Guide de l'utilisateur pour la version 1.17.0

AWS SimSpace Weaver

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

AWS SimSpace Weaver: Guide de l'utilisateur pour la version 1.17.0

Copyright © 2025 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

..... ix
Qu'est-ce que c'est SimSpace Weaver ? 1
Concepts clés 1
Comment SimSpace Weaver fonctionne 2
Comment utilisez-vous SimSpace Weaver 5
Schéma de simulation 6
Travailleurs et unités de ressources 6
Horloge de simulation 7
Partitions 7
State Fabric 7
Entités 8
Applications 8
Exemples de cas d’utilisation 11
AWS SimSpace Weaver fin du support 13
Configuration 14
Configurez votre compte 14
Inscrivez-vous pour un Compte AWS 14
Création d’un utilisateur doté d’un accès administratif 15
Ajouter des autorisations d'utilisation SimSpace Weaver 16
Configurez votre environnement local 18
AL2 dans Docker 18
AL2 dans WSL 20
Utilisation de logiciels sous licence 24
Premiers pas 25
Tutoriel de démarrage rapide 25
Étape 1 : activer la journalisation (facultatif) 26
Étape 2 : Démarrage rapide avec le client de console (option 1) 26
Étape 2 : Démarrage rapide avec le client Unreal Engine (option 2) 27
Arrêtez et supprimez votre simulation 27
Résolution des problèmes 27
Tutoriel détaillé 28
Étape 1 : activer la journalisation (facultatif) 29
Étape 2 : Lancez votre simulation 29
Étape 3 : Vérifiez les journaux (facultatif) 36

Étape 4 : Afficher votre simulation 38
Étape 5 : Arrêtez et supprimez votre simulation 39
Résolution des problèmes 40
Travailler avec SimSpace Weaver 41
Configuration de votre simulation 41
Paramètres de configuration de simulation 42
Version de SDK 43
Propriétés de simulation 43
Workers 44
Horloge 45
Stratégies de partitionnement 48
Domaines 49
Durée maximum 60
Valeur maximale 60
Valeur par défaut 60
Valeur minimale 60
Lancer une simulation à l'aide de la console 60
État d'une simulation qui atteint sa durée maximale 61
Développement d'applications 61
Applications spatiales 62
Apps personnalisées 62
Développement d'applications clientes 63
Obtenez l'adresse IP et le numéro de port 64
Lancement du client de visualisation Unreal Engine 67
Résolution des problèmes 68
Développement local 69
Étape 1 : Lancez votre simulation locale 69
Étape 2 : Afficher votre simulation locale 71
Étape 3 : Arrêtez votre simulation locale (facultatif sous Windows) 72
Résolution des problèmes liés au développement local 73
SimSpace Weaver SDK de l'application 73
Les méthodes d'API renvoient un Result 74
Interaction avec le SDK de l'application au plus haut niveau 75
Gestion des simulations 75
Abonnements 78
Entités 79

Événements organisés par les entités 91
Result et gestion des erreurs 98
Génériques et types de domaines 100
Opérations diverses du SDK de l'application 100
SimSpace Weaver framework de démonstration 103
Travailler avec des quotas 104
Déterminer les limites d'une application 105
Obtenez la quantité de ressources utilisées par une application 105
Réinitialiser les métriques 106
Dépassement d'une limite 107
À court de mémoire 107
Bonnes pratiques 107
Simulations de débogage 108
Utiliser SimSpace Weaver Local et regardez la sortie de la console 108
Consultez vos journaux dans Amazon CloudWatch Logs 109
Utiliser describe Appels d’API 109
Connect un client 110
Débogage de simulations locales 110
Conteneurs personnalisés 111
Création d'un conteneur personnalisé 112
Modifier un projet pour utiliser un conteneur personnalisé 113
FAQ 116
Résolution des problèmes 116
Travail avec Python 117
Création d'un projet Python 118
Démarrage d'une simulation Python 120
Exemple de client Python 120
FAQ 121
Résolution des problèmes 121
Support pour d'autres moteurs 123
Unity 123
Unreal Engine 123
Utilisation de logiciels sous licence 124
Gérer les ressources avec AWS CloudFormation 124
Instantanés 127
Instantanés 127

Cas d'utilisation pour les instantanés 128
SimSpace Weaver Console 129
AWS CLI 131
FAQ 133
Messagerie 134
Cas d'utilisation de la messagerie 135
Utilisation de la messagerie APIs 135
Quand utiliser la messagerie 143
Conseils relatifs à l'utilisation de la messagerie 146
Erreurs de messagerie et résolution des problèmes 148
Bonnes pratiques 150
Configurer des alarmes de facturation 150
Utiliser SimSpace Weaver Local 150
Arrêtez les simulations dont vous n'avez pas besoin 151
Supprimer les ressources dont vous n'avez pas besoin 151
Disposer de sauvegardes 152
Sécurité 153
Protection des données 154
Chiffrement au repos 155
Chiffrement en transit 155
Confidentialité du trafic inter-réseaux 156
Gestion de l’identité et des accès 156
Public ciblé 157
Authentification par des identités 157
Gestion des accès à l’aide de politiques 161
Comment AWS SimSpace Weaver fonctionne avec IAM 164
Exemples de politiques basées sur l’identité 172
Des autorisations SimSpace Weaver créées pour vous 176
Prévention du problème de l’adjoint confus entre services 178
Résolution des problèmes 181
Enregistrement et surveillance des événements de sécurité 184
Validation de la conformité 185
Résilience 186
Sécurité de l’infrastructure 187
Modèle de sécurité de connectivité réseau 187
Analyse de la configuration et des vulnérabilités 188

Bonnes pratiques de sécurité 189
Chiffrez les communications entre vos applications et leurs clients 189
Sauvegardez régulièrement l'état de votre simulation 189
Gérez vos applications et SDKs 189
Journalisation et surveillance 191
Se connecte CloudWatch 191
Accès à vos SimSpace Weaver journaux 191
SimSpace Weaver journaux 192
Surveillance avec CloudWatch 194
SimSpace Weaver statistiques au niveau du compte 195
CloudTrail journaux 195
SimSpace Weaver informations dans CloudTrail 196
Comprendre les entrées du fichier SimSpace Weaver journal 197
Points de terminaison et quotas de service 199
Points de terminaison de service 199
Quotas de service 200
Quotas de messagerie 203
Tarifs des horloges 203
Quotas de service pour SimSpace Weaver Local 203
Résolution des problèmes 205
AssumeRoleAccessDenied 205
InvalidBucketName 207
ServiceQuotaExceededException 208
TooManyBuckets 208
Autorisation refusée lors du démarrage de la simulation 209
Problèmes liés au temps d'utilisation Docker 210
Le client de console ne parvient pas à se connecter 210
Non `simspaceweaver` dans le AWS CLI 212
Référence de schéma 214
Exemple de schéma complet 214
Format du schéma 216
Version de SDK 217
Propriétés de simulation 217
Workers 219
Horloge 220
Stratégies de partitionnement 220

Domaines 222
Contraintes de placement 232
Références d'API 234
SimSpace Weaver versions 235
Dernière version 235
Comment trouver votre version actuelle 235
Téléchargez la dernière version 235
Résolution des problèmes liés aux téléchargements du SDK d'applications 236
Installez la dernière version 237
Versions du service 237
1.17.0 254
Changements majeurs pour la version 1.17.0 254
Mettre à jour un projet vers la version 1.17.0 256
Questions fréquemment posées sur la version 1.17.0 257
1.15.1 257
Mettre à jour un projet Python existant vers la version 1.15.1 257
Résolution des problèmes liés à la version 1.15.1 258
Questions fréquemment posées sur la version 1.15.1 258
Historique de la documentation 260
Glossaire 270

Avis de fin de support : le 20 mai 2026, AWS le support de AWS SimSpace Weaver. Après le 20 mai 2026, vous ne pourrez plus accéder à la SimSpace Weaver console ni aux SimSpace Weaver ressources. Pour plus d'informations, voir AWS SimSpace Weaver fin du support.

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.

Qu'est-ce que c'est AWS SimSpace Weaver ?

AWS SimSpace Weaver est un service que vous pouvez utiliser pour créer et exécuter des simulations spatiales à grande échelle dans le AWS Cloud. Par exemple, vous pouvez créer des simulations de foule, de grands environnements réels et des expériences immersives et interactives.

Vous pouvez SimSpace Weaver ainsi répartir les charges de travail de simulation sur plusieurs instances Amazon Elastic Compute Cloud (Amazon EC2). SimSpace Weaver déploie l' AWS infrastructure sous-jacente pour vous et gère la gestion des données de simulation et la communication réseau entre les EC2 instances Amazon exécutant votre simulation.

Concepts clés pour SimSpace Weaver

Une simulation ou un jeu est limité par l'ordinateur qui l'exécute. À mesure que la taille et la complexité de votre monde virtuel augmentent, les performances de traitement commencent à se dégrader. Les calculs prennent plus de temps, les systèmes manquent de mémoire et les fréquences d'images des clients chutent. Pour les simulations qui ne nécessitent pas de performances en temps réel, cela ne peut être que gênant. Il peut également s'agir d'une situation critique pour l'entreprise dans laquelle l'augmentation des délais de traitement entraîne une augmentation des coûts. Si votre simulation ou votre jeu nécessite des performances en temps réel, la dégradation des performances est certainement un problème.

Une solution courante pour une simulation qui atteint une limite de performance consiste à simplifier la simulation. Les jeux en ligne comptant de nombreux utilisateurs résolvent souvent les problèmes de mise à l'échelle en créant des copies de leur monde virtuel sur différents serveurs et en répartissant les utilisateurs entre eux.

SimSpace Weaver résout le problème de mise à l'échelle en divisant votre monde virtuel dans l'espace et en répartissant les éléments entre un cluster d'instances de calcul exécutées dans le AWS Cloud. Les instances de calcul fonctionnent ensemble pour traiter l'ensemble du monde de simulation en parallèle. Votre monde de simulation apparaît comme un espace intégré unique à tout ce qu'il contient et à tous les clients qui s'y connectent. Il n'est plus nécessaire de simplifier une simulation en raison d'une limite de performance matérielle. Vous pouvez plutôt ajouter de la capacité de calcul dans le cloud.

Rubriques

- Comment SimSpace Weaver fonctionne

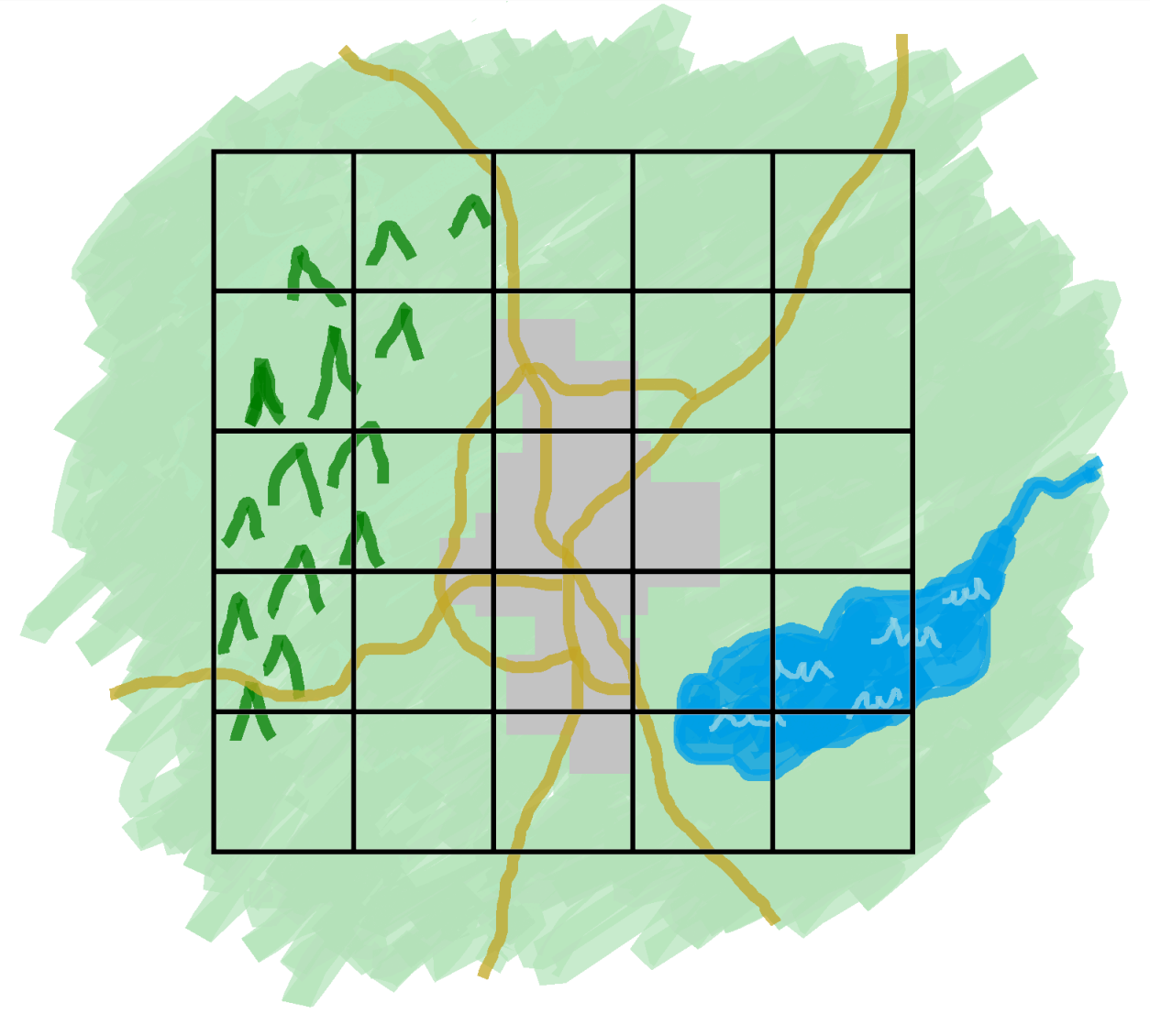
- Comment utilisez-vous SimSpace Weaver
- Schéma de simulation
- Travailleurs et unités de ressources
- Horloge de simulation
- Partitions
- State Fabric
- Entités
- Applications

Comment SimSpace Weaver fonctionne

Votre simulation consiste en un monde contenant des objets. Certains objets (tels que les personnes et les véhicules) se déplacent et font des choses. Les autres objets (tels que les arbres et les bâtiments) sont statiques. Dans SimSpace Weaver, une entité est un objet dans votre monde de simulation.

Vous définissez les limites de votre monde de simulation et vous le divisez en une grille. Au lieu de créer une logique de simulation qui fonctionne sur l'ensemble de la grille, vous créez une logique de simulation qui fonctionne sur une cellule de la grille. Dans SimSpace Weaver, une application spatiale est un programme que vous écrivez et qui implémente la logique de simulation pour une cellule de votre grille. Cela inclut la logique de toutes les entités de cette cellule. La zone de propriété d'une application spatiale est la cellule de grille contrôlée par l'application spatiale.

> **Note**
>
> Dans SimSpace Weaver, le terme « application » peut faire référence au code d'une application ou à une instance en cours d'exécution de ce code.



Votre monde de simulation divisé en une grille

Vous divisez votre monde de simulation en une grille. Chaque application spatiale implémente une logique de simulation pour une seule cellule de cette grille.

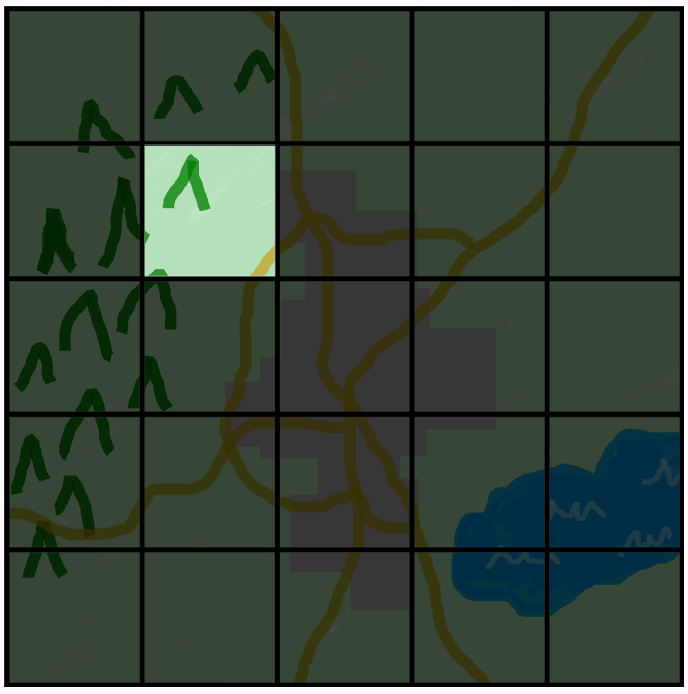
SimSpace Weaver exécute une instance du code de votre application spatiale pour chaque cellule de votre grille. Toutes les instances d'applications spatiales s'exécutent en parallèle. Il SimSpace Weaver divise essentiellement votre simulation globale en plusieurs simulations plus petites. Chacune des plus petites simulations gère une partie du monde global de la simulation. SimSpace Weaver peut distribuer et exécuter ces petites simulations sur plusieurs instances Amazon Elastic

Compute Cloud (Amazon EC2) (appelées workers) dans le AWS Cloud. Un seul utilisateur peut exécuter plusieurs applications spatiales.

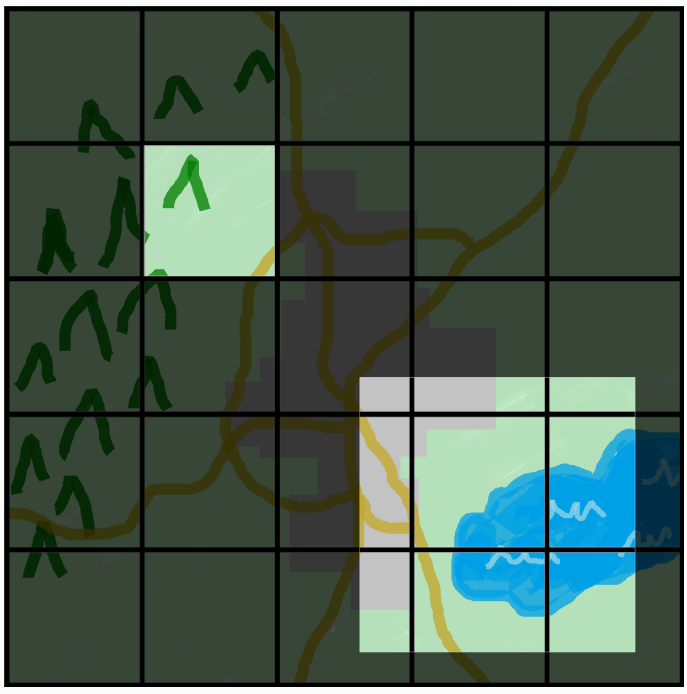
Les entités peuvent se déplacer dans le monde de la simulation. Si une entité entre dans la zone de propriété d'une autre application spatiale (une autre cellule de la grille), le propriétaire de l'application spatiale de la nouvelle zone prend le contrôle de l'entité. Si votre simulation s'exécute sur plusieurs travailleurs, une entité peut passer du contrôle d'une application spatiale sur un travailleur à une application spatiale sur un autre travailleur. Lorsqu'une entité se déplace vers un autre travailleur, SimSpace Weaver gère la communication réseau sous-jacente.

Abonnements

La vision du monde d'une application spatiale est son propre domaine de propriété. Pour découvrir ce qui se passe dans une autre partie du monde de la simulation, l'application spatiale crée un abonnement. La zone d'abonnement est un sous-ensemble de la zone globale du monde de simulation. Une zone d'abonnement peut inclure des parties de plusieurs zones de propriété, y compris la propre zone de propriété de l'application spatiale. SimSpace Weaver informe l'application spatiale de tous les événements d'entité (par exemple, entrée, sortie, création, mise à jour et suppression) qui se produisent dans la zone d'abonnement.



La vision du monde d'une application spatiale



Vue d'une application spatiale avec une zone d'abonnement ajoutée

La vision du monde d'une application spatiale correspond à sa zone de propriété , qui est une cellule de la grille mondiale.	Une application spatiale utilise un abonnement pour découvrir ce qui se passe dans une autre partie du monde de la simulation. La zone d'abonnement peut contenir plusieurs cellules de grille et des parties de cellules.

Par exemple, une application qui simule des entités interagissant physiquement peut avoir besoin de connaître les entités situées juste au-delà des limites spatiales de sa zone de propriété. Pour ce faire, l'application peut s'abonner à des zones situées en bordure de sa zone de propriété. Après avoir créé l'abonnement, l'application reçoit des notifications concernant les événements relatifs aux entités dans ces zones et peut lire les entités. Un autre exemple est un véhicule autonome qui doit voir toutes les entités à 200 mètres devant lui, quelle que soit l'application propriétaire de la zone. L'application pour le véhicule peut créer un abonnement avec un filtre servant de cadre de délimitation aligné sur l'axe (AABB) qui couvre la zone visible.

Vous pouvez créer une logique de simulation qui ne soit pas chargée de gérer les aspects spatiaux de votre simulation. Une application personnalisée est un programme exécutable qui s'exécute sur un seul outil de travail. Vous contrôlez le cycle de vie (démarrage et arrêt) d'une application personnalisée. Les clients de simulation peuvent se connecter à une application personnalisée pour visualiser ou interagir avec la simulation. Vous pouvez également créer une application de service qui s'exécute sur tous les travailleurs. SimSpace Weaver démarre une instance de votre application de service sur chaque collaborateur qui exécute votre simulation.

Les applications personnalisées et les applications de service créent des abonnements pour en savoir plus sur les événements des entités et lire les entités. Ces applications ne disposent pas de zones de propriété car elles ne sont pas spatiales. L'utilisation d'un abonnement est le seul moyen pour eux de découvrir ce qui se passe dans le monde de la simulation.

Comment utilisez-vous SimSpace Weaver

Lorsque vous l'utilisez SimSpace Weaver, voici les principales étapes que vous devez suivre :

1. Écrivez et construisez C++ applications qui intègrent le SDK de SimSpace Weaver l'application.
   a. Vos applications effectuent des appels d'API pour interagir avec l'état de la simulation.
2. Écrivez des clients qui visualisent et interagissent avec votre simulation via certaines applications.
3. Configurez votre simulation dans un fichier texte.
4. Téléchargez vos packages d'applications et votre configuration de simulation sur le service.

5. Lancez votre simulation.
6. Démarrez et arrêtez vos applications personnalisées selon vos besoins.
7. Connectez les clients à vos applications personnalisées ou de service pour visualiser ou interagir avec la simulation.
8. Consultez vos journaux de simulation dans Amazon CloudWatch Logs.
9. Arrêtez votre simulation.
10. Nettoyez votre simulation.

Schéma de simulation

Le schéma (ou schéma) de simulation est YAML-fichier texte formaté contenant les informations de configuration pour votre simulation. SimSpace Weaver utilise votre schéma lorsqu'il lance une simulation. Le package distribuable du SDK de l' SimSpace Weaver application inclut un schéma pour un exemple de projet. Vous pouvez l'utiliser comme point de départ pour votre propre schéma. Pour plus d'informations sur le schéma de simulation, consultezSimSpace Weaver référence du schéma de simulation.

Travailleurs et unités de ressources

Un worker est une EC2 instance Amazon qui exécute votre simulation. Vous spécifiez un type de travailleur dans votre schéma de simulation. SimSpace Weaver fait correspondre votre type de travailleur à un type d' EC2 instance Amazon spécifique utilisé par le service. SimSpace Weaver démarre et arrête vos travailleurs pour vous, et gère les communications réseau entre les travailleurs. SimSpace Weaver lance un ensemble de travailleurs pour chaque simulation. Les différentes simulations font appel à des travailleurs différents.

La capacité de calcul (processeur et mémoire) disponible sur un travailleur est divisée en unités logiques appelées unités de ressources de calcul (ou unités de ressources). Une unité de ressource représente une quantité fixe de capacité de processeur et de mémoire.

> **Note**
>
> Nous avons précédemment désigné une unité de ressource de calcul par le terme « slot ». Vous pouvez toujours voir ce terme précédent dans notre documentation.

Horloge de simulation

Chaque simulation possède sa propre horloge. Vous démarrez et arrêtez l'horloge à l'aide d'appels d'API ou de la SimSpace Weaver console. La simulation est mise à jour uniquement lorsque l'horloge tourne. Toutes les opérations de la simulation se déroulent dans des segments temporels appelés ticks. L'horloge annonce l'heure de début de chaque coche à tous les travailleurs.

La fréquence d'horloge (ou fréquence de tic-tac) est le nombre de ticks par seconde (hertz ou Hz) annoncés par l'horloge. La fréquence d'horloge souhaitée pour une simulation fait partie du schéma de simulation. Toutes les opérations relatives à une coche doivent être terminées avant le début de la prochaine coche. Pour cette raison, la fréquence d'horloge effective peut être inférieure à la fréquence d'horloge souhaitée. La fréquence d'horloge effective ne sera pas supérieure à la fréquence d'horloge souhaitée.

Partitions

Une partition est un segment de la mémoire partagée d'un travailleur. Chaque partition contient une partie des données d'état de simulation.

La partition d'une application spatiale (également appelée partition d'application spatiale ou partition spatiale) contient toutes les entités de la zone de propriété d'une application spatiale. SimSpace Weaver place les entités dans des partitions spatiales d'applications en fonction de l'emplacement spatial de chaque entité. Cela signifie qu'il SimSpace Weaver essaie de placer des entités spatialement proches les unes des autres sur le même travailleur. Cela minimise la quantité de connaissances dont une application a besoin sur les entités qu'elle ne possède pas pour simuler les entités qu'elle possède.

State Fabric

Le State Fabric est le système de mémoire partagée (la collection de toutes les partitions) sur tous les travailleurs. Il contient toutes les données d'état pour votre simulation.

Le State Fabric utilise un format binaire personnalisé qui décrit une entité sous la forme d'un ensemble de données initiales et d'un journal de mise à jour, pour chaque champ de données de cette entité. Avec ce format, vous pouvez accéder à l'état d'une entité à un moment antérieur de la simulation et la mapper à un point en temps réel. La taille de la mémoire tampon est limitée et il n'est pas possible de remonter dans le temps au-delà de ce qu'elle contient. SimSpace Weaver utilise un pointeur vers le décalage actuel dans le journal des mises à jour pour chaque champ, et met à jour

un pointeur dans le cadre d'une mise à jour de champ. SimSpace Weaver mappe ces journaux de mise à jour dans l'espace de traitement d'une application à l'aide de la mémoire partagée.

Ce format d'objet permet de réduire les frais généraux et d'éviter les coûts de sérialisation. SimSpace Weaver utilise également ce format d'objet pour analyser et identifier les champs d'index (tels que la position de l'entité).

Entités

Une entité est le plus petit bloc de données de votre simulation. Les exemples d'entités incluent les acteurs (tels que les personnes et les véhicules) et les objets statiques (tels que les bâtiments et les obstacles). Les entités possèdent des propriétés (telles que la position et l'orientation) dans lesquelles vous pouvez stocker des données persistantes SimSpace Weaver. Les entités existent au sein des partitions.

Applications

Une SimSpace Weaver application est un logiciel que vous écrivez et qui contient une logique personnalisée qui exécute chaque test de simulation. Le but de la plupart des applications est de mettre à jour les entités au fur et à mesure que la simulation s'exécute. Vos applications font appel APIs au SDK de l' SimSpace Weaver application pour effectuer des actions (telles que la lecture et la mise à jour) sur les entités de votre simulation.

Vous regroupez vos applications et les ressources dont elles ont besoin (telles que les bibliothèques) sous forme de fichiers .zip et vous les chargez dans SimSpace Weaver. Une application s'exécute dans un conteneur Docker sur un worker. SimSpace Weaver alloue à chaque application un nombre fixe d'unités de ressources au travailleur.

SimSpace Weaver attribue la propriété d'une (et une seule) partition à chaque application. Une application et sa partition se trouvent sur le même worker. Chaque partition n'a qu'un seul propriétaire d'application. Une application peut créer, lire, mettre à jour et supprimer des entités dans sa partition. Une application possède toutes les entités de sa partition.

Il existe trois types d'applications : les applications spatiales, les applications personnalisées et les applications de service. Ils diffèrent selon les cas d'utilisation et les cycles de vie.

> **Note**
>
> Dans SimSpace Weaver, le terme « application » peut faire référence au code d'une application ou à une instance en cours d'exécution de ce code.

Applications spatiales

Les applications spatiales mettent à jour l'état des entités qui existent dans l'espace dans votre simulation. Par exemple, vous pouvez définir une `Physics` application chargée de déplacer et de faire entrer en collision des entités à chaque tic-tac en fonction de leur vélocité, de leur forme et de leur taille. Dans ce cas, SimSpace Weaver exécute plusieurs instances de l'`Physics`application en parallèle pour gérer la taille de la charge de travail.

SimSpace Weaver gère le cycle de vie des applications spatiales. Vous spécifiez une disposition des partitions spatiales des applications dans votre schéma de simulation. Lorsque vous lancez votre simulation, SimSpace Weaver démarre une application spatiale pour chaque partition d'application spatiale. Lorsque vous arrêtez la simulation, SimSpace Weaver vos applications spatiales sont arrêtées.

D'autres types d'applications peuvent créer des entités, mais seules les applications spatiales peuvent mettre à jour des entités. Les autres types d'applications doivent transférer les entités qu'ils créent vers un domaine spatial. SimSpace Weaver utilise l'emplacement spatial d'une entité pour déplacer l'entité vers la partition d'une application spatiale. Cela transfère la propriété de l'entité à l'application spatiale.

Apps personnalisées

Vous utilisez des applications personnalisées pour interagir avec votre simulation. Une application personnalisée lit les données des entités à l'aide d'abonnements. Une application personnalisée peut créer des entités. Toutefois, l'application doit transférer une entité vers une application spatiale pour l'inclure dans la simulation et la mettre à jour. Vous pouvez avoir SimSpace Weaver attribué un point de terminaison réseau à une application personnalisée. Les clients de simulation peuvent se connecter au point de terminaison du réseau pour interagir avec la simulation. Vous définissez vos applications personnalisées dans votre schéma de simulation, mais il vous incombe de les démarrer et de les arrêter (à l'aide d'appels d' SimSpace Weaver API). Une fois que vous avez démarré une instance d'application personnalisée sur un worker, l'instance SimSpace Weaver n'est pas transférée à un autre worker.

Applications de service

Vous pouvez utiliser une application de service lorsque vous avez besoin d'un processus en lecture seule exécuté sur chaque travailleur. Par exemple, vous pouvez utiliser une application de service si vous avez une simulation de grande envergure et que vous avez besoin d'un client de visualisation qui parcourt la simulation et n'affiche que les entités visibles à l'utilisateur. Dans ce cas, une seule instance d'application personnalisée ne peut pas traiter toutes les entités de la simulation. Vous pouvez configurer une application de service pour qu'elle soit lancée sur chaque travailleur. Chacune de ces applications de service peut ensuite filtrer les entités sur le travailleur qui lui est assigné et n'envoyer que les entités pertinentes à ses clients connectés. Votre client de visualisation peut ensuite se connecter à différentes applications de service lorsqu'il se déplace dans l'espace de simulation. Vous configurez les applications de service dans votre schéma de simulation. SimSpace Weaver démarre et arrête vos applications de service pour vous.

Résumé de l'application

Le tableau suivant récapitule les caractéristiques des différents types d' SimSpace Weaver applications.

	Applications spatiales	Apps personnalisées	Applications de service
Lire les entités	Oui	Oui	Oui
Mettre à jour les entités	Oui	Non	Non
Création d'entités	Oui	Oui*	Oui*
Cycle de vie	Géré (le SimSpace Weaver contrôle)	Non géré (c'est vous qui le contrôlez.)	Géré (le SimSpace Weaver contrôle)
Méthode de démarrage	SimSpace Weaver démarre une instance d'application pour chaque partition spatiale, comme indiqué dans votre schéma.	Vous démarrez chaque instance d'application.	SimSpace Weaver démarre une ou plusieurs instances d'application sur chaque worker, comme indiqué dans votre schéma.

	Applications spatiales	Apps personnalisées	Applications de service
Les clients peuvent se connecter	Non	Oui	Oui

* Lorsqu'une application ou une application de service personnalisée crée une entité, l'application doit transférer la propriété de l'entité à une application spatiale afin que celle-ci puisse mettre à jour l'état de l'entité.

Domaines

Un SimSpace Weaver domaine est un ensemble d'instances d'application qui exécutent le même code d'application exécutable et possèdent les mêmes options et commandes de lancement. Nous faisons référence aux domaines en fonction des types d'applications qu'ils contiennent : domaines spatiaux, domaines personnalisés et domaines de service. Vous configurez vos applications au sein de domaines.

