</p> <p>Dimensions : APIName, DecisionName</p> <p>Statistiques valides : Minimum</p>
ProvisionedLimit	<p>Le montant de la limite générale allouée au compte.</p> <p>Dimensions : GeneralLimitType</p>

Dimension	Description
Domain	Filtre les données vers le domaine Amazon SWF dans lequel le flux de travail ou l'activité s'exécute.
ActivityTypeName	Filtre les données sur le nom du type d'activité.
ActivityTypeVersion	Filtre les données sur la version du type d'activité.
WorkflowTypeName	Filtre les données sur le nom du type de flux de travail pour l'exécution de ce flux de travail.
WorkflowTypeVersion	Filtre les données sur la version du type de flux de travail pour l'exécution de ce flux de travail.
APIName	Filtre les données sur une API correspondant au nom d'API spécifié.
DecisionName	Filtre les données sur le nom de décision spécifié.
TaskListName	Filtre les données sur le nom Liste des tâches spécifié.
TaskListClassification	Filtre les données selon la classification de la liste des tâches. La valeur est « D » pour les listes de tâches décisionnelles et « A » pour les listes de tâches d'activité.

Dimension	Description
ThrottlingScope	Filtre les données selon l'étendue de régulation spécifiée. La valeur est « Compte » en cas de dépassement du quota au niveau du compte, ou « Workflow » en cas de dépassement du quota au niveau du flux de travail.

## Noms et dimensions des ressources non ASCII Amazon SWF CloudWatch

Amazon SWF autorise les caractères non ASCII dans les noms de ressources tels que `TaskList` et `DomainName`. Toutefois, les valeurs dimensionnelles des CloudWatch métriques ne peuvent contenir que des caractères ASCII imprimables. Pour garantir qu'Amazon SWF utilise des valeurs de dimension compatibles avec les [CloudWatch exigences](#), les noms de ressources Amazon SWF qui ne répondent pas à ces exigences sont convertis et une somme de contrôle sera ajoutée comme suit :

- Tout caractère non ASCII est remplacé par `?`.
- La chaîne d'entrée ou la chaîne convertie sera, si nécessaire, tronquée. Cela garantit que lorsque la somme de contrôle est ajoutée, la nouvelle longueur de chaîne ne dépassera pas le CloudWatch maximum.
- Étant donné que tous les caractères non ASCII sont convertis, certaines valeurs de dimension CloudWatch métrique qui étaient différentes avant la conversion peuvent sembler identiques après la conversion. Pour aider à les différencier, un trait de soulignement (`_`) suivi des 16 premiers caractères de la SHA256 somme de contrôle du nom de ressource d'origine est ajouté au nom de la ressource.

Exemples de conversion :

- `test àppleserait` converti en `test ?pp1e_82cc5b8e3a771d12`
- `àààserait` converti en `???_2fec5edbb2c05c22`.
- Les TaskList noms `àpplé` et les deux `âpplè` seraient convertis en `?pp1?`, et seraient identiques. L'ajout de la somme de contrôle renvoie des valeurs distinctes, `et?pp1?_f39a36df9d85a69d. ?pp1?_da3efb4f11dd0f7f`

**i** Tip

Vous pouvez générer votre propre SHA256 somme de contrôle. Par exemple, pour utiliser l'outil de ligne de shasum commande :

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

## Affichage des métriques Amazon SWF pour CloudWatch l'utilisation du AWS Management Console

Amazon CloudWatch fournit un certain nombre de statistiques consultables pour les flux de travail et les activités Amazon SWF. Vous pouvez consulter les métriques et définir des alarmes pour les exécutions de vos flux de travail Amazon SWF à l'aide du [AWS Management Console](#) Pour ce faire, vous devez être connecté à la console.

Pour obtenir une description de chacune des métriques disponibles, consultez la section [Métriques Amazon SWF pour CloudWatch](#).

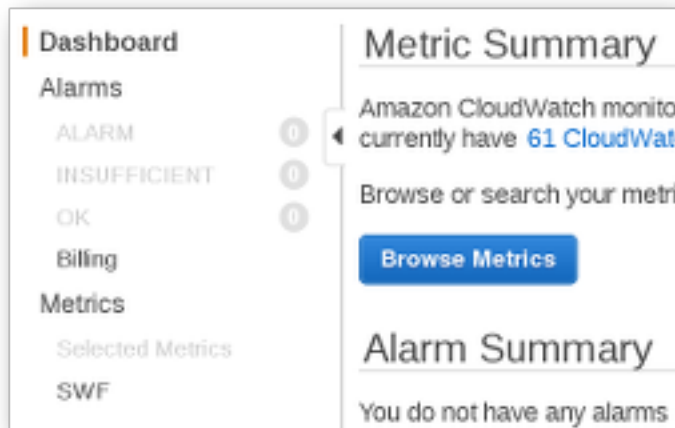
### Rubriques

- [Affichage des métriques](#)
- [Définition d'alarmes](#)

## Affichage des métriques

Pour consulter vos statistiques pour Amazon SWF

1. Connectez-vous à la CloudWatch console AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/cloudwatch/>.
2. Dans le volet de navigation, sous Metrics, sélectionnez SWF.



Si vous avez exécuté des exécutions de flux de travail récemment, deux listes de métriques sont présentées : les métriques sur les types de flux de travail et les métriques sur les types d'activité.

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

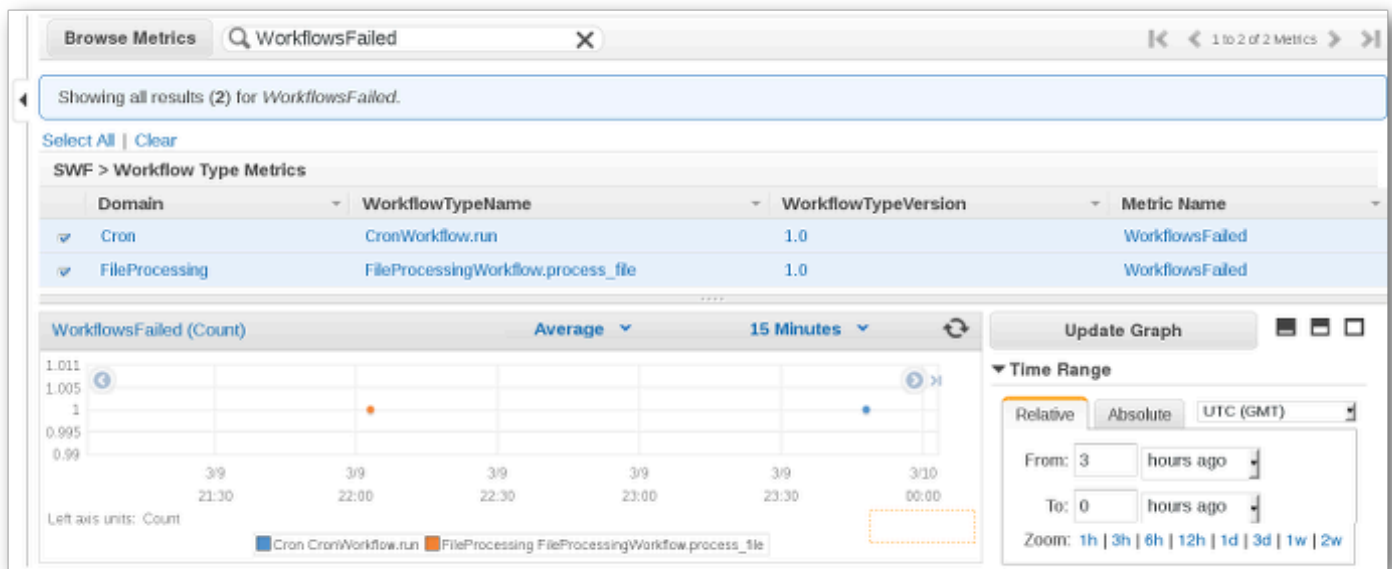
Initialement, vous ne verrez peut-être que les métriques sur les types de flux de travail. Les métriques sur les types d'activité se trouvent dans la même vue, mais vous devrez peut-être faire défiler la page vers le bas pour les voir.

Jusqu'à 50 des métriques les plus récentes sont présentées à la fois, réparties entre les types de flux de travail et les types d'activité.

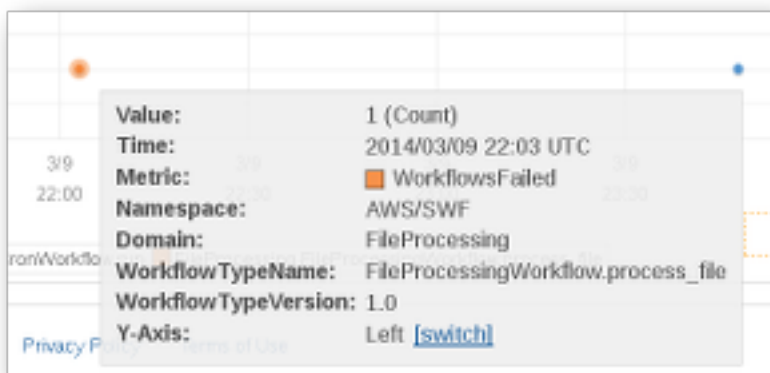
Vous pouvez utiliser les en-têtes interactifs au-dessus de chaque colonne de la liste pour trier les métriques à l'aide d'un des attributs fournis. Pour les flux de travail, les dimensions sont le domaine, `WorkflowTypeNameWorkflowTypeVersion`, et le nom de la métrique. Pour les activités, les dimensions sont le domaine, `ActivityTypeNameActivityTypeVersion`, et le nom de la métrique.

Les différents types de métriques sont décrits dans [Métriques Amazon SWF pour CloudWatch](#).

Pour afficher les graphiques des métriques, cochez les cases à correspondantes dans la liste et modifiez les paramètres de graphique à l'aide des contrôles Time Range situés à droite de la vue graphique.



Pour plus d'informations sur les différents points du graphique, placez le curseur dessus. Le détail des attributs du point s'affichera.



Pour plus d'informations sur l'utilisation des CloudWatch métriques, consultez [Visualisation, représentation graphique et publication des métriques](#) dans le guide de l' CloudWatch utilisateur Amazon.

## Définition d'alarmes

Vous pouvez utiliser les CloudWatch alarmes pour effectuer des actions telles que vous avertir lorsqu'un seuil d'alarme est atteint. Par exemple, vous pouvez définir une alarme afin d'envoyer une notification à une rubrique SNS ou d'envoyer un e-mail lorsque la métrique `WorkflowsFailed` dépasse un certain seuil.

Pour définir une alarme pour l'une de vos métriques

1. Choisissez une seule métrique en sélectionnant sa case.
2. A droite du graphique, dans les contrôles Tools, choisissez Create Alarm.
3. Sur l'écran Define Alarm, saisissez la valeur du seuil d'alarme, les paramètres de temps et les actions à effectuer.

**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

Pour plus d'informations sur le paramétrage et l'utilisation des CloudWatch alarmes, consultez [la section Création d' CloudWatch alarmes Amazon](#) dans le guide de CloudWatch l'utilisateur Amazon.

## Enregistrement des appels d'API avec AWS CloudTrail

Amazon Simple Workflow Service est intégré à [AWS CloudTrail](#) un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un Service AWS. CloudTrail capture tous les appels d'API pour Amazon SWF sous forme d'événements. Les appels capturés incluent des appels provenant de la console Amazon SWF et des appels de code vers les opérations de l'API Amazon SWF. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande envoyée à Amazon SWF, l'adresse IP à partir de laquelle la demande a été effectuée, la date à laquelle elle a été faite et des informations supplémentaires.

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer :

- Si la demande a été effectuée avec des informations d'identification d'utilisateur root ou d'utilisateur root.
- Si la demande a été faite au nom d'un utilisateur du centre d'identité IAM.
- Si la demande a été effectuée avec les informations d'identification de sécurité temporaires d'un rôle ou d'un utilisateur fédéré.
- Si la requête a été effectuée par un autre Service AWS.

CloudTrail est actif dans votre compte Compte AWS lorsque vous créez le compte et vous avez automatiquement accès à l'historique des CloudTrail événements. L'historique des CloudTrail événements fournit un enregistrement consultable, consultable, téléchargeable et immuable des 90 derniers jours des événements de gestion enregistrés dans un. Région AWS Pour plus d'informations, consultez la section [Utilisation de l'historique des CloudTrail événements](#) dans le guide de AWS CloudTrail l'utilisateur. La consultation de CloudTrail l'historique des événements est gratuite.

Pour un enregistrement continu des événements de vos 90 Compte AWS derniers jours, créez un magasin de données sur les événements de Trail ou [CloudTrailLake](#).

### CloudTrail sentiers

Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Tous les sentiers créés à l'aide du AWS Management Console sont multirégionaux. Vous ne pouvez

créer un journal de suivi en une ou plusieurs régions à l'aide de l' AWS CLI. Il est recommandé de créer un parcours multirégional, car vous capturez l'activité dans l'ensemble Régions AWS de votre compte. Si vous créez un journal de suivi pour une seule région, il convient de n'afficher que les événements enregistrés dans le journal de suivi pour une seule région Région AWS. Pour plus d'informations sur les journaux de suivi, consultez [Créez un journal de suivi dans vos Compte AWS](#) et [Création d'un journal de suivi pour une organisation](#) dans le AWS CloudTrail Guide de l'utilisateur.

Vous pouvez envoyer une copie de vos événements de gestion en cours dans votre compartiment Amazon S3 gratuitement CloudTrail en créant un journal. Toutefois, des frais de stockage Amazon S3 sont facturés. Pour plus d'informations sur la CloudTrail tarification, consultez la section [AWS CloudTrail Tarification](#). Pour obtenir des informations sur la tarification Amazon S3, consultez [Tarification Amazon S3](#).

## CloudTrail Stockages de données sur les événements du lac

CloudTrail Lake vous permet d'exécuter des requêtes SQL sur vos événements. CloudTrail Lake convertit les événements existants au format JSON basé sur les lignes au format [Apache ORC](#). ORC est un format de stockage en colonnes qui est optimisé pour une récupération rapide des données. Les événements sont agrégés dans des magasins de données d'événement. Ceux-ci constituent des collections immuables d'événements basées sur des critères que vous sélectionnez en appliquant des [sélecteurs d'événements avancés](#). Les sélecteurs que vous appliquez à un magasin de données d'événement contrôlent les événements qui persistent et que vous pouvez interroger. Pour plus d'informations sur CloudTrail Lake, consultez la section [Travailler avec AWS CloudTrail Lake](#) dans le guide de AWS CloudTrail l'utilisateur.

CloudTrail Les stockages et requêtes de données sur les événements de Lake entraînent des coûts. Lorsque vous créez un magasin de données d'événement, vous choisissez l'[option de tarification](#) que vous voulez utiliser pour le magasin de données d'événement. L'option de tarification détermine le coût d'ingestion et de stockage des événements, ainsi que les périodes de conservation par défaut et maximale pour le magasin de données d'événement. Pour plus d'informations sur la CloudTrail tarification, consultez la section [AWS CloudTrail Tarification](#).

## Événements liés aux données dans CloudTrail

Les [événements de données](#) fournissent des informations sur les opérations de ressources effectuées sur ou dans une ressource (par exemple, lecture ou écriture de données dans un objet Amazon S3). Ils sont également connus sous le nom opérations de plans de données. Les événements de données sont souvent des activités dont le volume est élevé. Par défaut, CloudTrail

n'enregistre pas les événements liés aux données. L'historique des CloudTrail événements n'enregistre pas les événements liés aux données.

Des frais supplémentaires s'appliquent pour les événements de données. Pour plus d'informations sur la CloudTrail tarification, consultez la section [AWS CloudTrail Tarification](#).

Vous pouvez enregistrer les événements de données pour les types de ressources Amazon SWF à l'aide de la CloudTrail console ou des AWS CLI opérations d' CloudTrail API. Pour plus d'informations sur la façon de journaliser les événements de données, consultez [Journalisation des événements de données avec la AWS Management Console](#) et [Journalisation des événements de données avec l' AWS Command Line Interface](#) dans le Guide de l'utilisateur AWS CloudTrail .

Le tableau suivant répertorie les types de ressources Amazon SWF pour lesquels vous pouvez enregistrer des événements de données. La colonne Type d'événement de données indique la valeur à choisir dans la liste des types d'événements de données de la CloudTrail console. La colonne de valeur `resources.type` indique la **resources.type** valeur que vous devez spécifier lors de la configuration de sélecteurs d'événements avancés à l'aide du ou. AWS CLI CloudTrail APIs La CloudTrail colonne Données APIs enregistrées indique les appels d'API enregistrés CloudTrail pour le type de ressource.

Vous pouvez configurer des sélecteurs d'événements avancés pour filtrer les champs `eventName`, `readOnly` et `resources.ARN` afin de ne journaliser que les événements importants pour vous. Pour plus d'informations sur ces champs, consultez [AdvancedFieldSelector](#) dans la Référence d'API AWS CloudTrail .