Abonnements et réplication

Une application crée un abonnement à une région spatiale pour connaître les événements relatifs aux entités (par exemple, entrée, sortie, création, mise à jour et suppression) dans cette région. Une application traite les événements d'entité liés à un abonnement avant de lire les données des entités situées dans des partitions qui ne lui appartiennent pas.

Une partition peut exister sur le même outil de travail que l'application (c'est ce qu'on appelle une partition locale), mais une autre application peut être propriétaire de la partition. Une partition peut également exister sur un autre serveur de travail (c'est ce qu'on appelle une partition distante). Si l'abonnement concerne une partition distante, le travailleur crée une copie locale de la partition distante par le biais d'un processus appelé réplication. Le travailleur lit ensuite la copie locale (partition distante répliquée). Si une autre application du programme de travail doit lire à partir de cette partition en même temps, le programme de travail lit la même copie locale.

Exemples de cas d'utilisation pour SimSpace Weaver

Vous pouvez l'utiliser SimSpace Weaver pour des modèles basés sur des agents et des simulations par étapes temporelles discrètes avec une composante spatiale.

Créez des simulations de grandes foules

Vous pouvez l'utiliser SimSpace Weaver pour simuler des foules dans des environnements réels. SimSpace Weaver vous permet d'adapter vos simulations à des millions d'objets dynamiques dotés de leurs propres comportements.

Créez des environnements à l'échelle d'une ville

SimSpace Weaver À utiliser pour créer le jumeau numérique d'une ville entière. Créez des simulations pour la planification urbaine, pour concevoir l'itinéraire du trafic et pour planifier la réponse aux risques environnementaux. Vous pouvez utiliser vos propres sources de données géospatiales comme éléments de base de vos environnements.

Créez des expériences immersives et interactives

Créez des expériences de simulation auxquelles plusieurs utilisateurs peuvent participer et interagir. Utilisez des outils de développement populaires tels que Unreal Engine et Unity pour créer des mondes virtuels tridimensionnels (3D). Personnalisez votre expérience 3D avec votre propre contenu et vos propres comportements.

AWS SimSpace Weaver fin du support

Après mûre réflexion, nous avons décidé de mettre fin au support à AWS SimSpace Weaver compter du 20 mai 2026. AWS SimSpace Weaver n'acceptera plus de nouveaux clients à compter du 20 mai 2025. En tant que client existant disposant d'un compte inscrit au service avant le 20 mai 2025, vous pouvez continuer à utiliser les AWS SimSpace Weaver fonctionnalités. Après le 20 mai 2026, vous ne pourrez plus utiliser AWS SimSpace Weaver.

Pour plus d'informations sur la transition AWS Batch afin de faciliter l'exécution de simulations conteneurisées, consultez ce billet de blog.

Configuration pour SimSpace Weaver

Pour être prêt à l'utiliser SimSpace Weaver pour la première fois, vous devez configurer votre environnement Compte AWS et votre environnement local. Lorsque vous aurez terminé ces tâches, vous serez prêt pour les didacticiels de mise en route.

Tâches de configuration

1. Configurez votre appareil Compte AWS pour l'utiliser SimSpace Weaver.
2. Configurez votre environnement local pour SimSpace Weaver.

Configurez votre appareil Compte AWS pour l'utiliser SimSpace Weaver

Effectuez les tâches suivantes pour Compte AWS configurer votre utilisation SimSpace Weaver.

Inscrivez-vous pour un Compte AWS

Si vous n'en avez pas Compte AWS, procédez comme suit pour en créer un.

Pour vous inscrire à un Compte AWS

1. Ouvrez l'https://portal.aws.amazon.com/billing/inscription.
2. Suivez les instructions en ligne.

   Une partie de la procédure d'inscription consiste à recevoir un appel téléphonique ou un message texte et à saisir un code de vérification sur le clavier du téléphone.

   Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWSest créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce compte. La meilleure pratique de sécurité consiste à attribuer un accès administratif à un utilisateur, et à utiliser uniquement l'utilisateur racine pour effectuer les tâches nécessitant un accès utilisateur racine.

AWS vous envoie un e-mail de confirmation une fois le processus d'inscription terminé. À tout moment, vous pouvez consulter l'activité actuelle de votre compte et gérer votre compte en accédant à https://aws.amazon.com/et en choisissant Mon compte.

Création d’un utilisateur doté d’un accès administratif

Après vous être inscrit à un Compte AWS, sécurisez Utilisateur racine d'un compte AWS AWS IAM Identity Center, activez et créez un utilisateur administratif afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisez votre Utilisateur racine d'un compte AWS

1. Connectez-vous en AWS Management Consoletant que propriétaire du compte en choisissant Utilisateur root et en saisissant votre adresse Compte AWS e-mail. Sur la page suivante, saisissez votre mot de passe.

   Pour obtenir de l’aide pour vous connecter en utilisant l’utilisateur racine, consultez Connexion en tant qu’utilisateur racine dans le Guide de l’utilisateur Connexion à AWS .
2. Activez l’authentification multifactorielle (MFA) pour votre utilisateur racine.

   Pour obtenir des instructions, voir Activer un périphérique MFA virtuel pour votre utilisateur Compte AWS root (console) dans le guide de l'utilisateur IAM.

Création d’un utilisateur doté d’un accès administratif

1. Activez IAM Identity Center.

   Pour obtenir des instructions, consultez Activation d’ AWS IAM Identity Center dans le Guide de l’utilisateur AWS IAM Identity Center .
2. Dans IAM Identity Center, octroyez un accès administratif à un utilisateur.

   Pour un didacticiel sur l'utilisation du Répertoire IAM Identity Center comme source d'identité, voir Configurer l'accès utilisateur par défaut Répertoire IAM Identity Center dans le Guide de AWS IAM Identity Center l'utilisateur.

Connexion en tant qu’utilisateur doté d’un accès administratif

- Pour vous connecter avec votre utilisateur IAM Identity Center, utilisez l’URL de connexion qui a été envoyée à votre adresse e-mail lorsque vous avez créé l’utilisateur IAM Identity Center.

  Pour obtenir de l'aide pour vous connecter en utilisant un utilisateur d'IAM Identity Center, consultez la section Connexion au portail AWS d'accès dans le guide de l'Connexion à AWS utilisateur.

Attribution d’un accès à d’autres utilisateurs

1. Dans IAM Identity Center, créez un ensemble d’autorisations qui respecte la bonne pratique consistant à appliquer les autorisations de moindre privilège.

   Pour obtenir des instructions, consultez Création d’un ensemble d’autorisations dans le Guide de l’utilisateur AWS IAM Identity Center .
2. Attribuez des utilisateurs à un groupe, puis attribuez un accès par authentification unique au groupe.

   Pour obtenir des instructions, consultez Ajout de groupes dans le Guide de l’utilisateur AWS IAM Identity Center .

Ajouter des autorisations d'utilisation SimSpace Weaver

Pour activer l’accès, ajoutez des autorisations à vos utilisateurs, groupes ou rôles :

- Utilisateurs et groupes dans AWS IAM Identity Center :

  Créez un jeu d’autorisations. Suivez les instructions de la rubrique Création d’un jeu d’autorisations du Guide de l’utilisateur AWS IAM Identity Center .
- Utilisateurs gérés dans IAM par un fournisseur d’identité :

  Créez un rôle pour la fédération d’identité. Suivez les instructions de la rubrique Création d’un rôle pour un fournisseur d’identité tiers (fédération) dans le Guide de l’utilisateur IAM.
- Utilisateurs IAM :
  - Créez un rôle que votre utilisateur peut assumer. Suivez les instructions de la rubrique Création d’un rôle pour un utilisateur IAM dans le Guide de l’utilisateur IAM.
  - (Non recommandé) Attachez une politique directement à un utilisateur ou ajoutez un utilisateur à un groupe d’utilisateurs. Suivez les instructions de la rubrique Ajout d’autorisations à un utilisateur (console) du Guide de l’utilisateur IAM.

Example Politique IAM pour accorder des autorisations d'utilisation SimSpace Weaver

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

```
            "Sid": "CreateAndRunSimulations",
            "Effect": "Allow",
            "Action": [
                "simspaceweaver:*",
                "iam:GetRole",
                "iam:ListRoles",
                "iam:CreateRole",
                "iam:DeleteRole",
                "iam:UpdateRole",
                "iam:CreatePolicy",
                "iam:AttachRolePolicy",
                "iam:PutRolePolicy",
                "iam:GetRolePolicy",
                "iam:DeleteRolePolicy",
                "s3:PutObject",
                "s3:GetObject",
                "s3:ListAllMyBuckets",
                "s3:PutBucketPolicy",
                "s3:CreateBucket",
                "s3:ListBucket",
                "s3:PutEncryptionConfiguration",
                "s3:DeleteBucket",
                "cloudformation:CreateStack",
                "cloudformation:UpdateStack",
                "cloudformation:DescribeStacks"
            ],
            "Resource": "*"
        },
        {
            "Sid": "PassAppRoleToSimSpaceWeaver",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "simspaceweaver.amazonaws.com"
                }
            }
        }
    ]
}
```

Configurez votre environnement local pour SimSpace Weaver

SimSpace Weaver simulations exécutées en conteneurisé Amazon Linux 2 (AL2) environnements. Vous devez disposer d'un AL2 environnement pour compiler et lier vos applications au SDK des SimSpace Weaver applications. L'environnement de développement local standard est un AL2 conteneur dans Docker. Si vous choisissez de ne pas utiliser Docker, nous fournissons des instructions alternatives pour exécuter un AL2 environnement dans Windows Subsystem for Linux (WSL). Vous pouvez également utiliser votre propre méthode pour créer un AL2 environnement local. Pour découvrir d'autres méthodes d'exécution en AL2 local, consultez la EC2 documentation Amazon.

> **Important**
>
> Docker on Microsoft Windows est l'environnement de développement standard. Pour vous faciliter la tâche, nous vous suggérons d'autres méthodes pour configurer votre environnement de développement local, mais elles ne sont pas standard et ne sont pas prises en charge.

Rubriques

- Configurez le package SimSpace Weaver de distribution pour Amazon Linux 2 (AL2) dans Docker
- Configurez le package SimSpace Weaver de distribution pour Amazon Linux 2 (AL2) dans Windows Subsystem for Linux (WSL)

Configurez le package SimSpace Weaver de distribution pour Amazon Linux 2 (AL2) dans Docker

Cette section fournit des instructions pour configurer votre fichier zip SimSpace Weaver de distribution local avec un AL2 environnement dans Docker. Pour obtenir des instructions de configuration avec AL2 in Windows Subsystem for Linux (WSL), voir Configurez le package SimSpace Weaver de distribution pour Amazon Linux 2 (AL2) dans Windows Subsystem for Linux (WSL).

Prérequis

- Microsoft Windows 10 ou supérieur, ou un système Linux compatible

- Microsoft Visual Studio 2019ou plus tard, avec Desktop development with C++charge de travail installée
- CMake3
- Git
- Docker Desktop
- AWS CLI
- Python 3.9

Pour configurer le zip SimSpace Weaver de distribution AL2 dans Docker

1. Si vous n'avez pas encore configuré vos AWS informations d'identification pour le AWS CLI, suivez ces instructions : Configuration de l'AWS CLI.
2. Téléchargez le package distribuable du SDK de l' SimSpace Weaver application. Il contient les éléments suivants :
   - Binaires et bibliothèques pour le développement d' SimSpace Weaver applications
   - Scripts d'assistance qui automatisent certaines parties du flux de travail de développement
   - Exemples d'applications illustrant SimSpace Weaver des concepts
3. Décompressez le fichier dans celui *sdk-folder* de votre choix.
4. Accédez au *sdk-folder*.
5. Entrez la commande suivante pour installer les packages Python requis :

   ```
   pip install -r PackagingTools/python_requirements.txt
   ```
6. Entrez la commande suivante pour configurer la SimSpace Weaver distribution avec une image Docker.

   ```
   python setup.py
   ```

   Cette commande exécute les opérations suivantes :

   - Crée une image AL2 docker avec toutes les exigences pour les SimSpace Weaver projets de construction installés.
   - Crée les CloudFormation ressources nécessaires au lancement d'une simulation.

- Le modèle de CloudFormation pile d'échantillons se trouve dans *sdk-folder*/PackagingTools/sample-stack-template.yaml
- Configure les exemples de projets fournis avec les chemins adaptés à votre système local.

Résolution des problèmes

- Docker semble bloqué
  - Si la sortie de la console semble bloquée après l'appel des commandes Docker, essayez de redémarrer le moteur Docker. Si cela ne fonctionne pas, redémarrez votre ordinateur.

Configurez le package SimSpace Weaver de distribution pour Amazon Linux 2 (AL2) dans Windows Subsystem for Linux (WSL)

Cette section fournit des instructions pour configurer votre fichier zip SimSpace Weaver de distribution avec un AL2 environnement dans Windows Subsystem for Linux (WSL). Pour obtenir des instructions de configuration AL2 dans Docker, voir Configurez le package SimSpace Weaver de distribution pour Amazon Linux 2 (AL2) dans Docker.

> **Important**
>
> Cette section décrit une solution qui utilise une version AL2 qui n'est pas détenue, développée ou prise en charge par Amazon. Cette solution est fournie pour votre commodité uniquement, si vous choisissez de ne pas utiliser Docker. Amazon AWS décline toute responsabilité si vous choisissez d'utiliser cette solution.

Prérequis

- Hyper-V on Windows 10
- Windows Subsystem for Linux (WSL)
- AL2 Distribution open source tierce pour WSL (télécharger la version 2.0.20200722.0-update.2) (voir les instructions)

> **Important**
>
> Notre WSL les instructions utilisent la version 2.0.20200722.0-update.2 de la distribution pour AL2 WSL. Vous risquez de rencontrer des erreurs si vous utilisez une autre version.

Pour configurer le SimSpace Weaver zip de distribution dans AL2 WSL

1. À l'invite de commande Windows, démarrez votre AL2 environnement dans WSL.

   ```
   wsl -d Amazon2
   ```

   > **Important**
   >
   > Pendant que vous courez WSL, incluez `--al2` cette option lors de l'exécution de l'un des scripts d'assistance `quick-start.py` Python situés à l'`sdky-folder/Samples/sample-name/tools/cloud/quick-start.py`adresse.

2. À l'invite du shell Linux, mettez à jour votre gestionnaire de paquets yum.

   ```
   yum update -y
   ```

   > **Important**
   >
   > Si cette étape expire, vous devrez peut-être passer à WSL1 et réessayez ces procédures. Quittez votre WSL AL2 session et entrez ce qui suit à l'invite de commande Windows :
   >
   > ```
   > wsl --set-version Amazon2 1
   > ```

3. Installez l'outil de décompression.

   ```
   yum install -y unzip
   ```

4. Supprimez tout AWS CLI ce qui `yum` est installé. Essayez les deux commandes suivantes si vous ne savez pas `yum` si un AWS CLI.

```
yum remove awscli
```

```
yum remove aws-cli
```

5. Créez un répertoire temporaire et accédez-y.

```
mkdir ~/temp
cd ~/temp
```

6. Téléchargez et installez AWS CLI :

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
./aws/install
```

7. Vous pouvez supprimer le répertoire temporaire.

```
cd ~
rm -rf temp
```

8. Redémarrez la session shell pour mettre à jour le chemin dans l'environnement.

```
exec
```

9. Configurez vos AWS informations d'identification pour le AWS CLI dans votre AL2 environnement. Pour plus d'informations, consultez Configuration de l'AWS CLI. Si vous en utilisez AWS IAM Identity Center, reportez-vous AWS CLI à la section Configuration du à utiliser AWS IAM Identity Center dans le guide de AWS Command Line Interface l'utilisateur.

```
aws configure
```

10. Installez Git.

```
yum install -y git
```

11. Installer `wget`.

```
yum install -y wget
```

12. Créez un dossier pour le SDK de SimSpace Weaver l'application.

```
mkdir sdk-folder
```

13. Accédez au dossier du SDK.

```
cd sdk-folder
```

14. Téléchargez le package distribuable du SDK de l' SimSpace Weaver application. Il contient les éléments suivants :

    - Binaires et bibliothèques pour le développement d' SimSpace Weaver applications
    - Scripts d'assistance qui automatisent certaines parties du flux de travail de développement
    - Exemples d'applications illustrant SimSpace Weaver des concepts

```
wget https://artifacts.simspaceweaver.us-east-2.amazonaws.com/latest/
SimSpaceWeaverAppSdkDistributable.zip
```

15. Décompressez le fichier.

```
unzip *.zip
```

16. Exécutez le script de configuration WSL.

```
source ./setup-wsl-distro.sh
```

17. Entrez la commande suivante pour installer les packages Python requis :

```
pip install -r PackagingTools/python_requirements.txt
```

18. Exécutez le script de configuration du fichier de SimSpace Weaver distribution zip :

```
python setup.py --samples --cloudformation
```

    Cette commande exécute les opérations suivantes :

    - Crée les CloudFormation ressources nécessaires pour lancer une simulation.
      - Le modèle de CloudFormation pile d'échantillons se trouve dans *sdk-folder*/PackagingTools/sample-stack-template.yaml
    - Configure les exemples de projets fournis avec les chemins adaptés à votre système local.

> **Note**
>
> Vous n'avez besoin de le faire qu'une seule fois pour votre AL2 environnement dans WSL.

Utilisation de logiciels sous licence avec AWS SimSpace Weaver

AWS SimSpace Weaver vous permet de créer des simulations avec le moteur de simulation et le contenu de votre choix. Dans le cadre de votre utilisation de SimSpace Weaver, vous êtes responsable de l'obtention, de la maintenance et du respect des termes de licence de tout logiciel ou contenu que vous utilisez dans vos simulations. Vérifiez que votre contrat de licence vous permet de déployer votre logiciel et votre contenu dans un environnement hébergé virtuel.

Commencer avec SimSpace Weaver

Cette section propose des didacticiels pour vous aider à démarrer SimSpace Weaver. Ces didacticiels vous présentent le flux de travail général pour créer des simulations avec SimSpace Weaver. Ces didacticiels montrent comment créer, déployer et exécuter des simulations dans SimSpace Weaver. Nous vous recommandons de commencer par le didacticiel de démarrage rapide pour exécuter une simulation en quelques minutes. Parcourez ensuite les autres didacticiels pour en savoir plus.

Ces didacticiels utilisent un exemple d'application (`PathfindingSample`) inclus dans le fichier .zip du SDK de SimSpace Weaver l'application que vous avez téléchargé lors des procédures de configuration. L'exemple d'application présente les concepts communs à toutes les SimSpace Weaver simulations, notamment le partitionnement spatial, le transfert d'entités entre partitions, les applications et les abonnements.

Dans les didacticiels, vous allez créer une simulation avec quatre partitions spatiales. Une instance distincte de l'application `PathfindingSample` spatiale gère chaque partition individuelle. Les applications spatiales créent des entités dans leurs propres partitions. Les entités se déplacent vers une position particulière dans le monde de simulation, évitant ainsi les obstacles lorsqu'elles se déplacent. Vous pouvez utiliser une application client distincte (incluse dans le SDK de SimSpace Weaver l'application) pour visualiser la simulation.

Rubriques

- Tutoriel de démarrage rapide pour SimSpace Weaver
- Tutoriel détaillé : découvrez les détails lors de la création de l'exemple d'application

Tutoriel de démarrage rapide pour SimSpace Weaver

Ce didacticiel vous guide tout au long du processus de création et d'exécution d'une simulation SimSpace Weaver en quelques minutes. Nous vous recommandons de commencer par ce didacticiel, puis de suivre le didacticiel détaillé par la suite.

Prérequis

Avant de commencer, assurez-vous d'avoir effectué les étapes deConfiguration pour SimSpace Weaver.

> **Note**
>
> Les scripts utilisés ici sont fournis pour votre commodité et ne sont PAS obligatoires. Consultez le didacticiel détaillé pour savoir comment effectuer ces étapes manuellement.

Étape 1 : activer la journalisation (facultatif)

Pour activer la journalisation

1. Naviguez vers :

   ```
   sdk-folder/Samples/PathfindingSample/tools
   ```

2. Ouvrez le fichier de schéma dans un éditeur de texte :

   ```
   pathfinding-single-worker-schema.yaml
   ```

3. Trouvez la `simulation_properties:` section au début du fichier :

   ```
   simulation_properties:
     default_entity_index_key_type: "Vector3<f32>"
   ```

4. Insérer les 2 lignes suivantes après la ligne `simulation_properties:` :

   ```
     log_destination_service: "logs"
     log_destination_resource_name: "MySimulationLogs"
   ```

5. Vérifiez que votre `simulation_properties:` section est identique à la suivante :

   ```
   simulation_properties:
     log_destination_service: "logs"
     log_destination_resource_name: "MySimulationLogs"
     default_entity_index_key_type: "Vector3<f32>"
   ```

6. Enregistrez le fichier et quittez votre éditeur de texte.

Étape 2 : Démarrage rapide avec le client de console (option 1)

Naviguez vers :

```
sdk-folder/Samples/PathfindingSample/tools/cloud
```

Exécutez une des commandes suivantes :

- Docker : `python quick-start.py --consoleclient`
- WSL : `python quick-start.py —-consoleclient --al2`

Par défaut, cela lancera une simulation avec une seule partition sur un seul travailleur. D'autres configurations peuvent être lancées en les transmettant `--schema {file name}.yaml` depuis le `/Samples/PathfindingSample/tools/` dossier.

> **Note**
>
> Consultez Tutoriel détaillé : découvrez les détails lors de la création de l'exemple d'application pour une explication détaillée de ce que fait ce script.

Étape 2 : Démarrage rapide avec le client Unreal Engine (option 2)

Consultez Lancement du client de visualisation Unreal Engine.

Arrêtez et supprimez votre simulation

Naviguez vers :

```
sdk-folder/Samples/PathfindingSample/tools/cloud
```

Trouvez le nom de vos simulations :

```
aws simspaceweaver list-simulations
```

Arrêter et supprimer la simulation

```
python stop-and-delete.py --simulation simulation-name
```

Résolution des problèmes

- FileNotFoundError: cmake

```
subprocess.run('cmake')
...
 FileNotFoundError: The system cannot find the file specified
```

  - Résolution : Le script ne trouve pas la commande`cmake`. Assurez-vous que la CMake version minimale recommandée est installée et qu'elle peut être appelée à l'aide de la `cmake` commande dans le PATH. Utilisez la commande `cmake -version` pour vérifier.
- ImportError: Le chargement de la DLL a échoué lors de l'importation de libweaver_app_sdk_python_v1 : le module spécifié est introuvable.
  - Résolution : Cette erreur se produit lorsque Python 3.9 n'est pas utilisé pour lancer le SDK Weaver Python. Assurez-vous que la version python associée à la commande « python » est Python 3.9. Vous pouvez vérifier en exécutant la `python --version` commande.
- Le script de démarrage rapide apparaît bloqué après le démarrage de Docker Build.
  - Résolution : Docker a parfois besoin de quelques minutes pour s'échauffer. Si le problème persiste pendant plus de 5 minutes, redémarrez Docker ou votre système.
- target_compile_features aucune fonctionnalité connue pour le compilateur CXX « GNU » :
  - Résolution : Videz votre cache Docker, supprimez l'image Docker de weaverappbuilder, supprimez les artefacts de construction de votre projet et réexécutez. `setup.py` Cela devrait réinitialiser votre environnement Docker et résoudre l'erreur.

Tutoriel détaillé : découvrez les détails lors de la création de l'exemple d'application

Le didacticiel de démarrage rapide expliquait comment créer, démarrer, arrêter et supprimer un exemple de simulation à l'aide de `quick-start.py` et`stop-and-delete.py`. Ce didacticiel décrit en détail le fonctionnement de ces scripts et les paramètres supplémentaires qu'ils peuvent prendre pour optimiser la flexibilité des simulations Weaver personnalisées.

Prérequis

Avant de commencer, assurez-vous d'avoir effectué les étapes deConfiguration pour SimSpace Weaver.

Étape 1 : activer la journalisation (facultatif)

Pour activer la journalisation

1. Naviguez vers :

```
sdk-folder/Samples/PathfindingSample/tools
```

2. Ouvrez le fichier de schéma dans un éditeur de texte :

```
pathfinding-single-worker-schema.yaml
```

3. Trouvez la `simulation_properties:` section au début du fichier :

```
simulation_properties:
  default_entity_index_key_type: "Vector3<f32>"
```

4. Insérer les 2 lignes suivantes après la ligne `simulation_properties:` :

```
  log_destination_service: "logs"
  log_destination_resource_name: "MySimulationLogs"
```

5. Vérifiez que votre `simulation_properties:` section est identique à la suivante :

```
simulation_properties:
  log_destination_service: "logs"
  log_destination_resource_name: "MySimulationLogs"
  default_entity_index_key_type: "Vector3<f32>"
```

6. Enregistrez le fichier et quittez votre éditeur de texte.

Étape 2 : Lancez votre simulation

Comme le montre le didacticiel de démarrage rapide, les étapes les plus élémentaires pour lancer un exemple de simulation sont les suivantes :

1. Naviguez vers :

```
sdk-folder/Samples/PathfindingSample/tools/cloud
```

2. Exécutez l'une des commandes suivantes :

- Docker : `python quick-start.py`
- WSL : `python quick-start.py --al2`

Ce script automatise les commandes de terminal courantes, qui peuvent toutes être exécutées manuellement à l'aide du AWS CLI. Ces étapes sont les suivantes :

1. Téléchargez le schéma Weaver sur S3.
   - SimSpace Weaver utilise un schéma pour configurer votre simulation. Le schéma est un fichier texte brut au format YAML. Pour plus d'informations, consultez Configuration de votre simulation.
2. Créez et téléchargez un conteneur personnalisé (facultatif).
   - Si votre schéma définit un conteneur personnalisé, le script de démarrage rapide créera l'image docker et la téléchargera sur Amazon ECR. Pour de plus amples informations, veuillez consulter Conteneurs personnalisés. Consultez le `PythonBubblesSample` schéma pour un exemple de cette fonctionnalité.
3. Générez le projet.
   - `quick-start.py`appelle la `build_project` fonction définie dans`build.py`. Cette étape varie en fonction du projet. Pour le PathfindingSample, CMake est utilisé. La commande CMake et Docker pour laquelle se trouve dans`build.py`.
4. Téléchargez les artefacts de construction sur S3.
   - Vous pouvez vérifier vos compartiments S3 pour vous assurer que tous les téléchargements ont bien été effectués. Pour plus d'informations sur l'utilisation d'Amazon S3, consultez la section Création, configuration et utilisation des compartiments Amazon S3 dans le guide de l'utilisateur d'Amazon Simple Storage Service.
   - Les exemples de zips d'application et de compartiment S3 utilisent le format de nom suivant :
     - `weaver-sample-bucket-account-number-region`
     - Application spatiale : `ProjectNameSpatial.zip`
     - Afficher l'application (personnalisée) : `ProjectNameView.zip`
5. Lancez la simulation.

- Ceci est une enveloppe de l'`aws simspaceweaver start-simulation` AWS CLI appel. Pour plus d'informations, consultez le manuel de référence des AWS CLI commandes pour SimSpace Weaver.
- Le script tournera en boucle jusqu'à ce que le statut de la simulation soit `STARTED` égal ou`FAILED`. Le démarrage d'une simulation peut prendre quelques minutes.

6. Obtenez les détails de la simulation.

   - L'DescribeSimulationAPI fournit des informations sur votre simulation, notamment son état. Une simulation peut présenter l'un des états suivants :

     États du cycle de vie de la simulation

     1. **`STARTING`**— État initial après l'appel StartSimulation
     2. **`STARTED`**— toutes les applications spatiales sont lancées et fonctionnent
     3. **`STOPPING`**— État initial après l'appel StopSimulation
     4. **`STOPPED`**— Toutes les ressources de calcul sont arrêtées
     5. **`DELETING`**— État initial après l'appel DeleteSimulation
     6. **`DELETED`**— Toutes les ressources affectées à la simulation sont supprimées
     7. **`FAILED`**— La simulation a présenté une erreur/un échec critique et s'est arrêtée
     8. **`SNAPSHOT_IN_PROGRESS`**— Un instantané est en cours

     Pour obtenir les détails de votre simulation

     1. Appelez l'API ListSimulations.

        ```
        aws simspaceweaver list-simulations
        ```

        Le script doit afficher les détails de chacune de vos simulations, comme suit :

        ```
        {
            "Status": "STARTED",
            "CreationTime": 1664921418.09,
            "Name": "MyProjectSimulation_22-10-04_22_10_15",
            "Arn": "arn:aws:simspaceweaver:us-west-2:111122223333:simulation/
        MyProjectSimulation_22-10-04_22_10_15",
            "TargetStatus": "STARTED"
        ```

```
}
```

2. Appelez DescribeSimulation pour obtenir les détails de votre simulation. *simulation-name*Remplacez-le par celui Name de votre simulation à partir du résultat de l'étape précédente.

```
aws simspaceweaver describe-simulation --simulation simulation-name
```

Le script doit afficher plus de détails sur la simulation que vous avez spécifiée, comme suit :

```
{
    "Status": "STARTED",
    "CreationTime": 1664921418.09,
    "Name": "MyProjectSimulation_22-10-04_22_10_15",
    "Arn": "arn:aws:simspaceweaver:us-west-2:111122223333:simulation/
MyProjectSimulation_22-10-04_22_10_15",
    "TargetStatus": "STARTED"
}
```

7. Lancez des applications personnalisées.

- SimSpace Weaver ne gère pas le cycle de vie des applications personnalisées. Vous devez démarrer vos applications personnalisées. Il est recommandé de démarrer vos applications personnalisées avant de démarrer votre horloge de simulation, mais vous pouvez démarrer des applications personnalisées une fois l'horloge démarrée.

  Vous pouvez appeler l'StartAppAPI pour démarrer vos applications personnalisées.

```
aws simspaceweaver start-app --simulation simulation-name --name app-name --
domain domain-name
```

  L'appel StartApp d'API créera et démarrera une nouvelle instance de l'application personnalisée en utilisant le nom que vous avez fourni. Si vous indiquez le nom d'une application qui existe déjà, vous recevrez un message d'erreur. Si vous souhaitez redémarrer une application (instance) en particulier, vous devez d'abord arrêter cette application et la supprimer.

> **Note**
>
> L'état de votre simulation doit être atteint `STARTED` avant que vous puissiez démarrer des applications personnalisées.

L'exemple d'application fournit l'application `ViewApp` personnalisée pour visualiser votre simulation. Cette application vous fournit une adresse IP statique et un numéro de port pour connecter les clients de simulation (vous le ferez dans une étape ultérieure de ce didacticiel). Vous pouvez considérer un domain comme une classe d'applications qui ont le même code exécutable et les mêmes options de lancement. app nameIdentifie l'instance de l'application. Pour plus d'informations sur SimSpace Weaver les concepts, voirConcepts clés pour SimSpace Weaver.

Vous pouvez utiliser l'DescribeAppAPI pour vérifier le statut d'une application personnalisée une fois que vous l'avez démarrée.

```
aws simspaceweaver describe-app --simulation simulation-name --app app-name --
domain domain-name
```

Pour démarrer l'application View dans ce didacticiel

1. Appelez StartApp pour`ViewApp`.

   ```
   aws simspaceweaver start-app --simulation simulation-name --name ViewApp --
   domain MyViewDomain
   ```

2. Appelez DescribeApp pour vérifier le statut de votre application personnalisée.

   ```
   aws simspaceweaver describe-app --simulation simulation-name --app ViewApp --
   domain MyViewDomain
   ```

Une fois que le statut de votre application personnalisée (instance) est `STARTED` atteint, la sortie de DescribeApp inclura l'adresse IP et le numéro de port de cette application personnalisée (instance). Dans l'exemple de sortie suivant, l'adresse IP est la valeur de `Address` et le numéro de port est la valeur de `Actual` dans le `EndpointInfo` bloc.

```
{
    "Status": "STARTED",
    "Domain": "MyViewDomain",
    "TargetStatus": "STARTED",
    "Simulation": "MyProjectSimulation_22-10-04_22_10_15",
    "LaunchOverrides": {
        "LaunchCommands": []
    },
    "EndpointInfo": {
        "IngressPortMappings": [
            {
                "Declared": 7000,
                "Actual": 4321
            }
        ],
        "Address": "198.51.100.135"
    },
    "Name": "ViewApp"
}
```

> **Note**
>
> La valeur de `Declared` est le numéro de port auquel le code de votre application doit être lié. La valeur de `Actual` est le numéro de port qui SimSpace Weaver permet aux clients de se connecter à votre application. SimSpace Weaver mappe le `Declared` port par `Actual` rapport au port.

> **Note**
>
> Vous pouvez utiliser la procédure décrite sur Obtenez l'adresse IP et le numéro de port d'une application personnalisée pour obtenir l'adresse IP et le numéro de port de toute application personnalisée démarrée.

8. Démarrez l'horloge.

- Lorsque vous créez votre simulation pour la première fois, elle possède une horloge, mais celle-ci ne tourne pas. Lorsque votre horloge ne fonctionne pas, l'état de votre simulation ne sera pas mis à jour. Une fois que vous aurez démarré l'horloge, elle commencera à envoyer des ticks à vos applications. À chaque coche, vos applications spatiales parcourent les entités qu'elles possèdent et valident les résultats dans SimSpace Weaver

  > **Note**
  >
  > Le démarrage de l'horloge peut prendre 30 à 60 secondes.

  Appelez l'API StartClock.