Type d'événement de données	valeur <code>resources.type</code>	Données APIs enregistrées sur CloudTrail
Domaine SWF	<code>AWS::SWF::Domain</code>	Événements du flux de travail <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> </ul>

Type d'événement de données	valeur resources.type	Données APIs enregistrées sur CloudTrail
		<ul style="list-style-type: none"> <li>• <a href="#">ListOpenWorkflowExecutions</a></li> <li>• <a href="#">GetWorkflowExecutionHistory</a></li> <li>• <a href="#">RequestCancelWorkflowExecution</a></li> <li>• <a href="#">SignalWorkflowExecution</a></li> <li>• <a href="#">StartWorkflowExecution</a></li> <li>• <a href="#">TerminateWorkflowExecution</a></li> </ul> <p>Événements liés aux tâches</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>Événements décisionnels</p> <ul style="list-style-type: none"> <li>• <a href="#">CancelTimer</a></li> <li>• <a href="#">CancelWorkflowExecution</a></li> </ul>

Type d'événement de données	valeur ressources.type	Données APIs enregistrées sur CloudTrail
		<ul style="list-style-type: none"> <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> <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 événements et RespondDecisionTaskCompleted

L'[RespondDecisionTaskCompleted](#) action prend une liste de décisions dans la charge utile de la demande. Un appel terminé émettra N+1 événements de CloudTrail données, un pour chaque décision et un pour l'appel d'API lui-même. Les événements de données et les événements d'API auront tous le même identifiant de demande.

## Événements de gestion dans CloudTrail

[Les événements de gestion](#) fournissent des informations sur les opérations de gestion effectuées sur les ressources de votre Compte AWS. Ils sont également connus sous le nom opérations de plan de contrôle. Par défaut, CloudTrail enregistre les événements de gestion.

Amazon Simple Workflow Service enregistre les opérations du plan de contrôle suivantes en CloudTrail tant qu'événements de gestion.

### Événements relatifs au domaine

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

### Activités et événements

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

### WorkflowType Événements

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

### Tag : événements

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

## Exemple d'évènement

Un événement représente une demande unique provenant de n'importe quelle source et inclut des informations sur l'opération d'API demandée, la date et l'heure de l'opération, les paramètres de la demande, etc. CloudTrail les fichiers journaux ne constituent pas une trace ordonnée des appels d'API publics. Les événements n'apparaissent donc pas dans un ordre spécifique.

L'exemple suivant montre un CloudTrail événement illustrant l'CountClosedWorkflowExecutions opération.

```
{
  "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-EXAMPLEebbbb",
  "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"
  }
}
```

Pour plus d'informations sur le contenu des CloudTrail enregistrements, voir [le contenu des CloudTrail enregistrements](#) dans le Guide de AWS CloudTrail l'utilisateur.

## EventBridge pour les modifications du statut d'exécution d'Amazon SWF

Vous utilisez Amazon EventBridge pour répondre à des changements d'état ou à des événements survenant dans une AWS ressource. Lorsqu'Amazon SWF émet un événement, celui-ci est toujours redirigé vers le bus d'EventBridge événements par défaut de votre compte. Vous pouvez créer une règle pour les événements, l'associer au bus d'événements par défaut et spécifier une action cible à effectuer lors de la EventBridge réception d'un événement correspondant à la règle. De cette façon, vous pouvez surveiller vos flux de travail sans avoir à effectuer des sondages en permanence à l'aide de l'[GetWorkflowExecutionHistory](#) API. En fonction des modifications apportées à l'exécution des flux de travail, vous pouvez utiliser une EventBridge cible pour appeler des AWS Lambda fonctions, publier des messages sur des rubriques Amazon Simple Notification Service (Amazon SNS), etc.

Vous pouvez consulter le contenu complet d'un événement de changement de statut d'exécution à l'aide de [DescribeWorkflowExecution](#).

Pour plus d'informations, consultez le [guide de EventBridge l'utilisateur Amazon](#).

## EventBridge événements

Les types d'événements historiques contiennent les modifications de l'état d'exécution. La `detail` section de chaque événement contient au moins les paramètres suivants :

- `eventId`: l'identifiant de l'événement affiché par `GetWorkflowExecutionHistory`.
- `workflowExecutionDetail`: l'état du flux de travail au moment de l'émission de l'événement.
- `eventType`: le type d'événement historique, l'un des suivants :
  - `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`

## Exemples d'événements Amazon SWF

Voici des exemples d'Amazon SWF envoyant des événements à : EventBridge

### Rubriques

- [Exécution commencée](#)
- [Exécution terminée](#)
- [Échec de l'exécution](#)
- [Expiration du délai d'exécution](#)
- [Exécution terminée](#)

Dans chaque cas, la section `detail` des données de l'événement fournit les mêmes informations que l'API [DescribeWorkflowExecution](#). Le `executionStatus` champ indique le statut de l'exécution au moment de l'envoi de l'événement, `OPEN` soit `CLOSED`.

### Exécution commencée

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

```
}
```

## Exécution terminée

```
{
  "version": "0",
  "id": "1111-2222-3333",
  "detail-type": "Simple Workflow Execution State Change",
  "source": "aws.swf",
  "account": "444455556666",
  "time": "2020-05-08T15:57:39Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:swf:us-east-1:444455556666:/domain/SimpleWorkflowUserSimulator"
  ],
  "detail": {
    "eventId": 35,
    "eventType": "WorkflowExecutionCompleted",
    "workflowExecutionDetail": {
      "executionInfo": {
        "execution": {
          "workflowId": "1234-5678-9012",
          "runId": "777788889999"
        },
        "workflowType": {
          "name": "SimpleWorkflowUserSimulator",
          "version": "myWorkflow"
        },
        "startTimestamp": 1588953458820,
        "closeTimestamp": 1588953459448,
        "executionStatus": "CLOSED",
        "closeStatus": "COMPLETED",
        "parent": null,
        "parentExecutionArn": null,
        "tagList": null,
        "cancelRequested": false
      },
      "executionConfiguration": {
        "taskStartToCloseTimeout": "60",
        "executionStartToCloseTimeout": "1000",
        "taskList": {
          "name": "1111-1111-1111"
        }
      }
    }
  }
}
```

```

    "taskPriority": null,
    "childPolicy": "ABANDON",
    "lambdaRole": "arn:aws:iam::444455556666:role/BasicSWFLambdaExecution"
  },
  "openCounts": {
    "openActivityTasks": 0,
    "openDecisionTasks": 0,
    "openTimers": 0,
    "openChildWorkflowExecutions": 0,
    "openLambdaFunctions": 0
  },
  "latestActivityTaskTimestamp": 1588953459402,
}
}
}

```

## Échec de l'exécution

```

{
  "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,
}
}
}

```

## Expiration du délai d'exécution

```

{
  "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,
  }
}
```

## Exécution terminée

```
{
  "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,
}
}
```

## Utilisation Notifications des utilisateurs AWS avec Amazon Simple Workflow Service

Vous pouvez l'utiliser [Notifications des utilisateurs AWS](#) pour configurer des canaux de diffusion afin d'être informé des événements d'Amazon Simple Workflow Service. Vous recevez une notification lorsqu'un événement correspond à une règle que vous avez spécifiée. Vous pouvez recevoir des notifications d'événements via plusieurs canaux, notamment par e-mail, [Amazon Q Developer dans les applications de chat](#), les notifications par chat ou les notifications [AWS Console Mobile Application](#) push. Vous pouvez également voir les notifications dans le [Centre de notifications de la console](#). Notifications des utilisateurs prend en charge l'agrégation, ce qui peut réduire le nombre de notifications que vous recevez lors d'événements spécifiques.

## Validation de conformité pour Amazon Simple Workflow Service

Des auditeurs tiers évaluent la sécurité et la conformité d'Amazon Simple Workflow Service dans le cadre de plusieurs programmes de AWS conformité. Il s'agit notamment des certifications SOC, PCI, FedRAMP, HIPAA et d'autres.

Pour une liste des AWS services concernés par des programmes de conformité spécifiques, voir [AWS Services concernés par programme de conformité AWS](#). Pour des informations générales, voir Programmes de [AWS conformité Programmes AWS](#) de .

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir [Téléchargement de rapports dans AWS Artifact](#).

Lorsque vous utilisez Amazon SWF, votre responsabilité en matière de conformité dépend de la sensibilité de vos données, des objectifs de conformité de votre entreprise et des lois et réglementations applicables. AWS fournit les ressources suivantes pour faciliter la mise en conformité :

- Guides [de démarrage rapide sur la sécurité et la conformité](#) [Guides](#) sur la sécurité et la conformité — Ces guides de déploiement abordent les considérations architecturales et fournissent des étapes pour déployer des environnements de base axés sur la sécurité et la conformité sur AWS.
- Livre blanc [sur l'architecture pour la sécurité et la conformité HIPAA](#) — [Ce livre blanc](#) décrit comment les entreprises peuvent créer des applications conformes à la loi HIPAA. AWS
- AWS Ressources de <https://aws.amazon.com/compliance/resources/> de conformité — Cette collection de classeurs et de guides peut s'appliquer à votre secteur d'activité et à votre région.
- [Évaluation des ressources à l'aide des règles](#) du guide du AWS Config développeur : le AWS Config service évalue dans quelle mesure les configurations de vos ressources sont conformes aux pratiques internes, aux directives du secteur et aux réglementations.
- [AWS Security Hub CSPM](#) — Ce AWS service fournit une vue complète de l'état de votre sécurité interne, AWS ce qui vous permet de vérifier votre conformité aux normes et aux meilleures pratiques du secteur de la sécurité.

## Résilience dans Amazon Simple Workflow Service

L'infrastructure AWS mondiale est construite autour des AWS régions et des zones de disponibilité. AWS Les régions fournissent plusieurs zones de disponibilité physiquement séparées et isolées, connectées par un réseau à faible latence, à haut débit et hautement redondant. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone à l'autre sans interruption. Les zones de disponibilité sont davantage disponibles, tolérantes aux pannes et ont une plus grande capacité de mise à l'échelle que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations sur AWS les régions et les zones de disponibilité, consultez la section [Infrastructure AWS mondiale](#).

Outre l'infrastructure AWS mondiale, Amazon SWF propose plusieurs fonctionnalités pour répondre à vos besoins en matière de résilience et de sauvegarde des données.

# Sécurité de l'infrastructure dans Amazon Simple Workflow Service

En tant que service géré, il est protégé par la sécurité du réseau AWS mondial. Pour plus d'informations sur les services AWS de sécurité et sur la manière dont AWS l'infrastructure est protégée, consultez la section [Sécurité du AWS cloud](#). Pour concevoir votre AWS environnement en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, consultez la section [Protection de l'infrastructure](#) dans le cadre AWS bien architecturé du pilier de sécurité.

Vous utilisez des appels d'API AWS publiés pour accéder via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Ses suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

Vous pouvez appeler ces opérations d'API depuis n'importe quel emplacement réseau, mais Amazon SWF prend en charge les politiques d'accès basées sur les ressources, qui peuvent inclure des restrictions basées sur l'adresse IP source. Vous pouvez également utiliser les politiques Amazon SWF pour contrôler l'accès depuis des points de terminaison Amazon Virtual Private Cloud (Amazon VPC) spécifiques ou spécifiques. VPCs En fait, cela isole l'accès réseau à une ressource Amazon SWF donnée uniquement du VPC spécifique au sein du réseau. AWS

## Analyse de configuration et de vulnérabilité dans Amazon Simple Workflow Service

La configuration et les contrôles informatiques sont une responsabilité partagée entre vous AWS et vous, notre client. Pour plus d'informations, consultez le [modèle de responsabilité AWS partagée](#).

# Utilisation du service AWS CLI with Amazon Simple Workflow

De nombreuses fonctionnalités d'Amazon Simple Workflow Service sont accessibles depuis le AWS CLI. AWS CLI II fournit une alternative à l'utilisation d'Amazon SWF avec AWS Management Console ou, dans certains cas, à la programmation avec l'API Amazon SWF et le AWS Flow Framework

Par exemple, vous pouvez utiliser le AWS CLI pour enregistrer un nouveau type de flux de travail :

```
aws swf register-workflow-type --domain MyDomain --name "MySimpleWorkflow" --workflow-version "v1"
```

Vous pouvez aussi afficher les types de flux de travail enregistrés :

```
aws swf list-workflow-types --domain MyDomain --registration-status REGISTERED
```

Vous trouverez ci-dessous un exemple de sortie par défaut au format 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"
      }
    }
  ]
}
```

Les commandes Amazon SWF incluses AWS CLI permettent de démarrer et de gérer les exécutions de flux de travail, d'interroger les tâches d'activité, d'enregistrer le rythme cardiaque des tâches, et bien plus encore ! Pour obtenir la liste complète des commandes Amazon SWF, avec des descriptions des arguments disponibles et des exemples illustrant leur utilisation, consultez la section Commandes [Amazon SWF](#) dans AWS CLI le manuel Command Reference.

Les AWS CLI commandes suivent de près l'API Amazon SWF. Vous pouvez donc les utiliser AWS CLI pour en savoir plus sur l'API Amazon SWF sous-jacente. Vous pouvez également utiliser vos connaissances actuelles en matière d'API pour prototyper du code ou exécuter des actions Amazon SWF sur la ligne de commande.

Pour en savoir plus AWS CLI, consultez le [guide de l'AWS Command Line Interface utilisateur](#).

# Utilisation d'Amazon SWF APIs

En plus d'utiliser AWS SDKs les éléments décrits dans [Développez avec AWS SDKs](#), vous pouvez utiliser directement l'API HTTP.

Pour utiliser l'API, vous envoyez des requêtes HTTP au [point de terminaison SWF](#) qui correspond à la région que vous souhaitez utiliser pour vos domaines, activités et flux de travail. Pour plus d'informations sur les requêtes HTTP pour Amazon SWF, consultez. [Envoi de demandes HTTP à Amazon SWF](#)

Cette section fournit des informations de base sur l'utilisation de l'API HTTP pour développer vos flux de travail avec Amazon SWF. Des fonctionnalités plus avancées, telles que l'utilisation de minuteries, la journalisation CloudTrail et le balisage de vos flux de travail, sont fournies dans la section. [Concepts de flux de travail de base dans Amazon SWF](#)

## Rubriques

- [Envoi de demandes HTTP à Amazon SWF](#)
- [Liste des actions Amazon SWF par catégorie](#)
- [Enregistrement d'un domaine auprès d'Amazon SWF](#)
- [Définition des valeurs de délai d'expiration dans Amazon SWF](#)
- [Enregistrement d'un type de flux de travail auprès d'Amazon SWF](#)
- [Enregistrement d'un type d'activité auprès d'Amazon SWF](#)
- [AWS Lambda tâches dans Amazon SWF](#)
- [Développement d'un Activity Worker dans Amazon SWF](#)
- [Développement de décideurs dans Amazon SWF](#)
- [Démarrage de flux de travail dans Amazon SWF](#)
- [Définition de la priorité des tâches dans Amazon SWF](#)
- [Gestion des erreurs dans Amazon SWF](#)

## Envoi de demandes HTTP à Amazon SWF

Si vous n'utilisez pas l'un d'entre eux AWS SDKs, vous pouvez effectuer des opérations Amazon Simple Workflow Service (Amazon SWF) via HTTP à l'aide de la méthode de requête POST. La

méthode POST nécessite que vous définissiez l'opération dans l'en-tête de la requête et que vous fournissiez les données de l'opération au format JSON dans le corps de la requête.

## Contenu de l'en-tête HTTP

Amazon SWF requiert les informations suivantes dans l'en-tête d'une requête HTTP :

- `host` Le point de terminaison Amazon SWF.
- `x-amz-date` Vous devez fournir l'horodatage dans l'`Date` en-tête HTTP ou dans le AWS `x-amz-date` header (certaines bibliothèques clientes HTTP ne vous permettent pas de définir l'`Date` en-tête). Lorsqu'un en-tête `x-amz-date` est présent, le système ignore tout en-tête `Date` lors de l'authentification de la demande.

La date doit être spécifiée dans l'un des formats suivants, comme indiqué dans le RFC HTTP/1.1 :

- Sun, 06 Nov 1994 08:49:37 GMT (RFC 822, mis à jour par RFC 1123)
- Sunday, 06-Nov-94 08:49:37 GMT (RFC 850, rendu obsolète par RFC 1036)
- Dim 6 nov 08:49:37 1994 (format `asctime()` ANSI C)
- `x-amzn-authorization` Les paramètres de requête signés au format :

```
AWS3 AWSAccessKeyId=####,Algorithm=HmacSHA256, [,SignedHeaders=Header1;Header2;...]
Signature=S(StringToSign)
```

**AWS3**— Il s'agit d'une balise AWS spécifique à l'implémentation qui indique la version d'authentification utilisée pour signer la demande (actuellement, cette valeur est toujours valable pour Amazon SWF). **AWS3**

**AWSAccessKeyId**— L'identifiant de votre clé d' AWS accès.

**Algorithm**— L'algorithme utilisé pour créer la valeur HMAC-SHA de string-to-sign, telle que `ou`.  
`HmacSHA256` `HmacSHA1`

**Signature**— Base64 (`Algorithme ( StringToSign, SigningKey )`). Pour en savoir plus, consultez la section [Calcul de la signature HMAC-SHA pour Amazon SWF](#).

**SignedHeaders**— (Facultatif) Le cas échéant, doit contenir une liste de tous les en-têtes HTTP utilisés dans le calcul canonisé `HttpHeaders` . Vous devez utiliser un point virgule (;) (caractère ASCII 59) pour délimiter les valeurs de la liste.

- `x-amz-target`— Le service de destination de la demande et le fonctionnement des données, au format

```
com.amazonaws.swf.service.model.SimpleWorkflowService. + <action>
```

Par exemple,

```
com.amazonaws.swf.service.model.SimpleWorkflowService.RegisterDomain
```

- `content-type`— Le type doit spécifier le JSON et le jeu de caractères, comme `application/json; charset=UTF-8`

Voici un exemple d'en-tête de requête HTTP utilisé pour créer un domaine.

```
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"}
```

Voici un exemple de la réponse HTTP correspondante.