  ```
  aws simspaceweaver start-clock --simulation simulation-name
  ```

  > **Note**
  >
  > L'StartClockAPI utilise votre*simulation-name*, que vous pouvez trouver à l'aide de l'ListSimulationsAPI :
  >
  > ```
  > aws simspaceweaver list-simulations
  > ```

paramètres de démarrage rapide

- -h, --help
  - Répertoriez ces paramètres.
- --propre
  - Supprimez le contenu du répertoire de construction avant de le compiler.
- --tous les 2
  - Construit directement sur la machine native au lieu de Docker. Utilisez-le uniquement si vous l'exécutez dans un environnement Amazon Linux 2, tel que WSL.
- --téléchargement uniquement
  - Importez uniquement le schéma et les fichiers compressés de l'application sur Amazon S3, ne lancez pas la simulation.

- --aucune construction
  - Ignorez la reconstruction du projet.
- --pas de conteneur
  - Ignorez la reconstruction du conteneur de simulation répertorié dans le schéma.
- --client de console
  - Créez et connectez automatiquement le client de console répertorié dans le fichier config.py.
- --schema SCHÉMA
  - Quel schéma utilisera cette invocation. La valeur par défaut est « SCHEMA » dans le fichier config.py.
- --name NOM
  - Quel nom portera la simulation ? La valeur par défaut est « Project_Name'-date-time » dans le fichier config.py.

Étape 3 : Vérifiez les journaux (facultatif)

SimSpace Weaver écrit les messages de gestion des simulations et les résultats de console de vos applications sur Amazon CloudWatch Logs. Pour plus d'informations sur l'utilisation des journaux, consultez la section Utilisation des groupes de journaux et des flux de CloudWatch journaux dans le guide de l'utilisateur Amazon Logs.

Chaque simulation que vous créez possède son propre groupe de CloudWatch journaux dans Logs. Le nom du groupe de journaux est spécifié dans le schéma de simulation. Dans l'extrait de schéma suivant, la valeur de `log_destination_service` est. `logs` Cela signifie que la valeur de `log_destination_resource_name` est le nom d'un groupe de journaux. Dans ce cas, le groupe de journaux est`MySimulationLogs`.

```
simulation_properties:
  log_destination_service: "logs"
  log_destination_resource_name: "MySimulationLogs"
  default_entity_index_key_type: "Vector3<f32>"
```

Vous pouvez également utiliser l'DescribeSimulationAPI pour trouver le nom du groupe de journaux pour la simulation une fois que vous l'avez démarrée.

```
aws simspaceweaver describe-simulation --simulation simulation-name
```

L'exemple suivant montre la partie de la sortie DescribeSimulation qui décrit la configuration de journalisation. Le nom du groupe de journaux est affiché à la fin du`LogGroupArn`.

```
    "LoggingConfiguration": {
        "Destinations": [
            {
                "CloudWatchLogsLogGroup": {
                    "LogGroupArn": "arn:aws:logs:us-west-2:111122223333:log-
group:MySimulationLogs"
                }
            }
        ]
    },
```

Chaque groupe de journaux de simulation contient plusieurs flux de journaux :

- Flux du journal de gestion : messages de gestion de simulation produits par le SimSpace Weaver service.

  ```
  /sim/management
  ```

- Flux du journal des erreurs : messages d'erreur produits par le SimSpace Weaver service. Ce flux de journal n'existe qu'en cas d'erreur. SimSpace Weaver stocke les erreurs écrites par vos applications dans leurs propres flux de journaux d'applications (voir les flux de journaux suivants).

  ```
  /sim/errors
  ```

- Flux de journaux d'applications spatiales (1 pour chaque application spatiale sur chaque travailleur) : sortie de console produite par les applications spatiales. Chaque application spatiale écrit dans son propre flux de log. Tous les caractères *spatial-app-id* se trouvent après la barre oblique finale à la fin du. *worker-id*

  ```
  /domain/spatial-domain-name/app/worker-worker-id/spatial-app-id
  ```

- Flux de journaux d'applications personnalisés (1 pour chaque instance d'application personnalisée) : sortie de console produite par des applications personnalisées. Chaque instance d'application personnalisée écrit dans son propre flux de journal.

  ```
  /domain/custom-domain-name/app/custom-app-name/random-id
  ```

- Flux de journal des applications de service (1 pour chaque instance d'application de service) : sortie de console produite par les applications de service. Chaque application de service écrit dans son propre flux de journal. Tous les caractères *service-app-id* se trouvent après la barre oblique finale à la fin du. *service-app-name*

  ```
  /domain/service-domain-name/app/service-app-name/service-app-id
  ```

> **Note**
>
> L'exemple d'application ne contient pas d'applications de service.

Étape 4 : Afficher votre simulation

Le SDK de SimSpace Weaver l'application propose différentes options pour afficher l'exemple d'application. Vous pouvez utiliser l'exemple de client de console si vous ne disposez d'aucun support local pour le développement d'Unreal Engine. Les instructions pour le client Unreal Engine supposent que vous utilisez Windows.

Le client de console affiche une liste des événements d'entité au fur et à mesure qu'ils se produisent. Le client obtient les informations relatives à l'événement de l'entité auprès du`ViewApp`. Si le client de console affiche la liste des événements, cela confirme la connectivité réseau avec l'activité `ViewApp` et dans votre simulation.

La `PathfindingSample` simulation crée des entités fixes et mobiles sur un plan bidimensionnel. Les entités mobiles se déplacent autour des entités stationnaires. Le client Unreal Engine fournit une visualisation des événements de l'entité.

Client de console

Le client de console peut être automatiquement créé et connecté lors du lancement d'un exemple `quick-start.py` si vous incluez l'`--consoleclient`option. Pour créer et connecter le client de console `quick-start.py` une fois qu'il a déjà été appelé, procédez comme suit :

Naviguez vers :

```
sdk-folder/Clients/TCP/CppConsoleClient
```

Exécutez le script pour créer et connecter le client :

```
python start_client.py --host ip-address --port port-number
```

Le script effectuera les opérations suivantes :

1. Créez le client de console avec CMake.
2. Lancez l'exécutable intégré avec l'adresse IP et le numéro de port indiqués.

```
.\WeaverNngConsoleClient.exe --url tcp://ip-address:port-number
```

Client Unreal Engine

Consultez Lancement du client de visualisation Unreal Engine.

Étape 5 : Arrêtez et supprimez votre simulation

Naviguez vers :

```
sdk-folder/Samples/PathfindingSample/tools/cloud
```

Trouvez le nom de vos simulations :

```
aws simspaceweaver list-simulations
```

Arrêtez et supprimez la simulation :

```
python stop-and-delete.py --simulation simulation-name
```

Le script `stop-and-delete.py` effectuera les opérations suivantes :

1. Appelez la AWS CLI commande pour arrêter une simulation.
   - `aws simspaceweaver stop-simulation`
   - Pour plus d'informations, consultez AWS CLI la section Référence des commandes pour SimSpace Weaver.
2. Appelez la AWS CLI commande pour supprimer une simulation.
   - `aws simpaceweaver delete-simulation`
   - Pour plus d'informations, consultez AWS CLI la section Référence des commandes pour SimSpace Weaver.

stop-and-delete paramètres

- -h, --help
  - Répertoriez ces paramètres.
- --simulation SIMULATION
  - Le nom de la simulation pour stop-and-delete
- --arrête
  - Arrêtez simplement la simulation. Ne le supprime pas.
- --supprimer
  - Supprimez uniquement une simulation. Ne fonctionnera que si la simulation est l'une `STOPPED` ou l'autre`FAILED`.

Résolution des problèmes

Consultez Résolution des problèmes le didacticiel de démarrage rapide.

Travailler avec SimSpace Weaver

Ce chapitre fournit des informations et des conseils pour vous aider à créer vos propres applications dans SimSpace Weaver.

Rubriques

- Configuration de votre simulation
- Durée maximale d'une simulation
- Développement d'applications
- Développement d'applications clientes
- Obtenez l'adresse IP et le numéro de port d'une application personnalisée
- Lancement du client de visualisation Unreal Engine
- Développement local en SimSpace Weaver
- AWS SimSpace Weaver SDK de l'application
- AWS SimSpace Weaver framework de démonstration
- Utilisation des quotas de service
- Simulations de débogage
- Conteneurs personnalisés
- Travail avec Python
- Support pour d'autres moteurs
- Utilisation de logiciels sous licence avec AWS SimSpace Weaver
- Gérez vos ressources avec AWS CloudFormation
- Instantanés
- Messagerie

Configuration de votre simulation

Un schéma (ou schéma) de simulation est YAML-fichier texte formaté qui spécifie la configuration d'une simulation. Vous pouvez utiliser le même schéma pour démarrer plusieurs simulations. Le fichier de schéma se trouve dans le dossier du projet pour votre simulation. Vous pouvez

utiliser n'importe quel éditeur de texte pour modifier le fichier. SimSpace Weaver lit votre schéma uniquement lorsqu'il démarre la simulation. Les modifications que vous apportez à un fichier de schéma n'affectent que les nouvelles simulations que vous commencez après les modifications.

Pour configurer votre simulation, modifiez le fichier de schéma de simulation (utilisez le séparateur de chemin approprié pour votre système d'exploitation) :

```
project-folder\tools\project-name-schema.yaml
```

Vous chargez le schéma de simulation lorsque vous créez une nouvelle simulation. Le script d'aide au démarrage rapide de votre projet téléchargera le schéma dans le cadre du processus de création de votre simulation :

```
project-folder\tools\windows\quick-start.py
```

Pour plus d'informations sur l'exécution du script de démarrage rapide, consultez Tutoriel détaillé le Premiers pas chapitre de ce guide.

Paramètres de configuration de simulation

Le schéma de simulation contient des informations d'amorçage, notamment :

- Propriétés de simulation : version du SDK et configuration de calcul (type et nombre de travailleurs)
- Horloges : taux de tic-tac et tolérances
- Stratégies de partitionnement spatial : topologie spatiale (telle qu'une grille), limites et groupes de placement (regroupement de partitions spatiales sur les travailleurs)
- Domaines et applications associées : compartiment d'applications, chemin et commande (s) de lancement

SimSpace Weaver utilise la configuration de votre schéma pour configurer et organiser les partitions spatiales, lancer des applications et faire avancer la simulation au rythme que vous avez spécifié.

> **Note**
>
> Le script de création de projet du SDK de l' SimSpace Weaver application génère automatiquement un schéma de simulation pour vous, sur la base de l'exemple d'application.

Les rubriques suivantes décrivent les paramètres du schéma de simulation. Pour une description complète du schéma de simulation, voirSimSpace Weaver référence du schéma de simulation.

Rubriques

- Version de SDK
- Propriétés de simulation
- Workers
- Horloge
- Stratégies de partitionnement
- Domaines

Version de SDK

Le `sdk_version` champ indique la version pour SimSpace Weaver laquelle le schéma est formaté. Valeurs valides: `1.17`, `1.16`, `1.15`, `1.14`, `1.13`, `1.12`

> **Important**
>
> La valeur de inclut `sdk_version` uniquement le numéro de version majeure et le numéro de première version secondaire. Par exemple, la valeur `1.12` spécifie toutes les versions`1.12.x`, telles que `1.12.0`1.12.1, et`1.12.2`.

Propriétés de simulation

La `simulation_properties` section de votre schéma indique la configuration de journalisation et le type de données pour le champ d'index (généralement l'emplacement spatial) des entités.

```
simulation_properties:
  log_destination_service: "logs"
  log_destination_resource_name: "MySimulationLogs"
  default_entity_index_key_type: "Vector3<f32>"
```

La valeur de `log_destination_service` détermine l'interprétation de la valeur de`log_destination_resource_name`. À l'heure actuelle, la seule valeur prise en charge est

`logs`. Cela signifie que la valeur de `log_destination_resource_name` est le nom d'un groupe de journaux dans Amazon CloudWatch Logs

> **Note**
>
> La journalisation est facultative. Si vous ne configurez pas les propriétés de destination des journaux, votre simulation ne produira pas de journaux.

Seule la propriété `default_entity_index_key_type` est obligatoire. La seule valeur valide est `Vector3<f32>`.

Workers

La `workers` section indique le type et le nombre de travailleurs que vous souhaitez utiliser pour votre simulation. SimSpace Weaver utilise ses propres types de travail qui correspondent aux types d' EC2 instances Amazon.

```
workers:
  MyComputeWorkers:
    type: "sim.c5.24xlarge"
    desired: 1
```

Permettre des simulations multi-opérateurs

Vous pouvez créer une simulation qui utilise plus d'un travailleur. Par défaut, les simulations utilisent 1 travailleur. Vous devez modifier le schéma de simulation avant de démarrer la simulation.

> **Note**
>
> Vous ne pouvez pas modifier une simulation qui a déjà commencé. Si vous souhaitez activer le mode multitravail pour une simulation en cours, vous devez d'abord arrêter et supprimer la simulation.

Pour utiliser plusieurs travailleurs, définissez le `desired` nombre d'instances de calcul sur une valeur supérieure à 1. Il existe un nombre maximum d'applications pour chaque collaborateur. Pour plus d'informations, consultezSimSpace Points de terminaison et quotas Weaver. SimSpace Weaver

n'utilisera plus d'un travailleur que lorsque le nombre d'applications d'un travailleur dépasse cette limite. SimSpace Weaver peut placer une application sur n'importe lequel des travailleurs disponibles. Le placement de l'application sur un collaborateur en particulier n'est pas garanti.

L'extrait de schéma suivant illustre une configuration pour une simulation qui demande 2 travailleurs. SimSpace Weaver tentera d'allouer le deuxième travailleur si le nombre d'applications dépasse le nombre maximum d'applications pour 1 travailleur.

```
workers:
  MyComputeWorkers:
    type: "sim.c5.24xlarge"
    desired: 2
```

Horloge

La `clock` section définit les propriétés de l'horloge de simulation. Actuellement, vous pouvez uniquement configurer le taux de ticks (le nombre de ticks par seconde que l'horloge envoie aux applications). Le taux de ticks est un taux maximum. Le taux de ticks effectif peut être inférieur car toutes les opérations (telles que les mises à jour d'entités) associées à un tick doivent être terminées avant que le prochain tick puisse commencer. Le taux de tic-tac est également appelé fréquence d'horloge.

Les valeurs valides pour `tick_rate` dépendent de celles `sdk_version` spécifiées dans votre schéma.

Valeurs valides pour le taux de ticking

- Versions antérieures à `"1.14"` :
  - `10`
  - `15`
  - `30`
- Version `"1.14"` ou ultérieure :
  - `"10"`
  - `"15"`
  - `"30"`
  - `"unlimited"`

Pour de plus amples informations, veuillez consulter Taux de ticks illimité.

> **Important**
>
> - Pour les schémas dont la valeur est `sdk_version` antérieure à `"1.14"` la valeur de `tick_rate` est un entier, tel que`30`.
> - Pour les schémas dotés d'un `"1.14"` ou `sdk_version` d'une version ultérieure, la valeur de `tick_rate` est une chaîne, telle que`"30"`. La valeur doit inclure les guillemets.
>
>   Si vous convertissez une version `"1.12"` ou un `"1.13"` schéma en version `"1.14"` ou ultérieure, vous devez placer la valeur de entre `tick_rate` guillemets.

Taux de ticks illimité

Vous pouvez définir la valeur `tick_rate` sur `"unlimited"` pour permettre à votre simulation de s'exécuter aussi vite que votre code peut s'exécuter. Avec un taux de ticks illimité, SimSpace Weaver envoie le prochain tick immédiatement après que toutes les applications ont terminé les validations pour le tick en cours.

> **Important**
>
> Le taux de ticks illimité n'est pas pris en charge dans SimSpace Weaver les versions antérieures à la version 1.14.0. La valeur minimale de `sdk_version` dans le schéma est`"1.14"`.

Taux de ticks illimité dans SimSpace Weaver Local

SimSpace Weaver Local implémente `"unlimited"` comme si le schéma spécifiait un taux de tic-tac de 10 kHz (10000). L'effet est le même qu'un taux de ticks illimité dans le AWS Cloud. Vous le spécifiez toujours `tick_rate: "unlimited"` dans votre schéma. Pour plus d'informations sur SimSpace Weaver Local, voir Développement local en SimSpace Weaver.

Questions fréquemment posées sur l'horloge

Q1. Puis-je modifier une simulation STARTED pour utiliser un taux de réussite différent ?

Vous ne pouvez modifier le taux de réussite d'une simulation qui existe déjà AWS Cloud à aucun stade de son cycle de vie. Vous ne pouvez pas non plus modifier le taux de réussite d'une simulation exécutée dans SimSpace Weaver Local. Vous pouvez le définir `tick_rate` dans le schéma et démarrer une nouvelle simulation à partir de ce schéma.

Q2. Puis-je exécuter ma simulation avec un taux de clics illimité dans une version antérieure à la version 1.14 ?

Non, le taux de ticking illimité n'est pas pris en charge dans les versions antérieures à la version 1.14.0.

Résolution des erreurs d'horloge

Si votre simulation ne démarre pas, vous pouvez vérifier la valeur de `"StartError"` dans la sortie de l'DescribeSimulationAPI. Une `tick_rate` valeur non valide dans votre schéma produira les erreurs suivantes.

> **Note**
>
> Le résultat d'erreur affiché ici est affiché sur plusieurs lignes pour améliorer la lisibilité. La sortie d'erreur réelle est une seule ligne.

- Le `sdk_version` est antérieur à `"1.14"` et la valeur de `tick_rate` est un entier non valide. Valeurs valides: `10`, `15`, `30`

```
"[{\"errorType\":\"SchemaFormatInvalid\",\"errorMessage\":
    \"$.clock.tick_rate: does not have a value in the enumeration [10, 15, 30]\"}]"
```

- Le `sdk_version` est antérieur à `"1.14"` et la valeur de `tick_rate` est une chaîne. Valeurs valides: `10`, `15`, `30`

```
"[{\"errorType\":\"SchemaFormatInvalid\",\"errorMessage\":
    \"$.clock.tick_rate: does not have a value in the enumeration [10, 15, 30]\"},
    {\"errorType\":\"SchemaFormatInvalid\",
    \"errorMessage\":\"$.clock.tick_rate: string found, integer expected\"}]"
```

- La chaîne `sdk_version` est `"1.14"` ou ultérieure et la valeur de `tick_rate` est une chaîne non valide. Valeurs valides : `"10"`, `"15"`, `"30"`, `"unlimited"`

```
"[{\"errorType\":\"SchemaFormatInvalid\",\"errorMessage\":
    \"$.clock.tick_rate: does not have a value in the enumeration [10, 15, 30,
 unlimited]\"}]"
```

- La valeur `sdk_version` est `"1.14"` ou ultérieure et la valeur de `tick_rate` est un entier. Valeurs valides : `"10"`, `"15"`, `"30"`, `"unlimited"`

```
"[{\"errorType\":\"SchemaFormatInvalid\",\"errorMessage\":
    \"$.clock.tick_rate: does not have a value in the enumeration [10, 15, 30,
 unlimited]\"},
    {\"errorType\":\"SchemaFormatInvalid\",
    \"errorMessage\":\"$.clock.tick_rate: integer found, string expected\"}]"
```

Stratégies de partitionnement

La `partitioning_strategies` section définit les propriétés de configuration pour les partitions des applications spatiales. Vous indiquez votre propre nom pour une stratégie de partitionnement (un ensemble de propriétés dans cette section) et vous l'utilisez dans la configuration de votre application spatiale.

```
partitioning_strategies:
  MyGridPartitioning:
    topology: "Grid"
    aabb_bounds:
      x: [0, 1000]
      y: [0, 1000]
    grid_placement_groups:
      x: 1
      y: 1
```

La `topology` propriété indique le type de système de coordonnées utilisé par votre simulation. La valeur `Grid` indique une grille bidimensionnelle (2D).

Pour une `Grid` topologie, l'espace de simulation est modélisé sous la forme d'un cadre de délimitation aligné sur un axe (AABB). Vous spécifiez les limites de coordonnées pour chaque axe de

votre AABB dans la propriété. `aabb_bounds` Toutes les entités qui existent dans l'espace dans votre simulation doivent avoir une position dans l'AABB.

Groupes de placement de grilles

Un groupe de placement est un ensemble de partitions d'applications spatiales que vous SimSpace Weaver souhaitez placer sur le même travailleur. Vous spécifiez le nombre et la disposition des groupes de placement (dans une grille) dans la `grid_placement_groups` propriété. SimSpace Weaver essaiera de répartir uniformément les partitions entre les groupes de placement. Les zones de propriété des applications spatiales dont les partitions appartiennent au même groupe de placement seront adjacentes dans l'espace.

Nous recommandons que x * y soit égal au nombre de travailleurs souhaité. Si ce n'est pas le cas, SimSpace Weaver vous essaierez d'équilibrer vos groupes de placement entre les travailleurs disponibles.

Si vous ne spécifiez pas de configuration de groupe de placement, SimSpace Weaver nous en calculerons une pour vous.

Domaines

Vous fournissez un nom pour un ensemble de propriétés de configuration pour un domaine. Le paramètre de lancement des applications d'un domaine détermine le type de domaine :

- **`launch_apps_via_start_app_call`**— domaine personnalisé
- **`launch_apps_by_partitioning_strategy`**— domaine spatial
- **`launch_apps_per_worker`**(non inclus dans l'exemple d'application) — domaine de service

> **Important**
>
> SimSpace Weaver prend en charge jusqu'à 5 domaines pour chaque simulation. Cela inclut tous les domaines spatiaux, personnalisés et de service.

```
domains:
  MyViewDomain:
    launch_apps_via_start_app_call: {}
    app_config:
```

```
      package: "s3://weaver-myproject-111122223333-us-west-2/MyViewApp.zip"
      launch_command: ["MyViewApp"]
      required_resource_units:
        compute: 1
      endpoint_config:
        ingress_ports:
          - 7000
  MySpatialDomain:
    launch_apps_by_partitioning_strategy:
      partitioning_strategy: "MyGridPartitioning"
      grid_partition:
        x: 2
        y: 2
    app_config:
      package: "s3://weaver-myproject-111122223333-us-west-2/MySpatialApp.zip"
      launch_command: ["MySpatialApp"]
      required_resource_units:
        compute: 1
```

> **Note**
>
> SimSpace Weaver Les projets du SDK d'application version 1.12.x utilisent des compartiments distincts pour les fichiers .zip de l'application et le schéma :
>
> - tisserand- *`lowercase-project-name`* - *`account-number`* -app-zips- *`region`*
> - tisserand- *`lowercase-project-name`* - *`account-number`* -schemas- *`region`*

Rubriques

- Configuration de l'application
- Configuration de domaines spatiaux
- Points de terminaison du réseau
- Configuration des domaines de service

Configuration de l'application

Vous spécifiez la configuration d'une application (`app_config`) dans le cadre de la configuration de son domaine. Tous les types de domaines utilisent les mêmes propriétés de configuration d'application.

```
app_config:
  package: "s3://weaver-myproject-111122223333-us-west-2/MyViewApp.zip"
  launch_command: ["MyViewApp"]
  required_resource_units:
    compute: 1
```

> **Note**
>
> SimSpace Weaver Les projets du SDK d'application version 1.12.x utilisent des compartiments distincts pour les fichiers .zip de l'application et le schéma :
>
> - tisserand- *`lowercase-project-name`* - *`account-number`* -app-zips- *`region`*
> - tisserand- *`lowercase-project-name`* - *`account-number`* -schemas- *`region`*

La `package` propriété spécifie l'URI S3 d'un fichier zip dans un compartiment S3. Le fichier zip contient le fichier exécutable de l'application (également appelé binaire) et toutes les autres ressources dont elle a besoin (telles que les bibliothèques). Chaque instance de l'exécutable de l'application s'exécute dans un Docker conteneur sur un travailleur.

La `launch_command` propriété indique le nom de l'exécutable et les options de ligne de commande permettant d'exécuter l'application. La valeur de `launch_command` est un tableau. Chaque jeton de la chaîne de commande de lancement complète est un élément du tableau.

exemple

- Pour la commande de lancement : `MyTestApp --option1 value1`
- Spécifiez : `launch_command: ["MyTestApp", "-option1", "value1"]`

La `required_resource_units` propriété indique le nombre d'unités de ressources de calcul qui SimSpace Weaver doivent être allouées à cette application. Une unité de ressource informatique est une quantité fixe de capacité de traitement (vCPU) et mémoire (RAM) sur un travailleur. Vous pouvez augmenter cette valeur pour augmenter la puissance de calcul disponible pour l'application lorsqu'elle est exécutée sur un utilisateur. Le nombre d'unités de ressources de calcul par travailleur est limité. Pour de plus amples informations, veuillez consulter SimSpace Points de terminaison et quotas Weaver.

Configuration de domaines spatiaux

Pour les domaines spatiaux, vous devez spécifier un`partitioning_strategy`. La valeur de cette propriété est le nom que vous avez donné à une stratégie de partitionnement que vous avez définie dans une autre partie du schéma.

```
MySpatialDomain:
  launch_apps_by_partitioning_strategy:
    partitioning_strategy: "MyGridPartitioning"
    grid_partition:
      x: 2
      y: 2
  app_config:
    package: "s3://weaver-myproject-111122223333-us-west-2/MySpatialApp.zip"
    launch_command: ["MySpatialApp"]
    required_resource_units:
      compute: 1
```

> **Note**
>
> SimSpace Weaver Les projets du SDK d'application version 1.12.x utilisent des compartiments distincts pour les fichiers .zip de l'application et le schéma :
>
> - tisserand- *`lowercase-project-name`* - *`account-number`* -app-zips- *`region`*
> - tisserand- *`lowercase-project-name`* - *`account-number`* -schemas- *`region`*

Une stratégie de partitionnement avec une `Grid` topologie (la seule topologie prise en charge dans cette version) indique d' SimSpace Weaver organiser les partitions d'applications spatiales de ce domaine dans une grille. La `grid_partition` propriété indique le nombre de lignes et de colonnes de la grille de partition.

SimSpace Weaver démarrera 1 instance de l'application spatiale pour chaque cellule de la grille de partition. Par exemple, si un domaine spatial comporte `x: 2` `y: 2` des `grid_partition` valeurs et qu'il comporte 2 * 2 = 4 partitions. SimSpace Weaver démarrera 4 instances de l'application configurées dans le domaine spatial et attribuera une partition à chaque instance d'application.

Rubriques

- Besoins en ressources pour les domaines spatiaux
- Domaines spatiaux multiples
- Questions fréquemment posées sur les domaines spatiaux
- Résolution des problèmes liés aux domaines spatiaux

Besoins en ressources pour les domaines spatiaux

Vous pouvez attribuer jusqu'à 17 unités de ressources de calcul à chaque travailleur. Vous spécifiez le nombre d'unités de ressources de calcul utilisées par chaque application spatiale dans la `app_config` section de votre domaine spatial.

Example extrait de schéma montrant les unités de ressources de calcul pour une application spatiale

```
  MySpatialDomain:
    launch_apps_by_partitioning_strategy:
      partitioning_strategy: "MyGridPartitioning"
      grid_partition:
        x: 2
        y: 2
    app_config:
      package: "s3://weaver-myproject-111122223333-artifacts-us-west-2/
MySpatialApp.zip"
      launch_command: ["MySpatialApp"]
      required_resource_units:
        compute: 1
```

Pour calculer le nombre d'unités de ressources de calcul requises par un domaine, multipliez le nombre de cellules de votre grille (dans votre`grid_partition`, x *y) par le nombre d'unités de ressources de calcul attribuées aux applications spatiales.

Dans l'exemple précédent, le domaine `MySpatialDomain` spécifie :

- x: 2
- y: 2
- `compute`: 1

La grille pour `MySpatialDomain` comporte 2 * 2 = 4 cellules. Le domaine spatial nécessite 4 * 1 = 4 unités de ressources de calcul.

Le nombre total d'unités de ressources de calcul pour tous les domaines spécifiés dans votre schéma doit être inférieur ou égal au `desired` nombre de travailleurs multiplié par le nombre maximum d'unités de ressources de calcul pour chaque travailleur (17).

Domaines spatiaux multiples

Vous pouvez configurer votre simulation pour utiliser plusieurs domaines spatiaux. Par exemple, vous pouvez utiliser un domaine spatial pour contrôler les principaux acteurs d'une simulation (tels que les personnes et les voitures) et un autre domaine spatial pour contrôler l'environnement.

Vous pouvez également utiliser plusieurs domaines spatiaux pour affecter différentes ressources aux différentes parties de votre simulation. Par exemple, si votre simulation comporte un type d'entité qui possède 10 fois plus d'instances d'entités qu'un autre type, vous pouvez créer différents domaines pour gérer chaque type d'entité et allouer plus de ressources au domaine comportant plus d'entités.

Important

SimSpace Weaver les versions antérieures à la version 1.14.0 ne prennent pas en charge plusieurs domaines spatiaux.

Important

AWS SimSpace Weaver Local ne prend actuellement pas en charge plusieurs domaines spatiaux. Pour plus d'informations sur SimSpace Weaver Local, voir Développement local en SimSpace Weaver.

Important

SimSpace Weaver prend en charge jusqu'à 5 domaines pour chaque simulation. Cela inclut tous les domaines spatiaux, personnalisés et de service.

Configuration de plusieurs domaines spatiaux

Pour configurer plusieurs domaines spatiaux, ajoutez les autres définitions de domaines spatiaux sous forme de sections nommées distinctes dans votre schéma. Chaque domaine doit spécifier la `launch_apps_by_partitioning_strategy` clé. Consultez l'exemple de schéma suivant.

```
sdk_version: "1.14"
workers:
  MyComputeWorkers:
    type: "sim.c5.24xlarge"
    desired: 1
clock:
  tick_rate: "30"
partitioning_strategies:
  MyGridPartitioning:
    topology: Grid
    aabb_bounds:
      x: [0, 1000]
      y: [0, 1000]
domains:
  MySpatialDomain:
    launch_apps_by_partitioning_strategy:
      partitioning_strategy: "MyGridPartitioning"
      grid_partition:
        x: 2
        y: 2
    app_config:
      package: "s3://weaver-myproject-111122223333-artifacts-us-west-2/
MySpatialApp.zip"
      launch_command: ["MySpatialApp"]
      required_resource_units:
        compute: 1
  MySecondSpatialDomain:
    launch_apps_by_partitioning_strategy:
      partitioning_strategy: "MyGridPartitioning"
      grid_partition:
        x: 2
        y: 2
    app_config:
      package: "s3://weaver-myproject-111122223333-artifacts-us-west-2/
MySpatialApp2.zip"
      launch_command: ["MySpatialApp2"]
      required_resource_units:
```

```
        compute: 1
```

Regroupement de domaines spatiaux

Dans certains scénarios, vous souhaiterez peut-être placer les partitions d'un domaine spatial sur des serveurs à côté des partitions d'un autre domaine. Cela peut améliorer les performances si ces partitions créent des abonnements interdomaines les unes aux autres.

Ajoutez la clé de niveau supérieur `placement_constraints` à votre schéma pour spécifier les domaines à SimSpace Weaver placer ensemble. La `on_workers` clé requise doit faire référence à une `workers` configuration nommée dans le schéma.

Example extrait de schéma montrant les domaines spatiaux placés ensemble

```
workers:
  MyComputeWorkers:
    type: "sim.c5.24xlarge"
    desired: 2
placement_constraints:
  - placed_together: ["MySpatialDomain", "MySecondSpatialDomain"]
    on_workers: ["MyComputeWorkers"]
```

⚠ Important

- Si vous utilisez des groupes de placement :
  - Assurez-vous que x * y est un multiple du nombre de travailleurs.
  - Assurez-vous que les valeurs des groupes de placement sont des diviseurs communs pour les dimensions de la grille des domaines que vous placez ensemble.
- Si vous n'utilisez pas de groupes de placement :
  - Assurez-vous qu'un axe de vos grilles de domaine spatial possède un diviseur commun égal au nombre de travailleurs.

Pour plus d'informations sur les groupes de placement, consultezStratégies de partitionnement.

Questions fréquemment posées sur les domaines spatiaux

Q1. Comment puis-je ajouter un autre domaine spatial à une simulation existante ?

- Pour une simulation en cours d'exécution : vous ne pouvez pas modifier la configuration d'une simulation en cours d'exécution. Modifiez la configuration du domaine dans le schéma, téléchargez le schéma et les fichiers compressés de l'application, puis lancez une nouvelle simulation.
- Pour une nouvelle simulation : ajoutez la configuration du domaine au schéma, téléchargez le schéma et les fichiers compressés de l'application, puis lancez la nouvelle simulation.

Résolution des problèmes liés aux domaines spatiaux

L'erreur suivante peut s'afficher lorsque vous essayez de démarrer votre simulation mais que la configuration de votre domaine n'est pas valide.

```
"StartError": "[{\"errorType\":\"SchemaFormatInvalid\",\"errorMessage\":
    \"We were unable to determine an arrangement of your domains that would fit
    within the provided set of workers. This can generally be resolved by
    increasing the number of workers if able, decreasing your domains\u0027
    [\u0027\u0027grid_partition\u0027\u0027] values, or adjusting the
    dimensions of your [\u0027\u0027grid_placement_groups\u0027\u0027].\"}]"
```

Causes potentielles

- Le schéma alloue plus d'unités de ressources de calcul aux applications que celles disponibles pour les travailleurs.
- SimSpace Weaver Impossible de déterminer un arrangement pour placer les domaines ensemble entre les mains des travailleurs. Cela se produit lorsque vous spécifiez plusieurs domaines spatiaux mais qu'il n'existe pas de diviseur commun (ou multiple entre les grilles de domaines, par exemple entre une grille 2x4 et une grille 3x5).

Points de terminaison du réseau

Les applications personnalisées et de service peuvent comporter des points de terminaison réseau auxquels les clients externes peuvent se connecter. Vous spécifiez une liste de numéros de port comme valeur pour `ingress_ports` le`endpoint_config`. Ces numéros de port sont à la fois TCP et UDP. L'application personnalisée ou de service doit être liée aux numéros de port que vous spécifiez dans`ingress_ports`. SimSpace Weaver alloue dynamiquement les numéros de port

lors de l'exécution et mappe ces ports aux ports dynamiques. Vous pouvez appeler le description de l'application API une fois que vos applications ont commencé à trouver les numéros de port dynamiques (réels). Pour plus d'informations, consultez le didacticiel Obtenez l'adresse IP et le numéro de port d'une application personnalisée de démarrage rapide.

```
domains:
  MyViewDomain:
    launch_apps_via_start_app_call: {}
    app_config:
      package: "s3://weaver-myproject-111122223333-us-west-2/MyViewApp.zip"
      launch_command: ["MyViewApp"]
      required_resource_units:
        compute: 1
      endpoint_config:
        ingress_ports:
          - 7000
```

> **Note**
>
> SimSpace Weaver Les projets du SDK d'application version 1.12.x utilisent des compartiments distincts pour les fichiers .zip de l'application et le schéma :
>
> - tisserand- *lowercase-project-name* - *account-number* -app-zips- *region*
> - tisserand- *lowercase-project-name* - *account-number* -schemas- *region*

> **Note**
>
> `endpoint_config`est une propriété facultative pour les applications personnalisées et les applications de service. Si vous n'en spécifiez pas`endpoint_config`, l'application n'aura pas de point de terminaison réseau.

Configuration des domaines de service

La présence de `launch_apps_per_worker:` dans une configuration de domaine indique qu'il s'agit d'un domaine de service qui possède des applications de service. SimSpace Weaver démarre et arrête les applications de service pour vous. Lors du SimSpace Weaver démarrage et de l'arrêt d'une

application, celle-ci est considérée comme ayant un cycle de vie géré. SimSpace Weaver prend actuellement en charge le démarrage d'une ou deux applications de service pour chaque travailleur.

Example Exemple de domaine configuré pour lancer une application de service sur chaque travailleur

```
domains:
  MyServiceDomain:
    launch_apps_per_worker:
      count: 1
    app_config:
      package: "s3://weaver-myproject-111122223333-app-zips-us-west-2/
PlayerConnectionServiceApp.zip"
      launch_command: ["PlayerConnectionServiceApp"]
      required_resource_units:
        compute: 1
      endpoint_config:
        ingress_ports:
          - 9000
          - 9001
```

Example Exemple de domaine configuré pour lancer 2 applications de service sur chaque travailleur

```
domains:
  MyServiceDomain:
    launch_apps_per_worker:
      count: 2
    app_config:
      package: "s3://weaver-myproject-111122223333-app-zips-us-west-2/
PlayerConnectionServiceApp.zip"
      launch_command: ["PlayerConnectionServiceApp"]
      required_resource_units:
        compute: 1
      endpoint_config:
        ingress_ports:
          - 9000
          - 9001
```

Durée maximale d'une simulation

Chaque simulation AWS SimSpace Weaver possède un paramètre de durée maximale qui indique la durée maximale pendant laquelle la simulation peut s'exécuter. Vous indiquez la durée maximale en tant que paramètre lorsque vous lancez une simulation. L'interface de programmation d'`StartSimulation`applications (API) possède un paramètre facultatif`MaximumDuration`. La valeur du paramètre est un nombre de minutes (m ou M), d'heures (h ou H) ou de jours (d ou D). Par exemple, `1h` ou `1H` signifie 1 heure. SimSpace Weaver arrête votre simulation lorsqu'elle atteint cette limite.

Valeur maximale

La valeur valide la plus élevée pour `MaximumDuration` est`14D`, ou son équivalent en heures (`336H`) ou minutes (`20160M`).

Valeur par défaut

Le paramètre `MaximumDuration` est facultatif. Si vous ne fournissez pas de valeur, SimSpace Weaver utilise une valeur de`14D`.

Valeur minimale

La valeur valide la plus basse pour `MaximumDuration` est une valeur numériquement équivalente à`0`. Par exemple, les valeurs `0M` `0H0D`, et sont toutes numériquement équivalentes à`0`.

Si vous indiquez la valeur minimale pour une durée maximale, votre simulation passe immédiatement à l'`STOPPING`état dès qu'elle atteint `STARTED` cet état.

Lancer une simulation à l'aide de la console

Vous pouvez fournir une valeur pour la durée maximale lorsque vous lancez une simulation dans la SimSpace Weaver console. Entrez la valeur dans le champ Durée maximale du formulaire des paramètres de simulation lorsque vous choisissez Démarrer la simulation.

> **Important**
>
> Si vous ne fournissez pas de valeur pour la durée maximale, SimSpace Weaver utilise la valeur par défaut (`14D`).

État d'une simulation qui atteint sa durée maximale

Lorsque s'arrête SimSpace Weaver automatiquement une simulation qui atteint sa durée maximale, le statut de la simulation est `STOPPING` (si elle est en cours) ou`STOPPED`. Dans la SimSpace Weaver console, l'état cible de la simulation est toujours`STARTED`, car il s'agit du dernier état demandé par un utilisateur.

Développement d'applications

SimSpace Weaver le développement nécessite un Amazon Linux 2 (AL2) environnement pour créer des applications car vos simulations s'exécutent sur Amazon Linux dans le AWS Cloud. Si vous utilisez Windows, vous pouvez utiliser des scripts dans le SDK de SimSpace Weaver l'application pour créer et lancer un Docker conteneur qui fonctionne AL2 avec les dépendances dont vous avez besoin pour créer des SimSpace Weaver applications. Vous pouvez également lancer un AL2 environnement utilisant Windows Subsystem for Linux (WSL), ou utilisez un natif AL2 système. Pour de plus amples informations, veuillez consulter Configurez votre environnement local pour SimSpace Weaver.

> **Note**
>
> Quelle que soit la façon dont vous configurez votre environnement de développement local, vos applications s'exécutent dans Docker conteneurs lorsque vous les téléchargez pour qu'ils s'exécutent dans le AWS Cloud. Vos applications n'ont pas d'accès direct au système d'exploitation hôte.

Flux général d'une SimSpace Weaver application

1. Créez une application
2. Boucle :
   a. Commencez la mise à jour en créant un`Transaction`.
      - Quittez la boucle si la simulation s'arrête.
   b. Traitez les événements liés à l'abonnement et à l'entité propriétaire.
   c. Mettez à jour la simulation.
   d. Validez le `Transaction` pour mettre fin à la mise à jour.

3. Détruisez l'application.

Applications spatiales

Chaque application spatiale possède une zone de propriété qui est une région spatiale du monde de la simulation. Les entités situées dans la zone de propriété d'une application spatiale sont stockées dans la partition attribuée à l'application. L'application spatiale unique possède la pleine propriété (autorisations de lecture et d'écriture) de toutes les entités au sein de la partition qui lui est attribuée. Aucune autre application ne peut écrire dans ces entités. L'application spatiale améliore l'état de ses entités. Chaque application spatiale ne possède qu'une seule partition. SimSpace Weaver utilise l'emplacement spatial d'une entité pour l'indexer et l'attribuer à une partition spatiale d'application.

Le SDK de SimSpace Weaver l'application fournit un exemple d'application. Le code source de l'application spatiale de l'exemple d'application se trouve dans le dossier suivant (utilisez le séparateur de chemin adapté à votre système d'exploitation) :

```
sdk-folder\Samples\PathfindingSample\src\SpatialApp
```

Apps personnalisées

Vous créez et utilisez des applications personnalisées pour interagir avec la simulation.

Les applications personnalisées peuvent

- Création d'entités
- S'abonner à d'autres partitions
- Valider les modifications

Flux général d'une application personnalisée

1. Créez une application
2. Abonnez-vous à une région spécifique dans la simulation :
    a. Créez un `Transaction` pour commencer la première mise à jour.
    b. Créez un abonnement pour la région spécifique.
    c. Validez le `Transaction` pour mettre fin à la première mise à jour.
3. Boucle :

a. Créez un `Transaction` pour commencer la mise à jour.

   - Quittez la boucle si la simulation s'arrête.

b. Modifications de l'état du processus.

c. Validez le `Transaction` pour mettre fin à la mise à jour.

4. Détruisez l'application.

Une fois qu'une application personnalisée a créé une entité, elle doit la transférer dans un domaine spatial pour que l'entité existe spatialement dans la simulation. SimSpace Weaver utilise l'emplacement spatial de l'entité pour placer l'entité dans la partition spatiale d'application appropriée. L'application personnalisée qui a créé l'entité ne peut ni mettre à jour ni supprimer l'entité après l'avoir transférée dans un domaine spatial.

Le SDK de SimSpace Weaver l'application fournit un exemple d'application. Vous pouvez utiliser les applications personnalisées incluses dans l'exemple d'application comme modèles pour vos propres applications personnalisées. Vous trouverez le code source de l'application View (une application personnalisée) de l'exemple d'application dans le dossier suivant (utilisez le séparateur de chemin adapté à votre système d'exploitation) :

```
sdk-folder\Samples\PathfindingSample\src\ViewApp
```

Développement d'applications clientes

Parmi les raisons pour lesquelles vous souhaiterez peut-être connecter un client à une simulation, citons :

- Injectez des informations de trafic en temps réel dans une simulation à l'échelle d'une ville.
- Créez human-in-the-loopdes simulations dans lesquelles un opérateur humain contrôle certains aspects de la simulation.
- Permettez aux utilisateurs d'interagir avec la simulation, par exemple pour une simulation d'entraînement.

Les applications personnalisées présentées dans ces exemples font office d'interface entre l'état de simulation et le monde extérieur. Les clients se connectent aux applications personnalisées pour interagir avec la simulation.

SimSpace Weaver ne gère pas les applications clientes et leur communication avec vos applications personnalisées. Vous êtes responsable de la conception, de la création, du fonctionnement et de la sécurité de vos applications clientes, ainsi que de leur communication avec vos applications personnalisées. SimSpace Weaver expose uniquement une adresse IP et un numéro de port pour chacune de vos applications personnalisées afin que les clients puissent s'y connecter.

Le SDK de SimSpace Weaver l'application fournit des clients pour son exemple d'application. Vous pouvez utiliser ces clients comme modèles pour vos propres applications clientes. Le code source des exemples de clients d'applications se trouve dans le dossier suivant :

Docker

```
sdk-folder\packaging-tools\clients\PathfindingSampleClients
```

WSL

> **Important**
>
> Nous vous fournissons ces instructions pour vous faciliter la tâche. Ils sont destinés à être utilisés avec Windows Subsystem for Linux (WSL), et ne sont pas pris en charge. Pour de plus amples informations, veuillez consulter Configurez votre environnement local pour SimSpace Weaver.

```
sdk-folder/packaging-tools/clients/PathfindingSampleClients
```

Pour plus d'informations sur la création et l'utilisation des exemples de clients d'applications, consultez les didacticiels dansCommencer avec SimSpace Weaver.

Obtenez l'adresse IP et le numéro de port d'une application personnalisée

Pour visualiser votre simulation, vous créez une application personnalisée et vous vous y connectez avec un client. Pour plus d'informations, consultez les didacticiels dansCommencer avec SimSpace Weaver. Vous pouvez utiliser la procédure suivante pour obtenir l'adresse IP et le numéro de port de votre application personnalisée. Utilisez le séparateur de chemin adapté à votre système d'exploitation (par exemple, \ sous Windows et / sous Linux).

Pour obtenir votre adresse IP et votre numéro de port

1. Utilisez l' ListSimulationsAPI pour obtenir le nom de votre simulation.

```
aws simspaceweaver list-simulations
```

Exemple de sortie :

```
{
    "Simulations": [
        {
            "Status": "STARTED",
            "CreationTime": 1664921418.09,
            "Name": "MyProjectSimulation_22-10-04_22_10_15",
            "Arn": "arn:aws:simspaceweaver:us-west-2: 111122223333:simulation/
MyProjectSimulation_22-10-04_22_10_15",
            "TargetStatus": "STARTED"
        }
    ]

}
```

2. Utilisez l' DescribeSimulationAPI pour obtenir la liste des domaines de votre simulation.

```
aws simspaceweaver describe-simulation --simulation simulation-name
```

Recherchez la `Domains` section dans la `LiveSimulationState` section de la sortie.

Exemple de sortie :

```
    "LiveSimulationState": {
        "Domains": [
            {
                "Type": "",
                "Name": "MySpatialSimulation",
                "Lifecycle": "Unknown"
            },
            {
```

```
                "Type": "",
                "Name": "MyViewDomain",
                "Lifecycle": "ByRequest"
            }
        ],
```

3. Utilisez l' ListAppsAPI pour obtenir la liste des applications personnalisées d'un domaine. Par exemple, le nom de domaine de l'application d'affichage (personnalisée) dans l'exemple de projet est`MyViewDomain`. Recherchez le nom de l'application dans le résultat.

```
aws simspaceweaver list-apps --simulation simulation-name --domain domain-name
```

Exemple de sortie :

```
{
    "Apps": [
        {
            "Status": "STARTED",
            "Domain": "MyViewDomain",
            "TargetStatus": "STARTED",
            "Name": "ViewApp",
            "Simulation": "MyProjectSimulation_22-10-04_22_10_15"
        }
    ]
}
```

4. Utilisez l' DescribeAppAPI pour obtenir l'adresse IP et le numéro de port. Pour l'exemple de projet, le nom de domaine est `MyViewDomain` et le nom de l'application est`ViewApp`.

```
aws simspaceweaver describe-app --simulation simulation-name --domain domain-name
 --app app-name
```

L'adresse IP et le numéro de port se trouvent dans le `EndpointInfo` bloc de sortie. L'adresse IP est la valeur de `Address` et le numéro de port est la valeur de`Actual`.

Exemple de sortie :

```
{
    "Status": "STARTED",
    "Domain": "MyViewDomain",
    "TargetStatus": "STARTED",
    "Simulation": "MyProjectSimulation_22-10-04_22_10_15",
    "LaunchOverrides": {
        "LaunchCommands": []
    },
    "EndpointInfo": {
        "IngressPortMappings": [
            {
                "Declared": 7000,
                "Actual": 4321
            }
        ],
        "Address": "198.51.100.135"
    },
    "Name": "ViewApp"
}
```

> **Note**
>
> La valeur de `Declared` est le numéro de port auquel le code de votre application doit être lié. La valeur de `Actual` est le numéro de port qui SimSpace Weaver permet aux clients de se connecter à votre application. SimSpace Weaver mappe le `Declared` port par `Actual` rapport au port.

Lancement du client de visualisation Unreal Engine

Naviguez vers :

```
sdk-folder/Samples/PathfindingSample/tools/cloud
```

1. Exécutez une des commandes suivantes :

- Docker : `python quick-start.py`
- WSL : `python quick-start.py --al2`

2. Obtenez l'adresse IP et le numéro de port « réel ». Ils figureront dans la sortie de console après l'exécution de quick-start.py, ou obtenez-les en suivant les procédures surObtenez l'adresse IP et le numéro de port d'une application personnalisée.
3. Naviguez vers :

```
sdk-folder/Clients/TCP/UnrealClient/lib
```

4. Exécutez les commandes suivantes pour créer la bibliothèque NNG :

```
cmake -S . -B build
cmake --build build --config RelWithDebInfo
cmake --install build
```

5. Dans un éditeur de texte, ouvrez`view_app_url.txt`.
6. Mettez à jour l'URL avec l'adresse IP et le numéro de port de votre application View : `tcp://ip-address:actual-port-number` (cela devrait ressembler à`tcp://198.51.100.135:1234`).
7. Dans l'éditeur Unreal, choisissez Play.

Résolution des problèmes

- L'étape d' CMake installation de NNG échoue avec le message « Peut-être avez-vous besoin de privilèges administratifs » :

```
CMake Error at build/_deps/nng-build/src/cmake_install.cmake:39 (file):
  file cannot create directory: C:/Program Files
  (x86)/ThirdPartyNngBuild/lib.  Maybe need administrative privileges.
Call Stack (most recent call first):
  build/_deps/nng-build/cmake_install.cmake:37 (include)
  build/cmake_install.cmake:73 (include)
```

  - Résolution : si elle `nng.lib` `nng.so` existe dans le répertoire UnrealClient /lib, cette erreur peut être ignorée en toute sécurité. Sinon, essayez d'exécuter les commandes cmake build dans un terminal avec des privilèges d'administrateur.
- « CMake pour trouver un fichier de configuration de package fourni par nng » :

```
CMake Error at CMakeLists.txt:23 (find_package):
By not providing "Findnng.cmake" in CMAKE_MODULE_PATH this project has
 asked CMake to find a package configuration file provided by "nng", but
 CMake did not find one.
```

- Résolution : vous CMake ne parvenez pas à trouver le `Findnng.cmake` fichier. Lorsque vous créez avec CMake, ajoutez l'argument`-DTHIRD_PARTY_LIB_PATH sdk-folder/ThirdParty`. Assurez-vous que le `Findnng.cmake` fichier se trouve toujours dans le `ThirdParty` répertoire avant de relancer la CMake compilation.

  ```
  cmake -S . -B build -DTHIRD_PARTY_LIB_PATH sdk-folder/ThirdParty
  cmake --build build --config RelWithDebInfo
  cmake --install build
  ```

Développement local en SimSpace Weaver

Vous pouvez déployer vos SimSpace Weaver applications localement pour des tests et un débogage rapides.

Prérequis

- Suivez les étapes de Configuration pour SimSpace Weaver.

Rubriques

- Étape 1 : Lancez votre simulation locale
- Étape 2 : Afficher votre simulation locale
- Étape 3 : Arrêtez votre simulation locale (facultatif sous Windows)
- Résolution des problèmes liés au développement local dans SimSpace Weaver

Étape 1 : Lancez votre simulation locale

1. Naviguez vers

   ```
   cd sdk-folder/Samples/sample-name/tools/local
   ```

2. Exécutez la commande suivante pour créer et lancer votre simulation localement.

```
python quick-start.py
```

Ce script effectuera les opérations suivantes :

1. Générez le projet.
   - `quick-start.py`appelle la `build_project` fonction définie dans le fichier build.py. Cette étape varie en fonction du projet. Pour le PathfindingSample, CMake est utilisé. La commande CMake et Docker pour laquelle se trouve dans le fichier build.py.
2. Lancez votre simulation locale
   - Le script lancera un processus local pour chaque partition spatiale définie dans le schéma.
   - Le script lancera un processus pour chaque application personnalisée définie dans le schéma.
   - Les applications spatiales seront lancées en premier, suivies des applications personnalisées, chacune dans l'ordre dans lequel elle apparaît dans le schéma.

> **Important**
>
> Lors du lancement dans un environnement qui ne prend pas en charge l'interface graphique, tel qu'une session SSH de console, utilisez l'`--noappwindow`option permettant de rediriger toutes les sorties vers le terminal actuel.

> **Important**
>
> Pour les utilisateurs de Linux, le script suppose que votre système possède la `xterm` commande. Si votre distribution Linux ne dispose pas de cette `xterm` commande, utilisez l'`--noappwindow`option pour rediriger toutes les sorties vers le terminal actuel.

- -h, --help
  - Répertoriez ces paramètres.
- --propre
  - Supprimez le contenu du répertoire de construction avant de le compiler.
- --aucune construction

  - Ignorez la reconstruction du projet.
- --aucune fenêtre d'application
  - N'ouvrez pas de nouvelle fenêtre pour chaque application. Redirigez plutôt la sortie standard vers le terminal actuel.
- --fichier journal
  - Écrire le résultat de la console dans un fichier journal.
- --client de console
  - Connectez automatiquement le client de console répertorié dans la configuration.
- --schema SCHÉMA
  - Quel schéma utilisera cette invocation. La valeur par défaut est « SCHEMA » dans le fichier config.py.

Étape 2 : Afficher votre simulation locale

Pour visualiser votre simulation locale, vous pouvez utiliser n'importe quel client inclus dans le SimSpaceWeaverAppSdkDistributable. Pour plus d'informations sur la création et l'utilisation des exemples de clients, consultez les didacticiels dansCommencer avec SimSpace Weaver.

Vous devez mettre à jour l'adresse IP et le numéro de port du client pour vous connecter à l'application View pour votre simulation locale. Utilisez toujours les valeurs suivantes avec SimSpace Weaver Local:

```
tcp://127.0.0.1:7000
```

En fonction du client sélectionné, vous pouvez mettre à jour l'adresse IP et le numéro de port comme suit :

- Unreal — Modifiez l'URL à la ligne 1 de `view_app_url.txt`
- Console — Lance le client avec l'adresse IP et l'URL du numéro de port comme paramètre

Étape 3 : Arrêtez votre simulation locale (facultatif sous Windows)

> **Note**
>
> Cette étape est obligatoire sous Linux mais facultative sous Windows.

1. Naviguez vers :

   ```
   sdk-folder/Samples/sample-name/tools/local
   ```

2. Exécutez la commande suivante pour arrêter votre simulation locale et supprimer toutes les ressources de mémoire partagée.

   ```
   python stop-and-delete.py
   ```

   Ce script effectuera les opérations suivantes :

   - Arrêtez les processus locaux.
   - Supprimez l'objet de mémoire partagée (uniquement nécessaire sous Linux).

stop-and-deleteparamètres .py

- -h, --help
  - Répertoriez ces paramètres.
- --arrête
  - Essayez uniquement d'arrêter les processus.
- --supprimer
  - Essayez uniquement de supprimer les ressources de mémoire partagée.
- --processus
  - Nom du processus à arrêter. Utilisez-le si le nom de votre processus ne correspond pas au nom du package indiqué dans le schéma.
- --schema SCHÉMA
  - Quel schéma utilisera cette invocation. La valeur par défaut est « SCHEMA » dans le fichier config.py.

Résolution des problèmes liés au développement local dans SimSpace Weaver

- Linux : commande xterm introuvable/impossible d'ouvrir
  - Les scripts locaux supposent que la commande xterm existe lorsqu'elle est exécutée sous Linux. Si vous ne disposez pas de la commande xterm ou si vous utilisez un environnement qui ne prend pas en charge l'interface graphique, utilisez cette `--noappwindow` option lors de l'exécution du script de démarrage rapide.
- Aucune fenêtre d'application ne s'ouvre !
  - Cela se produit lorsque la simulation locale se bloque immédiatement. Pour voir le résultat de la console après le crash, utilisez les `--logfile` options `--noappwindow` ou lorsque vous exécutez le script de démarrage rapide.
- La simulation ne démarre pas une fois que l'application View démarre ou que le client View se connecte !
  - L'utilisation de `–noappwindow` cette option permet généralement de résoudre ce type de problèmes. Sinon, le redémarrage à quelques reprises est également un succès (bien qu'à un rythme beaucoup plus faible).

AWS SimSpace Weaver SDK de l'application

Le SDK de SimSpace Weaver l'application APIs vous permet de contrôler les entités de votre simulation et de répondre aux SimSpace Weaver événements. Il inclut l'espace de noms suivant :

- API — définitions fondamentales de l'API et de son utilisation

Lien vers la bibliothèque suivante :

- `libweaver_app_sdk_cxx_v1_full.so`

> **Important**
>
> La bibliothèque est disponible pour les liens dynamiques lorsque vous exécutez vos applications dans le AWS Cloud. Vous n'avez pas besoin de le télécharger avec vos applications.

> **Note**
>
> Le SDK de SimSpace Weaver l'application APIs contrôle les données de votre simulation. Ils APIs sont distincts du SimSpace Weaver service APIs, qui contrôlent les ressources de votre SimSpace Weaver service (telles que les simulations, les applications et les horloges). AWS Pour de plus amples informations, veuillez consulter SimSpace Weaver Références d'API.

Rubriques

- Les méthodes d'API renvoient un Result
- Interaction avec le SDK de l'application au plus haut niveau
- Gestion des simulations
- Abonnements
- Entités
- Événements organisés par les entités
- Result et gestion des erreurs
- Génériques et types de domaines
- Opérations diverses du SDK de l'application

Les méthodes d'API renvoient un Result

La majorité des fonctions d' SimSpace Weaver API ont un type de retour`Aws::WeaverRuntime::Result<T>`. Si la fonction s'est exécutée avec succès, le `Result` contient`T`. Dans le cas contraire, `Aws::WeaverRuntime::ErrorCode` il `Result` contient un code d'erreur provenant du Rust App SDK.

Example exemple

```
Result<Transaction> BeginUpdate(Application& app)
```

Cette méthode :

- Renvoie `Transaction` si l'`BeginUpdate()`exécution est réussie.
- Renvoie en `Aws::WeaverRuntime::ErrorCode` cas d'`BeginUpdate()`échec.

Interaction avec le SDK de l'application au plus haut niveau

Cycle de vie

- Le SDK de SimSpace Weaver l'application gère le cycle de vie de l'application. Il n'est pas nécessaire de lire ou d'écrire l'état du cycle de vie d'une application.

Partitions

- `Result <PartitionSet> AssignedPartitions(Transaction& txn);` À utiliser pour obtenir des partitions possédées.
- `Result <PartitionSet> AllPartitions(Transaction& txn);` À utiliser pour obtenir toutes les partitions de la simulation.

Gestion des simulations

Cette section décrit les solutions pour les tâches courantes de gestion de simulation.

Rubriques

- Lancer une simulation
- Mettre à jour une simulation
- Mettre fin à une simulation

Lancer une simulation

`CreateApplication()` À utiliser pour créer une application.

Example exemple

```
Result<Application> applicationResult = Api::CreateApplication();

if (!applicationResult)
{
    ErrorCode errorCode = WEAVERRUNTIME_EXPECT_ERROR(applicationResult);

    std::cout << "Failed to create application. Error code " <<
        static_cast<std::underlying_type_t<ErrorCode>>(errorCode) <<
```

```
        " Last error message "<< Api::LastErrorMessage() << ".";

    return 1;
}

/**
* Run simulation
*/
RunSimulation(std::move(applicationResult.assume_value()));
```

Mettre à jour une simulation

Utilisez les `BeginUpdate` fonctions suivantes pour mettre à jour l'application :

- `Result<Transaction> BeginUpdate(Application& app)`
- `Result<bool> BeginUpdateWillBlock(Application& app)`— vous indique si vous `BeginUpdate()` allez bloquer ou non.

`Result<void> Commit(Transaction& txn)`À utiliser pour valider les modifications.

Example exemple

```
Result<void> AppDriver::RunSimulation(Api::Application app) noexcept
{
    while (true)
    {
        {
            bool willBlock;

            do
            {
                WEAVERRUNTIME_TRY(willBlock, Api::BeginUpdateWillBlock(m_app));
            } while (willBlock);
        }

        WEAVERRUNTIME_TRY(Transaction transaction, Api::BeginUpdate(app));

        /**
         * Simulate app.
         */
        WEAVERRUNTIME_TRY(Simulate(transaction));
```

```
        WEAVERRUNTIME_TRY(Api::Commit(std::move(transaction)));
    }

    return Success();
}
```

Mettre fin à une simulation

`Result<void> DestroyApplication(Application&& app)`À utiliser pour mettre fin à l'application et à la simulation.

D'autres applications découvrent que la simulation s'arrête lorsqu'elles reçoivent `ErrorCode::ShuttingDown` des appels vers `BeginUpdateWillBlock()` ou`BeginUpdate()`. Lorsqu'une application reçoit`ErrorCode::ShuttingDown`, elle peut appeler `Result<void> DestroyApplication(Application&& app)` pour se terminer elle-même.

Example exemple

```
Result<void> AppDriver::EncounteredAppError(Application&& application) noexcept
{
    const ErrorCode errorCode = WEAVERRUNTIME_EXPECT_ERROR(runAppResult);

    switch (errorCode)
    {
    case ErrorCode::ShuttingDown:
        {
            // insert custom shutdown process here.

            WEAVERRUNTIME_TRY(Api::DestroyApplication(std::move(application)));
            return Success();
        }
    default:
        {
            OnAppError(errorCode);
            return errorCode;
        }
    }
}
```

> **Important**
>
> N'appelez qu'`Result<void> DestroyApplication(Application&& app)`après`Api::Commit()`. La destruction d'une application lors d'une mise à jour peut entraîner un comportement indéfini.

> **Important**
>
> Vous devez appeler `DestroyApplication()` avant la fin du programme pour vous assurer que l'application indique qu'elle s'est terminée correctement.
> Si `DestroyApplication()` vous n'appelez pas à la fin du programme, le statut sera considéré comme`FATAL`.

Abonnements

Vous créez un abonnement avec un espace d'abonnement et un identifiant de domaine. L'ID de domaine représente le domaine propriétaire de cette zone d'abonnement. A `BoundingBox2F32` décrit la zone d'abonnement. Utilisez la fonction suivante pour créer un abonnement :

```
Result<SubscriptionHandle> CreateSubscriptionBoundingBox2F32(Transaction& txn, DomainId
 id, const BoundingBox2F32& boundingBox)
```

Example exemple

```
Result<void> CreateSubscriptionInSpatialDomain(Transaction& transaction)
{
    WEAVERRUNTIME_TRY(Api::PartitionSet partitionSet, Api::AllPartitions(transaction));


    Api::DomainId spatialDomainId;

    for (const Api::Partition& partition : partitionSet.partitions)
    {
        if (partition.domain_type == Api::DomainType::Spatial)
        {
            /**
            * Get the spatial domain ID.
```

```
            */
            spatialDomainId = partition.domain_id;
            break;
        }
    }

    constexpr Api::BoundingBox2F32 subscriptionBounds {
        /* min */ { /* x */ 0, /* y */ 0 },
        /* max */ { /* x */ 1000, /* y */ 1000 } }

    WEAVERRUNTIME_TRY(
        Api::SubscriptionHandle subscriptionHandle,
        Api::CreateSubscriptionBoundingBox2F32(
        transaction,
        spatialDomainId,
        subscriptionBounds));

    return Success();
}
```

Vous pouvez utiliser le `Api::SubscriptionHandle` code renvoyé `CreateSubscriptionBoundingBox2F32()` pour modifier l'abonnement. Vous le transmettez en argument aux fonctions suivantes :

```
Result<void> ModifySubscriptionBoundingBox2F32(Transaction& txn, SubscriptionHandle
 handle, const BoundingBox2F32& boundingBox)
```

```
Result<void> DeleteSubscription(Transaction& txn, SubscriptionHandle handle)
```

Entités

Vous appelez le `Store` et `Load` APIs en utilisant `Api:Entity` l'événement `Result<Api::Entity>` renvoyé par ou depuis `CreateEntity()` un changement de propriétaire lorsqu'une entité entre dans la zone d'abonnement de l'application (pour plus d'informations, voirÉvénements organisés par les entités). Nous vous recommandons de suivre vos `Api::Entity` objets afin de pouvoir les utiliser avec eux APIs.

Rubriques

- Création d'entités
- Transférer une entité vers un domaine spatial

- Écrire et lire les données des champs d'entités
- Mémoriser la position d'une entité
- Charger la position d'une entité

Création d'entités

`CreateEntity()`À utiliser pour créer une entité. Vous définissez le sens de ce `Api::TypeId` que vous transmettez à cette fonction.