```
HTTP/1.1 200 OK
Content-Length: 0
Content-Type: application/json
x-amzn-RequestId: 4ec4ac3f-3e16-11e1-9b11-7182192d0b57
```

## Contenu du corps HTTP

Le corps d'une requête HTTP contient les données de l'opération spécifiée dans l'en-tête de la requête HTTP. Utilisez le format de données JSON pour transmettre simultanément les valeurs de données et la structure de données. Pour imbriquer des éléments dans d'autres, utilisez la notation d'accolade. Par exemple, ce qui suit montre une demande visant à répertorier toutes les exécutions de flux de travail qui ont débuté entre deux moments spécifiés, à l'aide de la notation horaire Unix.

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

## Exemple de demande et de réponse Amazon SWF JSON

L'exemple suivant montre une demande adressée à Amazon SWF pour obtenir une description du domaine que nous avons créé précédemment. Il affiche ensuite la réponse Amazon SWF.

### Demande HTTP POST

```
POST http://swf.us-east-1.amazonaws.com/ HTTP/1.1
Host: swf.us-east-1.amazonaws.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.2.25) Gecko/20111212
  Firefox/3.6.25 ( .NET CLR 3.5.30729; .NET4.0E)
Accept: application/json, text/javascript, */*
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
```

```
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
Content-Type: application/json; charset=UTF-8
X-Requested-With: XMLHttpRequest
X-Amz-Date: Sun, 15 Jan 2012 03:13:33 GMT
X-Amz-Target: com.amazonaws.swf.service.model.SimpleWorkflowService.DescribeDomain
Content-Encoding: amz-1.0
X-Amzn-Authorization: AWS3
  AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE,Algorithm=HmacSHA256,SignedHeaders=Host;X-Amz-
Date;X-Amz-Target;Content-
Encoding,Signature=IFJtq3M366CHqMLTpyqYqd9z0ChCoKDC5SCJBsLifu4=
Referer: http://swf.us-east-1.amazonaws.com/explorer/index.html
Content-Length: 21
Pragma: no-cache
Cache-Control: no-cache

{"name": "867530901"}
```

## Réponse d'Amazon SWF

```
HTTP/1.1 200 OK
Content-Length: 137
Content-Type: application/json
x-amzn-RequestId: e86a6779-3f26-11e1-9a27-0760db01a4a8

{"configuration":
  {"workflowExecutionRetentionPeriodInDays": "60"},
 "domainInfo":
  {"description": "music",
   "name": "867530901",
   "status": "REGISTERED"}
}
```

Notez que le protocole (HTTP/1.1) est suivi d'un code d'état (200). La valeur 200 indique une opération réussie.

Amazon SWF ne sérialise pas les valeurs nulles. Si votre analyseur JSON est configuré pour sérialiser les valeurs nulles pour les requêtes, Amazon SWF les ignore.

## Calcul de la signature HMAC-SHA pour Amazon SWF

Chaque demande adressée à Amazon SWF doit être authentifiée. Ils signent AWS SDKs automatiquement vos demandes et gèrent votre authentification basée sur des jetons. Toutefois, si vous souhaitez écrire vos propres demandes HTTP POST, vous devez créer une valeur `x-amzn-authorization` pour le contenu de l'en-tête HTTP POST Header dans le cadre de l'authentification de votre demande.

Pour en savoir plus sur la mise en forme de ces en-têtes, consultez la section [Contenu de l'en-tête HTTP](#). Pour l' AWS SDK pour Java implémentation de la signature de AWS la version 3, consultez la classe [AWSSigner.java](#).

### Création d'une signature de demande

Avant de créer une signature de demande HMAC-SHA, vous devez obtenir vos informations d'identification AWS (ID de clé d'accès et clé secrète).

#### Important

Vous pouvez utiliser l'un SHA1 ou SHA256 l'autre ou pour signer vos demandes. Cependant, veuillez à utiliser la même méthode pour tout le processus de signature. La méthode que vous choisissez doit correspondre à la valeur du nom `Algorithm` de l'en-tête HTTP.

Pour créer la signature de demande

1. Créez une forme canonique des en-têtes de requête HTTP. La forme canonique de l'en-tête HTTP comprend les éléments suivants :
  - `host`
  - N'importe quel élément d'en-tête commençant par `x-amz-`

Pour en savoir plus sur les en-têtes inclus, consultez la section [Contenu de l'en-tête HTTP](#).

- a. Pour chaque paire nom-valeur d'en-tête, convertissez le nom d'en-tête (pas la valeur d'en-tête) en minuscules.
- b. Combinez les champs d'en-tête avec le même nom dans une paire « nom d'en-tête:liste de valeurs séparées par des virgules ».

```
x-amz-example: value1
x-amz-example: value2 => x-amz-example:value1,value2
```

Pour en savoir plus, consultez la [section 4.2 sur la page RFC 2616](#).

- c. Convertissez chaque paire nom-valeur d'en-tête en chaîne au format `headerName:headerValue`. Supprimez les espaces de début et de fin des valeurs `headerName` et `headerValue`, et n'utilisez aucun espace de chaque côté des deux points.

```
x-amz-example1:value1,value2
x-amz-example2:value3
```

- d. Insérez une nouvelle ligne (U+000A) après chaque chaîne convertie, y compris la dernière chaîne.
  - e. Triez la collection de chaînes converties dans l'ordre alphabétique par nom d'en-tête.
2. Créez une string-to-sign valeur qui inclut les éléments suivants :
    - Ligne 1 : la méthode HTTP (POST), suivie d'une nouvelle ligne.
    - Ligne 2 : l'URI de la demande (/), suivi d'une nouvelle ligne.
    - Ligne 3 : une chaîne vide suivie d'une nouvelle ligne.

#### Note

Généralement, la chaîne de requête apparaît ici, mais Amazon SWF n'utilise pas de chaîne de requête.

- Ligne 4–n : la chaîne représentant les en-têtes de demandes canoniques calculés à l'étape 1, suivie d'une nouvelle ligne. Cette nouvelle ligne crée une ligne vide entre les en-têtes et le corps de la demande HTTP. Pour plus d'informations, consultez [RFC 2616](#).
  - Le corps de la demande, non suivie d'une nouvelle ligne.
3. Calculez SHA256 SHA1 ou résumez la string-to-sign valeur. Utilisez la même méthode SHA tout au long du processus.
  4. Calculez et encodez le HMAC-SHA en utilisant un résumé SHA256 ou un SHA1 résumé (selon la méthode que vous avez utilisée) de la valeur résultant de l'étape précédente et de la clé d'accès secrète temporaire du AWS Security Token Service à l'aide de l'action API.

[GetSessionToken](#)

**Note**

Amazon SWF attend un signe égal (=) à la fin de la valeur HMAC-SHA codée en Base64. Si votre routine d'encodage Base64 n'inclut pas l'ajout de signes égal, insérez-en un à la fin de la valeur.

Pour plus d'informations sur l'utilisation d'informations d'identification de sécurité temporaires avec Amazon SWF et d'autres AWS services, consultez AWS la section Services That [Work with IAM dans le guide de l'utilisateur IAM](#).

5. Placez la valeur résultante comme valeur du Signature nom dans l'`x-amzn-authorization` en-tête de la requête HTTP envoyée à Amazon SWF.
6. Amazon SWF vérifie la demande et exécute l'opération spécifiée.

## Liste des actions Amazon SWF par catégorie

Cette section répertorie les rubriques de référence relatives aux actions Amazon SWF dans l'interface de programmation d'applications (API) Amazon SWF. Elles sont classées par catégorie fonctionnelle.

Pour une liste alphabétique des actions, consultez le manuel [Amazon Simple Workflow Service API Reference](#).

### Rubriques

- [Actions liées aux activités](#)
- [Actions liées aux décideurs](#)
- [Actions liées aux exécutions de flux de travail](#)
- [Actions liées à l'administration](#)
- [Actions de visibilité](#)

## Actions liées aux activités

Les travailleurs d'activité utilisent la commande `PollForActivityTask` pour obtenir les nouvelles tâches d'activité. Une fois qu'un travailleur reçoit une tâche d'activité d'Amazon SWF,

il exécute la tâche et répond en `RespondActivityTaskCompleted` cas de réussite ou `RespondActivityTaskFailed` d'échec.

Voici les actions exécutées par les outils de traitement d'activité.

- [PollForActivityTask](#)
- [RespondActivityTaskCompleted](#)
- [RespondActivityTaskFailed](#)
- [RespondActivityTaskCanceled](#)
- [RecordActivityTaskHeartbeat](#)

## Actions liées aux décideurs

Les décideurs utilisent la commande `PollForDecisionTask` pour obtenir les tâches de décision. Une fois qu'un décideur reçoit une tâche de décision d'Amazon SWF, il examine l'historique d'exécution de son flux de travail et décide de la marche à suivre. Il appelle `RespondDecisionTaskCompleted` pour terminer la tâche de décision et fournit zéro ou plusieurs décisions suivantes.

Voici les actions exécutées par les décideurs.

- [PollForDecisionTask](#)
- [RespondDecisionTaskCompleted](#)

## Actions liées aux exécutions de flux de travail

Les actions suivantes font partie d'une exécution de flux de travail.

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

## Actions liées à l'administration

Bien que vous puissiez effectuer des tâches administratives depuis la console Amazon SWF, vous pouvez utiliser les actions décrites dans cette section pour automatiser des fonctions ou créer vos propres outils d'administration.

### Gestion des activités

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

### Gestion du flux de travail

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

### Gestion de domaine

Ces actions vous permettent d'enregistrer et de déconseiller des domaines Amazon SWF.

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

Pour obtenir plus d'informations, ainsi que des exemples de ces actions de gestion de domaine, consultez la section [Enregistrement d'un domaine auprès d'Amazon SWF](#).

### Gestion de l'exécution du flux de travail

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

## Actions de visibilité

Bien que vous puissiez effectuer des actions de visibilité depuis la console Amazon SWF, vous pouvez utiliser les actions décrites dans cette section pour créer votre propre console ou vos propres outils d'administration.

### Visibilité de l'activité

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

### Visibilité du flux de travail

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

### Visibilité de l'exécution du flux de travail

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

### Visibilité du domaine

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

### Visibilité de la liste de tâches

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

## Enregistrement d'un domaine auprès d'Amazon SWF

Vos types de flux de travail et d'activités, ainsi que l'exécution elle-même du flux de travail sont tous limités à un domaine. Les domaines séparent un ensemble de types, d'exécutions et de listes de tâches des autres dans le même compte.

Vous pouvez enregistrer un domaine en utilisant AWS Management Console ou en utilisant l'`RegisterDomain` dans l'API Amazon SWF. L'exemple suivant utilise l'API.

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

Les paramètres sont spécifiés au format JSON ( JavaScript Object Notation). Ici, la période de conservation est définie sur 60 jours. Pendant la période de conservation, toutes les informations relatives à l'exécution du flux de travail sont disponibles via des opérations de visibilité à l'aide de l'API Amazon SWF AWS Management Console ou de l'API Amazon SWF.

Après l'enregistrement du domaine, vous devez enregistrer le type de flux de travail et les types d'activité utilisés par le flux de travail. Vous devez tout d'abord enregistrer le domaine, car l'enregistrement d'un nom de domaine est nécessaire à l'enregistrement des types de flux de travail et d'activités.

consultez aussi

[RegisterDomain](#) dans la référence d'API Amazon Simple Workflow Service

## Définition des valeurs de délai d'expiration dans Amazon SWF

Rubriques

- [Quotas sur les valeurs de délai d'expiration](#)
- [Délais des exécutions de flux de travail et des tâches de décision](#)
- [Délais des tâche d'activité](#)
- [consultez aussi](#)

## Quotas sur les valeurs de délai d'expiration

Les valeurs de délai d'expiration sont toujours déclarées en secondes et peuvent être définies sur un nombre quelconque de secondes jusqu'à un an (31536 000 secondes), soit la limite d'exécution maximale pour tout flux de travail ou activité. La valeur spéciale NONE permet de définir un paramètre de délai « sans délai », ou infini, mais la durée maximale d'une année continue à s'appliquer.

## Délais des exécutions de flux de travail et des tâches de décision

Vous pouvez définir des valeurs de délai pour le flux de travail et les tâches de décision lorsque vous enregistrez le type de flux de travail. Par exemple :

```
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"
}
```

Cet enregistrement définit la valeur [defaultTaskStartToCloseTimeout](#) sur 600 secondes (10 minutes), et [defaultExecutionStartToCloseTimeout](#) sur 3 600 secondes (1 heure).

Pour plus d'informations sur l'enregistrement du type de flux de travail [Enregistrement d'un type de flux de travail auprès d'Amazon SWF](#), consultez et [RegisterWorkflowType](#) dans le manuel Amazon Simple Workflow Service API Reference.

Pour remplacer la valeur définie pour `defaultExecutionStartToCloseTimeout`, spécifiez [executionStartToCloseTimeout](#) .

## Délais des tâche d'activité

Vous pouvez définir des valeurs de délai pour les tâches d'activité lors de l'enregistrement du type d'activité. Par exemple :

```
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"
}
```

Cet enregistrement de type d'activité définit la valeur [defaultTaskStartToCloseTimeout](#) sur 600 secondes (10 minutes), [defaultTaskHeartbeatTimeout](#) sur 120 secondes (2 minutes), [defaultTaskScheduleToStartTimeout](#) sur 1 800 secondes (30 minutes) et [defaultTaskScheduleToCloseTimeout](#) sur 5 400 secondes (1,5 heure).

Pour plus d'informations sur l'enregistrement du type d'activité [Enregistrement d'un type d'activité auprès d'Amazon SWF](#), consultez et [RegisterActivityType](#) dans le manuel Amazon Simple Workflow Service API Reference.

Pour remplacer la valeur définie pour `defaultTaskStartToCloseTimeout` spécifiez [taskStartToCloseTimeout](#) lors de la planification de la tâche d'activité.

consultez aussi

[Types de délai d'expiration Amazon SWF](#)

## Enregistrement d'un type de flux de travail auprès d'Amazon SWF

L'exemple présenté dans cette section enregistre un type de flux de travail à l'aide de l'API Amazon SWF. Le nom et la version que vous spécifiez lors de l'enregistrement forment un identifiant unique pour le type de flux de travail. Le domaine défini doit être enregistré au préalable à l'aide de l'action d'API [RegisterDomain](#).

Les paramètres de délai d'attente de l'exemple suivant sont des valeurs de durée définies en secondes. Pour le paramètre `defaultTaskStartToCloseTimeout`, vous pouvez utiliser le spécificateur de durée `NONE` pour n'indiquer aucun délai d'attente. Cependant, vous ne pouvez pas spécifier la valeur `NONE` pour `defaultExecutionStartToCloseTimeout`. La durée

d'exécution d'un flux de travail est limitée à un an. Passé ce délai, cette limite entraîne toujours l'expiration de l'exécution du flux de travail. Si vous spécifiez une valeur supérieure à un an pour `defaultExecutionStartToCloseTimeout`, l'enregistrement échoue.

```
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"
}
```

consultez aussi

[RegisterWorkflowType](#) dans la référence d'API Amazon Simple Workflow Service

## Enregistrement d'un type d'activité auprès d'Amazon SWF

L'exemple suivant enregistre un type d'activité à l'aide de l'API Amazon SWF. Le nom et la version que vous spécifiez lors de l'enregistrement forment un identifiant unique pour le type d'activité au sein du domaine. Le domaine défini doit être enregistré au préalable à l'aide de l'action `RegisterDomain`.

Les paramètres de délai de cet exemple sont des durées spécifiées en secondes. Vous pouvez utiliser le spécificateur de durée `NONE` pour indiquer qu'aucun délai ne s'applique.