```
Namespace
{
    constexpr Api::TypeId k_entityTypeId { /* value */ 512 };
}

Result<void> CreateEntity(Transaction& transaction)
{
    WEAVERRUNTIME_TRY(
        Api::Entity entity,
        Api::CreateEntity(
            transaction, Api::BuiltinTypeIdToTypeId(k_entityTypeId )));
}
```

> **Note**
>
> Les valeurs 0 à 511 pour `Api::BuiltinTypeId` sont réservées. Votre entité TypeID (`k_entityTypeId`dans cet exemple) doit avoir une valeur supérieure ou égale à 512.

Transférer une entité vers un domaine spatial

Une fois qu'une application ou une application de service personnalisée a créé une entité, l'application doit transférer l'entité vers un domaine spatial pour que l'entité existe spatialement dans la simulation. Les entités d'un domaine spatial peuvent être lues par d'autres applications et mises à jour par une application spatiale. Utilisez l'`ModifyEntityDomain()`API pour transférer une entité vers un domaine spatial.

```
AWS_WEAVERRUNTIME_API Result<void> ModifyEntityDomain(Transaction& txn, const Entity&
 entity, DomainId domainId) noexcept;
```

Si le `DomainId` ne correspond pas à celui attribué par `Partition` l'application d'appel, cela `DomainId` doit être pour un `DomainType::SpatialDomain`. Le transfert de propriété vers le nouveau `Domain` a lieu pendant le`Commit(Transaction&&)`.

Paramètres

`txn`

Le courant`Transaction`.

`entity`

L'objectif `Entity` du changement de`Domain`.

`domainId`

Le `DomainId` de la destination `Domain` pour le`Entity`.

Cette API est renvoyée `Success` si le domaine de l'entité a été modifié avec succès.

Écrire et lire les données des champs d'entités

Tous les champs de données d'entité sont de type blob. Vous pouvez écrire jusqu'à 1 024 octets de données dans une entité. Nous vous recommandons de réduire au maximum la taille des blobs, car des tailles plus importantes réduiront les performances. Lorsque vous écrivez dans un blob, vous passez SimSpace Weaver un pointeur sur les données et leur longueur. Lorsque vous lisez à partir d'un blob, il vous SimSpace Weaver fournit un pointeur et une longueur à lire. Toutes les lectures doivent être terminées avant l'appel de l'application`Commit()`. Les pointeurs renvoyés par un appel de lecture sont invalidés lorsque l'application appelle. `Commit()`

> ⚠ Important
>
> - La lecture à partir d'un pointeur blob mis en cache après a `Commit()` n'est pas prise en charge et peut entraîner l'échec de la simulation.
> - L'écriture sur un pointeur blob renvoyé par un appel de lecture n'est pas prise en charge et peut entraîner l'échec de la simulation.

Rubriques

- Stocker les données de champ d'une entité

- Charger les données de champ d'une entité
- Chargement des données de champ des entités supprimées

Stocker les données de champ d'une entité

Les exemples suivants montrent comment stocker (écrire dans le State Fabric) les données de champ d'une entité détenue par l'application. Ces exemples utilisent la fonction suivante :

```
AWS_WEAVERRUNTIME_API Result<void> StoreEntityField(
    Transaction& txn,
    const Entity& entity,
    TypeId keyTypeId,
    FieldIndex index,
    std::int8_t* src,
    std::size_t length) noexcept;
```

Le `Api::TypeId keyTypeId` paramètre représente le type de données des données transmises.

Le `Api::TypeId keyTypeId` paramètre doit recevoir le formulaire correspondant `Api::TypeIdApi::BuiltinTypeId`. S'il n'y a pas de conversion appropriée, vous pouvez utiliser`Api::BuiltinTypeId::Dynamic`.

Pour les types de données complexes, utilisez`Api::BuiltInTypeId::Dynamic`.

> **Note**
>
> La valeur de `FieldIndex index` doit être supérieure à 0. La valeur 0 est réservée à la clé d'index (voir`StoreEntityIndexKey()`).

Example Exemple d'utilisation de types de données primitifs

```
namespace
{
    constexpr Api::FieldIndex k_isTrueFieldId { /* value */ 1 };
}

Result<void> SetEntityFields(
    Api::Entity& entity,
    Transaction& transaction)
```

```
{
    bool value = true;

    auto* src = reinterpret_cast<std::int8_t*>(value);
    size_t length = sizeof(*value);

    WEAVERRUNTIME_TRY(Api::StoreEntityField(
        transaction,
        entity,
        Api::BuiltinTypeIdToTypeId(
            Aws::WeaverRuntime::Api::BuiltinTypeId::Bool),
        k_isTrueFieldId,
        src,
        length));
}
```

Example Exemple utilisant un struct pour conserver les données

```
namespace
{
    constexpr Api::FieldIndex k_dataFieldId { /* value */ 1 };
}

struct Data
{
    bool boolData;
    float floatData;
};

Result<void> SetEntityFields(
    Api::Entity& entity,
    Transaction& transaction)
{
    Data data = { /* boolData */ false, /* floatData */ -25.93 };

    auto* src = reinterpret_cast<std::int8_t*>(data);
    size_t length = sizeof(*data);

    WEAVERRUNTIME_TRY(Api::StoreEntityField(
        transaction,
        entity,
        Api::BuiltinTypeIdToTypeId(
            Aws::WeaverRuntime::Api::BuiltinTypeId::Dynamic),
```

```
        k_dataFieldId,
        src,
        length));
}
```

Charger les données de champ d'une entité

Les exemples suivants montrent comment charger (lire depuis le State Fabric) les données de champ d'une entité. Ces exemples utilisent la fonction suivante :

```
Result<std::size_t> LoadEntityField(
    Transaction& txn,
    const Entity& entity,
    TypeId keyTypeId,
    FieldIndex index,
    std::int8_t** dest) noexcept;
```

Le `Api::TypeId keyTypeId` paramètre doit recevoir le formulaire correspondant `Api::TypeIdApi::BuiltinTypeId`. S'il n'y a pas de conversion appropriée, vous pouvez utiliser`Api::BuiltinTypeId::Dynamic`.

> **Note**
>
> La valeur de l'`FieldIndex`index doit être supérieure à 0. La valeur 0 est réservée à la clé d'index (voir`StoreEntityIndexKey()`).

Example Exemple d'utilisation de types de données primitifs

```
namespace
{
    constexpr Api::FieldIndex k_isTrueFieldId { /* value */ 1 };
}

Result<void> LoadEntityFields(
    Api::Entity& entity,
    Transaction& transaction)
{
    std::int8_t* dest = nullptr;

    WEAVERRUNTIME_TRY(Api::LoadEntityField(
        transaction,
```

```
        entity,
        Api::BuiltinTypeIdToTypeId(
            Aws::WeaverRuntime::Api::BuiltinTypeId::Bool),
        k_isTrueFieldId,
        &dest));

    bool isTrueValue = *reinterpret_cast<bool*>(dest);
}
```

Example Exemple utilisant un struct pour conserver les données

```
namespace
{
    constexpr Api::FieldIndex k_dataFieldId { /* value */ 1 };
}

struct Data
{
    bool boolData;
    float floatData;
};

Result<void> LoadEntityFields(
    Api::Entity& entity,
    Transaction& transaction)
{
    std::int8_t* dest = nullptr;

    WEAVERRUNTIME_TRY(Api::LoadEntityField(
        transaction,
        entity,
        Api::BuiltinTypeIdToTypeId(
            Aws::WeaverRuntime::Api::BuiltinTypeId::Dynamic),
        k_dataFieldId,
        &dest));

    Data dataValue = *reinterpret_cast<Data*>(dest);
}
```

Chargement des données de champ des entités supprimées

Vous ne pouvez pas charger (lire depuis le State Fabric) les données des champs d'entité pour les entités qui ont été supprimées des zones de propriété et d'abonnement de l'application.

L'exemple suivant génère une erreur car il appelle `Api::LoadIndexKey()` une entité à la suite d'un`Api::ChangeListAction::Remove`. Le deuxième exemple montre une méthode correcte pour stocker et charger les données d'entité directement dans l'application.

Example Exemple de code incorrect

```
Result<void> ProcessSubscriptionChanges(Transaction& transaction)
{
    /* ... */

    WEAVERRUNTIME_TRY(Api::SubscriptionChangeList subscriptionChangeList,
        Api::AllSubscriptionEvents(transaction));

    for (const Api::SubscriptionEvent& event :
        subscriptionChangeList.changes)
    {
        switch (event.action)
        {
        case Api::ChangeListAction::Remove:
            {
                std::int8_t* dest = nullptr;

                /**
                 * Error!
                 * This calls LoadEntityIndexKey on an entity that
                 * has been removed from the subscription area.
                 */
                WEAVERRUNTIME_TRY(Api::LoadEntityIndexKey(
                    transaction,
                    event.entity,
                    Api::BuiltinTypeIdToTypeId(
                        Api::BuiltinTypeId::Vector3F32),
                    &dest));

                AZ::Vector3 position =
                    *reinterpret_cast<AZ::Vector3*>(dest);
                break;
            }
        }

    }

    /* ... */
```

```
}
```

Example Exemple de méthode correcte pour stocker et charger les données d'entité dans l'application

```
Result<void> ReadAndSaveSubscribedEntityPositions(Transaction& transaction)
{
    static std::unordered_map<Api::EntityId, AZ::Vector3>
        positionsBySubscribedEntity;

    WEAVERRUNTIME_TRY(Api::SubscriptionChangeList subscriptionChangeList,
        Api::AllSubscriptionEvents(transaction));

    for (const Api::SubscriptionEvent& event :
        subscriptionChangeList.changes)
    {
        switch (event.action)
        {
        case Api::ChangeListAction::Add:
            {
                std::int8_t* dest = nullptr;

                /**
                 * Add the position when the entity is added.
                 */
                WEAVERRUNTIME_TRY(Api::LoadEntityIndexKey(
                    transaction,
                    event.entity,
                    Api::BuiltinTypeIdToTypeId(
                        Api::BuiltinTypeId::Vector3F32),
                    &dest));

                AZ::Vector3 position =
                    *reinterpret_cast<AZ::Vector3*>(dest);
                positionsBySubscribedEntity.emplace(
                    event.entity.descriptor->id, position);

                break;
            }
        case Api::ChangeListAction::Update:
            {
                std::int8_t* dest = nullptr;
```

```
                /**
                 * Update the position when the entity is updated.
                 */
                WEAVERRUNTIME_TRY(Api::LoadEntityIndexKey(
                    transaction,
                    event.entity,
                    Api::BuiltinTypeIdToTypeId(
                        Api::BuiltinTypeId::Vector3F32),
                    &dest));

                AZ::Vector3 position =
                    *reinterpret_cast<AZ::Vector3*>(dest);
                positionsBySubscribedEntity[event.entity.descriptor->id] =
                    position;

                break;
            }
        case Api::ChangeListAction::Remove:
            {
                /**
                 * Load the position when the entity is removed.
                 */
                AZ::Vector3 position = positionsBySubscribedEntity[
                    event.entity.descriptor->id];

                /**
                 * Do something with position...
                 */
                break;
            }
        }
    }

    /* ... */
}
```

Mémoriser la position d'une entité

Vous pouvez enregistrer (écrire dans le State Fabric) la position d'une entité à l'aide d'une structure de données entière. Ces exemples utilisent la fonction suivante :

```
Result<void> StoreEntityIndexKey(
    Transaction& txn,
```

```
    const Entity& entity,
    TypeId keyTypeId,
    std::int8_t* src,
    std::size_t length)
```

> **Note**
>
> Vous devez fournir `Api::BuiltinTypeId::Vector3F32` à`Api::StoreEntityIndexKey()`, comme indiqué dans les exemples suivants.

Example Exemple d'utilisation d'un tableau pour représenter la position

```
Result<void> SetEntityPositionByFloatArray(
    Api::Entity& entity,
    Transaction& transaction)
{
    std::array<float, 3> position = { /* x */ 25, /* y */ 21, /* z */ 0 };

    auto* src = reinterpret_cast<std::int8_t*>(position.data());
    std::size_t length = sizeof(position);

    WEAVERRUNTIME_TRY(Api::StoreEntityIndexKey(
        transaction,
        entity,
        Api::BuiltinTypeIdToTypeId(Api::BuiltinTypeId::Vector3F32),
        src,
        length));
}
```

Example Exemple utilisant un struct pour représenter la position

```
struct Position
{
   float x;
   float y;
   float z;
};

Result<void> SetEntityPositionByStruct(
```

```
    Api::Entity& entity,
    Transaction& transaction)
{
    Position position = { /* x */ 25, /* y */ 21, /* z */ 0 };

    auto* src = reinterpret_cast<std::int8_t*>(&position);
    std::size_t length = sizeof(position);

    WEAVERRUNTIME_TRY(Api::StoreEntityIndexKey(
        transaction,
        entity,
        Api::BuiltinTypeIdToTypeId(Api::BuiltinTypeId::Vector3F32),
        src,
        length));
}
```

Charger la position d'une entité

Vous pouvez charger (lire à partir de la structure d'état) la position d'une entité à l'aide d'une structure de données entière. Ces exemples utilisent la fonction suivante :

> **Note**
>
> Vous devez fournir `Api::BuiltinTypeId::Vector3F32` à`Api::LoadEntityIndexKey()`, comme indiqué dans les exemples suivants.

Example Exemple d'utilisation d'un tableau pour représenter la position

```
Result<void> GetEntityPosition(Api::Entity& entity,
    Transaction& transaction)
{
    std::int8_t* dest = nullptr;

    WEAVERRUNTIME_TRY(Aws::WeaverRuntime::Api::LoadEntityIndexKey(
        transaction,
        entity,
        Api::BuiltinTypeIdToTypeId(
            Aws::WeaverRuntime::Api::BuiltinTypeId::Vector3F32),
        &dest));

    std::array<float, 3> position =
```

```
        *reinterpret_cast<std::array<float, 3>*>(dest);
}
```

Example Exemple utilisant un struct pour représenter la position

```
struct Position
{struct
   float x;
   float y;
   float z;
};

Result<void> GetEntityPosition(Api::Entity& entity, Transaction& transaction)
{
    std::int8_t* dest = nullptr;

    WEAVERRUNTIME_TRY(Aws::WeaverRuntime::Api::LoadEntityIndexKey(
        transaction,
        entity,
        Api::BuiltinTypeIdToTypeId(
            Aws::WeaverRuntime::Api::BuiltinTypeId::Vector3F32),
        &dest));

    Position position = *reinterpret_cast<Position*>(dest);
}
```

Événements organisés par les entités

Vous pouvez utiliser les fonctions suivantes du SDK de l' SimSpace Weaver application pour obtenir tous les événements relatifs à la propriété et à l'abonnement :

- `Result<OwnershipChangeList> OwnershipChanges(Transaction& txn)`
- `Result<SubscriptionChangeList> AllSubscriptionEvents(Transaction& txn)`

Vous pouvez utiliser le framework de SimSpace Weaver démonstration si vous avez besoin d'un traitement des événements d'entité piloté par rappel. Pour plus d'informations, consultez le fichier d'en-tête suivant :

- *`sdk-folder`*`/packaging-tools/samples/ext/DemoFramework/include/DemoFramework/EntityEventProcessor.h`

Vous pouvez également créer votre propre traitement des événements d'entité.

Rubriques

- Parcourez les événements pour les entités détenues
- Parcourez les événements pour les entités abonnées
- Itérer à travers les événements de changement de propriétaire pour les entités

Parcourez les événements pour les entités détenues

`OwnershipChanges()`Utilisez-le pour obtenir une liste des événements relatifs aux entités détenues (entités situées dans la zone de propriété de l'application). La fonction possède la signature suivante :

```
Result<OwnershipChangeList> OwnershipChanges(Transaction& txn)
```

Effectuez ensuite une itération sur les entités à l'aide d'une boucle, comme illustré dans l'exemple suivant.

Example exemple

```
WEAVERRUNTIME_TRY(Result<Api::OwnershipChangeList> ownershipChangesResult,
 Api::OwnershipChanges(transaction));

for (const Api::OwnershipChange& event : ownershipChangeList.changes)
{
    Api::Entity entity = event.entity;
    Api::ChangeListAction action = event.action;

    switch (action)
    {
    case Api::ChangeListAction::None:
        // insert code to handle the event
        break;
    case Api::ChangeListAction::Remove:
        // insert code to handle the event
        break;
    case Api::ChangeListAction::Add:
        // insert code to handle the event
        break;
    case Api::ChangeListAction::Update:
```

```
        // insert code to handle the event
        break;
    case Api::ChangeListAction::Reject:
        // insert code to handle the event
        break;
    }
}
```

Types d'événements

- None— L'entité se trouve dans la zone et ses données de position et de champ n'ont pas été modifiées.
- Remove— L'entité a été supprimée de la zone.
- Add— L'entité a été ajoutée à la zone.
- Update— L'entité se trouve dans la zone et a été modifiée.
- Reject— L'application n'a pas réussi à supprimer l'entité de la zone.

> **Note**
>
> Dans le cas d'un `Reject` événement, l'application tentera à nouveau le transfert à la prochaine case.

Parcourez les événements pour les entités abonnées

Permet `AllSubscriptionEvents()` d'obtenir une liste des événements pour les entités abonnées (entités situées dans la zone d'abonnement de l'application). La fonction possède la signature suivante :

```
Result<SubscriptionChangeList> AllSubscriptionEvents(Transaction& txn)
```

Effectuez ensuite une itération sur les entités à l'aide d'une boucle, comme illustré dans l'exemple suivant.

Example exemple

```
WEAVERRUNTIME_TRY(Api::SubscriptionChangeList subscriptionChangeList,
 Api::AllSubscriptionEvents(transaction));
```

```
for (const Api::SubscriptionEvent& event : subscriptionChangeList.changes)
{
    Api::Entity entity = event.entity;
    Api::ChangeListAction action = event.action;

    switch (action)
    {
    case Api::ChangeListAction::None:
        // insert code to handle the event
        break;
    case Api::ChangeListAction::Remove:
        // insert code to handle the event
        break;
    case Api::ChangeListAction::Add:
        // insert code to handle the event
        break;
    case Api::ChangeListAction::Update:
        // insert code to handle the event
        break;
    case Api::ChangeListAction::Reject:
        // insert code to handle the event
        break;
    }
}
```

Types d'événements

- `None`— L'entité se trouve dans la zone et ses données de position et de champ n'ont pas été modifiées.
- `Remove`— L'entité a été supprimée de la zone.
- `Add`— L'entité a été ajoutée à la zone.
- `Update`— L'entité se trouve dans la zone et a été modifiée.
- `Reject`— L'application n'a pas réussi à supprimer l'entité de la zone.

> **Note**
>
> Dans le cas d'un `Reject` événement, l'application tentera à nouveau le transfert à la prochaine case.

Itérer à travers les événements de changement de propriétaire pour les entités

Pour obtenir les événements dans lesquels une entité passe d'une zone de propriété à une zone d'abonnement, comparez les modifications entre les événements de propriété et d'abonnement actuels et précédents de l'entité.

Vous pouvez gérer ces événements en lisant :

- `Api::SubscriptionChangeList`
- `Api::OwnershipEvents`

Vous pouvez ensuite comparer les modifications aux données enregistrées précédemment.

L'exemple suivant montre comment gérer les événements de changement de propriétaire d'une entité. Cet exemple suppose que pour les entités faisant la transition entre des entités abonnées et des entités détenues (dans les deux sens), l'remove/add event occurs first followed by the subscription remove/addévénement de propriété est coché à la case suivante.

Example exemple

```
Result<void> ProcessOwnershipEvents(Transaction& transaction)
{
    using EntityIdsByAction =
        std::unordered_map<Api::ChangeListAction,
        std::vector<Api::EntityId>>;
    using EntityIdSetByAction =
        std::unordered_map<Api::ChangeListAction,
        std::unordered_set<Api::EntityId>>;

    static EntityIdsByAction m_entityIdsByPreviousOwnershipAction;

    EntityIdSetByAction entityIdSetByAction;

    /**
     * Enumerate Api::SubscriptionChangeList items
     * and store Add and Remove events.
     */
    WEAVERRUNTIME_TRY(Api::SubscriptionChangeList subscriptionEvents,
        Api::AllSubscriptionEvents(transaction));

    for (const Api::SubscriptionEvent& event : subscriptionEvents.changes)
    {
```

```
        const Api::ChangeListAction action = event.action;

        switch (action)
        {
        case Api::ChangeListAction::Add:
        case Api::ChangeListAction::Remove:

            {
                entityIdSetByAction[action].insert(
                    event.entity.descriptor->id);
                break;
            }
        case Api::ChangeListAction::None:
        case Api::ChangeListAction::Update:
        case Api::ChangeListAction::Reject:
            {
                break;
            }
        }
    }

    EntityIdsByAction entityIdsByAction;

    /**
     * Enumerate Api::OwnershipChangeList items
     * and store Add and Remove events.
     */

    WEAVERRUNTIME_TRY(Api::OwnershipChangeList ownershipChangeList,
        Api::OwnershipChanges(transaction));

    for (const Api::OwnershipChange& event : ownershipChangeList.changes)
    {
        const Api::ChangeListAction action = event.action;

        switch (action)
        {
        case Api::ChangeListAction::Add:
        case Api::ChangeListAction::Remove:
            {
                entityIdsByAction[action].push_back(
                    event.entity.descriptor->id);
                break;
            }
```

```
        case Api::ChangeListAction::None:
        case Api::ChangeListAction::Update:
        case Api::ChangeListAction::Reject:
            {
                break;
            }
        }

    }

    std::vector<Api::EntityId> fromSubscribedToOwnedEntities;
    std::vector<Api::EntityId> fromOwnedToSubscribedEntities;

    /**
     * Enumerate the *previous* Api::OwnershipChangeList Remove items
     * and check if they are now in
     * the *current* Api::SubscriptionChangeList Add items.
     *
     * If true, then that means
     * OnEntityOwnershipChanged(bool isOwned = false)
     */
    for (const Api::EntityId& id : m_entityIdsByPreviousOwnershipAction[
        Api::ChangeListAction::Remove])
    {
        if (entityIdSetBySubscriptionAction[
            Api::ChangeListAction::Add].find(id) !=
                entityIdSetBySubscriptionAction[
                Api::ChangeListAction::Add].end())
        {
            fromOwnedToSubscribedEntities.push_back(id);
        }
    }


    /**
     * Enumerate the *previous* Api::OwnershipChangeList Add items
     * and check if they are now in
     * the *current* Api::SubscriptionChangeList Remove items.
     *
     * If true, then that means
     * OnEntityOwnershipChanged(bool isOwned = true)
     */
    for (const Api::EntityId& id : m_entityIdsByPreviousOwnershipAction[
        Api::ChangeListAction::Add])
```

```
    {
        if (entityIdSetBySubscriptionAction[
            Api::ChangeListAction::Remove].find(id) !=

                entityIdSetBySubscriptionAction[
                Api::ChangeListAction::Remove].end())
        {
            fromSubscribedToOwnedEntities.push_back(id);
        }
    }

    m_entityIdsByPreviousOwnershipAction = entityIdsByOwnershipAction;

    return Success();
}
```

Result et gestion des erreurs

La `Aws::WeaverRuntime::Result<T>` classe utilise une `Outcome` bibliothèque tierce. Vous pouvez utiliser le modèle suivant pour vérifier `Result` et détecter les erreurs renvoyées par les appels d'API.

```
void DoBeginUpdate(Application& app)
{
    Result<Transaction> transactionResult = Api::BeginUpdate(app);

    if (transactionResult)
    {
        Transaction transaction =
            std::move(transactionResult).assume_value();

        /**
         * Do things with transaction ...
         */
    }
    else
    {
        ErrorCode errorCode = WEAVERRUNTIME_EXPECT_ERROR(transactionResult);
        /**
         * Macro compiles to:
         * ErrorCode errorCode = transactionResult.assume_error();
         */
    }
```

```
}
```

Result macro d'instruction de contrôle

Dans une fonction avec un type de retour`Aws::WeaverRuntime::Result<T>`, vous pouvez utiliser la `WEAVERRUNTIME_TRY` macro au lieu du modèle de code précédent. La macro exécutera la fonction qui lui a été transmise. Si la fonction transmise échoue, la macro fera en sorte que la fonction englobante renvoie une erreur. Si la fonction transmise réussit, l'exécution passe à la ligne suivante. L'exemple suivant montre une réécriture de la `DoBeginUpdate()` fonction précédente. Cette version utilise la `WEAVERRUNTIME_TRY` macro au lieu du if-else structure de commande. Notez que le type de retour de la fonction est`Aws::WeaverRuntime::Result<void>`.

```
Aws::WeaverRuntime::Result<void> DoBeginUpdate(Application& app)
{
    /**
     * Execute Api::BeginUpdate()
     * and return from DoBeginUpdate() if BeginUpdate() fails.
     * The error is available as part of the Result.
     */
    WEAVERRUNTIME_TRY(Transaction transaction, Api::BeginUpdate(m_app));

    /**
     * Api::BeginUpdate executed successfully.
     *
     * Do things here.
     */

    return Aws::Success();
}
```

En cas d'`BeginUpdate()`échec, la macro `DoBeginUpdate()` revient plus tôt en cas d'échec. Vous pouvez utiliser la `WEAVERRUNTIME_EXPECT_ERROR` macro pour `Aws::WeaverRuntime::ErrorCode` obtenir `BeginUpdate()` le L'exemple suivant montre comment la `Update()` fonction appelle `DoBeginUpdate()` et obtient le code d'erreur en cas d'échec.

```
void Update(Application& app)
{
    Result<void> doBeginUpdateResult = DoBeginUpdate(app);
```

```
    if (doBeginUpdateResult)
    {
        /**
         * Successful.
         */
    }
    else
    {
        /**
         * Get the error from Api::BeginUpdate().
         */
        ErrorCode errorCode = WEAVERRUNTIME_EXPECT_ERROR(doBeginUpdateResult);

    }
}
```

Vous pouvez rendre le code d'erreur `BeginUpdate()` accessible à une fonction qui appelle `Update()` en modifiant le type de retour `Update()` en`Aws::WeaverRuntime::Result<void>`. Vous pouvez répéter ce processus pour continuer à envoyer le code d'erreur plus bas dans la pile d'appels.

Génériques et types de domaines

Le SDK de SimSpace Weaver l'application fournit les types de données à simple précision `Api::Vector2F32` et`Api::BoundingBox2F32`, ainsi que les types de données à double `Api::Vector2F64` précision et. `Api::BoundingBox2F64` Ces types de données sont des structures de données passives dépourvues de méthodes pratiques. Notez que l'API utilise uniquement `Api::Vector2F32` et`Api::BoundingBox2F32`. Vous pouvez utiliser ces types de données pour créer et modifier des abonnements.

Le framework de SimSpace Weaver démonstration fournit une version minimale du AzCore bibliothèque mathématique, qui contient `Vector3` et`Aabb`. Pour plus d'informations, consultez les fichiers d'en-tête dans :

- *`sdk-folder`*`/packaging-tools/samples/ext/DemoFramework/include/AzCore/Math`

Opérations diverses du SDK de l'application

Rubriques

- AllSubscriptionEvents and OwnershipChanges contenir les événements du dernier appel

- Relâchez les verrous de lecture après le traitement SubscriptionChangeList
- Création d'une instance d'application autonome à des fins de test

AllSubscriptionEvents and OwnershipChanges contenir les événements du dernier appel

Les valeurs renvoyées par les appels vers `Api::AllSubscriptionEvents()` et `Api::OwnershipChanges()` contenant des événements survenus lors du dernier appel, et non de la dernière coche. Dans l'exemple suivant, `secondSubscriptionEvents` et `secondOwnershipChangeList` sont vides car leurs fonctions sont appelées immédiatement après les premiers appels.

Si vous attendez 10 ticks puis que vous appelez `Api::AllSubscriptionEvents()` et`Api::OwnershipChanges()`, leurs résultats contiendront à la fois les événements et les changements survenus lors des 10 derniers ticks (et non le dernier).

Example exemple

```
Result<void> ProcessOwnershipChanges(Transaction& transaction)
{
    WEAVERRUNTIME_TRY(
        Api::SubscriptionChangeList firstSubscriptionEvents,
        Api::AllSubscriptionEvents(transaction));
    WEAVERRUNTIME_TRY(
        Api::OwnershipChangeList firstOwnershipChangeList,
        Api::OwnershipChanges(transaction));

    WEAVERRUNTIME_TRY(
        Api::SubscriptionChangeList secondSubscriptionEvents,
        Api::AllSubscriptionEvents(transaction));
    WEAVERRUNTIME_TRY(
        Api::OwnershipChangeList secondOwnershipChangeList,
        Api::OwnershipChanges(transaction));

    /**
     * secondSubscriptionEvents and secondOwnershipChangeList are
     * both empty because there are no changes since the last call.
     */
}
```

> **Note**
>
> La fonction `AllSubscriptionEvents()` est implémentée, mais elle ne l'`SubscriptionEvents()`est pas.

Relâchez les verrous de lecture après le traitement SubscriptionChangeList

Lorsque vous commencez une mise à jour, des segments de mémoire partagée contiennent des segments de mémoire partagée pour les données validées dans d'autres partitions lors de la vérification précédente. Ces segments de mémoire partagée peuvent être verrouillés par les lecteurs. Une application ne peut pas être entièrement validée tant que tous les lecteurs n'ont pas relâché les verrous. À titre d'optimisation, une application doit appeler `Api::ReleaseReadLeases()` pour libérer les verrous après le traitement des `Api::SubscriptionChangelist` éléments. Cela réduit le contentieux au moment de la validation. `Api::Commit()`publie les baux de lecture par défaut, mais il est recommandé de les publier manuellement après le traitement des mises à jour des abonnements.

Example exemple

```
Result<void> ProcessSubscriptionChanges(Transaction& transaction)
{
    WEAVERRUNTIME_TRY(ProcessSubscriptionChanges(transaction));

    /**
     * Done processing Api::SubscriptionChangeList items.
     * Release read locks.
     */

    WEAVERRUNTIME_EXPECT(Api::ReleaseReadLeases(transaction));

    ...
}
```

Création d'une instance d'application autonome à des fins de test

Vous pouvez l'utiliser `Api::CreateStandaloneApplication()` pour créer une application autonome afin de tester la logique de l'application avant d'exécuter le code dans une simulation réelle.

Example exemple

```
int main(int argc, char* argv[])
{
    Api::StandaloneRuntimeConfig config = {
        /* run_for_seconds (the lifetime of the app) */ 3,
        /* tick_hertz (the app clock rate) */ 10 };

    Result<Application> applicationResult =
        Api::CreateStandaloneApplication(config);

    ...
}
```

AWS SimSpace Weaver framework de démonstration

Le framework de AWS SimSpace Weaver démonstration (framework de démonstration) est une bibliothèque d'utilitaires que vous pouvez utiliser pour développer des SimSpace Weaver applications.

Le framework de démonstration fournit

- Exemples de code et modèles de programmation que vous pouvez utiliser et examiner
- Abstractions et fonctions utilitaires qui rationalisent le développement d'applications simples
- Un moyen plus simple de tester les fonctionnalités expérimentales du SDK de l' SimSpace Weaver application

Nous avons conçu le SDK de SimSpace Weaver l'application avec un accès de bas niveau SimSpace Weaver APIs afin d'offrir de meilleures performances. En revanche, nous avons conçu le framework de démonstration pour fournir des abstractions de haut niveau et un accès à celles-ci pour en APIs SimSpace Weaver faciliter l'utilisation. Le coût de la facilité d'utilisation est un niveau de performance inférieur à celui de l'utilisation directe du SDK de l' SimSpace Weaver application. Les simulations qui peuvent tolérer des performances inférieures (telles que celles qui ne sont pas soumises à des exigences de performance en temps réel) peuvent être de bonnes candidates pour utiliser le framework de démonstration. Nous vous recommandons d'utiliser les fonctionnalités natives du SDK de l' SimSpace Weaver application pour les applications complexes, car le framework de démonstration n'est pas une boîte à outils complète.

Le framework de démonstration inclut

- Exemples de code de travail qui prennent en charge et démontrent :
  - Gestion du flux d'applications
  - Traitement des événements d'entité piloté par le rappel
- Un ensemble de bibliothèques d'utilitaires tierces :
  - spdlog (une bibliothèque de journalisation)
  - Une version minimale de AZCore (une bibliothèque mathématique) qui contient uniquement :
    - Vector3
    - Aabb
  - cxxopts (une bibliothèque d'analyseurs d'options en ligne de commande)
- Fonctions utilitaires spécifiques à SimSpace Weaver

Le framework de démonstration se compose d'une bibliothèque, de fichiers sources et CMakeLists. Les fichiers sont inclus dans le package distribuable du SDK de SimSpace Weaver l'application.

Utilisation des quotas de service

Cette section décrit comment utiliser les quotas de service pour SimSpace Weaver. Les quotas sont également appelés limites. Pour obtenir la liste des quotas de service, consultezSimSpace Points de terminaison et quotas Weaver. Les APIs éléments de cette section proviennent de l'ensemble de l'application APIs. APIsLes applications sont différentes du service APIs. L'application APIs fait partie du SDK de l' SimSpace Weaver application. Vous trouverez la documentation de l'application APIs dans le dossier du SDK de l'application sur votre système local :

```
sdk-folder\SimSpaceWeaverAppSdk-sdk-version\documentation\index.html
```

Rubriques

- Déterminer les limites d'une application
- Obtenez la quantité de ressources utilisées par une application
- Réinitialiser les métriques
- Dépassement d'une limite
- À court de mémoire
- Bonnes pratiques

Déterminer les limites d'une application

Vous pouvez utiliser le plugin RuntimeLimits API d'application pour demander les limites d'une application.

```
Result<Limit> RuntimeLimit(Application& app, LimitType type)
```

Paramètres

Applicationet application

Une référence à l'application.

LimitType type

Une énumération avec les types de limites suivants :

```
enum LimitType {
    Unset = 0,
    EntitiesPerPartition = 1,
    RemoteEntityTransfers = 2,
    LocalEntityTransfers = 3
};
```

L'exemple suivant interroge la limite du nombre d'entités.

```
WEAVERRUNTIME_TRY(auto entity_limit,
    Api::RuntimeLimit(m_app, Api::LimitType::EntitiesPerPartition))
Log::Info("Entity count limit", entity_limit.value);
```

Obtenez la quantité de ressources utilisées par une application

Vous pouvez appeler le RuntimeMetrics API d'application pour obtenir la quantité de ressources utilisées par une application :

```
Result<std::reference_wrapper<const AppRuntimeMetrics>> RuntimeMetrics(Application&
 app) noexcept
```

Paramètres

Applicationet application

Une référence à l'application.

L'API renvoie une référence à un struct qui contient les métriques. Une contre-métrique contient une valeur totale cumulée et ne fait qu'augmenter. Une métrique de jauge contient une valeur qui peut augmenter ou diminuer. Le moteur d'exécution de l'application met à jour un compteur chaque fois qu'un événement augmente la valeur. Le moteur d'exécution met à jour les jauges uniquement lorsque vous appelez l'API. SimSpace Weaver garantit que la référence est valide pendant toute la durée de vie de l'application. Les appels répétés à l'API ne modifieront pas la référence.

```
struct AppRuntimeMetrics {
    uint64_t total_committed_ticks_gauge,

    uint32_t active_entity_gauge,
    uint32_t ticks_since_reset_counter,

    uint32_t load_field_counter,
    uint32_t store_field_counter,

    uint32_t created_entity_counter,
    uint32_t deleted_entity_counter,

    uint32_t entered_entity_counter,
    uint32_t exited_entity_counter,

    uint32_t rejected_incoming_transfer_counter,
    uint32_t rejected_outgoing_transfer_counter
}
```

Réinitialiser les métriques

Le ResetRuntimeMetrics L'API de l'application réinitialise les valeurs du `AppRuntimeMetrics` struct.

```
Result<void> ResetRuntimeMetrics(Application& app) noexcept
```

L'exemple suivant montre comment vous pouvez appeler ResetRuntimeMetrics dans votre application.

```
if (ticks_since_last_report > 100)
{
    auto metrics = WEAVERRUNTIME_EXPECT(Api::RuntimeMetrics(m_app));
    Log::Info(metrics);

    ticks_since_last_report = 0;

    WEAVERRUNTIME_EXPECT(Api::ResetRuntimeMetrics(m_app));
}
```

Dépassement d'une limite

Un appel d'API d'application qui dépasse une limite renverra un`ErrorCode::CapacityExceeded`, sauf pour les transferts d'entités. SimSpace Weaver gère les transferts d'entités de manière asynchrone dans le cadre de Commit et BeginUpdate opérations d'API d'application, il n'existe donc aucune opération spécifique qui renvoie une erreur si un transfert échoue en raison de la limite de transfert d'entités. Pour détecter les échecs de transfert, vous pouvez comparer les valeurs actuelles de `rejected_incoming_transfer_counter` et `rejected_outgoing_transfer_counter` (dans le `AppRuntimeMetrics` struct) avec leurs valeurs précédentes. Les entités rejetées ne figureront pas dans la partition, mais l'application peut toujours les simuler.

À court de mémoire

SimSpace Weaver utilise un processus de collecte des déchets pour nettoyer et libérer la mémoire libérée. Il est possible d'écrire des données plus rapidement que le ramasse-miettes ne peut libérer de la mémoire. Dans ce cas, les opérations d'écriture risquent de dépasser la limite de mémoire réservée de l'application. SimSpace Weaver renverra une erreur interne avec un message contenant `OutOfMemory` (et des détails supplémentaires). Pour de plus amples informations, veuillez consulter Répartissez les écritures dans le temps.

Bonnes pratiques

Les bonnes pratiques suivantes sont des directives générales pour concevoir vos applications afin d'éviter de dépasser les limites. Ils peuvent ne pas s'appliquer à la conception spécifique de votre application.

Surveillez fréquemment et ralentissez

Vous devez surveiller fréquemment vos indicateurs et ralentir les opérations qui approchent d'une limite.

Évitez de dépasser les limites d'abonnement et les limites de transfert

Si possible, concevez votre simulation de manière à réduire le nombre d'abonnements à distance et de transferts d'entités. Vous pouvez utiliser des groupes de placement pour placer plusieurs partitions sur le même travailleur et réduire le besoin de transferts d'entités à distance entre les travailleurs.

Répartissez les écritures dans le temps

Le nombre et la taille des mises à jour contenues dans un tick peuvent avoir un impact significatif sur le temps et la mémoire nécessaires pour valider une transaction. Des besoins de mémoire importants peuvent entraîner une pénurie de mémoire lors de l'exécution de l'application. Vous pouvez répartir les écritures dans le temps afin de réduire la taille totale moyenne des mises à jour par tick. Cela peut contribuer à améliorer les performances et à éviter de dépasser les limites. Nous vous recommandons de ne pas écrire plus de 12 Mo en moyenne sur chaque case ou 1,5 Ko pour chaque entité.

Simulations de débogage

Vous pouvez utiliser les méthodes suivantes pour obtenir des informations sur vos simulations.

Rubriques

- Utiliser SimSpace Weaver Local et regardez la sortie de la console
- Consultez vos journaux dans Amazon CloudWatch Logs
- Utiliser describe Appels d'API
- Connect un client

Utiliser SimSpace Weaver Local et regardez la sortie de la console

Nous vous recommandons de développer d'abord vos simulations localement, puis de les exécuter dans le AWS Cloud. Vous pouvez afficher la sortie de la console directement lorsque vous exécutez avec SimSpace Weaver Local. Pour plus d'informations, consultezDéveloppement local en SimSpace Weaver.

Consultez vos journaux dans Amazon CloudWatch Logs

Lorsque vous exécutez votre simulation dans la console, AWS Cloud la sortie de vos applications est envoyée aux flux de CloudWatch journaux dans Amazon Logs. Votre simulation écrit également d'autres données de journal. Vous devez activer la journalisation dans votre schéma de simulation si vous souhaitez que votre simulation écrive des données de journal. Pour de plus amples informations, veuillez consulter SimSpace Weaver journaux dans Amazon CloudWatch Logs.

> **Warning**
>
> Votre simulation peut produire de grandes quantités de données de log. Les données du journal peuvent augmenter très rapidement. Vous devez surveiller attentivement vos journaux et arrêter vos simulations lorsque vous n'en avez plus besoin. L'exploitation forestière peut entraîner des coûts importants.

Utiliser describe Appels d'API

Vous pouvez utiliser le service suivant APIs pour obtenir des informations sur vos simulations dans le AWS Cloud.

- ListSimulations— obtenez une liste de toutes vos simulations dans le AWS Cloud.

  Example exemple

  ```
  aws simspaceweaver list-simulations
  ```

- DescribeSimulation— obtenir des informations sur une simulation.

  Example exemple

  ```
  aws simspaceweaver describe-simulation --simulation MySimulation
  ```

- DescribeApp— obtenir des informations sur une application.

  Example exemple

  ```
  aws simspaceweaver describe-app --simulation MySimulation --domain MyCustomDomain --
  app MyCustomApp
  ```

Pour plus d'informations sur le SimSpace Weaver APIs, consultezSimSpace Weaver Références d'API.

Connect un client

Vous pouvez connecter un client à une application personnalisée ou de service en cours d'exécution que vous avez définie `endpoint_config` dans votre schéma de simulation. Le SDK de SimSpace Weaver l'application inclut des exemples de clients que vous pouvez utiliser pour visualiser l'exemple d'application. Vous pouvez consulter le code source de ces exemples de clients et de l'exemple d'application pour voir comment créer vos propres clients. Pour plus d'informations sur la création et l'exécution des exemples de clients, consultez les didacticiels dansCommencer avec SimSpace Weaver.

Le code source des exemples de clients se trouve dans le dossier suivant :

- *`sdk-folder`*`\packaging-tools\clients\PathfindingSampleClients\`

Débogage de simulations locales

Vous pouvez déboguer vos SimSpace Weaver Local applications avecMicrosoft Visual Studio. Pour plus d'informations sur la procédure de débogage avecVisual Studio, consultez leMicrosoft Visual Studio documentation.

Pour déboguer votre simulation locale

1. Assurez-vous que vous vous `schema.yaml` trouvez dans votre répertoire de travail.
2. Dans Visual Studio, ouvrez le menu contextuel de chaque application que vous souhaitez déboguer (par exemple `PathfindingSampleLocalSpatial` ou`PathfindingSampleLocalView`) et définissez le répertoire de travail dans la section de débogage.
3. Ouvrez le menu contextuel de l'application que vous souhaitez déboguer et sélectionnez Définir comme projet de démarrage.
4. Choisissez F5 de démarrer le débogage de l'application.

Les conditions requises pour débugger une simulation sont les mêmes que celles requises pour exécuter une simulation normalement. Vous devez démarrer le nombre d'applications spatiales spécifié dans le schéma. Par exemple, si votre schéma spécifie une grille 2x2 et que vous lancez une

application spatiale en mode débogage, la simulation ne s'exécutera pas tant que vous n'aurez pas démarré 3 autres applications spatiales (en mode débogage ou non en mode débogage).

Pour déboguer une application personnalisée, vous devez d'abord démarrer vos applications spatiales, puis démarrer l'application personnalisée dans le débogueur.

Notez que votre simulation s'exécute en mode « lock step ». Dès qu'une application atteint un point d'arrêt, toutes les autres applications s'interrompent. Une fois que vous aurez continué à partir de ce point d'arrêt, les autres applications continueront.

Conteneurs personnalisés

AWS SimSpace Weaver applications exécutées en conteneur Amazon Linux 2 (AL2) environnements. Dans le AWS Cloud, SimSpace Weaver exécute vos simulations dans des conteneurs Docker créés à partir d'une `amazonlinux:2` image fournie par Amazon Elastic Container Registry (Amazon ECR). Vous pouvez créer une image Docker personnalisée, la stocker dans Amazon ECR et utiliser cette image pour votre simulation au lieu de l'image Docker par défaut que nous fournissons.

Vous pouvez utiliser un conteneur personnalisé pour gérer vos dépendances logicielles et inclure des composants logiciels supplémentaires qui ne figurent pas dans l'image Docker standard. Par exemple, vous pouvez ajouter au conteneur les bibliothèques logicielles accessibles au public utilisées par votre application et ne placer votre code personnalisé que dans le fichier zip de l'application.

> **Important**
>
> Nous prenons uniquement en charge les images AL2 Docker hébergées dans les référentiels Amazon ECR, que ce soit dans la galerie publique Amazon ECR ou dans votre registre Amazon ECR privé. Nous ne prenons pas en charge les images Docker hébergées en dehors d'Amazon ECR. Pour plus d'informations sur Amazon ECR, consultez la documentation Amazon Elastic Container Registry.

Rubriques

- Création d'un conteneur personnalisé
- Modifier un projet pour utiliser un conteneur personnalisé

- Questions fréquemment posées sur les conteneurs personnalisés
- Résolution des problèmes liés aux conteneurs personnalisés

Création d'un conteneur personnalisé

Ces instructions supposent que vous savez comment utiliser Docker et Amazon Elastic Container Registry (Amazon ECR). Pour plus d'informations sur la sécurité dans Amazon ECR, veuillez consulter le Guide de l'utilisateur Amazon ECR.

Prérequis

- L'identité IAM (utilisation ou rôle) que vous utilisez pour effectuer ces actions dispose des autorisations appropriées pour utiliser Amazon ECR
- Docker est installé sur votre système local

Pour créer un conteneur personnalisé

1. Créez votre `Dockerfile`.

   A `Dockerfile` pour exécuter AWS SimSpace Weaver des applications, commencez par le Amazon Linux 2 image dans Amazon ECR.

   ```
   # parent image required to run AWS SimSpace Weaver apps
   FROM public.ecr.aws/amazonlinux/amazonlinux:2
   ```

2. Construisez votre`Dockerfile`.
3. Téléchargez l'image de votre conteneur sur Amazon ECR.
   - Utilisez le AWS Management Console.
   - Utilisez le AWS Command Line Interface.

   > **Note**
   >
   > Si un `AccessDeniedException` message d'erreur s'affiche lorsque vous essayez de télécharger votre image de conteneur sur Amazon ECR, il est possible que votre identité IAM (utilisateur ou rôle) ne dispose pas des autorisations nécessaires pour utiliser Amazon ECR. Vous pouvez associer la politique `AmazonEC2ContainerRegistryPowerUser` AWS gérée à votre identité IAM et

réessayer. Pour plus d'informations sur la façon d'associer une politique, consultez la section Ajout et suppression d'autorisations d'identité IAM dans le Guide de l'AWS Identity and Access Management utilisateur.

Modifier un projet pour utiliser un conteneur personnalisé

Ces instructions supposent que vous savez déjà comment utiliser vos processus de stockage AWS SimSpace Weaver et de développement d'applications et que vous souhaitez les rendre AWS Cloud plus efficaces.

Prérequis

- Vous avez un conteneur personnalisé dans Amazon Elastic Container Registry (Amazon ECR). Pour plus d'informations sur la création d'un conteneur personnalisé, consultezCréation d'un conteneur personnalisé.

Pour modifier votre projet afin d'utiliser un conteneur personnalisé

1. Ajoutez des autorisations au rôle d'application de simulation de votre projet pour utiliser Amazon ECR.

   a. Si vous ne disposez pas encore d'une stratégie IAM avec les autorisations suivantes, créez-la. Nous suggérons le nom de la politique`simspaceweaver-ecr`. Pour plus d'informations sur la création d'une stratégie IAM, consultez la section Création de politiques IAM dans le Guide de l'AWS Identity and Access Management utilisateur.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement",
            "Effect": "Allow",
            "Action": [
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer",
                "ecr:GetAuthorizationToken"
            ],
            "Resource": "*"
        }
```

```
    ]
}
```

b. Trouvez le nom du rôle de l'application de simulation de votre projet :

i. Dans un éditeur de texte, ouvrez le AWS CloudFormation modèle :

```
sdk-folder\PackagingTools\sample-stack-template.yaml
```

ii. Trouvez la `RoleName` propriété ci-dessous`WeaverAppRole`. La valeur est le nom du rôle de l'application de simulation de votre projet.

Example

```
AWSTemplateFormatVersion: "2010-09-09"
Resources:
  WeaverAppRole:
    Type: 'AWS::IAM::Role'
    Properties:
      RoleName: 'weaver-MySimulation-app-role'
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
        - Effect: Allow
          Principal:
            Service:
              - 'simspaceweaver.amazonaws.com'
```

c. Associez la `simspaceweaver-ecr` politique au rôle d'application de simulation du projet. Pour plus d'informations sur la façon d'associer une politique, consultez la section Ajout et suppression d'autorisations d'identité IAM dans le Guide de l'AWS Identity and Access Management utilisateur.

d. Accédez à la pile d'échantillons *sdk-folder* et exécutez-la pour mettre à jour la SimSpace Weaver pile d'échantillons :

```
python setup.py --cloudformation
```

2. Spécifiez les images de vos conteneurs dans le schéma de simulation du projet.

- Vous pouvez ajouter la `default_image` propriété facultative ci-dessous `simulation_properties` pour spécifier une image de conteneur personnalisée par défaut pour tous les domaines.
- Ajoutez la `image` propriété dans le `app_config` pour le domaine dans lequel vous souhaitez utiliser une image de conteneur personnalisée. Spécifiez l'URI du référentiel Amazon ECR comme valeur. Vous pouvez spécifier une image différente pour chaque domaine.
  - Si `image` ce n'est pas spécifié pour un domaine mais si `default_image` c'est le cas, les applications de ce domaine utilisent l'image par défaut.
  - Si `image` ce n'est pas spécifié pour un domaine et `default_image` ne l'est pas, les applications de ce domaine s'exécutent dans un SimSpace Weaver conteneur standard.

Example Extrait de schéma incluant des paramètres de conteneur personnalisés

```
sdk_version: "1.17.0"
simulation_properties:
  log_destination_service: "logs"
  log_destination_resource_name: "MySimulationLogs"
  default_entity_index_key_type: "Vector3<f32>"
  default_image: "111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-
repository:latest" # image to use if no image specified for a domain
domains:
  MyCustomDomain:
    launch_apps_via_start_app_call: {}
    app_config:
      package: "s3://weaver-myproject-111122223333-us-west-2/MyViewApp.zip"
      launch_command: ["MyViewApp"]
      required_resource_units:
        compute: 1
      endpoint_config:
        ingress_ports:
          - 7000
      image: "111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-
repository:latest" # custom container image to use for this domain
  MySpatialDomain:
    launch_apps_by_partitioning_strategy:
      partitioning_strategy: "MyGridPartitioning"
      grid_partition:
        x: 2
        y: 2
```

```
    app_config:
      package: "s3://weaver-myproject-111122223333-us-west-2/MySpatialApp.zip"
      launch_command: ["MySpatialApp"]
      required_resource_units:
        compute: 1
      image: "111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-
repository:latest" # custom container image to use for this domain
```

3. Créez et téléchargez votre projet comme d'habitude.

Questions fréquemment posées sur les conteneurs personnalisés

Q1. Que dois-je faire si je souhaite modifier le contenu de mon contenant ?

- Pour une simulation en cours d'exécution : vous ne pouvez pas modifier le conteneur d'une simulation en cours d'exécution. Vous devez créer un nouveau conteneur et démarrer une nouvelle simulation utilisant ce conteneur.
- Pour une nouvelle simulation : créez un nouveau conteneur, chargez-le sur Amazon Elastic Container Registry (Amazon ECR) et lancez une nouvelle simulation utilisant ce conteneur.

Q2. Comment puis-je modifier l'image du conteneur pour ma simulation ?

- Pour une simulation en cours d'exécution : vous ne pouvez pas modifier le conteneur d'une simulation en cours d'exécution. Vous devez démarrer une nouvelle simulation utilisant le nouveau conteneur.
- Pour une nouvelle simulation : spécifiez la nouvelle image du conteneur dans le schéma de simulation de votre projet. Pour de plus amples informations, veuillez consulter Modifier un projet pour utiliser un conteneur personnalisé.

Résolution des problèmes liés aux conteneurs personnalisés

Rubriques

- AccessDeniedException lors du chargement de votre image sur Amazon Elastic Container Registry (Amazon ECR)
- Une simulation utilisant un conteneur personnalisé ne démarre pas

AccessDeniedException lors du chargement de votre image sur Amazon Elastic Container Registry (Amazon ECR)

Si un `AccessDeniedException` message d'erreur s'affiche lorsque vous essayez de télécharger votre image de conteneur sur Amazon ECR, il est possible que votre identité IAM (utilisateur ou rôle) ne dispose pas des autorisations nécessaires pour utiliser Amazon ECR. Vous pouvez associer la politique `AmazonEC2ContainerRegistryPowerUser` AWS gérée à votre identité IAM et réessayer. Pour plus d'informations sur la façon d'associer une politique, consultez la section Ajout et suppression d'autorisations d'identité IAM dans le Guide de l'AWS Identity and Access Management utilisateur.

Une simulation utilisant un conteneur personnalisé ne démarre pas

Conseils pour le dépannage

- Si la journalisation est activée pour votre simulation, consultez vos journaux d'erreurs.
- Testez votre simulation sans conteneur personnalisé.
- Testez votre simulation localement. Pour de plus amples informations, veuillez consulter Développement local en SimSpace Weaver.

Travail avec Python

Vous pouvez utiliser Python pour vos SimSpace Weaver applications et vos clients. Le kit de développement logiciel Python (SDK Python) est inclus dans le package distribuable standard du SDK d' SimSpace Weaver applications. Le développement avec Python fonctionne de la même manière que le développement dans les autres langages pris en charge.

> **Important**
>
> SimSpace Weaver ne supporte que la version 3.9 de Python.

> **Important**
>
> SimSpace Weaver le support de Python nécessite SimSpace Weaver la version 1.15.0 ou ultérieure.

Rubriques

- Création d'un projet Python
- Démarrage d'une simulation Python
- Exemple de client Python
- Questions fréquemment posées sur l'utilisation de Python
- Résolution des problèmes liés à Python

Création d'un projet Python

Conteneur personnalisé en Python

Pour exécuter votre SimSpace Weaver simulation basée sur Python dans le AWS Cloud, vous pouvez créer un conteneur personnalisé qui inclut les dépendances nécessaires. Pour de plus amples informations, veuillez consulter Conteneurs personnalisés.

Un conteneur personnalisé Python doit inclure les éléments suivants :

- gcc
- openssl-devel
- bzip2-devel
- libffi-devel
- wget
- tar
- gzip
- make
- Python (version 3.9)

Si vous utilisez le `PythonBubblesSample` modèle pour créer votre projet, vous pouvez exécuter le `quick-start.py` script (situé dans le `tools` dossier de votre projet) pour créer une image Docker avec les dépendances nécessaires. Le script télécharge l'image sur Amazon Elastic Container Registry (Amazon ECR).

Le `quick-start.py` script utilise les éléments suivants `Dockerfile` :

```
FROM public.ecr.aws/amazonlinux/amazonlinux:2
```

```
RUN yum -y install gcc openssl-devel bzip2-devel libffi-devel
RUN yum -y install wget
RUN yum -y install tar
RUN yum -y install gzip
RUN yum -y install make
WORKDIR /opt
RUN wget https://www.python.org/ftp/python/3.9.0/Python-3.9.0.tgz
RUN tar xzf Python-3.9.0.tgz
WORKDIR /opt/Python-3.9.0
RUN ./configure --enable-optimizations
RUN make altinstall
COPY requirements.txt ./
RUN python3.9 -m pip install --upgrade pip
RUN pip3.9 install -r requirements.txt
```

Vous pouvez ajouter vos propres dépendances aux éléments suivants `Dockerfile` :

```
RUN yum -y install dependency-name
```

Le `requirements.txt` fichier contient la liste des packages Python requis pour l'`PythonBubblesSample`exemple de simulation :

```
Flask==2.1.1
```

Vous pouvez ajouter vos propres dépendances de package Python aux éléments suivants `requirements.txt` :

```
package-name==version-number
```

Les `Dockerfile` et `requirements.txt` se trouvent dans le `tools` dossier de votre projet.

> **Important**
>
> Techniquement, vous n'êtes pas obligé d'utiliser un conteneur personnalisé avec votre simulation Python, mais nous vous recommandons vivement d'utiliser un conteneur personnalisé. Le conteneur Amazon Linux 2 (AL2) standard que nous fournissons ne contient pas Python. Par conséquent, si vous n'utilisez pas de conteneur personnalisé contenant Python, vous devez inclure Python et les dépendances requises dans chaque fichier zip d'application vers lequel vous le téléchargez SimSpace Weaver.

Démarrage d'une simulation Python

Vous pouvez démarrer votre simulation basée sur Python de la même manière qu'une SimSpace Weaver simulation classique, à la fois dans SimSpace Weaver Local et SimSpace Weaver dans le AWS Cloud. Pour plus d'informations, consultez les didacticiels dansCommencer avec SimSpace Weaver.

`PythonBubblesSample`Il inclut son propre exemple de client Python. Pour de plus amples informations, veuillez consulter Exemple de client Python.

Exemple de client Python

Si vous utilisez le `PythonBubblesSample` modèle pour créer un projet, celui-ci contient un exemple de client Python. Vous pouvez utiliser l'exemple de client pour visualiser la `PythonBubblesSample` simulation. Vous pouvez également utiliser l'exemple de client comme point de départ pour créer votre propre client Python.

La procédure suivante part du principe que vous avez créé un `PythonBubblesSample` projet et lancé sa simulation.

Pour démarrer le client Python

1. Dans une fenêtre d'invite de commande, accédez au dossier `PyBubbleClient` d'exemple de projet.

   ```
   cd sdk-folder\Clients\HTTP\PyBubbleClient
   ```

2. Exécutez le client Python.

   ```
   python tkinter_client.py --host ip-address --port port-number
   ```

   Paramètres

   host

   L'adresse IP de votre simulation. Pour une simulation démarrée dans le AWS Cloud, vous pouvez trouver l'adresse IP de votre simulation dans la SimSpace Weaver console ou utiliser la procédure décrite dans le didacticiel Obtenez l'adresse IP et le numéro de port d'une application personnalisée de démarrage rapide. Pour une simulation locale, `127.0.0.1` utilisez-le comme adresse IP.

port

Le numéro de port de votre simulation. Pour une simulation démarrée dans le AWS Cloud, il s'agit du numéro de `Actual` port. Vous pouvez trouver le numéro de port de votre simulation dans la SimSpace Weaver console ou utiliser la procédure décrite dans le didacticiel Obtenez l'adresse IP et le numéro de port d'une application personnalisée de démarrage rapide. Pour une simulation locale, utilisez-le `7000` comme numéro de port.

simsize

Le nombre maximum d'entités à afficher dans le client.

Questions fréquemment posées sur l'utilisation de Python

Q1. Quelles sont les versions de Python prises en charge ?

SimSpace Weaver ne supporte que la version 3.9 de Python.

Résolution des problèmes liés à Python

Rubriques

- Échec lors de la création d'un conteneur personnalisé
- Votre simulation Python ne démarre pas
- Un client de visualisation ou de simulation Python génère une ModuleNotFound erreur

Échec lors de la création d'un conteneur personnalisé

Si un message d'erreur s'affiche `no basic auth credentials` après l'exécution`quick-start.py`, il se peut qu'il y ait un problème avec vos informations d'identification temporaires pour Amazon ECR. Exécutez la commande suivante avec votre Région AWS identifiant et votre numéro de AWS compte :

```
aws ecr get-login-password --region region | docker login --username AWS --password-
stdin account_id.dkr.ecr.region.amazonaws.com
```

Example

```
aws ecr get-login-password --region us-west-2 | docker login --username AWS --password-
stdin 111122223333.dkr.ecr.region.amazonaws.com
```

> **Important**
>
> Assurez-vous que celui que Région AWS vous spécifiez est le même que celui que vous utilisez pour votre simulation. Utilisez l'un Régions AWS des SimSpace Weaver supports. Pour de plus amples informations, veuillez consulter SimSpace Points de terminaison et quotas Weaver.

Après avoir exécuté la `aws ecr` commande, exécutez-la `quick-start.py` à nouveau.

Autres ressources de résolution des problèmes à vérifier

- Résolution des problèmes liés aux conteneurs personnalisés
- Résolution des problèmes liés à Amazon ECR dans le guide de l'utilisateur Amazon ECR
- Configuration avec Amazon ECR dans le guide de l'utilisateur Amazon ECR

Votre simulation Python ne démarre pas

Il se peut qu'une `Unable to start app` erreur apparaisse dans le journal de gestion de votre simulation. Cela peut se produire si la création de votre conteneur personnalisé a échoué. Pour de plus amples informations, veuillez consulter Échec lors de la création d'un conteneur personnalisé. Pour de plus amples informations sur les journaux, veuillez consulter SimSpace Weaver journaux dans Amazon CloudWatch Logs.

Si vous êtes certain qu'il n'y a aucun problème avec votre conteneur, vérifiez le code source Python de votre application. Vous pouvez utiliser … SimSpace Weaver Local pour tester votre application. Pour de plus amples informations, veuillez consulter Développement local en SimSpace Weaver.

Un client de visualisation ou de simulation Python génère une ModuleNotFound erreur

Python génère une `ModuleNotFound` erreur lorsqu'il ne parvient pas à localiser le package Python requis.

Si votre simulation se trouve dans le AWS Cloud, assurez-vous que votre conteneur personnalisé possède toutes les dépendances requises répertoriées dans votre`requirements.txt`. N'oubliez pas de l'exécuter `quick-start.py` à nouveau si vous modifiez`requirements.txt`.

Si le message d'erreur s'affiche pour le `PythonBubblesSample` client, utilisez `pip` pour installer le package indiqué :

```
pip install package-name==version-number
```

Support pour d'autres moteurs

Vous pouvez utiliser votre propre coutume C++ moteur avec SimSpace Weaver. Nous développons actuellement le support pour les moteurs suivants. Il existe une documentation distincte pour chacun de ces moteurs.

> **Important**
>
> Les intégrations avec les moteurs listés ici sont expérimentales. Ils sont disponibles en avant-première.

Moteurs

- Unity(version minimale 2022.3.19.F1)
- Unreal Engine(version minimale 5.0)

Unity

Vous devez avoir le Unity environnement de développement déjà installé avant de créer SimSpace Weaver des simulations avec Unity. Pour plus d'informations, consultez les instructions séparées :

```
sdk-folder\Unity-Guide.pdf
```

Unreal Engine

Vous devez construire un Unreal Engine serveur dédié à partir du code source. SimSpaceWeaverAppSdkDistributable Il inclut une version du PathfindingSample pour Unreal Engine. Pour plus d'informations, consultez les instructions séparées :

```
sdk-folder\Unreal-Engine-Guide.pdf
```

Utilisation de logiciels sous licence avec AWS SimSpace Weaver

AWS SimSpace Weaver vous permet de créer des simulations avec le moteur de simulation et le contenu de votre choix. Dans le cadre de votre utilisation de SimSpace Weaver, vous êtes responsable de l'obtention, de la maintenance et du respect des termes de licence de tout logiciel ou contenu que vous utilisez dans vos simulations. Vérifiez que votre contrat de licence vous permet de déployer votre logiciel et votre contenu dans un environnement hébergé virtuel.

Gérez vos ressources avec AWS CloudFormation

Vous pouvez l'utiliser AWS CloudFormation pour gérer vos AWS SimSpace Weaver ressources. AWS CloudFormation est un AWS service distinct qui vous aide à spécifier, à approvisionner et à gérer votre AWS infrastructure sous forme de code. Avec AWS CloudFormation vous, créez un fichier JSON ou YAML, appelé modèle. Votre modèle précise les détails de votre infrastructure. AWS CloudFormation utilise votre modèle pour provisionner votre infrastructure sous la forme d'une unité unique, appelée pile. Lorsque vous supprimez votre pile, vous pouvez avoir à AWS CloudFormation supprimer tous les éléments de la pile en même temps. Vous pouvez gérer votre modèle à l'aide de processus de gestion de code source standard (par exemple, en le suivant dans un système de contrôle de version tel que Git). Pour plus d'informations AWS CloudFormation, consultez le guide de AWS CloudFormation l'utilisateur.

Votre ressource de simulation

Dans AWS, une ressource est une entité avec laquelle vous pouvez travailler. Les exemples incluent une EC2 instance Amazon, un compartiment Amazon S3 ou un rôle IAM. Votre SimSpace Weaver simulation est une ressource. Dans les configurations, vous spécifiez généralement une AWS ressource dans le formulaire`AWS::service::resource`. Pour SimSpace Weaver, vous spécifiez votre ressource de simulation en tant que`AWS::SimSpaceWeaver::Simulation`. Pour plus d'informations sur votre ressource de simulation dans AWS CloudFormation, consultez la SimSpace Weaversection du guide de l'AWS CloudFormation utilisateur.

Comment puis-je l'utiliser AWS CloudFormation avec SimSpace Weaver ?

Vous pouvez créer un AWS CloudFormation modèle qui indique les AWS ressources que vous souhaitez mettre en service. Votre modèle peut spécifier une architecture complète, une partie d'une architecture ou une petite solution. Par exemple, vous pouvez spécifier une architecture pour votre

SimSpace Weaver solution qui inclut des compartiments Amazon S3, des autorisations IAM, une base de données compatible dans Amazon Relational Database Service ou Amazon DynamoDB, ainsi que votre ressource. `Simulation` Vous pouvez ensuite utiliser AWS CloudFormation pour fournir toutes ces ressources en tant qu'unité, et en même temps.

Example modèle qui crée des ressources IAM et lance une simulation

L'exemple de modèle suivant crée un rôle IAM et des autorisations SimSpace Weaver nécessaires pour effectuer des actions dans votre compte. Les scripts du SDK de l' SimSpace Weaver application créent le rôle et les autorisations d'un projet spécifique Région AWS , mais vous pouvez utiliser un AWS CloudFormation modèle pour déployer la simulation sur un autre Région AWS sans exécuter à nouveau les scripts. Par exemple, vous pouvez le faire pour configurer une simulation de sauvegarde à des fins de reprise après sinistre.