```
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"
}
```

consultez aussi

[RegisterActivityType](#) dans la référence d'API Amazon Simple Workflow Service

## AWS Lambda tâches dans Amazon SWF

Rubriques

- [À propos AWS Lambda](#)
- [Avantages et limites de l'utilisation des tâches Lambda](#)
- [Utilisation de tâches Lambda dans vos flux de travail](#)

### À propos AWS Lambda

AWS Lambda est un service de calcul entièrement géré qui exécute votre code en réponse à des événements générés par du code personnalisé ou par divers AWS services tels qu'Amazon S3, DynamoDB, Amazon Kinesis, Amazon SNS et Amazon Cognito. Pour plus d'informations sur Lambda, consultez le [guide du développeur AWS Lambda](#).

Amazon Simple Workflow Service fournit une tâche Lambda qui vous permet d'exécuter des fonctions Lambda à la place ou en parallèle des activités Amazon SWF traditionnelles.

#### Important

Votre AWS compte sera débité pour les exécutions Lambda (requêtes) exécutées par Amazon SWF en votre nom. [Pour plus de détails sur la tarification Lambda, consultez la section tarification/https://aws.amazon.com/lambda/](https://aws.amazon.com/lambda/).

### Avantages et limites de l'utilisation des tâches Lambda

L'utilisation de tâches Lambda au lieu d'une activité Amazon SWF traditionnelle présente de nombreux avantages :

- Les tâches Lambda n'ont pas besoin d'être enregistrées ou versionnées comme les types d'activité Amazon SWF.
- Vous pouvez utiliser toutes les fonctions Lambda existantes que vous avez déjà définies dans vos flux de travail.
- Les fonctions Lambda sont appelées directement par Amazon SWF ; il n'est pas nécessaire d'implémenter un programme de travail pour les exécuter, comme c'est le cas pour les activités traditionnelles.
- Lambda vous fournit des métriques et des journaux pour suivre et analyser les exécutions de vos fonctions.

Vous devez également connaître les quelques limites qui s'appliquent aux tâches Lambda :

- Les tâches Lambda ne peuvent être exécutées que dans AWS les régions qui prennent en charge Lambda. Consultez la section [Régions et points de terminaison Lambda](#) dans le manuel Amazon Web Services General Reference pour en savoir plus sur les régions actuellement prises en charge pour Lambda.
- Les tâches Lambda ne sont actuellement prises en charge que par l'API HTTP SWF de base et par l'API for Java. AWS Flow Framework Les tâches Lambda ne sont actuellement pas prises en charge dans AWS Flow Framework for Ruby.

## Utilisation de tâches Lambda dans vos flux de travail

Pour utiliser les tâches Lambda dans vos flux de travail Amazon SWF, vous devez :

1. Configurez des rôles IAM pour autoriser Amazon SWF à invoquer des fonctions Lambda.
2. Attachez les rôles IAM à vos flux de travail.
3. Appelez la fonction Lambda pendant une exécution de flux de travail.

### Configuration d'un rôle IAM

Avant de pouvoir invoquer des fonctions Lambda depuis Amazon SWF, vous devez fournir un rôle IAM qui permet d'accéder à Lambda depuis Amazon SWF. Vous avez le choix entre les options suivantes :

- choisissez un rôle prédéfini, Role, AWSLambda pour autoriser vos flux de travail à invoquer n'importe quelle fonction Lambda associée à votre compte.

- définissez votre propre politique et le rôle associé pour autoriser les flux de travail à invoquer des fonctions Lambda spécifiques, spécifiées par leur Amazon Resource Names (ARNs).

### Limiter les autorisations sur un rôle IAM

Vous pouvez limiter les autorisations sur un rôle IAM que vous fournissez à Amazon SWF en utilisant les clés de contexte `SourceAccount` et `SourceArn` de votre politique de confiance en matière de ressources. Ces clés limitent l'utilisation d'une politique IAM afin qu'elle ne soit utilisée qu'à partir des exécutions d'Amazon Simple Workflow Service appartenant à l'ARN du domaine spécifié. Si vous utilisez les deux clés contextuelles de condition globale, la `aws:SourceAccount` valeur et le compte référencés dans la `aws:SourceArn` valeur doivent utiliser le même identifiant de compte lorsqu'ils sont utilisés dans la même déclaration de politique.

Dans l'exemple suivant, la clé de `SourceArn` contexte restreint l'utilisation du rôle de service IAM uniquement dans les exécutions d'Amazon Simple Workflow Service appartenant `someDomain` au compte, `123456789012`

- Déclaration 1

```
Principal : "Service": "swf.amazonaws.com"
```

```
Action : sts:AssumeRole
```

```
"Condition": {
  "ArnLike": {
    "aws:SourceArn": "arn:aws:swf:*:123456789012:/domain/someDomain"
  }
}
```

Dans l'exemple suivant, la clé de `SourceAccount` contexte restreint l'utilisation du rôle de service IAM uniquement dans le cadre des exécutions d'Amazon Simple Workflow Service dans le compte, `123456789012`

```
"Condition": {
  "StringLike": {
    "aws:SourceAccount": "123456789012"
  }
}
```

Fournir à Amazon SWF l'accès lui permettant d'invoquer n'importe quel rôle Lambda

Vous pouvez utiliser le rôle prédéfini, Role, AWSLambda pour permettre à vos flux de travail Amazon SWF d'invoquer n'importe quelle fonction Lambda associée à votre compte.

Pour utiliser AWSLambda Role pour autoriser Amazon SWF à invoquer des fonctions Lambda

1. Ouvrez la [console Amazon IAM](#).
2. Choisissez Rôles, puis Créer un rôle.
3. Attribuez un nom à votre rôle, tel que `swf-lambda`, puis choisissez Étape suivante.
4. Sous Rôles de AWS service, sélectionnez Amazon SWF, puis Next Step.
5. Sur l'écran Attach Policy, choisissez Role AWSLambda dans la liste.
6. Choisissez Étape suivante, puis Créer un rôle une fois que vous avez vérifié le rôle.

Définition d'un rôle IAM pour fournir un accès permettant d'invoquer une fonction Lambda spécifique

Si vous souhaitez autoriser l'appel d'une fonction Lambda spécifique à partir de votre flux de travail, vous devez définir votre propre politique IAM.

Pour créer une politique IAM afin de fournir l'accès à une fonction Lambda particulière

1. Ouvrez la [console Amazon IAM](#).
2. Choisissez Stratégies, puis Créer une stratégie.
3. Choisissez Copier une politique AWS gérée et sélectionnez AWSLambdaRôle dans la liste. Une stratégie sera générée pour vous. Au besoin, modifiez son nom et sa description.
4. Dans le champ Ressource du document de politique, ajoutez l'ARN de vos fonctions Lambda.

Par exemple :

- Ressource : `arn:aws:lambda:us-east-1:111122223333:function:hello_lambda_function`

#### Note

Pour une description complète de la manière de spécifier les ressources dans un rôle IAM, voir [Présentation des politiques IAM](#) dans Using IAM.

5. Choisissez Créer une stratégie afin de finaliser la création de la stratégie.

Vous pouvez ensuite sélectionner cette politique lors de la création d'un nouveau rôle IAM et utiliser ce rôle pour donner un accès d'appel à vos flux de travail Amazon SWF. Cette procédure est très similaire à la création d'un rôle avec la politique des AWSLambda rôles. Choisissez plutôt votre propre politique lors de la création du rôle.

Pour créer un rôle Amazon SWF à l'aide de votre politique Lambda

1. Ouvrez la [console Amazon IAM](#).
2. Choisissez Rôles, puis Créer un rôle.
3. Attribuez un nom à votre rôle, tel que `swf-lambda-fonction`, puis choisissez Étape suivante.
4. Sous Rôles de AWS service, sélectionnez Amazon SWF, puis Next Step.
5. Sur l'écran Attach Policy, choisissez votre politique spécifique à la fonction Lambda dans la liste.
6. Choisissez Étape suivante, puis Créer un rôle une fois que vous avez vérifié le rôle.

## Attacher le rôle IAM à votre flux de travail

Une fois que vous avez défini votre rôle IAM, vous devez l'associer au flux de travail qui l'utilisera pour appeler les fonctions Lambda auxquelles vous avez donné accès à Amazon SWF.

Vous pouvez attacher le rôle au flux de travail à deux niveaux :

- Lors de l'enregistrement de type de flux de travail. Ce rôle pourra ensuite être utilisé comme rôle Lambda par défaut pour chaque exécution de ce type de flux de travail.
- Lors du lancement d'une exécution de flux de travail. Ce rôle sera utilisé uniquement au cours de l'exécution de ce flux de travail (et tout au long de l'exécution).

Pour fournir un rôle Lambda par défaut pour un type de flux de travail

- Lorsque vous appelez `RegisterWorkflowType`, définissez le `defaultLambdaRole` champ sur l'ARN du rôle que vous avez défini.

Pour fournir un rôle Lambda à utiliser pendant une exécution de flux de travail

- Lors de l'appel `StartWorkflowExecution`, définissez le champ `LambdaRole` sur l'ARN du rôle que vous avez défini.

**Note**

si le compte appelle RegisterWorkflowType ou StartWorkflowExecution n'est pas autorisé à utiliser le rôle donné, l'appel échouera avec un OperationNotPermittedFault.

## Appelez votre fonction Lambda depuis un flux de travail Amazon SWF

Vous pouvez utiliser le type de ScheduleLambdaFunctionDecisionAttributes données pour identifier la fonction Lambda à appeler lors de l'exécution d'un flux de travail.

Lors d'un appel à RespondDecisionTaskCompleted, fournissez une liste de décisions ScheduleLambdaFunctionDecisionAttributes à prendre. Par exemple :

```
{
  "decisions": [{
    "ScheduleLambdaFunctionDecisionAttributes": {
      "id": "lambdaTaskId",
      "name": "myLambdaFunctionName",
      "input": "inputToLambdaFunction",
      "startToCloseTimeout": "30"
    },
  ]
}
```

Définissez les paramètres suivants :

- ID avec un identifiant pour la tâche Lambda. Il doit s'agir d'une chaîne qui contient entre 1 et 256 caractères, sans les caractères : (deux points), / (barre oblique), | (barre verticale) ni les caractères de contrôle (\u0000 - \u001f et \u007f - \u009f), ni la chaîne littérale a.r.n.
- nom avec le nom de votre fonction Lambda. Votre flux de travail Amazon SWF doit être doté d'un rôle IAM lui permettant d'appeler la fonction Lambda. Le nom fourni doit respecter les contraintes du FunctionName paramètre, comme dans l'action Lambda Invoke.
- input avec les données d'entrée facultatives de la fonction. S'il est défini, il doit respecter les contraintes du ClientContext paramètre, comme dans l'action Lambda Invoke.
- startToCloseDélai d'expiration avec une période maximale facultative, en secondes, que la fonction peut prendre pour s'exécuter avant que la tâche n'échoue, avec une exception de délai d'expiration. La valeur NONE peut être utilisée pour spécifier une durée illimitée.

Pour plus d'informations, voir [Implémentation de AWS Lambda tâches](#)

## Développement d'un Activity Worker dans Amazon SWF

Un outil de traitement d'activité fournit l'implémentation d'un ou de plusieurs types d'activités. Un travailleur d'activité communique avec Amazon SWF pour recevoir des tâches d'activité et les exécuter. Plusieurs outils de traitement d'activité peuvent réaliser des tâches dont le type d'activité est le même.

Amazon SWF met une tâche d'activité à la disposition des travailleurs lorsque le décideur planifie la tâche d'activité. Lorsqu'un décideur planifie une tâche d'activité, il fournit les données (que vous déterminez) dont le travailleur d'activité a besoin pour effectuer cette tâche d'activité. Amazon SWF insère ces données dans la tâche d'activité avant de les envoyer au responsable de l'activité.

Les outils de traitement d'activité sont gérés par vous. Ils peuvent être écrits dans n'importe quel langage. Un programme de travail peut être exécuté n'importe où, à condition qu'il puisse communiquer avec Amazon SWF via l'API. Amazon SWF fournissant toutes les informations nécessaires à l'exécution d'une tâche d'activité, tous les intervenants peuvent être apatrides. L'absence d'état permet aux flux de travail d'être hautement évolutifs. Pour gérer les besoins accrus en capacité, il suffit d'ajouter des outils de traitement d'activité.

Cette section explique comment mettre en œuvre un outil de traitement d'activité. Les outils de traitement d'activité doivent régulièrement effectuer les opérations suivantes.

1. Interrogez Amazon SWF pour une tâche d'activité.
2. Commencer l'exécution de la tâche
3. Signalez régulièrement un battement de cœur à Amazon SWF si la tâche est de longue durée.
4. Signalez que la tâche s'est terminée ou a échoué et renvoyez les résultats à Amazon SWF.

### Rubriques

- [Recherche de tâches d'activité](#)
- [Réalisation de la tâche d'activité](#)
- [Création de rapports sur les pulsations des tâches d'activité](#)
- [Réussite ou échec d'une tâche d'activité](#)
- [Lancement des outils de traitement d'activité](#)

## Recherche de tâches d'activité

Pour effectuer des tâches d'activité, chaque intervenant doit interroger Amazon SWF en appelant régulièrement l'`PollForActivityTask`.

Dans l'exemple suivant, l'outil de traitement d'activité `ChargeCreditCardWorker01` recherche une tâche dans la liste des tâches, `ChargeCreditCard-v0.1`. Si aucune tâche d'activité n'est disponible, Amazon SWF renvoie une réponse vide au bout de 60 secondes. Une réponse vide est une structure `Task` dans laquelle la valeur `taskToken` est une chaîne vide.

```
https://swf.us-east-1.amazonaws.com
PollForActivityTask
{
  "domain" : "867530901",
  "taskList" : { "name": "ChargeCreditCard-v0.1" },
  "identity" : "ChargeCreditCardWorker01"
}
```

Si une tâche d'activité devient disponible, Amazon SWF la renvoie au responsable de l'activité. La tâche contient les données que le décideur spécifie quand il planifie l'activité.

Une fois qu'un outil de traitement d'activité reçoit une tâche d'activité, il peut l'effectuer. La section suivante décrit comment réaliser une tâche d'activité.

## Réalisation de la tâche d'activité

Après avoir reçu une tâche d'activité, l'outil de traitement d'activité est prêt à l'exécuter.

Pour effectuer une tâche d'activité

1. Programmez l'outil de traitement d'activité pour interpréter le contenu du champ d'entrée de la tâche. Ce champ contient les données spécifiées par le décideur lorsque la tâche a été programmée.
2. Programmez l'outil de traitement d'activité pour commencer à traiter les données et à exécuter votre logique.

La section suivante explique comment programmer vos agents d'activité pour qu'ils fournissent des mises à jour de statut à Amazon SWF pour les activités de longue durée.

## Création de rapports sur les pulsations des tâches d'activité

Si un délai de pulsation a été associé au type d'activité, l'outil de traitement d'activité doit enregistrer une pulsation avant que ce délai n'expire. Si une tâche d'activité ne fournit pas de battement de cœur dans le délai imparti, la tâche expire, Amazon SWF la ferme et planifie une nouvelle tâche de décision pour informer le décideur du délai d'expiration. Le décideur peut ensuite replanifier la tâche d'activité ou effectuer une autre action.

Si, après expiration du délai imparti, le responsable de l'activité tente de contacter Amazon SWF, par exemple en appelant `RespondActivityTaskCompleted`, Amazon SWF renverra un message d'erreur. `UnknownResource`

Cette section décrit comment fournir une pulsation d'activité.

Pour enregistrer une pulsation pour la tâche d'activité, programmez l'outil de traitement d'activité pour appeler l'action `RecordActivityTaskHeartbeat`. Cette action fournit également un champ de chaîne que vous pouvez utiliser pour stocker les données de forme libre afin de quantifier la progression de la façon qui convient à votre application.

Dans cet exemple, le responsable de l'activité indique le rythme cardiaque à Amazon SWF et utilise le champ de détails pour indiquer que la tâche d'activité est terminée à 40 %. Pour signaler la pulsation, l'outil de traitement d'activité doit spécifier le jeton de la tâche d'activité.

```
https://swf.us-east-1.amazonaws.com
RecordActivityTaskHeartbeat
{
  "taskToken" : "12342e17-80f6-FAKE-TASK-TOKEN32f0223",
  "details" : "40"
}
```

Cette action ne crée pas en soi un événement dans l'historique d'exécution du flux de travail. Toutefois, si la tâche expire, l'historique d'exécution du flux de travail contient un événement `ActivityTaskTimedOut` qui comprend les informations provenant de la dernière pulsation générée par l'outil de traitement d'activité.

## Réussite ou échec d'une tâche d'activité

Après l'exécution d'une tâche, l'outil de traitement d'activité doit indiquer si celle-ci a réussi ou échoué.

## Finalisation d'une tâche d'activité

Pour finaliser une tâche d'activité, programmez l'outil de traitement d'activité pour appeler l'action `RespondActivityTaskCompleted` une fois que la tâche d'activité aboutit, en spécifiant le jeton correspondant.

Dans cet exemple, l'outil de traitement d'activité indique que la tâche s'est correctement déroulée.

```
https://swf.us-east-1.amazonaws.com
RespondActivityTaskCompleted
{
  "taskToken": "12342e17-80f6-FAKE-TASK-TOKEN32f0223",
  "results": "40"
}
```

Lorsque l'activité est terminée, Amazon SWF planifie une nouvelle tâche de décision pour l'exécution du flux de travail auquel l'activité est associée.

Programmez l'outil de traitement d'activité pour rechercher une autre tâche d'activité une fois qu'une tâche est terminée. Avec ce processus en boucle, l'outil de traitement d'activité recherche et finalise constamment les tâches.

Si l'activité ne répond pas dans le `StartToCloseTimeout` délai imparti, ou si `ScheduleToCloseTimeout` a été dépassée, Amazon SWF expire la tâche d'activité et planifie une tâche de décision. Cette approche permet à un décideur d'effectuer une action appropriée, telle que la reprogrammation de la tâche.

Par exemple, si une EC2 instance Amazon exécute une tâche d'activité et que l'instance échoue avant que la tâche ne soit terminée, le décideur reçoit un événement de temporisation dans l'historique d'exécution du flux de travail. Si la tâche d'activité utilise un battement de cœur, le décideur reçoit l'événement lorsque la tâche ne parvient pas à délivrer le battement de cœur suivant après l'échec de l'instance Amazon EC2. Si ce n'est pas le cas, le décideur finit par recevoir l'événement lorsque la tâche d'activité n'aboutit pas dans les délais spécifiés. Il appartient ensuite au décideur de réaffecter la tâche ou d'effectuer une autre action.

## Échec d'une tâche d'activité

Si, pour une raison ou une autre, un agent d'activité ne peut pas exécuter une tâche d'activité, mais qu'il peut toujours communiquer avec Amazon SWF, vous pouvez le programmer pour qu'il échoue.

Pour que l'outil de traitement d'activité abandonne une tâche d'activité, programmez-le pour qu'il appelle l'action `RespondActivityTaskFailed`, qui spécifie le jeton de la tâche.

```
https://swf.us-east-1.amazonaws.com
RespondActivityTaskFailed
{
  "taskToken" : "12342e17-80f6-FAKE-TASK-TOKEN32f0223",
  "reason" : "CC-Invalid",
  "details" : "Credit Card Number Checksum Failed"
}
```

En tant que développeur, vous définissez les valeurs qui sont stockées dans les champs `Reason` et `Details`. Il s'agit de chaînes de forme libre ; vous pouvez utiliser toutes les conventions de code d'erreur applicables à votre application. Amazon SWF ne traite pas ces valeurs. Amazon SWF peut toutefois afficher ces valeurs dans la console.

Lorsqu'une tâche d'activité échoue, Amazon SWF planifie une tâche de décision pour l'exécution du flux de travail auquel la tâche d'activité est associée afin d'informer le décideur de l'échec. Programmez le décideur pour gérer les activités ayant échoué, par exemple en replanifiant l'activité ou en abandonnant l'exécution du flux de travail, selon la nature de l'échec.

## Lancement des outils de traitement d'activité

Pour lancer les outils de traitement d'activité, compressez votre logique dans un fichier exécutable que vous pourrez utiliser sur la plateforme de ces outils de traitement d'activité. Par exemple, vous pouvez compresser votre code activité sous forme de fichier exécutable Java que vous pouvez exécuter sur des serveurs Linux et Windows.

Une fois lancés, les outils de traitement d'activité se mettent à rechercher des tâches. Toutefois, tant que le décideur ne planifie pas les tâches d'activité, ces recherches expirent sans tâche et les outils de traitement d'activité continuent simplement de rechercher des tâches.

Les sondages étant des demandes sortantes, Activity Worker peut être exécuté sur n'importe quel réseau ayant accès au point de terminaison Amazon SWF.

Vous pouvez lancer autant de outils de traitement d'activité que vous le souhaitez. Au fur et à mesure que le décideur planifie les tâches d'activité, Amazon SWF distribue automatiquement les tâches d'activité aux personnes chargées du sondage.

# Développement de décideurs dans Amazon SWF

Les décideurs sont des implémentations de la logique de coordination qui est appliquée lors de l'exécution de votre flux de travail. Vous pouvez exécuter plusieurs décideurs pour un seul type de flux de travail.

Comme l'état d'exécution d'un flux de travail est enregistré dans l'historique du flux de travail, les décideurs peuvent être apatrides. Amazon SWF conserve l'historique d'exécution du flux de travail et le fournit à un décideur pour chaque tâche de décision. Cela vous permet, le cas échéant, d'ajouter et de supprimer les décideurs, ce qui optimise l'évolutivité du traitement des flux de travail. Lorsque la charge système augmente, il vous suffit d'ajouter des décideurs pour gérer l'accroissement de la capacité. Notez, toutefois, qu'une seule tâche de décision à la fois peut être ouverte pour une exécution de workflow donnée.

Chaque fois qu'un changement d'état survient pour l'exécution d'un flux de travail, Amazon SWF planifie une tâche de décision. Chaque fois qu'un décideur reçoit une tâche de décision, il effectue les opérations suivantes :

- Il interprète l'historique d'exécution du flux de travail fourni avec la tâche de la décision.
- Il applique la logique de coordination en fonction de l'historique d'exécution du flux de travail et émet des décisions sur ce qu'il faut faire ensuite. Chacune d'elles est représentée par une structure de décision.
- Termine la tâche de décision et fournit une liste des décisions à Amazon SWF.

Cette section décrit comment développer un décideur, ce qui implique les actions suivantes :

- Programmation du décideur pour qu'il recherche les tâches de décision
- Programmation du décideur pour qu'il interprète l'historique d'exécution du flux de travail et pour qu'il prenne des décisions
- Programmation du décideur pour qu'il réponde à une tâche de décision

Les exemples de cette section illustrent comment programmer un décideur pour le flux de travail d'un commerce en ligne.

Vous pouvez implémenter le décideur dans le langage de votre choix et l'exécuter n'importe où, à condition qu'il puisse communiquer avec Amazon SWF via son API de service.

## Rubriques

- [Définition de la logique de coordination](#)
- [Recherche des tâches de décision](#)
- [Application de la logique de coordination](#)
- [Renvoi des décisions](#)
- [Fermeture d'une exécution de flux de travail](#)
- [Lancement des décideurs](#)

## Définition de la logique de coordination

La première étape à suivre lors du développement d'un décideur consiste à définir la logique de coordination. Pour un commerce en ligne, la logique de coordination qui planifie chaque activité une fois que l'activité précédente se termine peut ressembler à cela :

```
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
```

Le décideur applique la logique de coordination de l'historique d'exécution du flux de travail et crée une liste des décisions lorsqu'il finalise la tâche de décision à l'aide de l'action `RespondDecisionTaskCompleted`.

## Recherche des tâches de décision

Chaque décideur recherche les tâches de décision. Ces dernières contiennent les informations que le décideur utilise pour générer des décisions telles que la planification des tâches d'activité. Pour rechercher les tâches de décision, le décideur utilise l'action `PollForDecisionTask`.

Dans cet exemple, le décideur recherche une tâche de décision, en spécifiant la liste de tâches `customerOrderWorkflow-0.1`.

```
https://swf.us-east-1.amazonaws.com
PollForDecisionTask
{
  "domain": "867530901",
  "taskList": {"name": "customerOrderWorkflow-v0.1"},
  "identity": "Decider01",
  "maximumPageSize": 50,
  "reverseOrder": true
}
```

Si une tâche de décision est disponible dans la liste de tâches spécifiée, Amazon SWF la renvoie immédiatement. Si aucune tâche de décision n'est disponible, Amazon SWF maintient la connexion ouverte pendant 60 secondes au maximum et renvoie une tâche dès qu'elle est disponible. Si aucune tâche n'est disponible, Amazon SWF renvoie une réponse vide. Une réponse vide est une structure `Task` dans laquelle la valeur `taskToken` est une chaîne vide. Prenez soin de programmer le décideur de sorte qu'il recherche une autre tâche s'il reçoit une réponse vide.

Si une tâche de décision est disponible, Amazon SWF renvoie une réponse contenant la tâche de décision ainsi qu'une vue paginée de l'historique d'exécution du flux de travail.

Dans cet exemple, le type de l'événement le plus récent indique que l'exécution du flux de travail a commencé et que l'élément d'entrée contient les informations nécessaires pour effectuer la première tâche.

```
{
  "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"
      }
    }
  }
}
],
...
}

```

Après la réception de l'historique d'exécution du flux de travail, le décideur interprète l'historique et prend des décisions en fonction de sa logique de coordination.

Comme le nombre d'événements de l'historique d'une seule exécution de flux de travail peut être élevé, le résultat renvoyé est parfois réparti sur plusieurs pages. Pour récupérer les pages suivantes, effectuez des appels supplémentaires en `PollForDecisionTask` utilisant le `nextPageToken` renvoyé par l'appel initial. Notez que vous n'appellez pas `GetWorkflowExecutionHistory` avec ce `nextPageToken`. Au lieu de cela, appelez `PollForDecisionTask` à nouveau.

## Application de la logique de coordination

Une fois que le décideur reçoit une tâche de décision, programmez-le pour qu'il interprète l'historique d'exécution du flux de travail afin de déterminer les événements qui ont eu lieu jusque-là. Il devrait se baser sur ces informations pour générer une liste des décisions.

Dans l'exemple de commerce en ligne que nous avons choisi, seul le dernier événement de l'historique du flux de travail nous intéresse. Nous définissons donc la logique suivante.

```
IF lastEvent = "StartWorkflowInstance"  
  addToDecisions ScheduleVerifyOrderActivity  
  
ELSIF lastEvent = "CompleteVerifyOrderActivity"  
  addToDecisions ScheduleChargeCreditCardActivity  
  
ELSIF lastEvent = "CompleteChargeCreditCardActivity"  
  addToDecisions ScheduleCompleteShipOrderActivity  
  
ELSIF lastEvent = "CompleteShipOrderActivity"  
  addToDecisions ScheduleRecordOrderCompletion  
  
ELSIF lastEvent = "CompleteRecordOrderCompletion"  
  addToDecisions CloseWorkflow  
  
ENDIF
```

Si le dernier événement correspond à `CompleteVerifyOrderActivity`, vous devez ajouter l'activité `ScheduleChargeCreditCardActivity` à la liste des décisions.

Une fois que le décideur a déterminé la ou les décisions à prendre, il peut répondre à Amazon SWF en prenant les décisions appropriées.

## Renvoi des décisions

Après avoir interprété l'historique du flux de travail et généré une liste de décisions, le décideur est prêt à répondre à Amazon SWF avec ces décisions.

Programmez le décideur pour qu'il extraie les données dont il a besoin dans l'historique d'exécution du flux de travail, puis créez des décisions qui spécifient les actions appropriées suivantes pour ce dernier. Le décideur transmet ces décisions à Amazon SWF à l'aide

RespondDecisionTaskCompleted de l'action. Consultez le manuel Amazon Simple Workflow Service API Reference pour obtenir la liste des [types de décisions](#) disponibles.

Dans l'exemple de commerce en ligne, lorsque le décideur renvoie l'ensemble des décisions qu'il a générées, il inclut également l'entrée de la carte de crédit provenant de l'historique d'exécution du flux de travail. L'outil de traitement d'activité dispose ensuite des informations dont il a besoin pour effectuer la tâche d'activité.

Lorsque toutes les activités de l'exécution de flux de travail sont terminées, le décideur ferme celle-ci.

```
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"
      }
    }
  ]
}
```

## Fermeture d'une exécution de flux de travail

Lorsque le décideur détermine que le processus est terminé (autrement dit, qu'il n'y a plus d'activités à effectuer), le décideur génère une décision pour fermer l'exécution de flux de travail.

Afin de fermer une exécution du flux de travail, programmez le décideur pour qu'il interprète les événements de l'historique de flux de travail afin de déterminer ce qui s'est passé lors de l'exécution du flux de travail et pour voir si cette dernière doit être fermée.

Si le flux de travail est terminé sans erreur, fermez l'exécution du flux de travail en appelant `RespondDecisionTaskCompleted` avec la décision `CompleteWorkflowExecution`. Si l'exécution est erronée, vous pouvez l'arrêter à l'aide de la décision `FailWorkflowExecution`.

Dans cet exemple de commerce en ligne, le décideur examine l'historique et se base sur la logique de coordination pour ajouter une décision de fermeture de l'exécution de flux de travail à la liste des décisions, puis initie une action `RespondDecisionTaskCompleted` avec une décision de fermeture du flux de travail.

#### Note

Dans certains cas, fermeture d'une exécution de flux de travail peut échouer. Par exemple, si un signal est reçu pendant que le décideur ferme l'exécution du flux de travail, la décision de fermeture échoue. Pour gérer cette possibilité, vérifiez que le décideur continue de rechercher les tâches de décision. Assurez-vous également que le décideur qui reçoit la prochaine tâche de décision répond à l'événement (dans ce cas, un signal) qui a empêché la clôture de l'exécution.

Si vous le souhaitez, vous pouvez également annuler les exécutions de flux de travail. Cette approche peut s'avérer particulièrement utile pour les flux de travail de longue durée. Pour permettre l'annulation, le décideur doit gérer l'événement `WorkflowExecutionCancelRequested` dans l'historique. Cet événement indique que l'annulation de l'exécution a été demandée. Le décideur doit effectuer les opérations de nettoyage appropriées, telles que l'annulation des tâches d'activité en cours et la fermeture du flux de travail en appelant l'action `RespondDecisionTaskCompleted` avec la décision `CancelWorkflowExecution`.

L'exemple suivant appelle `RespondDecisionTaskCompleted` pour spécifier que l'exécution de flux de travail en cours est annulée.

```
https://swf.us-east-1.amazonaws.com
RespondDecisionTaskCompleted
{
  "taskToken" : "12342e17-80f6-FAKE-TASK-TOKEN32f0223",
  "decisions" : [
```

```
{
  "decisionType":"CancelWorkflowExecution",
  "CancelWorkflowExecutionAttributes":{
    "Details": "Customer canceled order"
  }
}
]
```

Amazon SWF vérifie que la décision de fermer ou d'annuler l'exécution du flux de travail est la dernière décision envoyée par le décideur. Autrement dit, vous ne pouvez pas avoir de décisions après celle qui ferme le flux de travail.

## Lancement des décideurs

Une fois que les décideurs sont développés, vous êtes prêt à en lancer un ou plusieurs.

Pour lancer les décideurs, compressez votre logique de coordination dans un fichier exécutable que vous pouvez utiliser sur la plateforme du décideur. Par exemple, vous pouvez compresser le code du décideur sous forme de fichier exécutable Java que vous pouvez exécuter sur des ordinateurs Linux et Windows.

Une fois lancé, vos décideurs devraient commencer à interroger Amazon SWF pour trouver des tâches. Jusqu'à ce que vous commenciez à exécuter les flux de travail et qu'Amazon SWF ne planifie pas les tâches décisionnelles, ces sondages expireront et recevront des réponses vides. Une réponse vide est une structure Task dans laquelle la valeur `taskToken` est une chaîne vide. Les décideurs doivent simplement poursuivre les recherches.

Amazon SWF garantit qu'une seule tâche de décision peut être active à la fois pour l'exécution d'un flux de travail. Cela empêche les problèmes tels que des décisions contradictoires. En outre, Amazon SWF garantit qu'une seule tâche de décision est attribuée à un seul décideur, quel que soit le nombre de décideurs en cours d'exécution.

Si un événement génère une tâche de décision alors qu'un décideur traite une autre tâche de décision, Amazon SWF met la nouvelle tâche en file d'attente jusqu'à ce que la tâche en cours soit terminée. Une fois la tâche en cours terminée, Amazon SWF rend la nouvelle tâche de décision disponible. En outre, les tâches de décision sont groupées dans le sens où, si plusieurs activités sont effectuées pendant qu'un décideur traite une tâche de décision, Amazon SWF ne créera qu'une seule nouvelle tâche de décision pour tenir compte des multiples tâches achevées. Toutefois, chaque tâche terminée reçoit un événement individuel dans l'historique d'exécution du flux de travail.

Les sondages étant des demandes sortantes, les décideurs peuvent être exécutés sur n'importe quel réseau ayant accès au point de terminaison Amazon SWF.

Pour que les exécutions de flux de travail progressent, au moins un décideur doit être exécuté. Vous pouvez lancer autant de décideurs que vous le souhaitez. Amazon SWF prend en charge l'interrogation de plusieurs décideurs sur la même liste de tâches.

## Démarrage de flux de travail dans Amazon SWF

Vous pouvez lancer l'exécution d'un type de flux de travail enregistré à partir de n'importe quelle application via l'action `StartWorkflowExecution`. Lorsque vous lancez l'exécution, vous y associez un identifiant, appelé `workflowId`. La valeur `workflowId` peut être une chaîne spécifique à votre application, telle que le numéro de commande dans une application de traitement des commandes. Vous ne pouvez pas utiliser la même valeur `workflowId` pour plusieurs exécutions de flux de travail ouvertes dans le même domaine. Par exemple, si vous lancez deux exécutions de flux de travail avec la valeur `workflowId` `Customer Order 01`, la deuxième exécution de flux de travail ne démarre pas et la demande échoue. Vous pouvez toutefois réutiliser `workflowId` une exécution fermée. Amazon SWF associe également un identifiant unique généré par le système, appelé `runId`, à chaque exécution de flux de travail.

Une fois que les types de flux de travail et d'activités sont enregistrés, appelez l'action `StartWorkflowExecution` pour lancer le flux de travail. La valeur du paramètre `input` peut être une chaîne spécifiée par l'application qui initie le flux de travail. La valeur `executionStartToCloseTimeout` correspond au temps, en secondes, que l'exécution de flux de travail peut prendre entre le début à la fermeture. Passé ce délai, cette limite entraîne l'expiration de l'exécution du flux de travail. Contrairement à certains autres paramètres de délai d'expiration d'Amazon SWF, vous ne pouvez pas spécifier `NONE` de valeur de pour ce délai ; la durée maximale d'exécution d'un flux de travail est d'un an. De même, `taskStartToCloseTimeout` s'agit de la durée en secondes qu'une tâche de décision associée à l'exécution de ce flux de travail peut prendre avant l'expiration du délai imparti.