Dans cet exemple, le nom de la simulation d'origine est`MySimulation`. Un compartiment pour le schéma existe déjà dans le répertoire Région AWS où AWS CloudFormation sera construite la pile. Le bucket contient une version du schéma correctement configurée pour exécuter la simulation dans celui-ci Région AWS. N'oubliez pas que le schéma indique l'emplacement des fichiers zip de votre application, qui sont un compartiment Amazon S3 Région AWS identique à celui de la simulation. L'application zips bucket et les fichiers doivent déjà exister dans le fichier Région AWS when AWS CloudFormation build the stack, sinon votre simulation ne démarrera pas. Notez que le nom du compartiment dans cet exemple inclut le Région AWS, mais cela ne détermine pas où se trouve réellement le compartiment. Vous devez vous assurer que le compartiment s'y trouve réellement Région AWS (vous pouvez vérifier les propriétés du compartiment dans la console Amazon S3, avec Amazon S3 APIs ou avec les commandes Amazon S3 dans le AWS CLI).

Cet exemple utilise des fonctions et des paramètres intégrés AWS CloudFormation pour effectuer une substitution de variables. Pour plus d'informations, reportez-vous aux sections Référence intrinsèque aux fonctions et Référence aux pseudo-paramètres dans le Guide de AWS CloudFormation l'utilisateur.

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  WeaverAppRole:
    Type: AWS::IAM::Role
    Properties:
      RoleName: SimSpaceWeaverAppRole
      AssumeRolePolicyDocument:
        Version: 2012-10-17
```

```
        Statement:
        - Effect: Allow
          Principal:
            Service:
              - simspaceweaver.amazonaws.com
          Action:
            - sts:AssumeRole
      Path: /
      Policies:
        - PolicyName: SimSpaceWeaverAppRolePolicy
          PolicyDocument:
            Version: 2012-10-17
            Statement:
            - Effect: Allow
              Action:
                - logs:PutLogEvents
                - logs:DescribeLogGroups
                - logs:DescribeLogStreams
                - logs:CreateLogGroup
                - logs:CreateLogStream
              Resource: *
            - Effect: Allow
              Action:
                - cloudwatch:PutMetricData
              Resource: *
            - Effect: Allow
              Action:
                - s3:ListBucket
                - s3:PutObject
                - s3:GetObject
              Resource: *
  MyBackupSimulation:
    Type: AWS::SimSpaceWeaver::Simulation
    Properties:
      Name: !Sub 'mySimulation-${AWS::Region}'
      RoleArn: !GetAtt WeaverAppRole.Arn
      SchemaS3Location:
        BucketName: !Sub 'weaver-mySimulation-${AWS::AccountId}-schemas-${AWS::Region}'
        ObjectKey: !Sub 'schema/mySimulation-${AWS::Region}-schema.yaml'
```

Utilisation d'instantanés avec AWS CloudFormation

Un instantané est une sauvegarde d'une simulation. L'exemple suivant démarre une nouvelle simulation à partir d'un instantané plutôt que d'un schéma. L'instantané de cet exemple a été créé à partir d'une simulation de projet du SDK d' SimSpace Weaver application. AWS CloudFormation crée la nouvelle ressource de simulation et l'initialise avec les données de l'instantané. La nouvelle simulation peut être `MaximumDuration` différente de la simulation d'origine.

Nous vous recommandons de créer et d'utiliser une copie du rôle d'application de votre simulation d'origine. Le rôle d'application de la simulation d'origine peut être supprimé si vous supprimez la AWS CloudFormation pile de cette simulation.

```
Description: "Example - Start a simulation from a snapshot"
Resources:
  MyTestSimulation:
    Type: "AWS::SimSpaceWeaver::Simulation"
    Properties:
      MaximumDuration: "2D"
      Name: "MyTestSimulation_from_snapshot"
      RoleArn: "arn:aws:iam::111122223333:role/weaver-MyTestSimulation-app-role-copy"

      SnapshotS3Location:
        BucketName: "weaver-mytestsimulation-111122223333-artifacts-us-west-2"
        ObjectKey: "snapshot/MyTestSimulation_22-12-15_12_00_00-230428-1207-13.zip"
```

Instantanés

Vous pouvez créer un instantané pour sauvegarder les données de vos entités de simulation à tout moment. SimSpace Weaver crée un fichier .zip dans un compartiment Amazon S3. Vous pouvez créer une nouvelle simulation avec l'instantané. SimSpace Weaver initialise le State Fabric de votre nouvelle simulation avec les données d'entité stockées dans l'instantané, démarre les applications spatiales et de service qui étaient en cours d'exécution lorsque le cliché a été créé et met l'horloge sur le tic-tac approprié. SimSpace Weaver obtient la configuration de votre simulation à partir de l'instantané plutôt que d'un fichier de schéma. Les fichiers .zip de votre application doivent se trouver au même emplacement dans Amazon S3 que dans la simulation d'origine. Vous devez démarrer les applications personnalisées séparément.

Rubriques

- Cas d'utilisation pour les instantanés
- Utiliser la SimSpace Weaver console pour travailler avec des instantanés
- Utilisez le AWS CLI pour travailler avec des instantanés
- Utilisation d'instantanés avec AWS CloudFormation
- Questions fréquemment posées sur les instantanés

Cas d'utilisation pour les instantanés

Revenez à un état précédent et explorez les scénarios de branchement

Vous pouvez créer un instantané de votre simulation pour l'enregistrer dans un état spécifique. Vous pouvez ensuite créer plusieurs nouvelles simulations à partir de cet instantané et explorer différents scénarios susceptibles de découler de cet état.

Meilleures pratiques en matière de reprise après sinistre et de sécurité

Nous vous recommandons de sauvegarder régulièrement votre simulation, en particulier pour les simulations qui s'exécutent pendant plus d'une heure ou qui font appel à plusieurs intervenants. Les sauvegardes peuvent vous aider à vous remettre en cas de sinistre ou d'incident de sécurité. Les instantanés vous permettent de sauvegarder votre simulation. Les instantanés nécessitent que les fichiers .zip de votre application existent au même emplacement dans Amazon S3 qu'auparavant. Si vous devez être en mesure de déplacer les fichiers .zip de votre application vers un autre emplacement, vous devez utiliser une solution de sauvegarde personnalisée.

Pour plus d'informations sur les autres meilleures pratiques, consultez Bonnes pratiques en matière de travail avec SimSpace Weaver etBonnes pratiques de sécurité pour SimSpace Weaver.

Prolongez la durée de votre simulation

Votre ressource de simulation est la représentation de votre simulation dans SimSpace Weaver. Toutes les ressources de simulation ont un `MaximumDuration` paramètre. Une ressource de simulation s'arrête automatiquement lorsqu'elle atteint sa valeur`MaximumDuration`. La valeur maximale de `MaximumDuration` est `14D` (14 jours).

Si vous souhaitez que votre simulation persiste plus longtemps que sa ressource `MaximumDuration` de simulation, vous pouvez créer un instantané avant que la ressource de

simulation n'atteigne sa limite`MaximumDuration`. Vous pouvez démarrer une nouvelle simulation (créer une nouvelle ressource de simulation) avec votre instantané. SimSpace Weaver initialise les données de votre entité à partir de l'instantané, lance les mêmes applications spatiales et de service que celles qui s'exécutaient auparavant et rétablit l'horloge. Vous pouvez démarrer vos applications personnalisées et effectuer toute initialisation personnalisée supplémentaire. Vous pouvez définir une valeur différente pour la nouvelle ressource de simulation lorsque vous la démarrez. `MaximumDuration`

Utiliser la SimSpace Weaver console pour travailler avec des instantanés

Vous pouvez utiliser la SimSpace Weaver console pour créer un instantané de votre simulation.

Rubriques

- Utiliser la console pour créer un instantané
- Utiliser la console pour démarrer une simulation à partir d'un instantané

Utiliser la console pour créer un instantané

Pour créer un instantané

1. Connectez-vous à la SimSpace Weaver console AWS Management Console et connectez-vous à celle-ci.
2. Choisissez Simulations dans le volet de navigation.
3. Sélectionnez le bouton radio à côté du nom de la simulation. Le statut de votre simulation doit être démarré.
4. En haut de la page, choisissez Créer un instantané.
5. Dans Paramètres des instantanés, pour Destination des instantanés, entrez l'URI Amazon S3 d'un compartiment ou d'un compartiment et d'un dossier dans lesquels vous SimSpace Weaver souhaitez créer votre instantané. Vous pouvez choisir Browse S3 si vous préférez parcourir les compartiments disponibles et sélectionner un emplacement.

> **Important**
>
> Le compartiment Amazon S3 doit se trouver dans le même emplacement Région AWS que celui de la simulation.

> **Note**
>
> SimSpace Weaver crée un `snapshot` dossier dans la destination de capture d'écran que vous avez sélectionnée. SimSpace Weaver crée le fichier .zip de capture d'écran dans ce `snapshot` dossier.

6. Choisissez Créer un instantané.

Utiliser la console pour démarrer une simulation à partir d'un instantané

Pour démarrer une simulation à partir d'un instantané, votre fichier .zip d'instantané doit exister dans un compartiment Amazon S3 auquel votre simulation peut accéder. Votre simulation utilise les autorisations définies dans le rôle d'application que vous sélectionnez lorsque vous démarrez la simulation. Tous les fichiers .zip de l'application issus de la simulation d'origine doivent se trouver au même emplacement que lors de la création de l'instantané.

Pour démarrer une simulation à partir d'un instantané

1. Connectez-vous à la SimSpace Weaver console AWS Management Console et connectez-vous à celle-ci.
2. Choisissez Simulations dans le volet de navigation.
3. En haut de la page, choisissez Démarrer la simulation.
4. Sous Paramètres de simulation, entrez un nom et une description facultative pour votre simulation. Le nom de votre simulation doit être unique dans votre Compte AWS.
5. Pour la méthode de démarrage de la simulation, choisissez Utiliser un instantané dans Amazon S3.
6. Pour l'URI Amazon S3 pour le snapshot, entrez l'URI Amazon S3 de votre fichier de snapshot, ou choisissez Browse S3 pour parcourir et sélectionner le fichier.

> **Important**
>
> Le compartiment Amazon S3 doit se trouver dans le même emplacement Région AWS que celui de la simulation.

7. Pour le rôle IAM, sélectionnez le rôle d'application que votre simulation utilisera.

8. Dans Durée maximale, entrez la durée maximale pendant laquelle votre ressource de simulation doit s'exécuter. La valeur maximale est `14D`. Pour plus d'informations sur la durée maximale, consultez.
9. Sous Balises - facultatif, choisissez Ajouter une nouvelle étiquette si vous souhaitez ajouter une étiquette.
10. Choisissez Démarrer la simulation.

Utilisez le AWS CLI pour travailler avec des instantanés

Vous pouvez utiliser le AWS CLI pour appeler le SimSpace Weaver APIs depuis une invite de commande. Vous devez l'avoir AWS CLI installé et configuré correctement. Pour plus d'informations, consultez la section Installation ou mise à jour de la dernière version de la AWS CLI dans le guide de AWS Command Line Interface l'utilisateur pour la version 2.

Rubriques

- Utilisez le AWS CLI pour créer un instantané
- Utilisez le AWS CLI pour démarrer une simulation à partir d'un instantané

Utilisez le AWS CLI pour créer un instantané

Pour créer un instantané

- À l'invite de commande, appelez l'`CreateSnapshot`API.

```
aws simspaceweaver create-snapshot --simulation simulation-name –destination s3-destination
```

Paramètres

simulation

Le nom d'une simulation démarrée. Vous pouvez l'utiliser `aws simspaceweaver list-simulations` pour voir les noms et les statuts de vos simulations.

destination

Chaîne qui indique le compartiment Amazon S3 de destination et le préfixe de clé d'objet facultatif pour votre fichier de capture instantanée. Le préfixe de votre clé d'objet est

généralement un dossier de votre bucket. SimSpace Weaver crée votre instantané dans un `snapshot` dossier à cette destination.

> **Important**
>
> Le compartiment Amazon S3 doit se trouver dans le même emplacement Région AWS que celui de la simulation.

Exemple

```
aws simspaceweaver create-snapshot –simulation
 MyProjectSimulation_23-04-29_12_00_00 –destination BucketName=weaver-
myproject-111122223333-artifacts-us-west-2,ObjectKeyPrefix=myFolder
```

Pour plus d'informations sur l'`CreateSnapshot`API, consultez CreateSnapshotla référence de l'AWS SimSpace Weaver API.

Utilisez le AWS CLI pour démarrer une simulation à partir d'un instantané

Pour démarrer une simulation à partir d'un instantané

- À l'invite de commande, appelez l'`StartSimulation`API.

```
aws simspaceweaver start-simulation --name simulation-name --role-arn role-arn --
snapshot-s3-location s3-location
```

Paramètres

nom

Nom de la nouvelle simulation. Le nom de la simulation doit être unique dans votre Compte AWS. Vous pouvez l'utiliser `aws simspaceweaver list-simulations` pour voir les noms de vos simulations existantes.

role-arn

Le nom de ressource Amazon (ARN) du rôle d'application que votre simulation utilisera.

emplacement de snapshot-s3

Chaîne qui spécifie le compartiment Amazon S3 et la clé d'objet de votre fichier de capture instantanée.

> **Important**
>
> Le compartiment Amazon S3 doit se trouver dans le même emplacement Région AWS que celui de la simulation.

Exemple

```
aws simspaceweaver start-simulation –name MySimulation –role-arn
 arn:aws:iam::111122223333:role/weaver-MyProject-app-role –snapshot-s3-location
 BucketName=weaver-myproject-111122223333-artifacts-us-west-2,ObjectKey=myFolder/
snapshot/MyProjectSimulation_23-04-29_12_00_00-230429-1530-27.zip
```

Pour plus d'informations sur l'`StartSimulation`API, consultez StartSimulationla référence de l'AWS SimSpace Weaver API.

Questions fréquemment posées sur les instantanés

Ma simulation continue-t-elle de s'exécuter pendant un instantané ?

Vos ressources de simulation continuent de s'exécuter pendant un instantané et vous continuez à recevoir des frais de facturation pour cette période. Le temps est pris en compte dans la durée maximale de votre simulation. Vos applications ne reçoivent pas de ticks lorsque l'instantané est en cours de création. Si l'état de votre horloge était `STARTED` celui où la création de l'instantané a commencé, votre horloge indiquera toujours `STARTED` l'état. Vos applications reçoivent à nouveau des coches une fois l'instantané terminé. Si l'état de votre horloge était le `STOPPED` même, votre état d'horloge restera le mêmeSTOPPED. Notez qu'une simulation avec un `STARTED` statut est en cours d'exécution même si son état d'horloge l'estSTOPPED.

Que se passe-t-il si un instantané est en cours et que ma simulation atteint sa durée maximale ?

Votre simulation terminera l'instantané, puis s'arrêtera dès que le processus de capture d'écran sera terminé (avec ou sans succès). Nous vous recommandons de tester le processus de capture d'écran

au préalable pour déterminer sa durée, la taille du fichier de capture à laquelle vous pouvez vous attendre et s'il doit s'exécuter correctement.

Que se passe-t-il si j'arrête une simulation dont un instantané est en cours ?

Un instantané en cours s'arrête immédiatement lorsque vous arrêtez la simulation. Cela ne créera pas de fichier instantané.

Comment puis-je arrêter un instantané en cours ?

Le seul moyen d'arrêter un instantané en cours est d'arrêter la simulation. Vous ne pouvez pas redémarrer une simulation après l'avoir arrêtée.

Combien de temps me faudra-t-il pour terminer mon instantané ?

Le temps nécessaire pour créer un instantané dépend de votre simulation. Nous vous recommandons de tester le processus de capture d'écran au préalable pour savoir combien de temps prendra votre simulation.

Quelle sera la taille de mon fichier de capture d'écran ?

La taille d'un fichier de capture dépend de votre simulation. Nous vous recommandons de tester le processus de capture d'écran au préalable afin de déterminer la taille du fichier pour votre simulation.

Messagerie

L'API de messagerie simplifie la communication entre applications dans le cadre de la simulation. APIs pour envoyer et recevoir des messages font partie du SDK de l' SimSpace Weaver application. La messagerie utilise actuellement une approche basée sur le meilleur effort pour envoyer et recevoir des messages. SimSpace Weaver tente d'envoyer/recevoir des messages lors de la prochaine simulation, mais il n'existe aucune garantie quant à la livraison, à la commande ou à l'heure d'arrivée.

Rubriques

- Cas d'utilisation de la messagerie
- Utilisation de la messagerie APIs
- Quand utiliser la messagerie
- Conseils relatifs à l'utilisation de la messagerie
- Erreurs de messagerie et résolution des problèmes

Cas d'utilisation de la messagerie

Communiquez entre les applications de simulation

Utilisez l'API de messagerie pour communiquer entre les applications de votre simulation. Utilisez-le pour modifier l'état des entités à distance, modifier le comportement des entités ou diffuser des informations sur l'ensemble de la simulation.

Accusez réception d'un message

Les messages envoyés contiennent des informations sur l'expéditeur dans l'en-tête du message. Utilisez ces informations pour renvoyer un accusé de réception à la réception d'un message.

Transférer les données reçues par une application personnalisée vers d'autres applications dans le cadre de la simulation

La messagerie ne remplace pas la façon dont les clients se connectent aux applications personnalisées en cours d'exécution SimSpace Weaver. Cependant, la messagerie permet aux utilisateurs de transférer des données depuis des applications personnalisées recevant des données client vers d'autres applications qui ne disposent pas d'une connexion externe. Le flux de messages peut également fonctionner en sens inverse, permettant aux applications sans connexion externe de transférer des données vers une application personnalisée puis vers un client.

Utilisation de la messagerie APIs

Les messages APIs sont contenus dans le SDK de l' SimSpace Weaver application (version minimale 1.16.0). La messagerie est prise en charge en C++, Python et dans nos intégrations avec Unreal Engine 5 et Unity.

Deux fonctions gèrent les transactions de messages : `SendMessage` et`ReceiveMessages`. Tous les messages envoyés contiennent une destination et une charge utile. L'`ReceiveMessages`API renvoie une liste des messages actuellement présents dans la file d'attente des messages entrants d'une application.

C++

Envoyer un message

```
AWS_WEAVERRUNTIME_API Result<void> SendMessage(
```

```
    Transaction& txn,
    const MessagePayload& payload,
    const MessageEndpoint& destination,
    MessageDeliveryType deliveryType = MessageDeliveryType::BestEffort
    ) noexcept;
```

Recevoir des messages

```
AWS_WEAVERRUNTIME_API Result<MessageList> ReceiveMessages(
    Transaction& txn) noexcept;
```

Python

Envoyer un message

```
api.send_message(
 txn, # Transaction
 payload, # api.MessagePayload
 destination, # api.MessageDestination
 api.MessageDeliveryType.BestEffort # api.MessageDeliveryType
)
```

Recevoir des messages

```
api.receive_messages(
 txn, # Transaction
) -> api.MessageList
```

Rubriques

- Envoi de messages
- Réception de messages
- Répondre à l'expéditeur

Envoi de messages

Les messages se composent d'une transaction (similaire aux autres appels d'API Weaver), d'une charge utile et d'une destination.

Charge utile du message

La charge utile des messages est une structure de données flexible de 256 octets maximum. Nous vous recommandons de suivre les bonnes pratiques suivantes pour créer les charges utiles de vos messages.

Pour créer la charge utile du message

1. Créez une structure de données (telle qu'une structure `struct` en C++) qui définit le contenu du message.
2. Créez la charge utile du message contenant les valeurs à envoyer dans votre message.
3. Créez l'`MessagePayload`objet.

Destination du message

La destination d'un message est définie par l'`MessageEndpoint`objet. Cela inclut à la fois un type de point de terminaison et un identifiant de point de terminaison. Le seul type de point de terminaison actuellement pris en charge est `Partition` celui qui vous permet d'adresser des messages à d'autres partitions dans la simulation. L'ID du point de terminaison est l'ID de partition de votre destination cible.

Vous ne pouvez fournir qu'une seule adresse de destination dans un message. Créez et envoyez plusieurs messages si vous souhaitez envoyer des messages à plusieurs partitions en même temps.

Pour obtenir des conseils sur la résolution d'un point de terminaison de message à partir d'une position, consultezConseils relatifs à l'utilisation de la messagerie.

Envoyer le message

Vous pouvez utiliser l'`SendMessage`API après avoir créé les objets de destination et de charge utile.

C++

```
Api::SendMessage(transaction, payload, destination,
 MessageDeliveryType::BestEffort);
```

Python

```
api.send_message(txn, payload, destination, api.MessageDeliveryType.BestEffort)
```

Exemple complet d'envoi de messages

L'exemple suivant montre comment créer et envoyer un message générique. Cet exemple envoie 16 messages individuels. Chaque message contient une charge utile d'une valeur comprise entre 0 et 15, et la simulation en cours fonctionne.

Example

C++

```
// Message struct definition
struct MessageTickAndId
{
    uint32_t id;
    uint32_t tick;
};

Aws::WeaverRuntime::Result<void> SendMessages(Txn& txn) noexcept
{
     // Fetch the destination MessageEndpoint with the endpoint resolver
    WEAVERRUNTIME_TRY(
        Api::MessageEndpoint destination,
        Api::Utils::MessageEndpointResolver::ResolveFromPosition(
        txn,
            "MySpatialSimulation",
            Api::Vector2F32 {231.3, 654.0}
        )
    );
    Log::Info("destination: ", destination);

    WEAVERRUNTIME_TRY(auto tick, Api::CurrentTick(txn));

    uint16_t numSentMessages = 0;
    for (std::size_t i=0; i<16; i++)
    {
        // Create the message that'll be serialized into payload
        MessageTickAndId message {i, tick.value};

        // Create the payload out of the struct
        const Api::MessagePayload& payload = Api::Utils::CreateMessagePayload(
            reinterpret_cast<const std::uint8_t*>(&message),
            sizeof(MessageTickAndId)
        );
```

```
        // Send the payload to the destination
        Result<void> result = Api::SendMessage(txn, payload, destination);
        if (result.has_failure())
        {
            // SendMessage has failure modes, log them
            auto error = result.as_failure().error();
            std::cout<< "SendMessage failed, ErrorCode: " << error << std::endl;
            continue;
        }

        numSentMessages++;
    }

    std::cout << numSentMessages << " messages is sent to endpoint"
       << destination << std::endl;
    return Aws::WeaverRuntime::Success();
}
```

Python

```
# Message data class
@dataclasses.dataclass
class MessageTickAndId:
    tick: int = 0
    id: int = 0

# send messages
def _send_messages(self, txn):
    tick = api.current_tick(txn)
    num_messages_to_send = 16

    # Fetch the destination MessageEndpoint with the endpoint resolver
    destination = api.utils.resolve_endpoint_from_domain_name_position(
       txn,
       "MySpatialSimulation",
       pos
   )
    Log.debug("Destination_endpoint = %s", destination_endpoint)

   for id in range(num_messages_to_send):
       # Message struct that'll be serialized into payload
        message_tick_and_id = MessageTickAndId(id = id, tick = tick.value)
```

```
       # Create the payload out of the struct
        message_tick_and_id_data = struct.pack(
           '<ii',
           message_tick_and_id.id,
           message_tick_and_id.tick
       )
        payload = api.MessagePayload(list(message_tick_and_id_data))

        # Send the payload to the destination
        Log.debug("Sending message: %s, endpoint: %s",
           message_tick_and_id,
           destination
       )
        api.send_message(
           txn,
           payload,
           destination,
           api.MessageDeliveryType.BestEffort
       )

    Log.info("Sent %s messages to %s", num_messages_to_send, destination)
    return True
```

Réception de messages

SimSpace Weaver envoie les messages dans la file d'attente des messages entrants d'une partition. Utilisez l'`ReceiveMessages`API pour obtenir un `MessageList` objet contenant les messages de la file d'attente. Traitez chaque message avec l'`ExtractMessage`API pour obtenir les données du message.

Example

C++

```
Result<void> ReceiveMessages(Txn& txn) noexcept
{
     // Fetch all the messages sent to the partition owned by the app
    WEAVERRUNTIME_TRY(auto messages, Api::ReceiveMessages(txn));
    std::cout << "Received" << messages.messages.size() << " messages" << std::endl;
    for (Api::Message& message : messages.messages)
    {
```

```
        std::cout << "Received message: " << message << std::endl;

         // Deserialize payload to the message struct
        const MessageTickAndId& receivedMessage
            = Api::Utils::ExtractMessage<MessageTickAndId>(message);
        std::cout << "Received MessageTickAndId, Id: " << receivedMessage.id
            <<", Tick: " << receivedMessage.tick << std::endl;
    }

    return Aws::WeaverRuntime::Success();
}
```

Python

```
# process incoming messages
def _process_incoming_messages(self, txn):
    messages = api.receive_messages(txn)
    for message in messages:
        payload_list = message.payload.data
        payload_bytes = bytes(payload_list)
        message_tick_and_id_data_struct
           = MessageTickAndId(*struct.unpack('<ii', payload_bytes))

        Log.debug("Received message. Header: %s, message: %s",
                    message.header, message_tick_and_id_data_struct)

    Log.info("Received %s messages", len(messages))
    return True
```

Répondre à l'expéditeur

Chaque message reçu contient un en-tête contenant des informations sur l'expéditeur d'origine du message. Vous pouvez utiliser le message.header.source_endpoint pour envoyer une réponse.

Example

C++

```
Result<void> ReceiveMessages(Txn& txn) noexcept
{
     // Fetch all the messages sent to the partition owned by the app
    WEAVERRUNTIME_TRY(auto messages, Api::ReceiveMessages(txn));
```

```
    std::cout << "Received" << messages.messages.size() << " messages" << std::endl;
    for (Api::Message& message : messages.messages)
    {
        std::cout << "Received message: " << message << std::endl;

         // Deserialize payload to the message struct
        const MessageTickAndId& receivedMessage
            = Api::Utils::ExtractMessage<MessageTickAndId>(message);
        std::cout << "Received MessageTickAndId, Id: " << receivedMessage.id
            <<", Tick: " << receivedMessage.tick << std::endl;

        // Get the sender endpoint and payload to bounce the message back
        Api::MessageEndpoint& sender = message.header.source_endpoint;
        Api::MessagePayload& payload = message.payload;
        Api::SendMessage(txn, payload, sender);
    }

    return Aws::WeaverRuntime::Success();
}
```

Python

```
# process incoming messages
def _process_incoming_messages(self, txn):
    messages = api.receive_messages(txn)
    for message in messages:
        payload_list = message.payload.data
        payload_bytes = bytes(payload_list)
        message_tick_and_id_data_struct
           = MessageTickAndId(*struct.unpack('<ii', payload_bytes))

        Log.debug("Received message. Header: %s, message: %s",
                    message.header, message_tick_and_id_data_struct)
       # Get the sender endpoint and payload
       # to bounce the message back
       sender = message.header.source_endpoint
       payload = payload_list
       api.send_message(
           txn,
           payload_list,
           sender,
           api.MessageDeliveryType.BestEffort
```

```
    Log.info("Received %s messages", len(messages))
    return True
```

Quand utiliser la messagerie

La messagerie SimSpace Weaver entrante offre un autre modèle d'échange d'informations entre les applications de simulation. Les abonnements fournissent un mécanisme d'extraction pour lire les données provenant d'applications ou de domaines spécifiques de la simulation ; les messages fournissent un mécanisme d'extraction pour envoyer des données à des applications ou à des domaines spécifiques de la simulation.

Vous trouverez ci-dessous deux cas d'utilisation dans lesquels il est plus utile d'envoyer des données par messagerie plutôt que de les extraire ou de les lire via un abonnement.

Example 1 : Envoi d'une commande à une autre application pour modifier la position d'une entité

```
// Message struct definition
struct MessageMoveEntity
{
     uint64_t entityId;
    std::array<float, 3> destinationPos;
};

// Create the message
MessageMoveEntity message {45, {236.67, 826.22, 0.0} };

// Create the payload out of the struct
const Api::MessagePayload& payload = Api::Utils::CreateMessagePayload(
    reinterpret_cast<const std::uint8_t*>(&message),
    sizeof(MessageTickAndId)
);

// Grab the MessageEndpoint of the recipient app.
Api::MessageEndpoint destination = ...

// One way is to resolve it from the domain name and position
WEAVERRUNTIME_TRY(
    Api::MessageEndpoint destination,
    Api::Utils::MessageEndpointResolver::ResolveFromPosition(
    txn,
        "MySpatialSimulation",
```

```
        Api::Vector2F32 {200.0, 100.0}
    )
);

// Then send the message
Api::SendMessage(txn, payload, destination);
```

Du côté récepteur, l'application met à jour la position de l'entité et l'écrit dans State Fabric.

```
Result<void> ReceiveMessages(Txn& txn) noexcept
{
    WEAVERRUNTIME_TRY(auto messages, Api::ReceiveMessages(txn));
    for (Api::Message& message : messages.messages)
    {
        std::cout << "Received message: " << message << std::endl;
         // Deserialize payload to the message struct
        const MessageMoveEntity& receivedMessage
            = Api::Utils::ExtractMessage<MessageMoveEntity>(message);

        ProcessMessage(txn, receivedMessage);
    }

    return Aws::WeaverRuntime::Success();
}

void ProcessMessage(Txn& txn, const MessageMoveEntity& receivedMessage)
{
     // Get the entity corresponding to the entityId
    Entity entity = EntityFromEntityId (receivedMessage.entityId);

    // Update the position and write to StateFabric
    WEAVERRUNTIME_TRY(Api::StoreEntityIndexKey(
            txn,
            entity,
            k_vector3f32TypeId, // type id of the entity
            reinterpret_cast<std::int8_t*>(&receivedMessage.destinationPos),
            sizeof(receivedMessage.destinationPos)));

}
```

Example 2 : Envoyer un message de création d'entité à une application spatiale

```
struct WeaverMessage
```

```
{
    const Aws::WeaverRuntime::Api::TypeId messageTypeId;
};

const Aws::WeaverRuntime::Api::TypeId k_createEntityMessageTypeId = { 1 };

struct CreateEntityMessage : WeaverMessage
{
    const Vector3 position;
   const Aws::WeaverRuntime::Api::TypeId typeId;
};


CreateEntityMessage messageData {
    k_createEntityMessageTypeId,
    Vector3{ position.GetX(), position.GetY(), position.GetZ() },
    Api::TypeId { 0 }
}

WEAVERRUNTIME_TRY(Api::MessageEndpoint destination,
 Api::Utils::MessageEndpointResolver::ResolveFromPosition(
    transaction, "MySpatialDomain", DemoFramework::ToVector2F32(position)
));

Api::MessagePayload payload = Api::Utils::CreateMessagePayload(
    reinterpret_cast<const uint8_t*>(&messageData),
    sizeof(CreateEntityMessage));

Api::SendMessage(transaction, payload, destination);
```

Du côté récepteur, l'application crée une nouvelle entité dans le State Fabric et met à jour sa position.

```
Result<void> ReceiveMessages(Txn& txn) noexcept
{
    WEAVERRUNTIME_TRY(auto messageList, Api::ReceiveMessages(transaction));
    WEAVERRUNTIME_TRY(auto tick, Api::CurrentTick(transaction));
    for (auto& message : messageList.messages)
    {
        // cast to base WeaverMessage type to determine MessageTypeId
        WeaverMessage weaverMessageBase =
 Api::Utils::ExtractMessage<WeaverMessage>(message);
        if (weaverMessageBase.messageTypeId == k_createEntityMessageTypeId)
        {
```

```
            CreateEntityMessage createEntityMessageData =
                Api::Utils::ExtractMessage<CreateEntityMessage>(message);
        CreateActorFromMessage(transaction, createEntityMessageData));
        }
        else if (weaverMessageBase.messageTypeId == k_tickAndIdMessageTypeId)
        {
            ...
        }
    }
}

void ProcessMessage(Txn& txn, const CreateEntityMessage& receivedMessage)
{
    // Create entity
    WEAVERRUNTIME_TRY(
        Api::Entity entity,
        Api::CreateEntity(transaction, receivedMessage.typeId)
    );

    // Update the position and write to StateFabric
    WEAVERRUNTIME_TRY(Api::StoreEntityIndexKey(
        transaction,
        entity,
        receivedMessage.typeId,
        reinterpret_cast<std::int8_t*>(&receivedMessage.position),
        sizeof(receivedMessage.position)));
}
```

Conseils relatifs à l'utilisation de la messagerie

Résoudre un point de terminaison à partir d'une position ou d'un nom d'application

Vous pouvez utiliser cette `AllPartitions` fonction pour obtenir les limites spatiales et l'ID de domaine dont vous avez besoin pour déterminer la partition IDs et les destinations des messages. Toutefois, si vous connaissez la position à laquelle vous souhaitez envoyer un message, mais pas son ID de partition, vous pouvez utiliser la MessageEndpointResolver fonction.