```
https://swf.us-east-1.amazonaws.com
StartWorkflowExecution
{
  "domain" : "867530901",
  "workflowId" : "20110927-T-1",
  "workflowType" : {
    "name" : "customerOrderWorkflow", "version" : "1.1"
```

```
},  
"taskList" : { "name" : "specialTaskList" },  
"input" : "arbitrary-string-that-is-meaningful-to-the-workflow",  
"executionStartToCloseTimeout" : "1800",  
"tagList" : [ "music purchase", "digital", "ricoh-the-dog" ],  
"taskStartToCloseTimeout" : "1800",  
"childPolicy" : "TERMINATE"  
}
```

Si l'`StartWorkflowExecution` action aboutit, Amazon SWF renvoie le `runId` pour l'exécution du flux de travail. Le `runId` pour une exécution de flux de travail est unique dans une région spécifique. Enregistrez-le `runId` au cas où vous auriez besoin ultérieurement de spécifier l'exécution de ce flux de travail lors d'un appel à Amazon SWF. Par exemple, vous aurez besoin du `runId` si vous devez envoyer plus tard un signal à l'exécution de flux de travail.

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

## Définition de la priorité des tâches dans Amazon SWF

Par défaut, les tâches d'une liste des tâches dépendent de leur heure d'arrivée : celles qui sont planifiées en premier sont généralement exécutées en premier, aussi loin que possible. En définissant une priorité de tâche facultative, vous pouvez donner la priorité à certaines tâches : Amazon SWF essaiera de fournir les tâches les plus prioritaires d'une liste de tâches avant celles dont la priorité est inférieure.

### Note

Les tâches qui sont planifiées en premier sont généralement exécutées en premier, mais ce n'est pas garanti.

Vous pouvez définir des priorités de tâche pour les flux de travail et les activités. La priorité de tâche d'un flux de travail n'a pas d'incidence sur la priorité des tâches d'activité qu'il planifie, ni sur les flux de travail enfants qu'il démarre. La priorité par défaut d'une activité ou d'un flux de travail est définie (par vous ou par Amazon SWF) lors de l'enregistrement, et la priorité de tâche enregistrée est toujours utilisée, sauf si elle est remplacée lors de la planification de l'activité ou du démarrage d'une exécution de flux de travail.

Les valeurs de priorité des tâches peuvent aller de « -2147483648 » à « 2147483647 » (le nombre le plus élevé indique une priorité supérieure). Si vous ne définissez la priorité de tâche d'une activité ou d'un flux de travail, la priorité zéro (« 0 ») lui est attribuée.

## Rubriques

- [Définition d'une priorité de tâche pour les flux de travail](#)
- [Définition d'une priorité de tâche pour les activités](#)
- [Actions renvoyant les informations relatives aux priorités de tâche](#)

## Définition d'une priorité de tâche pour les flux de travail

Vous pouvez définir la priorité de tâche d'un flux de travail lorsque vous l'enregistrez ou lorsque vous le lancez. La priorité de tâche définie lorsque le type de flux de travail est enregistré est utilisée comme valeur par défaut pour toutes les exécutions de flux de travail de ce type, sauf si elle est remplacée lors du lancement de l'exécution de flux de travail.

Pour enregistrer un type de flux de travail avec une priorité de tâche par défaut, définissez l'`defaultTaskPriorityOption` lorsque vous utilisez l'[RegisterWorkflowType](#) action :

```
{
  "domain": "867530901",
  "name": "expeditedOrderWorkflow",
  "version": "1.0",
  "description": "Expedited customer orders workflow",
  "defaultTaskStartToCloseTimeout": "600",
  "defaultExecutionStartToCloseTimeout": "3600",
  "defaultTaskList": {"name": "mainTaskList"},
  "defaultTaskPriority": "10",
  "defaultChildPolicy": "TERMINATE"
}
```

Vous pouvez annuler la priorité de tâche enregistrée d'un type de flux de travail lorsque vous démarrez l'exécution d'un flux de travail avec [StartWorkflowExecution](#):

```
{
  "childPolicy": "TERMINATE",
  "domain": "867530901",
  "executionStartToCloseTimeout": "1800",
  "input": "arbitrary-string-that-is-meaningful-to-the-workflow",
}
```

```

"tagList": ["music purchase", "digital", "ricoh-the-dog"],
"taskList": {"name": "specialTaskList"},
"taskPriority": "-20",
"taskStartToCloseTimeout": "600",
"workflowId": "20110927-T-1",
"workflowType": {"name": "customerOrderWorkflow", "version": "1.0"},
}

```

Vous pouvez également annuler la priorité des tâches enregistrée lorsque vous démarrez un flux de travail enfant ou lorsque vous poursuivez un flux de travail en tant que nouveau, par exemple lorsque vous répondez à une décision par [RespondDecisionTaskCompleted](#).

Pour définir la priorité de tâche d'un flux de travail enfant, indiquez la valeur dans `startChildWorkflowExecutionDecisionAttributes` :

```

{
  "taskToken": "AAAAKgAAAAEAAAAAAAAAAAA...",
  "decisions": [
    {
      "decisionType": "StartChildWorkflowExecution",
      "startChildWorkflowExecutionDecisionAttributes": {
        "childPolicy": "TERMINATE",
        "control": "digital music",
        "executionStartToCloseTimeout": "900",
        "input": "201412-Smith-011x",
        "taskList": {"name": "specialTaskList"},
        "taskPriority": "5",
        "taskStartToCloseTimeout": "600",
        "workflowId": "verification-workflow",
        "workflowType": {
          "name": "MyChildWorkflow",
          "version": "1.0"
        }
      }
    }
  ]
}

```

Lorsque vous poursuivez un flux de travail comme s'il était nouveau, définissez la priorité de la tâche dans `continueAsNewWorkflowExecutionDecisionAttributes` :

```

{

```

```
"taskToken": "AAAAKgAAAAEAAAAAAAAAAAA...",
"decisions": [
  {
    "decisionType": "ContinueAsNewWorkflowExecution",
    "continueAsNewWorkflowExecutionDecisionAttributes": {
      "childPolicy": "TERMINATE",
      "executionStartToCloseTimeout": "1800",
      "input": "5634-0056-4367-0923,12/12,437",
      "taskList": {"name": "specialTaskList"},
      "taskStartToCloseTimeout": "600",
      "taskPriority": "100",
      "workflowTypeVersion": "1.0"
    }
  }
]
```

## Définition d'une priorité de tâche pour les activités

Vous pouvez définir la priorité de tâche d'une activité lors de son enregistrement ou de sa planification. La priorité de tâche définie lors de l'enregistrement d'un type d'activité est utilisée comme priorité par défaut lorsque l'activité est exécutée, sauf si elle est remplacée lors de la planification de l'activité.

Pour définir la priorité des tâches lors de l'enregistrement d'un type d'activité, définissez l'`defaultTaskPriorityOption` lorsque vous utilisez l'[RegisterActivityType](#) action :

```
{
  "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"
}
```

Pour planifier une tâche avec une priorité de tâche, utilisez l'option `TaskPriority` lorsque vous planifiez l'activité avec l'[RespondDecisionTaskCompleted](#) action :

```
{
  "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"
      }
    }
  ]
}
```

## Actions renvoyant les informations relatives aux priorités de tâche

Vous pouvez obtenir des informations sur la priorité des tâches définie (ou définir la priorité des tâches par défaut) à partir des actions Amazon SWF suivantes :

- [DescribeActivityType](#) renvoie le type `defaultTaskPriority` d'activité dans la `configuration` section de la réponse.
- [DescribeWorkflowExecution](#) renvoie le `TaskPriority` de l'exécution du flux de travail dans la `executionConfiguration` section de la réponse.
- [DescribeWorkflowType](#) renvoie le type `defaultTaskPriority` de flux de travail dans la `configuration` section de la réponse.
- [GetWorkflowExecutionHistory](#) et [PollForDecisionTask](#) fournissent des informations sur la priorité des tâches dans les `workflowExecutionStartedEventAttributes` sections `activityTaskScheduledEventAttributes` `decisionTaskScheduledEventAttributes` `workflowExecutionContinuedAsNewEventAttributes` et de la réponse.

# Gestion des erreurs dans Amazon SWF

Différents types d'erreurs peuvent survenir au cours d'une exécution de flux de travail.

## Rubriques

- [Erreurs de validation](#)
- [Erreur lors de l'application d'actions ou de décisions](#)
- [Délais](#)
- [Erreurs déclenchées par le code utilisateur](#)
- [Erreurs liées à la fermeture d'une exécution de flux de travail](#)

## Erreurs de validation

Des erreurs de validation se produisent lorsqu'une demande adressée à Amazon SWF échoue parce qu'elle n'est pas correctement formée ou qu'elle contient des données non valides. Dans ce contexte, il peut s'agir d'une action comme `DescribeDomain` ou d'une décision comme `StartTimer`. Si la demande est une action, Amazon SWF renvoie un code d'erreur dans la réponse. Vérifiez ce code d'erreur, car il peut indiquer la cause de l'échec. Par exemple, il se peut qu'un ou plusieurs des arguments transmis avec la requête ne soient pas valides. Pour obtenir la liste des codes d'erreur courants, consultez la rubrique relative à l'action dans le manuel Amazon Simple Workflow Service API Reference.

Si la requête qui a échoué est une décision, un événement approprié figure dans l'historique d'exécution du flux de travail. Par exemple, si la décision `StartTimer` a échoué, l'événement `StartTimerFailed` apparaît dans l'historique. Le décideur doit vérifier ces événements lorsqu'il reçoit l'historique en réponse à l'action `PollForDecisionTask` ou `GetWorkflowExecutionHistory`. Voici une liste des événements d'échec qui peuvent se produire lorsque la décision n'est pas correctement formée ou lorsqu'elle contient des données non valides.

## Erreur lors de l'application d'actions ou de décisions

Même si la demande est correctement formulée, des erreurs peuvent se produire lorsqu'Amazon SWF tente de l'exécuter. Dans ce cas, l'un des événements suivants de l'historique indique qu'une erreur est survenue. Examinez le champ `reason` de l'événement afin de déterminer la cause de l'échec.

- [CancelTimerFailed](#)

- [RequestCancelActivityTaskFailed](#)
- [RequestCancelExternalWorkflowExecutionFailed](#)
- [ScheduleActivityTaskFailed](#)
- [SignalExternalWorkflowExecutionFailed](#)
- [StartChildWorkflowExecutionFailed](#)
- [StartTimerFailed](#)

## Délais

Les [décideurs](#), les [outils de traitement d'activité](#) et les [exécutions de flux de travail](#) sont tous soumis à des délais d'expiration. Avec ce type d'erreur, une tâche ou un flux de travail enfant expire. Un événement qui décrit l'expiration s'affiche dans l'historique. Pour gérer cet événement, le décideur doit, par exemple, replanifier la tâche ou redémarrer le flux de travail enfant. Pour plus d'informations sur les délais d'expiration, consultez la section [Types de délai d'expiration Amazon SWF](#).

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

## Erreurs déclenchées par le code utilisateur

Ce type de condition d'erreur correspond, par exemple, à l'échec d'une tâche d'activité ou d'un flux de travail enfant. Comme pour les erreurs de temporisation, Amazon SWF ajoute un événement approprié à l'historique d'exécution du flux de travail. Pour gérer cet événement, le décideur doit essayer de replanifier la tâche ou de redémarrer le flux de travail enfant.

- [ActivityTaskFailed](#)
- [ChildWorkflowExecutionFailed](#)

## Erreurs liées à la fermeture d'une exécution de flux de travail

Les décideurs peuvent également voir les événements suivants s'ils essaient de fermer un flux de travail avec une tâche de décision en attente.

- [FailWorkflowExecutionFailed](#)
- [CompleteWorkFlowExecutionFailed](#)
- [ContinueAsNewWorkflowExecutionFailed](#)
- [CancelWorkflowExecutionFailed](#)

Pour plus d'informations sur les événements répertoriés ci-dessus, consultez la section [Historique des événements](#) dans le manuel Amazon SWF API Reference.

# Quotas Amazon SWF

Amazon SWF impose des quotas sur la taille de certains paramètres du flux de travail, tels que le nombre de domaines par compte et la taille de l'historique d'exécution du flux de travail. Ces quotas sont conçus pour empêcher les flux de travail erronés de consommer toutes les ressources du système, mais ils ne constituent pas des limites strictes. Si vous constatez que votre application dépasse fréquemment ces quotas, vous pouvez [demander une augmentation des quotas de service](#).

## Table des matières

- [Quotas de compte généraux pour Amazon SWF](#)
- [Quotas d'exécution des flux de travail](#)
- [Quotas relatifs à l'exécution des tâches](#)
- [Limitation des quotas Amazon SWF](#)
  - [Limiter les quotas pour toutes les régions](#)
  - [Quotas décisionnels pour toutes les régions](#)
  - [Quotas au niveau du flux de travail](#)
- [Demande d'augmentation de quota](#)

## Quotas de compte généraux pour Amazon SWF

- Nombre maximum de domaines enregistrés : 100

Ce quota inclut à la fois les domaines enregistrés et les domaines obsolètes.

- Nombre maximum de flux de travail et de types d'activités : 10 000 par domaine

Ce quota inclut à la fois les types enregistrés et les types déconseillés.

- Quota d'appels d'API — Au-delà des pics peu fréquents, les applications peuvent être limitées si elles effectuent un grand nombre d'appels d'API en très peu de temps.
- Taille maximale des demandes : 1 Mo par demande

Il s'agit de la taille totale des données par demande d'API Amazon SWF, y compris l'en-tête de demande et toutes les autres données de demande associées.

- Réponses tronquées pour le décompte APIs : indique qu'un quota interne a été atteint et que la réponse ne correspond pas au décompte complet.

Certaines requêtes atteindront en interne le quota de 1 Mo mentionné ci-dessus avant de renvoyer une réponse complète. Les requêtes suivantes peuvent renvoyer une réponse tronquée au lieu du nombre total.

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

Pour chacune de ces requêtes, si la réponse `truncated` est définie sur `true`, le nombre est inférieur au montant total. Ce quota interne ne peut pas être augmenté.

- Nombre maximum de balises : 50 balises par ressource.

Toute tentative d'ajout de plus de 50 balises entraînera une erreur 400, `TooManyTagsFault`.

## Quotas d'exécution des flux de travail

- Nombre maximum d'exécutions de flux de travail ouverts : 100 000 par domaine

Ce nombre inclut les exécutions de flux de travail enfants.

- Durée maximale d'exécution du flux de travail : 1 an. Il s'agit d'un quota strict qui ne peut pas être modifié.
- Taille maximale de l'historique d'exécution du flux de travail : 25 000 événements. Il s'agit d'un quota strict qui ne peut pas être modifié.

La bonne pratique consiste à structurer chaque flux de travail de sorte que son historique ne contienne pas plus de 10 000 événements. Comme le décideur doit récupérer l'historique du flux de travail, un historique réduit lui permet de terminer plus rapidement. Si vous utilisez le [Flow Framework](#), vous pouvez l'utiliser `ContinueAsNew` pour poursuivre un flux de travail avec un nouvel historique.

- Nombre maximum d'exécutions de flux de travail Open Child : 1 000 par exécution de flux de travail

Si votre cas d'utilisation vous oblige à dépasser ces quotas, vous pouvez utiliser les fonctionnalités proposées par Amazon SWF pour poursuivre les exécutions et structurer vos applications à l'aide d'exécutions de [flux de travail secondaires](#). Si vous constatez que vous avez toujours besoin d'une augmentation de quota, consultez [Demande d'augmentation de quota](#).

## Quotas relatifs à l'exécution des tâches

- Nombre maximum de sondes par liste de tâches : 1 000 par liste de tâches

1 000 observateurs au maximum peuvent interroger simultanément une liste de tâches particulière. Si vous dépassez 1 000, vous recevez une exception `LimitExceededException`.

### Note

Bien que le maximum soit de 1 000, il est possible que vous rencontriez `LimitExceededException` des erreurs bien avant ce quota. Cette erreur ne signifie pas que vos tâches sont retardées. Cela signifie plutôt que vous avez le nombre maximum de sondes inactifs sur une liste de tâches. Amazon SWF définit cette limite pour économiser des ressources à la fois côté client et côté serveur. La fixation de la limite empêche un nombre excessif de sondes d'attendre inutilement. Vous pouvez réduire les `LimitExceededException` erreurs en utilisant plusieurs listes de tâches pour distribuer les sondages.

- Nombre maximum de tâches planifiées par seconde : 2 000 par liste de tâches

Vous pouvez planifier un maximum de 2 000 tâches par seconde sur une liste de tâches donnée. Si vous dépassez les 2 000, vos `ScheduleActivityTask` décisions échoueront par `ACTIVITY_CREATION_RATE_EXCEEDED` erreur.

### Note

Bien que le maximum soit de 2 000, vous risquez de rencontrer `ACTIVITY_CREATION_RATE_EXCEEDED` des erreurs bien avant ce quota. Pour réduire ces erreurs, utilisez plusieurs listes de tâches pour répartir la charge.

- Durée maximale d'exécution des tâches : 1 an (limitée par la durée maximale d'exécution du flux de travail)

Vous pouvez configurer des [délais d'activité](#) pour déclencher un événement d'expiration une étape particulière de votre [tâche d'activité](#) est trop longue.

- Durée maximale pendant laquelle SWF conservera une tâche dans la file d'attente : 1 an (limité par le quota de temps d'exécution du flux de travail)

Vous pouvez configurer par défaut des [délais d'activité](#) lors de l'enregistrement de l'activité afin de déclencher un événement d'expiration lorsqu'une étape particulière de votre [tâche d'activité](#) est trop longue. Vous pouvez également remplacer les délais d'activité par défaut lorsque vous planifiez une tâche d'activité dans le code du décideur.

- Nombre maximum de tâches ouvertes : 1 000 par exécution du flux de travail.

Ce quota inclut à la fois les tâches d'activité planifiées et celles traitées par les travailleurs.

- Nombre maximum de minuterics ouvertes : 1 000 par exécution du flux de travail
- Taille maximale input/result des données : 32 768 caractères

Ce quota affecte les données de résultat de l'activité ou de l'exécution du flux de travail, les données d'entrée lors de la planification des tâches d'activité ou des exécutions de flux de travail, et les entrées envoyées avec un [signal d'exécution du flux](#) de travail.

- Nombre maximum de décisions dans une réponse à une tâche de décision — variable

En raison du quota de 1 Mo fixé à la [taille maximale des demandes d'API](#), le nombre de décisions renvoyées par un seul appel [RespondDecisionTaskCompleted](#) sera limité en fonction de la taille des données utilisées par chaque décision, y compris la taille des données d'entrée fournies pour les tâches d'activité planifiées ou les exécutions de flux de travail.

## Limitation des quotas Amazon SWF

[Outre les quotas de service décrits précédemment, certains appels d'API Amazon SWF et certains événements décisionnels sont limités afin de maintenir la bande passante du service, à l'aide d'un schéma de compartiments à jetons.](#) Si votre taux de demandes dépasse régulièrement les taux indiqués ici, vous pouvez [demander une augmentation du quota d'accélération](#).

Les quotas de limitation et de décision sont les mêmes dans toutes les régions.

## Limiter les quotas pour toutes les régions

Les quotas suivants sont applicables au niveau de chaque compte. Vous pouvez également demander une augmentation des quotas suivants. Pour plus d'informations à ce sujet, consultez [Demande d'augmentation de quota](#).

Nom d'API	Taille de compartiment	Taux de remplissage par seconde
CountClosedWorkflowExecutions	2000	6
CountOpenWorkflowExecutions	2000	6
CountPendingActivityTasks	200	6
CountPendingDecisionTasks	200	6
DeleteActivityType	200	6
DeleteWorkflowType	200	6
DeprecateActivityType	200	6
DeprecateDomain	100	6
DeprecateWorkflowType	200	6
DescribeActivityType	2000	6
DescribeDomain	200	6
DescribeWorkflowExecution	2000	6
DescribeWorkflowType	2000	6
GetWorkflowExecutionHistory	2000	60
ListActivityTypes	200	6
ListClosedWorkflowExecutions	200	6
ListDomains	100	6
ListOpenWorkflowExecutions	200	48
ListTagsForResource	50	30

Nom d'API	Taille de compartiment	Taux de remplissage par seconde
ListWorkflowTypes	200	6
PollForActivityTask	2000	200
PollForDecisionTask	2000	200
RecordActivityTaskHeartbeat	2000	160
RegisterActivityType	200	60
RegisterDomain	100	6
RegisterWorkflowType	200	60
RequestCancelWorkflowExecution	2000	30
RespondActivityTaskCanceled	2000	200
RespondActivityTaskCompleted	2000	200
RespondActivityTaskFailed	2000	200
RespondDecisionTaskCompleted	2000	200
SignalWorkflowExecution	2000	30
StartWorkflowExecution	2000	200
TagResource	50	30
TerminateWorkflowExecution	2000	60
UndeprecateActivityType	200	6
UndeprecateDomain	100	6
UndeprecateWorkflowType	200	6
UntagResource	50	30

## Quotas décisionnels pour toutes les régions

Les quotas suivants sont applicables au niveau de chaque compte. Vous pouvez également demander une augmentation des quotas suivants. Pour plus d'informations à ce sujet, consultez [Demande d'augmentation de quota](#).

Nom d'API	Taille de compartiment	Taux de remplissage par seconde
RequestCancelExternalWorkflowExecution	1200	120
ScheduleActivityTask	1 000	200
SignalExternalWorkflowExecution	1200	120
StartChildWorkflowExecution	500	12
StartTimer	2000	200

## Quotas au niveau du flux de travail

Les quotas suivants sont applicables au niveau du flux de travail et ne peuvent pas être augmentés.

Nom d'API	Taille de compartiment	Taux de remplissage par seconde
GetWorkflowExecutionHistory	400	200
SignalWorkflowExecution	1 000	1 000
RecordActivityTaskHeartbeat	1 000	1 000
RequestCancelWorkflowExecution	200	200

## Demande d'augmentation de quota

Pour plus d'informations, consultez la section [Quotas du service AWS](#) dans le Références générales AWS.

# Ressources supplémentaires et informations de référence pour Amazon SWF

Ce chapitre fournit des ressources supplémentaires et des informations de référence utiles lors du développement de flux de travail avec Amazon SWF.

## Rubriques

- [Types de délai d'expiration Amazon SWF](#)
- [Points de terminaison Amazon Simple Workflow Service](#)
- [Documentation supplémentaire pour le service Amazon Simple Workflow](#)
- [Ressources Web pour le service Amazon Simple Workflow](#)
- [Options de migration pour Ruby Flow](#)

## Types de délai d'expiration Amazon SWF

Pour garantir que les exécutions des flux de travail s'exécutent correctement, vous pouvez définir différents types de délais d'expiration avec Amazon SWF. Certains délais spécifient la durée totale d'exécution du flux de travail. D'autres délais définissent combien de temps les tâches d'activité peuvent prendre avant d'être affectées à un outil de traitement, ainsi que combien de temps elles peuvent prendre pour se terminer à compter de l'heure où elles sont planifiées. Tous les délais d'expiration de l'API Amazon SWF sont spécifiés en secondes. Amazon SWF prend également en charge la chaîne en NONE tant que valeur de délai d'attente, ce qui indique l'absence de délai d'expiration.

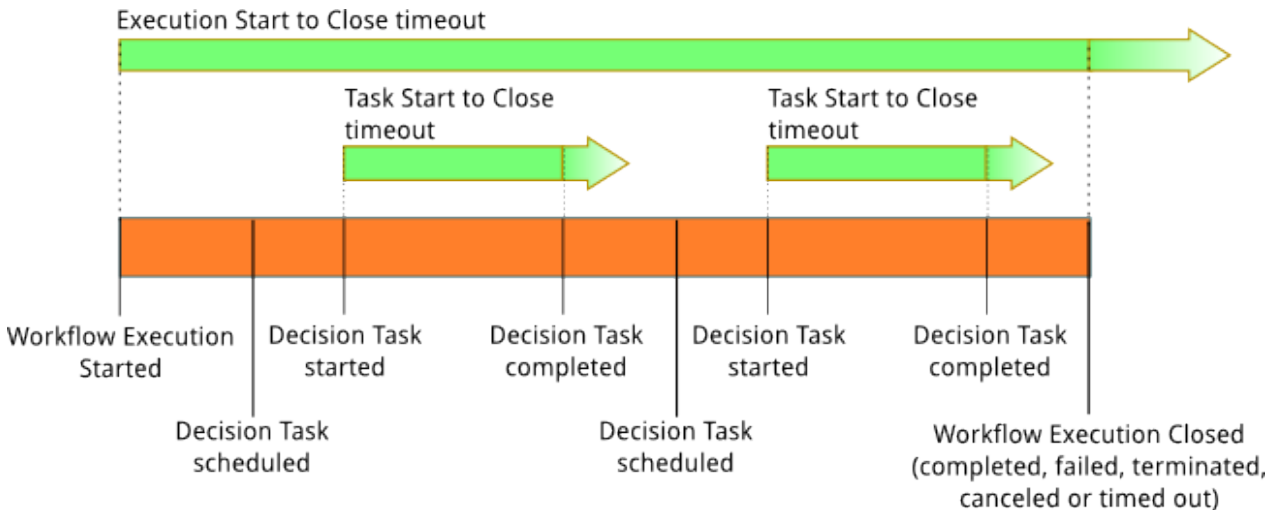
Pour les délais liés aux tâches de décision et aux tâches d'activité, Amazon SWF ajoute un événement à l'historique d'exécution du flux de travail. Les attributs de l'événement fournissent des informations sur le type de délai d'attente qui s'est produit et sur la tâche de décision ou d'activité affectée. Amazon SWF planifie également une tâche de décision. Lorsque le décideur reçoit la nouvelle tâche de décision, il voit l'événement de temporisation dans l'historique et prend l'action appropriée en l'[RespondDecisionTaskCompleted](#) appelant.

Une tâche est considérée comme ouverte depuis le moment où elle est planifiée jusqu'à ce qu'elle soit fermée. Par conséquent, une tâche est indiquée comme ouverte lorsqu'un outil de traitement s'en occupe. Une tâche est fermée lorsqu'un outil de traitement la signale comme [terminée](#), comme

[annulée](#) ou comme [ayant échoué](#). Une tâche peut également être fermée par Amazon SWF en raison d'un délai d'attente.

## Délais liés au flux de travail et aux tâches de décision

Le schéma suivant illustre comment les délais de flux de travail et de décision sont liés à la durée de vie d'un flux de travail :



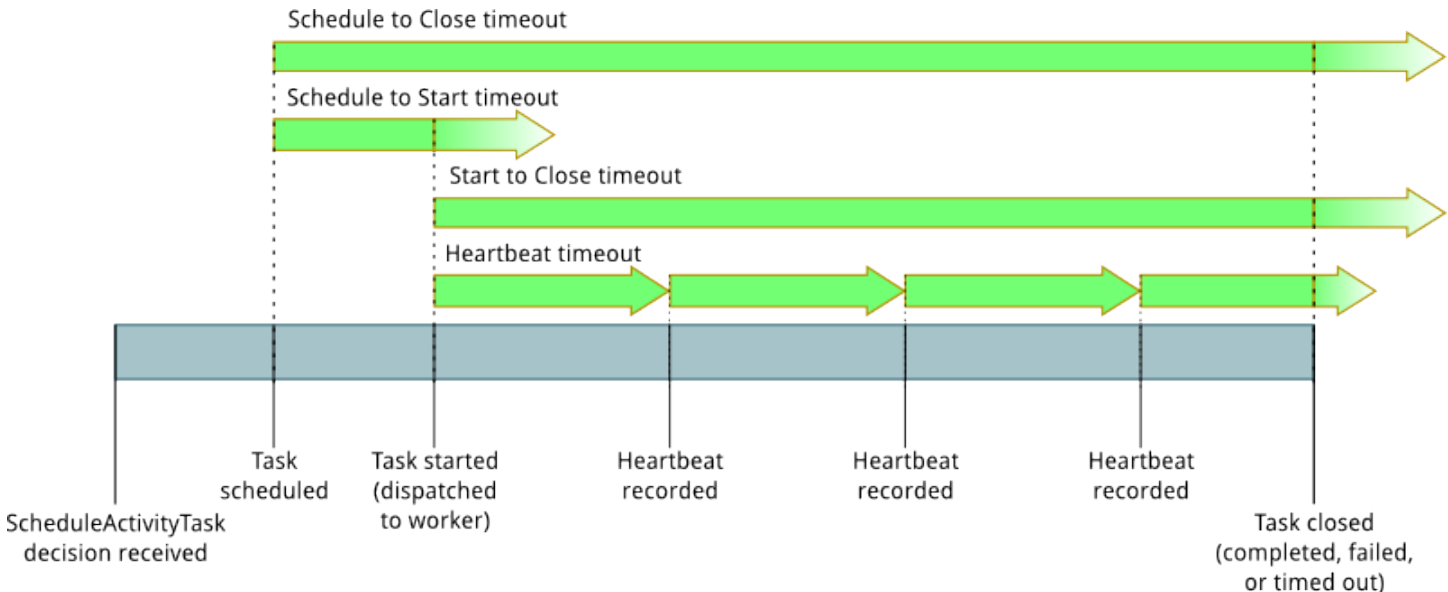
Il existe deux types de délai qui s'appliquent aux tâches du flux de travail et aux tâches de décision :

- Du début à la fermeture du flux de travail (**timeoutType: START\_TO\_CLOSE**) : ce délai indique la durée maximale d'exécution d'un flux de travail. Il est défini comme valeur par défaut lors de l'enregistrement du flux de travail, mais peut être remplacé par une autre valeur lorsque le flux de travail est lancé. Si ce délai est dépassé, Amazon SWF ferme l'exécution du flux de travail et ajoute [un](#) événement de [WorkflowExecutionTimedOut](#) type à l'historique d'exécution du flux de travail. Outre `timeoutType`, les attributs de l'événement spécifient la stratégie `childPolicy` qui est en vigueur pour cette exécution de flux de travail. La stratégie enfant définit comment les exécutions de flux de travail enfant sont gérées si l'exécution de flux de travail parent expire ou si elle est arrêtée. Par exemple, si la stratégie `childPolicy` est définie sur `TERMINATE`, les exécutions de flux de travail enfant sont arrêtées. Une fois qu'une exécution de flux de travail expire, vous ne pouvez plus effectuer que les appels de visibilité.
- Début de la fin de la tâche de décision (**timeoutType: START\_TO\_CLOSE**) : ce délai indique le temps maximum que le décideur correspondant peut prendre pour terminer une tâche de décision. Il est défini lors de l'enregistrement du type de flux de travail. Si ce délai est dépassé, la tâche est marquée comme expirée dans l'historique d'exécution du flux de travail, et Amazon SWF ajoute un événement de [DecisionTaskTimedOut](#) type à l'historique du flux de travail. Les attributs de l'événement incluront IDs les événements correspondant au moment où cette tâche de décision

a été planifiée (`scheduledEventId`) et à son lancement (`startedEventId`). Outre l'ajout de l'événement, Amazon SWF planifie également une nouvelle tâche de décision pour avertir le décideur que le délai imparti pour cette tâche de décision a expiré. Après expiration de ce délai, toute tentative de finalisation de la décision avec `RespondDecisionTaskCompleted` échoue.

## Délais des tâches d'activité

Le schéma suivant illustre le lien entre les délais d'attente et la durée de vie d'une tâche d'activité :



Quatre types de délai s'appliquent aux tâches d'activité :

- Début de la fin de la tâche d'activité (**timeoutType: START\_TO\_CLOSE**) : ce délai indique le temps maximum qu'un agent d'activité peut prendre pour traiter une tâche une fois qu'il l'a reçue. Les tentatives de clôture d'une tâche d'activité dont le délai imparti a expiré à l'aide de [RespondActivityTaskCanceled](#), [RespondActivityTaskCompleted](#), et [RespondActivityTaskFailed](#) échoueront.
- Activity Task Heartbeat (**timeoutType: HEARTBEAT**) : ce délai indique la durée maximale pendant laquelle une tâche peut être exécutée avant de fournir sa progression au cours de l'`RecordActivityTaskHeartbeat` action.
- Planification des tâches d'activité jusqu'au début (**timeoutType: SCHEDULE\_TO\_START**) : ce délai indique la durée pendant laquelle Amazon SWF attend avant de mettre fin à la tâche d'activité si aucun collaborateur n'est disponible pour effectuer la tâche. Lorsque la tâche arrive à expiration, elle n'est pas attribuée à un autre outil de traitement.

- **Calendrier de clôture des tâches d'activité (`timeoutType: SCHEDULE_TO_CLOSE`)** : ce délai indique le temps que la tâche peut prendre entre le moment où elle est planifiée et le moment où elle est terminée. Il est recommandé que cette valeur ne soit pas supérieure à la somme du délai d'expiration de la tâche et du `schedule-to-start` délai d'expiration de la tâche `start-to-close`.

#### Note

Chacun des types de délai a une valeur par défaut, généralement définie sur `NONE` (infini). Toutefois, la durée maximale de toute exécution d'activité est limitée à un an.

Vous définissez ces valeurs par défaut lors de l'enregistrement du type d'activité, mais vous pouvez les remplacer par d'autres valeurs lorsque vous [planifiez](#) la tâche d'activité. Lorsque l'un de ces délais se produit, Amazon SWF ajoute [un](#) événement de `ActivityTaskTimedOut` type à l'historique du flux de travail. La valeur d'attribut `timeoutType` de cet événement indique quel délai a expiré. Pour chaque délai, la valeur `timeoutType` est indiquée entre parenthèses. Les attributs d'événement incluront également IDs les événements correspondant au moment où la tâche d'activité a été planifiée (`scheduledEventId`) et à son lancement (`startedEventId`). Outre l'ajout de l'événement, Amazon SWF planifie également une nouvelle tâche de décision pour avertir le décideur que le délai imparti s'est écoulé.

## Points de terminaison Amazon Simple Workflow Service

Une liste des [régions et points de terminaison Amazon SWF actuels est fournie dans le Référence générale d'Amazon Web Services, ainsi que les points](#) de terminaison des autres services.

Les domaines Amazon SWF et tous les flux de travail et activités associés doivent exister au sein de la même région pour communiquer entre eux. En outre, les domaines enregistrés, les flux de travail et les activités d'une région n'existent pas dans les autres régions. Par exemple, si vous créez un domaine nommé « `MySampleDomain` » à la fois dans `us-east-1` et dans `us-west-2`, ils existent en tant que domaines distincts : aucun des flux de travail, listes de tâches, activités ou données associés à vos domaines n'est partagé entre les régions.

Si vous utilisez d'autres AWS ressources dans vos flux de travail, telles que EC2 des instances Amazon, celles-ci doivent également exister dans la même région que vos ressources Amazon SWF. Les seules exceptions à cette règle concernent les services couvrant plusieurs régions,

tels qu'Amazon S3 et IAM. Vous pouvez accéder à ces services à partir des flux de travail, indépendamment de la région qui les prend en charge.

## Documentation supplémentaire pour le service Amazon Simple Workflow

En plus de ce guide du développeur, la documentation suivante peut également vous être utile.

### Référence de l'API Amazon Simple Workflow Service

La [référence d'API Amazon Simple Workflow Service](#) fournit des informations détaillées sur l'API HTTP Amazon SWF, notamment les actions, les structures de demande et de réponse et les codes d'erreur.

### AWS Flow Framework Documentation

[AWS Flow Framework](#) Il s'agit d'un framework de programmation qui simplifie le processus de mise en œuvre d'applications asynchrones distribuées qui utilisent Amazon SWF pour gérer leurs flux de travail et leurs activités, afin que vous puissiez vous concentrer sur la mise en œuvre de votre logique de flux de travail.

Chacun AWS Flow Framework est conçu pour fonctionner de manière idiomatique dans la langue pour laquelle il a été conçu, afin que vous puissiez travailler naturellement avec la langue de votre choix afin de mettre en œuvre des flux de travail avec tous les avantages d'Amazon SWF.

Il existe un framework AWS Flow pour Java. Le [guide du développeur AWS Flow Framework pour Java](#) fournit des informations sur la façon d'obtenir, de configurer et d'utiliser le AWS Flow Framework pour Java.

### AWS Documentation du SDK

Les kits de développement AWS logiciel (SDKs) permettent d'accéder à Amazon SWF dans de nombreux langages de programmation différents. Ils SDKs suivent de près l'API HTTP, mais fournissent également des interfaces de programmation spécifiques au langage pour certaines fonctionnalités d'Amazon SWF. Pour plus d'informations sur chaque kit SDK, cliquez sur les liens ci-dessous.

**Note**

Seuls SDKs ceux qui étaient compatibles avec Amazon SWF au moment de la rédaction de cet article sont répertoriés ici. Pour obtenir la liste complète des outils disponibles AWS SDKs, rendez-vous sur la page [Outils pour Amazon Web Services](#).

## Java

AWS SDK pour Java fournit une API Java pour les services AWS d'infrastructure.

Pour afficher la documentation disponible, consultez la page [Documentation AWS SDK pour Java](#). Vous pouvez également accéder directement aux sections Amazon SWF de la référence du SDK en suivant ces liens :

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

## JavaScript

AWS SDK pour JavaScript Cela permet aux développeurs de créer des bibliothèques ou des applications qui utilisent des AWS services à l'aide d'une easy-to-use API simple disponible à la fois dans le navigateur ou dans les applications Node.js sur le serveur.

Pour afficher la documentation disponible, consultez la page [Documentation AWS SDK pour JavaScript](#). Vous pouvez également accéder directement à la section Amazon SWF dans la référence du SDK en suivant ce lien :

- [Class: AWS.SimpleWorkflow](#)

## .NET

AWS SDK pour .NET Il s'agit d'un package unique téléchargeable qui inclut des modèles de projet Visual Studio, la bibliothèque AWS .NET, des exemples de code C# et de la documentation. AWS SDK pour .NET Cela permet aux développeurs Windows de créer plus facilement des applications .NET pour Amazon SWF et d'autres services.

Pour afficher la documentation disponible, consultez la page [Documentation AWS SDK pour .NET](#). Vous pouvez également accéder directement aux sections Amazon SWF de la référence du SDK en suivant ces liens :

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

## PHP

AWS SDK for PHP fournit une interface de programmation PHP à Amazon SWF.

Pour afficher la documentation disponible, consultez la page [Documentation AWS SDK for PHP](#). Vous pouvez également accéder directement à la section Amazon SWF dans la référence du SDK en suivant ce lien :

- [Class: SwfClient](#)

## Python

AWS SDK pour Python (Boto) fournit une interface de programmation Python à Amazon SWF.

Pour consulter la documentation disponible, consultez la page [boto : A Python interface to Amazon Web Services](#). Vous pouvez également accéder directement aux sections Amazon SWF de la documentation en suivant ces liens :

- [Tutoriel Amazon SWF](#)
- [Référence Amazon SWF](#)

## Ruby

AWS SDK pour Ruby fournit une interface de programmation Ruby à Amazon SWF.

Pour afficher la documentation disponible, consultez la page [Documentation AWS SDK pour Ruby](#). Vous pouvez également accéder directement à la section Amazon SWF dans la référence du SDK en suivant ce lien :

- [Classe : AWS::Simple Workflow](#)

## AWS CLI Documentation

Le AWS Command Line Interface (AWS CLI) est un outil unifié permettant de gérer vos AWS services. Avec un seul outil à télécharger et à configurer, vous pouvez contrôler plusieurs AWS services à partir de la ligne de commande et les automatiser par le biais de scripts.

Pour plus d'informations à ce sujet AWS CLI, consultez la [AWS Command Line Interface](#) page.

Pour un aperçu des commandes disponibles pour Amazon SWF, consultez [swf](#) dans le manuel Command Reference.AWS CLI

# Ressources Web pour le service Amazon Simple Workflow

Il existe un certain nombre de ressources Web que vous pouvez utiliser pour en savoir plus sur Amazon SWF ou pour obtenir de l'aide concernant l'utilisation du service et le développement de flux de travail.

## Forum Amazon SWF

Le forum Amazon SWF vous permet de communiquer avec d'autres développeurs Amazon SWF et les membres de l'équipe de développement Amazon SWF d'Amazon pour poser des questions et obtenir des réponses.

Vous pouvez consulter le forum à l'adresse suivante : [Forum : Amazon Simple Workflow Service](#).

## FAQ sur Amazon SWF

La FAQ Amazon SWF fournit des réponses aux questions fréquemment posées sur Amazon SWF, notamment un aperçu des cas d'utilisation courants, des différences entre Amazon SWF et d'autres services, et bien plus encore.

Vous pouvez accéder à la FAQ ici : FAQ [Amazon SWF](#).

## Vidéos Amazon SWF

La chaîne [Amazon Web Services](#) YouTube propose une formation vidéo pour tous les services Web d'Amazon, y compris Amazon SWF. Pour obtenir la liste complète des vidéos liées à Amazon SWF, utilisez la requête suivante : [Simple Workflow in Amazon Web Services](#)

## Options de migration pour Ruby Flow

Le AWS Flow Framework for Ruby n'est plus en cours de développement actif. Le code existant continuera de fonctionner indéfiniment, mais il n'y aura aucune nouvelle fonction ni version. Cette rubrique traite des options d'utilisation et de migration permettant de continuer à travailler avec Amazon SWF, ainsi que des informations sur la façon de migrer vers Step Functions.

Option	Description
<a href="#">Continuez à utiliser le framework Ruby Flow</a>	Pour l'instant, le Ruby Flow Framework continuera de fonctionner. Si vous ne faites rien, votre code continuera de fonctionner de la même

Option	Description
	manière. Prévoyez de migrer hors de AWS Flow Framework for Ruby dans un futur proche.
<a href="#">Migrer vers le framework Java Flow</a>	Le Java Flow Framework est toujours en cours de développement et continuera de recevoir de nouvelles fonctionnalités et mises à jour.
<a href="#">Migrer vers Step Functions</a>	Step Functions permet de coordonner les composants des applications distribuées à l'aide de flux de travail visuels contrôlés par une machine à états.
<a href="#">Utilisez l'API SWF directement</a> , sans le Flow Framework	Vous pouvez continuer à travailler dans Ruby et à utiliser l'API SWF directement, au lieu d'AWS Flow Framework pour Ruby.

L'avantage du Flow Framework, que ce soit pour Ruby ou Java, est qu'il vous permet de vous concentrer sur la logique de votre flux de travail. L'infrastructure gère la plupart des informations de communication et de coordination, et la complexité est abstraite. Vous pouvez continuer à avoir le même niveau d'abstraction en migrant vers le Java Flow Framework, ou vous pouvez interagir directement avec le SDK Amazon SWF.

## Continuer à utiliser Flow Framework pour Ruby

Le AWS Flow Framework for Ruby continuera à fonctionner comme il le fait actuellement à court terme. Si vous avez écrit des flux de travail dans AWS Flow Framework for Ruby, ils continueront de fonctionner. Sans mises à jour, support ou correctifs de sécurité, il est préférable d'avoir un plan ferme AWS Flow Framework pour migrer hors de Ruby dans un futur proche.

## Migrer vers Flow Framework pour Java

Le AWS Flow Framework pour Java restera en développement actif. Sur le AWS Flow Framework plan conceptuel, Java est similaire à AWS Flow Framework Ruby : vous pouvez toujours vous concentrer sur la logique de votre flux de travail, et le framework vous aidera à gérer la logique de votre décideur et facilitera la gestion d'autres aspects d'Amazon SWF.

- [AWS Flow Framework pour Java](#)
- [AWS Flow Framework pour la référence de l'API Java](#)

## Migrer vers Step Functions

AWS Step Functions fournit un service similaire à Amazon SWF, mais dans lequel la logique de votre flux de travail est contrôlée par une machine à états. Step Functions vous permet de coordonner les composants des applications distribuées et des microservices à l'aide de flux de travail visuels. Vous développez les applications à partir de composants individuels qui exécutent chacun une fonction discrète, ou tâche, vous permettant de mettre à l'échelle et de modifier les applications rapidement. Step Functions fournit un moyen fiable de coordonner les composants et de parcourir les fonctions de votre application. Une console graphique permet de visualiser les composants de votre application en une série d'étapes. Il déclenche et suit automatiquement chaque étape, et réessaie en cas d'erreur, afin que votre application s'exécute dans l'ordre et comme prévu, à chaque fois. Step Functions enregistre l'état de chaque étape. Ainsi, en cas de problème, vous pouvez diagnostiquer et corriger rapidement les problèmes.

Dans Step Functions, vous gérez la coordination de vos tâches à l'aide d'une machine à états, écrite en JSON déclaratif, définie à l'aide du [langage Amazon States](#). En utilisant une machine à états, vous n'avez pas besoin d'écrire et de gérer un programme de décision pour contrôler la logique de votre application. Step Functions propose une approche intuitive, productive et agile pour coordonner les composants de l'application à l'aide de flux de travail visuels. Vous devriez envisager de l'utiliser AWS Step Functions pour toutes vos nouvelles applications, et Step Functions fournit une excellente plateforme de migration pour les flux de travail que vous avez actuellement implémentés AWS Flow Framework dans Ruby.

Pour vous aider à migrer vos tâches vers Step Functions, tout en continuant à tirer parti de vos compétences en langage Ruby, Step Functions fournit un exemple de gestionnaire d'activité Ruby. Cet exemple utilise les meilleures pratiques pour implémenter un travailleur d'activité et peut être utilisé comme modèle pour migrer votre logique de tâches vers Step Functions. Pour plus d'informations, consultez la rubrique [Example Activity Worker in Ruby](#) du [Guide du AWS Step Functions développeur](#).

### Note

Pour de nombreux clients, migrer vers Step Functions depuis AWS Flow Framework for Ruby est la meilleure option. Toutefois, si vous souhaitez que des signaux interviennent dans vos processus, ou si vous devez lancer des processus enfants renvoyant un résultat à un parent, pensez à utiliser directement l'API Amazon SWF ou à migrer vers l' AWS Flow Framework API pour Java.

Pour plus d'informations sur AWS Step Functions, voir :

- [AWS Step Functions Manuel du développeur](#)
- [AWS Step Functions API Reference](#)
- [AWS Step Functions Référence de ligne de commande](#)

## Utilisez directement l'API Amazon SWF

Bien que AWS Flow Framework for Ruby gère une partie de la complexité d'Amazon SWF, vous pouvez également utiliser directement l'API Amazon SWF. Cela vous permet de créer des flux de travail là où vous contrôlez intégralement l'implémentation et la coordination des tâches, sans vous soucier des complexités sous-jacentes telles que le suivi de leur progression ou la tenue à jour de leur état.

- [Guide du développeur du service Amazon Simple Workflow](#)
- [Référence de l'API Amazon Simple Workflow Service](#)

## Historique du document

Le tableau suivant décrit les modifications importantes apportées à la documentation depuis la dernière version du manuel Amazon Simple Workflow Service Developer Guide.

Modification	Description	Date de modification
Mise à jour relative à la documentation uniquement	Amazon SWF inclut désormais une section sur les notifications AWS utilisateur, Service AWS qui sert d'emplacement central pour vos AWS notifications dans le. AWS Management Console Pour de plus amples informations, veuillez consulter <a href="#">Utilisation Notifications des utilisateurs AWS avec Amazon Simple Workflow Service</a> .	4 mai 2023
Mettre à jour	Amazon SWF propose désormais une nouvelle expérience de console pour gérer les flux de travail SWF et leurs actions liées à l'exécution. Pour plus d'informations, consultez les <a href="#">didacticiels de la console Amazon SWF</a> .	12 septembre 2022
Mettre à jour	Mise à jour de la <a href="#">Quotas relatifs à l'exécution des tâches</a> section pour inclure Maximum tasks scheduled per second et de la <a href="#">Métriques Amazon SWF pour CloudWatch</a> page pour inclure des informations sur <a href="#">l'utilisation de noms de ressources non ASCII</a> avec. CloudWatch	12 mai 2021
Nouvelle fonctionnalité	Amazon Simple Workflow Service est désormais compatible avec Amazon EventBridge. Pour plus d'informations, consultez : <ul style="list-style-type: none"><li>• <a href="#">EventBridge pour Amazon SWF</a></li><li>• <a href="#">EventBridge Guide de l'utilisateur</a></li></ul>	18 décembre 2020
Nouvelle fonctionnalité	Amazon Simple Workflow Service prend en charge les autorisations IAM à l'aide de balises. Pour plus d'informations, consultez les rubriques suivantes. <ul style="list-style-type: none"><li>• <a href="#">Balises dans Amazon SWF</a></li></ul>	20 juin 2019

Modification	Description	Date de modification
	<ul style="list-style-type: none"> <li>• <a href="#">Gérer les balises</a></li> <li>• <a href="#">Étiqueter les exécutions de workflows</a></li> <li>• <a href="#">Contrôlez l'accès aux domaines à l'aide de tags</a></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>	
Nouvelle fonctionnalité	Amazon Simple Workflow Service est désormais disponible dans la région Europe (Stockholm).	12 décembre 2018
Mettre à jour	Amélioration de la rubrique Amazon Simple Workflow Service sur CloudTrail l'intégration. Consultez <a href="#">Enregistrement des appels d'API avec AWS CloudTrail</a> .	7 août 2018
Mettre à jour	Ajout d'informations sur la nouvelle PendingTasks métrique pour CloudWatch. Pour de plus amples informations, veuillez consulter <a href="#">Métriques Amazon SWF</a> .	18 juin 2018
Mettre à jour	Mise en évidence de la syntaxe améliorée dans les exemples de code.	29 mars 2018
Mettre à jour	Ajout d'une rubrique décrivant les options disponibles pour les utilisateurs d'AWS Flow pour Ruby afin de quitter cette plateforme. Pour de plus amples informations, veuillez consulter <a href="#">Options de migration pour Ruby Flow</a> .	9 mars 2018
Mettre à jour	Amélioration de la navigation de la rubrique sur les concepts avancés. Consultez <a href="#">Concepts de flux de travail avancés dans Amazon SWF</a> .	19 février 2018
Mettre à jour	Documentation améliorée CloudWatch des métriques en ajoutant des informations statistiques valides. Consultez <a href="#">Métriques Amazon SWF pour CloudWatch</a> .	4 décembre 2017

Modification	Description	Date de modification
Mettre à jour	Modification de la tables des matières pour améliorer la structure du document. Ajout de nouvelles informations sur <a href="#">Métriques des événements d'API et de décision</a> .	9 novembre 2017
Mettre à jour	La section <a href="#">Quotas Amazon SWF</a> a été mise à jour pour inclure les limitations pour toutes les régions.	18 octobre 2017
Mettre à jour	<code>task_list</code> a été modifié en <code>workflowId</code> dans le <a href="#">Commencer à utiliser Amazon SWF</a> pour éviter toute confusion avec <code>activity_list</code> .	25 juillet 2017
Mettre à jour	Les exemples de code ont été nettoyés dans ce guide.	5 juin 2017
Mettre à jour	Simplification et amélioration de l'organisation et du contenu de ce guide.	19 mai 2017
Mettre à jour	Mises à jour et correctifs de liens.	16 mai 2017
Mettre à jour	Mises à jour et correctifs de liens.	1 octobre 2016
Assistance aux tâches Lambda	Vous pouvez spécifier des tâches Lambda en plus des tâches d'activité traditionnelles dans vos flux de travail. Pour de plus amples informations, veuillez consulter <a href="#">AWS Lambda tâches dans Amazon SWF</a> .	21 juillet 2015
Prise en charge de la définition de la priorité des tâches	Amazon SWF inclut désormais la prise en charge de la définition de la priorité des tâches d'une liste de tâches et essaiera de fournir les tâches les plus prioritaires avant les tâches les moins prioritaires. Pour découvrir comment définir la priorité des tâches pour les flux de travail et les activités, consultez la section <a href="#">Définition de la priorité des tâches dans Amazon SWF</a> .	17 décembre 2014
Mettre à jour	Ajout d'une nouvelle rubrique qui décrit comment enregistrer les appels d'API Amazon SWF à l'aide CloudTrail de :. <a href="#">Enregistrement des appels d'API avec AWS CloudTrail</a>	8 mai 2014

Modification	Description	Date de modification
Mettre à jour	Deux nouvelles rubriques relatives aux CloudWatch métriques pour Amazon SWF ont été ajoutées : <a href="#">Métriques Amazon SWF pour CloudWatch</a> , qui fournit une liste et des descriptions des métriques prises en charge <a href="#">Affichage des métriques Amazon SWF pour CloudWatch l'utilisation du AWS Management Console</a> , et qui fournit des informations sur la façon de consulter les métriques et de définir des alarmes avec le. AWS Management Console	28 avril 2014
Mettre à jour	Une section a été ajoutée : <a href="#">Ressources supplémentaires et informations de référence pour Amazon SWF</a> . Cette section fournit des informations de référence sur les services et fournit des informations sur la documentation supplémentaire, les exemples, le code et d'autres ressources Web pour les développeurs Amazon SWF.	19 mars 2014
Mettre à jour	Un didacticiel sur le flux de travail a été ajouté. Consultez <a href="#">Commencer à utiliser Amazon SWF</a> .	le 25 octobre 2013
Mettre à jour	Ajout <a href="#">d'informations et d'exemple AWS CLI</a> .	26 août 2013
Mettre à jour	Mises à jour et correctifs.	1er août 2013
Mettre à jour	Le document a été mis à jour pour décrire comment utiliser IAM pour le contrôle d'accès.	22 février 2013
Première version	Il s'agit de la première version du guide du développeur Amazon Simple Workflow Service.	16 octobre 2012

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.