```
/**
* Resolves MessageEndpoint's from various inputs
**/
class MessageEndpointResolver
{
```

```
    public:
    /**
    * Resolves MessageEndpoint from position information
    **/
    Result<MessageEndpoint> ResolveEndpointFromPosition(
        const DomainId& domainId,
        const weaver_vec3_f32_t& pos);

    /**
    * Resolves MessageEndpoint from custom app name
    **/
    Result<MessageEndpoint> ResolveEndpointFromCustomAppName(
        const DomainId& domainId,
        const char* agentName);
};
```

Sérialisation et désérialisation de la charge utile des messages

Vous pouvez utiliser les fonctions suivantes pour créer et lire des charges utiles de messages. Pour plus d'informations, consultez MessagingUtils .h dans la bibliothèque du SDK de l'application sur votre système local.

```
/**
* Utility function to create MessagePayload from a custom type
*
* @return The @c MessagePayload.
*/
template <class T>
AWS_WEAVERRUNTIME_API MessagePayload CreateMessagePayload(const T& message) noexcept
{
    const std::uint8_t* raw_data = reinterpret_cast<const std::uint8_t*>(&message);

    MessagePayload payload;
    std::move(raw_data, raw_data + sizeof(T), std::back_inserter(payload.data));

    return payload;
}

/**
* Utility function to convert MessagePayload to custom type
*/
template <class T>
```

```
AWS_WEAVERRUNTIME_API T ExtractMessage(const MessagePayload& payload) noexcept
{
    return *reinterpret_cast<const T*>(payload.data.data());
}
```

Erreurs de messagerie et résolution des problèmes

Les erreurs suivantes peuvent se produire lorsque vous utilisez la messagerie APIs.

Erreurs de résolution des terminaux

Ces erreurs peuvent se produire avant qu'une application n'envoie un message.

Vérification du nom de domaine

L'envoi d'un message à un point de terminaison non valide entraîne l'erreur suivante :

```
ManifoldError::InvalidArgument {"No DomainId found for the given domain name" }
```

Cela peut se produire lorsque vous essayez d'envoyer un message à une application personnalisée et que cette application personnalisée n'a pas encore rejoint la simulation. Utilisez l'`DescribeSimulation`API pour vous assurer que votre application personnalisée a été lancée avant de lui envoyer un message. Ce comportement est le même dans SimSpace Weaver Local et le AWS Cloud.

Contrôle de position

La tentative de résolution d'un point de terminaison avec un nom de domaine valide mais une position non valide entraîne l'erreur suivante.

```
ManifoldError::InvalidArgument {"Could not resolve endpoint from domain : DomainId
 { value: domain-id } and position: Vector2F32 { x: x-position, y: y-position}" }
```

Nous vous suggérons d'utiliser le `MessageEndpointResolver` dans la `MessageUtils` bibliothèque contenue dans le SDK de SimSpace Weaver l'application.

Erreurs d'envoi de messages

Les erreurs suivantes peuvent se produire lorsqu'une application envoie un message.

Limite d'envoi de messages par application, par tick, dépassée

La limite actuelle du nombre de messages pouvant être envoyés par application et par test de simulation est de 128. Les appels suivants sur la même case échoueront avec l'erreur suivante :

```
ManifoldError::CapacityExceeded {"At Max Outgoing Message capacity: {}", 128}
```

SimSpace Weaver essaie d'envoyer des messages non envoyés à la prochaine case. Réduisez la fréquence d'envoi pour résoudre ce problème. Combinez des charges utiles de messages inférieures à la limite de 256 octets pour réduire le nombre de messages sortants.

Ce comportement est le même dans SimSpace Weaver Local et dans le AWS Cloud.

Limite de charge utile des messages dépassée

La limite actuelle de la taille de charge utile des messages est de 256 octets dans les deux SimSpace Weaver Local et dans le AWS Cloud. L'envoi d'un message dont la charge utile est supérieure à 256 octets entraîne l'erreur suivante :

```
ManifoldError::CapacityExceeded {"Message data too large! Max size: {}", 256}
```

SimSpace Weaver vérifie chaque message et rejette uniquement ceux qui dépassent la limite. Par exemple, si votre application essaie d'envoyer 10 messages et que l'un d'entre eux échoue, seul ce message est rejeté. SimSpace Weaver envoie les 9 autres messages.

Ce comportement est le même dans SimSpace Weaver Local et le AWS Cloud.

La destination est la même que la source

Les applications ne peuvent pas envoyer de messages aux partitions qu'elles possèdent. L'erreur suivante s'affiche si une application envoie un message à une partition dont elle est propriétaire.

```
ManifoldError::InvalidArgument { "Destination is the same as source" }
```

Ce comportement est le même dans SimSpace Weaver Local et le AWS Cloud.

Messagerie basée sur le meilleur effort

SimSpace Weaver ne garantit pas la livraison du message. Le service essaiera de terminer la livraison des messages lors de la simulation suivante, mais les messages risquent de se perdre ou d'être retardés.

Bonnes pratiques en matière de travail avec SimSpace Weaver

Nous vous recommandons de suivre les meilleures pratiques suivantes lorsque vous travaillez avec SimSpace Weaver.

Rubriques

- Configurer des alarmes de facturation
- Utiliser SimSpace Weaver Local
- Arrêtez les simulations dont vous n'avez pas besoin
- Supprimer les ressources dont vous n'avez pas besoin
- Disposer de sauvegardes

Configurer des alarmes de facturation

Il est facile de fournir des ressources AWS et de les laisser fonctionner en permanence, même lorsqu'elles ne sont plus nécessaires. Cela peut entraîner des coûts exorbitants qui peuvent être une surprise lorsque vous recevez votre facture. Vous pouvez configurer une alarme dans Amazon CloudWatch qui se déclenchera et vous avertira lorsque vos coûts dépassent un seuil que vous avez défini. Vous pouvez examiner vos coûts à l'aide d'outils de gestion des coûts. Pour plus d'informations, consultez :

- Créez une alarme de facturation pour surveiller vos AWS frais estimés
- Qu'est-ce que AWS Cost Management

Utiliser SimSpace Weaver Local

Nous vous recommandons de les utiliser SimSpace Weaver Local pour développer et tester vos simulations avant de les télécharger sur le SimSpace Weaver service dans le AWS Cloud. Les avantages du développement avec SimSpace Weaver Local sont les suivants :

- Pas besoin d'attendre de gros téléchargements
- Aucune limite quant au nombre de simulations locales que vous pouvez créer

- Le temps de calcul sur votre ordinateur local ne vous est pas facturé
- Accès direct aux sorties de console depuis vos applications
- Modifiez, reconstruisez et redémarrez votre simulation locale sans avoir à la recréer dans AWS Cloud

Arrêtez les simulations dont vous n'avez pas besoin

Vous recevez des frais de facturation pour une simulation en cours d'exécution. Vous devez arrêter une simulation pour ne plus être facturée. L'exécution de simulations est également prise en compte dans votre quota pour le nombre maximum de simulations. Une simulation en cours d'exécution dans laquelle la journalisation est configurée peut également générer de grandes quantités de journaux, pour lesquels vous devez également payer des frais de facturation. Vous devez arrêter toute simulation dont vous n'avez pas besoin pour ne plus recevoir de frais supplémentaires.

> **Important**
>
> L'arrêt de l'horloge de simulation n'arrête pas la simulation, l'horloge arrête simplement de publier des ticks dans vos applications. Vous ne pouvez pas redémarrer une simulation après l'avoir arrêtée.

Supprimer les ressources dont vous n'avez pas besoin

Chaque simulation que vous créez crée SimSpace Weaver également des ressources dans d'autres AWS services. Vous pouvez obtenir des frais de facturation pour les ressources et les données dans le cadre de ces autres services. Les simulations en cours et celles qui ont échoué sont prises en compte dans votre quota pour le nombre maximum de simulations. Vous devez supprimer les simulations infructueuses inutiles afin de pouvoir démarrer de nouvelles simulations. Lorsque vous supprimez une simulation, les ressources de votre simulation qui existent dans d'autres AWS services peuvent ne pas être supprimées. Par exemple, toutes les données du journal de simulation dans Amazon CloudWatch Logs y resteront jusqu'à ce que vous les supprimiez. Des frais de facturation vous seront facturés pour ces données de journal. Vous devez nettoyer toutes les ressources associées à vos simulations si vous n'en avez plus besoin.

Disposer de sauvegardes

Il est conseillé de disposer de sauvegardes et de plans de sauvegarde pour tout. Vous ne devez pas partir du principe que vous n'avez pas à AWS les sauvegarder simplement parce que vos données s'y trouvent. Vous devez créer votre propre système si vous devez sauvegarder l'état de votre simulation. Envisagez d'en utiliser plusieurs Régions AWS et de mettre en place un plan pour pouvoir rapidement transférer votre charge de travail de production à une autre Région AWS si nécessaire. Pour plus d'informations sur Régions AWS ce support SimSpace Weaver, consultezSimSpace Points de terminaison et quotas Weaver.

Sécurité dans AWS SimSpace Weaver

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez de centres de données et d'architectures réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le modèle de responsabilité partagée décrit cela comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS Cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de AWS conformité Programmes de de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent à AWS SimSpace Weaver, voir AWS Services concernés par programme de conformitéAWS .
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris de la sensibilité de vos données, des exigences de votre entreprise, ainsi que de la législation et de la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de son utilisation SimSpace Weaver. Les rubriques suivantes expliquent comment procéder à la configuration SimSpace Weaver pour atteindre vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui vous aident à surveiller et à sécuriser vos SimSpace Weaver ressources.

Rubriques

- Protection des données dans AWS SimSpace Weaver
- Identity and Access Management pour AWS SimSpace Weaver
- Enregistrement et surveillance des événements de sécurité dans AWS SimSpace Weaver
- Validation de conformité pour AWS SimSpace Weaver
- Résilience dans AWS SimSpace Weaver
- Sécurité de l'infrastructure dans AWS SimSpace Weaver
- Analyse de configuration et de vulnérabilité dans AWS SimSpace Weaver
- Bonnes pratiques de sécurité pour SimSpace Weaver

Protection des données dans AWS SimSpace Weaver

Le modèle de responsabilité AWS partagée de s'applique à la protection des données dans AWS SimSpace Weaver. 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 AWS Identity and Access Management (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 SimSpace Weaver ou d'autres Services

AWS utilisateurs de la console, de l'API ou AWS SDKs. AWS CLI 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 au repos

Les données sont considérées comme inactives lorsqu'elles se trouvent dans un espace de stockage de données non volatil (persistant), tel qu'un disque. Les données situées dans des stockages de données volatils, tels que la mémoire et les registres, ne sont pas considérées comme étant au repos.

Lorsque vous l'utilisez SimSpace Weaver, les seules données inactives sont les suivantes :

- Applications et schémas que vous chargez sur Amazon Simple Storage Service (Amazon S3)
- Données du journal de simulation stockées sur Amazon CloudWatch

Les autres données SimSpace Weaver utilisées en interne ne sont pas conservées une fois que vous avez arrêté votre simulation.

Pour savoir comment chiffrer vos données au repos, consultez :

- Chiffrez vos données dans Amazon S3
- Chiffrez les données de votre journal

Chiffrement en transit

Vos connexions à l' SimSpace Weaver API via le AWS Command Line Interface (AWS CLI), le AWS SDK et le SDK de SimSpace Weaver l'application utilisent le cryptage TLS avec le processus de signature Signature Version 4. AWS gère l'authentification à l'aide des politiques d'accès définies par IAM pour les informations d'identification de sécurité que vous utilisez pour vous connecter.

En interne, SimSpace Weaver utilise le protocole TLS pour se connecter aux autres AWS services qu'il utilise.

> **Important**
>
> Les communications entre vos applications et leurs clients n'impliquent rien SimSpace Weaver. Il est de votre responsabilité de chiffrer les communications avec les clients de simulation, si nécessaire. Nous vous recommandons de créer une solution pour chiffrer toutes les données en transit via les connexions client.

Pour en savoir plus sur les AWS services qui peuvent prendre en charge vos solutions de chiffrement, consultez le blog sur la AWS sécurité.

Confidentialité du trafic inter-réseaux

SimSpace Weaver les ressources de calcul se trouvent au sein d'un Amazon VPC partagé par tous les SimSpace Weaver clients. Tout le trafic des SimSpace Weaver services internes reste sur le AWS réseau et ne transite pas par Internet. La communication entre les clients de simulation et vos applications passe par Internet.

Identity and Access Management pour AWS SimSpace Weaver

AWS Identity and Access Management (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 SimSpace Weaver les ressources. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Rubriques

- Public ciblé
- Authentification par des identités
- Gestion des accès à l’aide de politiques
- Comment AWS SimSpace Weaver fonctionne avec IAM
- Exemples de politiques basées sur l'identité pour AWS SimSpace Weaver
- Des autorisations SimSpace Weaver créées pour vous
- Prévention du problème de l’adjoint confus entre services
- Résolution des problèmes AWS SimSpace Weaver d'identité et d'accès

Public ciblé

La façon dont vous utilisez AWS Identity and Access Management (IAM) varie en fonction du travail que vous effectuez. SimSpace Weaver

Utilisateur du service : si vous utilisez le SimSpace Weaver service pour effectuer votre travail, votre administrateur vous fournit les informations d'identification et les autorisations dont vous avez besoin. Au fur et à mesure que vous utilisez de nouvelles SimSpace Weaver fonctionnalités pour effectuer votre travail, vous aurez peut-être besoin d'autorisations supplémentaires. En comprenant bien la gestion des accès, vous saurez demander les autorisations appropriées à votre administrateur. Si vous ne pouvez pas accéder à une fonctionnalité dans SimSpace Weaver, consultez Résolution des problèmes AWS SimSpace Weaver d'identité et d'accès.

Administrateur du service — Si vous êtes responsable des SimSpace Weaver ressources de votre entreprise, vous avez probablement un accès complet à SimSpace Weaver. C'est à vous de déterminer les SimSpace Weaver fonctionnalités et les ressources auxquelles les utilisateurs de votre service doivent accéder. Vous devez ensuite soumettre les demandes à votre administrateur IAM pour modifier les autorisations des utilisateurs de votre service. Consultez les informations sur cette page pour comprendre les concepts de base d'IAM. Pour en savoir plus sur la manière dont votre entreprise peut utiliser IAM avec SimSpace Weaver, voirComment AWS SimSpace Weaver fonctionne avec IAM.

Administrateur IAM – Si vous êtes un administrateur IAM, vous souhaiterez peut-être en savoir plus sur la façon d'écrire des politiques pour gérer l'accès à SimSpace Weaver. Pour consulter des exemples de politiques SimSpace Weaver basées sur l'identité que vous pouvez utiliser dans IAM, consultez. Exemples de politiques basées sur l'identité pour AWS SimSpace Weaver

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 (connecté à AWS) en tant qu'utilisateur IAM ou en assumant un rôle IAM. Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en AWS tant qu'identité fédérée en utilisant les informations d'identification fournies par le biais d'une source d'identité. AWS IAM Identity Center Les utilisateurs (IAM Identity Center), l'authentification unique de votre entreprise et vos informations d'identification Google ou Facebook sont des exemples d'identités fédérées. Lorsque vous vous connectez avec une identité fédérée, votre administrateur aura précédemment configuré une fédération d'identités avec

des rôles IAM. Lorsque vous accédez à AWS l'aide de la fédération, vous assumez indirectement un rôle.

Selon le type d'utilisateur que vous êtes, vous pouvez vous connecter au portail AWS Management Console ou au portail AWS d'accès. Pour plus d'informations sur la connexion à AWS, consultez la section Comment vous connecter à votre compte Compte AWS dans le guide de Connexion à AWS l'utilisateur.

Si vous y accédez AWS par programmation, AWS fournit un kit de développement logiciel (SDK) et une interface de ligne de commande (CLI) pour signer cryptographiquement vos demandes à l'aide de vos informations d'identification. Si vous n'utilisez pas d' AWS outils, vous devez signer vous-même les demandes. Pour plus d'informations sur l'utilisation de la méthode recommandée pour signer des demandes vous-même, consultez AWS Signature Version 4 pour les demandes d'API dans le Guide de l'utilisateur IAM.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être fournir des informations de sécurité supplémentaires. Par exemple, il vous AWS recommande d'utiliser l'authentification multifactorielle (MFA) pour renforcer la sécurité de votre compte. Pour plus d'informations, consultez Authentification multifactorielle dans le Guide de l'utilisateur AWS IAM Identity Center et Authentification multifactorielle AWS dans IAM dans le Guide de l'utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une identité de connexion unique qui donne un accès complet à toutes Services AWS les ressources du compte. Cette identité est appelée utilisateur Compte AWS root et est accessible en vous connectant avec l'adresse e-mail et le mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas utiliser l'utilisateur racine pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur racine et utilisez-les pour effectuer les tâches que seul l'utilisateur racine peut effectuer. Pour obtenir la liste complète des tâches qui vous imposent de vous connecter en tant qu'utilisateur racine, consultez Tâches nécessitant des informations d'identification d'utilisateur racine dans le Guide de l'utilisateur IAM.

Identité fédérée

La meilleure pratique consiste à obliger les utilisateurs humains, y compris ceux qui ont besoin d'un accès administrateur, à utiliser la fédération avec un fournisseur d'identité pour accéder à l'aide Services AWS d'informations d'identification temporaires.

Une identité fédérée est un utilisateur de l'annuaire des utilisateurs de votre entreprise, d'un fournisseur d'identité Web AWS Directory Service, du répertoire Identity Center ou de tout utilisateur qui y accède à l'aide des informations d'identification fournies Services AWS par le biais d'une source d'identité. Lorsque des identités fédérées y accèdent Comptes AWS, elles 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. Vous pouvez créer des utilisateurs et des groupes dans IAM Identity Center, ou vous pouvez vous connecter et synchroniser avec un ensemble d'utilisateurs et de groupes dans votre propre source d'identité afin de les utiliser dans toutes vos applications Comptes AWS et applications. Pour obtenir des informations sur IAM Identity Center, 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 au sein de vous Compte AWS qui possède des autorisations spécifiques pour une seule personne ou une seule application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des utilisateurs IAM ayant des informations d'identification à long terme telles que des mots de passe et des clés d'accès. Toutefois, si certains cas d'utilisation spécifiques nécessitent des informations d'identification à long terme avec les utilisateurs IAM, nous vous recommandons d'effectuer une rotation des clés d'accès. Pour plus d'informations, consultez Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification dans le Guide de l'utilisateur IAM.

Un groupe IAM est une identité qui concerne un ensemble d'utilisateurs IAM. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez nommer un groupe IAMAdminset lui donner les autorisations nécessaires pour administrer les ressources IAM.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. 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 au sein de vous Compte AWS dotée d'autorisations spécifiques. Le concept ressemble à celui d'utilisateur IAM, mais le rôle IAM n'est pas associé à une personne en particulier. Pour assumer temporairement un rôle IAM dans le AWS Management Console, vous pouvez passer d'un rôle d'utilisateur à un rôle IAM (console). Vous pouvez assumer un rôle en appelant une opération d' AWS API AWS CLI ou en utilisant une URL personnalisée. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez Méthodes pour endosser un rôle dans le Guide de l'utilisateur IAM.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- Accès utilisateur fédéré : pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie, l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour obtenir des informations sur les rôles pour la fédération, consultez Création d'un rôle pour un fournisseur d'identité tiers (fédération) dans le Guide de l'utilisateur IAM. Si vous utilisez IAM Identity Center, vous configurez un jeu d'autorisations. IAM Identity Center met en corrélation le jeu d'autorisations avec un rôle dans IAM afin de contrôler à quoi vos identités peuvent accéder après leur authentification. Pour plus d'informations sur les jeux d'autorisations, consultez Jeux d'autorisations dans le Guide de l'utilisateur AWS IAM Identity Center .
- Autorisations d'utilisateur IAM temporaires : un rôle ou un utilisateur IAM peut endosser un rôle IAM pour profiter temporairement d'autorisations différentes pour une tâche spécifique.
- Accès intercompte : vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (principal de confiance) d'un compte différent d'accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Toutefois, dans certains Services AWS cas, vous pouvez associer une politique directement à une ressource (au lieu d'utiliser un rôle comme proxy). Pour en savoir plus sur la différence entre les rôles et les politiques basées sur les ressources pour l'accès intercompte, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.
- Accès multiservices — Certains Services AWS utilisent des fonctionnalités dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du principal, un rôle de service ou un rôle lié au service.
  - Sessions d'accès direct (FAS) : lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains

services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur une politique lors de la formulation de demandes FAS, consultez Transmission des sessions d'accès.

- Rôle de service : il s'agit d'un rôle IAM attribué à un service afin de réaliser 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.
- Rôle lié à un service — 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.

- Applications exécutées sur Amazon EC2 : vous pouvez utiliser un rôle IAM pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une EC2 instance et qui envoient des demandes AWS CLI d' AWS API. Cela est préférable au stockage des clés d'accès dans l' EC2 instance. Pour attribuer un AWS rôle à une EC2 instance et le rendre disponible pour toutes ses applications, vous devez créer un profil d'instance attaché à l'instance. Un profil d'instance contient le rôle et permet aux programmes exécutés sur l' EC2 instance d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez Utiliser un rôle IAM pour accorder des autorisations aux applications exécutées sur des EC2 instances Amazon dans le guide de l'utilisateur IAM.

Gestion des 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 est un objet AWS qui, lorsqu'il est associé à une identité ou à une ressource, définit leurs autorisations. AWS évalue ces politiques lorsqu'un principal (utilisateur, utilisateur root ou session de rôle) fait une demande. Les autorisations dans les politiques déterminent si la demande est autorisée ou refusée. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations sur la structure et le contenu des documents de politique JSON, consultez Vue d'ensemble des politiques JSON dans le Guide de l'utilisateur IAM.

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.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. 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. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Les politiques IAM définissent les autorisations d'une action, quelle que soit la méthode que vous utilisez pour exécuter l'opération. Par exemple, supposons que vous disposiez d'une politique qui autorise l'action `iam:GetRole`. Un utilisateur appliquant cette politique peut obtenir des informations sur le rôle à partir de AWS Management Console AWS CLI, de ou de l' AWS API.

Politiques basées sur l'identité

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.

Les politiques basées sur l'identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez associer à plusieurs utilisateurs, groupes et rôles au sein de votre Compte AWS. Les politiques gérées incluent les politiques AWS gérées et les politiques gérées par le client. Pour découvrir comment choisir entre une politique gérée et une politique 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. 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

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.

Listes de contrôle d'accès (ACLs)

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.

Amazon S3 et AWS WAF Amazon VPC sont des exemples de services compatibles. ACLs Pour en savoir plus ACLs, consultez la présentation de la liste de contrôle d'accès (ACL) dans le guide du développeur Amazon Simple Storage Service.

Autres types de politique

AWS prend en charge d'autres types de politiques moins courants. Ces types de politiques peuvent définir le nombre maximum d’autorisations qui vous sont accordées par des types de politiques plus courants.

- Limite d’autorisations : une limite d’autorisations est une fonctionnalité avancée dans laquelle vous définissez le nombre maximal d’autorisations qu’une politique basée sur l’identité peut accorder à une entité IAM (utilisateur ou rôle IAM). Vous pouvez définir une limite d’autorisations pour une entité. Les autorisations en résultant représentent la combinaison des politiques basées sur l’identité d’une entité et de ses limites d’autorisation. Les politiques basées sur les ressources qui spécifient l’utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d’autorisations. Un refus explicite dans l’une de ces politiques annule l’autorisation. Pour plus d’informations sur les limites d’autorisations, consultez Limites d’autorisations pour des entités IAM dans le Guide de l’utilisateur IAM.
- Politiques de contrôle des services (SCPs) : SCPs politiques JSON qui spécifient les autorisations maximales pour une organisation ou une unité organisationnelle (UO) dans AWS Organizations. AWS Organizations est un service permettant de regrouper et de gérer de manière centralisée Comptes AWS les multiples propriétés de votre entreprise. Si vous activez toutes les fonctionnalités d'une organisation, vous pouvez appliquer des politiques de contrôle des services (SCPs) à l'un ou à l'ensemble de vos comptes. Le SCP limite les autorisations pour les entités

figurant dans les comptes des membres, y compris chacune Utilisateur racine d'un compte AWS d'entre elles. Pour plus d'informations sur les Organizations et consultez SCPs les politiques de contrôle des services dans le Guide de AWS Organizations l'utilisateur.

- Politiques de contrôle des ressources (RCPs) : RCPs politiques JSON que vous pouvez utiliser pour définir le maximum d'autorisations disponibles pour les ressources de vos comptes sans mettre à jour les politiques IAM associées à chaque ressource que vous possédez. Le RCP limite les autorisations pour les ressources des comptes membres et peut avoir un impact sur les autorisations effectives pour les identités, y compris Utilisateur racine d'un compte AWS, qu'elles appartiennent ou non à votre organisation. Pour plus d'informations sur les Organizations RCPs, y compris une liste de ces Services AWS supports RCPs, consultez la section Resource control policies (RCPs) dans le guide de AWS Organizations l'utilisateur.
- Politiques de séance : les politiques de séance sont des politiques avancées que vous utilisez en tant que paramètre lorsque vous créez par programmation une séance temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de séance en résultant sont une combinaison des politiques basées sur l'identité de l'utilisateur ou du rôle et des politiques de séance. Les autorisations peuvent également provenir d'une politique basée sur les ressources. Un refus explicite dans l'une de ces politiques annule l'autorisation. 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.

Comment AWS SimSpace Weaver fonctionne avec IAM

Avant d'utiliser IAM pour gérer l'accès à SimSpace Weaver, découvrez les fonctionnalités IAM disponibles. SimSpace Weaver

Fonctionnalités IAM que vous pouvez utiliser avec AWS SimSpace Weaver

Fonctionnalité IAM	SimSpace Weaver soutien
Politiques basées sur l'identité	Oui

Fonctionnalité IAM	SimSpace Weaver soutien
Politiques basées sur les ressources	Non
Actions de politique	Oui
Ressources de politique	Oui
Clés de condition de politique (spécifiques au service)	Oui
ACLs	Non
ABAC (étiquettes dans les politiques)	Oui
Informations d’identification temporaires	Oui
Autorisations de principal	Oui
Rôles de service	Oui
Rôles liés à un service	Non

Pour obtenir une vue d'ensemble de la façon dont SimSpace Weaver les autres AWS services fonctionnent avec la plupart des fonctionnalités IAM, consultez la section AWS Services compatibles avec IAM dans le Guide de l'utilisateur IAM.

Politiques basées sur l'identité pour SimSpace Weaver

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. Vous ne pouvez pas spécifier le principal dans une politique basée sur une identité, car celle-ci s'applique à l'utilisateur ou au rôle auquel elle est attachée. 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 SimSpace Weaver

Pour consulter des exemples de politiques SimSpace Weaver basées sur l'identité, consultez. Exemples de politiques basées sur l'identité pour AWS SimSpace Weaver

Politiques basées sur les ressources au sein de SimSpace Weaver

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. L'ajout d'un principal intercompte à une politique basée sur les ressources ne représente qu'une partie de l'instauration de la relation d'approbation. Lorsque le principal et la ressource sont différents Comptes AWS, un administrateur IAM du compte sécurisé doit également accorder à l'entité principale (utilisateur ou rôle) l'autorisation d'accéder à la ressource. Pour ce faire, il attache une politique basée sur une identité à l'entité. Toutefois, si une politique basée sur des ressources accorde l'accès à un principal dans le même compte, aucune autre politique basée sur l'identité n'est requise. Pour plus d'informations, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.

Actions politiques pour SimSpace Weaver

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. Les actions de stratégie portent généralement le même nom que l'opération AWS d'API associée. Il existe quelques exceptions, telles que les actions avec autorisations uniquement qui n'ont pas d'opération API correspondante. Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d'actions dans une politique afin d'accorder l'autorisation d'exécuter les opérations associées.

Pour consulter la liste des SimSpace Weaver actions, reportez-vous à la section Actions définies par AWS SimSpace Weaver dans la référence d'autorisation de service.

Les actions de politique en SimSpace Weaver cours utilisent le préfixe suivant avant l'action :

```
simspaceweaver
```

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

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

Pour consulter des exemples de politiques SimSpace Weaver basées sur l'identité, consultez. Exemples de politiques basées sur l'identité pour AWS SimSpace Weaver

Ressources politiques pour SimSpace Weaver

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. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de définir

une ressource à l'aide de son Amazon Resource Name (ARN). Vous pouvez le faire pour des actions qui prennent en charge un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

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

Pour consulter la liste des types de SimSpace Weaver ressources et leurs caractéristiques ARNs, consultez la section Ressources définies par AWS SimSpace Weaver dans la référence d'autorisation de service. Pour savoir grâce à quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez Actions définies par AWS SimSpace Weaver.

Pour consulter des exemples de politiques SimSpace Weaver basées sur l'identité, consultez. Exemples de politiques basées sur l'identité pour AWS SimSpace Weaver

Clés de conditions de politique pour SimSpace Weaver

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` (ou le bloc `Condition`) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément `Condition` est facultatif. 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.

Si vous spécifiez plusieurs éléments `Condition` dans une instruction, ou plusieurs clés dans un seul élément `Condition`, AWS les évalue à l'aide d'une opération `AND` logique. Si vous spécifiez plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une `OR` opération logique. Toutes les conditions doivent être remplies avant que les autorisations associées à l'instruction ne soient accordées.

Vous pouvez aussi utiliser des variables d'espace réservé quand vous spécifiez des conditions. Par exemple, vous pouvez accorder à un utilisateur IAM l'autorisation d'accéder à une ressource

uniquement si elle est balisée avec son nom d'utilisateur IAM. Pour plus d'informations, consultez Éléments d'une politique IAM : variables et identifications dans le Guide de l'utilisateur IAM.

AWS prend en charge les clés de condition globales et les clés de condition spécifiques au service. 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 SimSpace Weaver condition, reportez-vous à la section Clés de condition pour AWS SimSpace Weaver la référence d'autorisation de service. Pour savoir avec quelles actions et ressources vous pouvez utiliser une clé de condition, consultez la section Actions définies par AWS SimSpace Weaver.

Pour consulter des exemples de politiques SimSpace Weaver basées sur l'identité, consultez. Exemples de politiques basées sur l'identité pour AWS SimSpace Weaver

Listes de contrôle d'accès (ACLs) dans SimSpace Weaver

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.

Contrôle d'accès basé sur les attributs (ABAC) avec SimSpace Weaver

Prise en charge d'ABAC (balises dans les politiques) : Oui

Le contrôle d'accès par attributs (ABAC) est une stratégie d'autorisation qui définit des autorisations en fonction des attributs. Dans AWS, ces attributs sont appelés balises. Vous pouvez associer des balises aux entités IAM (utilisateurs ou rôles) et à de nombreuses AWS ressources. L'étiquetage des entités et des ressources est la première étape d'ABAC. Vous concevez ensuite des politiques ABAC pour autoriser des opérations quand l'identification du principal correspond à celle de la ressource à laquelle il tente d'accéder.

L'ABAC est utile dans les environnements qui connaissent une croissance rapide et pour les cas où la gestion des politiques devient fastidieuse.

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 SimSpace Weaver

Prend en charge les informations d'identification temporaires : oui

Certains Services AWS ne fonctionnent pas lorsque vous vous connectez à l'aide d'informations d'identification temporaires. Pour plus d'informations, y compris celles qui Services AWS fonctionnent avec des informations d'identification temporaires, consultez Services AWS la section relative à l'utilisation d'IAM dans le guide de l'utilisateur d'IAM.

Vous utilisez des informations d'identification temporaires si vous vous connectez à l' AWS Management Console aide d'une méthode autre qu'un nom d'utilisateur et un mot de passe. Par exemple, lorsque vous accédez à AWS l'aide du lien d'authentification unique (SSO) de votre entreprise, ce processus crée automatiquement des informations d'identification temporaires. Vous créez également automatiquement des informations d'identification temporaires lorsque vous vous connectez à la console en tant qu'utilisateur, puis changez de rôle. Pour plus d'informations sur le changement de rôle, consultez Passage d'un rôle utilisateur à un rôle IAM (console) dans le Guide de l'utilisateur IAM.

Vous pouvez créer manuellement des informations d'identification temporaires à l'aide de l' AWS API AWS CLI or. Vous pouvez ensuite utiliser ces informations d'identification temporaires pour y accéder AWS. 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.

Autorisations principales interservices pour SimSpace Weaver

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

Lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une

action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur une politique lors de la formulation de demandes FAS, consultez Transmission des sessions d'accès.

Rôles de service pour SimSpace Weaver

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 SimSpace Weaver les fonctionnalités. Modifiez les rôles de service uniquement lorsque SimSpace Weaver vous recevez des instructions à cet effet.

Les scripts du SDK de SimSpace Weaver l'application utilisent un AWS CloudFormation modèle pour créer des ressources dans d'autres AWS services afin de soutenir votre simulation. L'une de ces ressources est le rôle de l'application pour votre simulation. SimSpace Weaver assume le rôle d'application pour effectuer des actions en votre Compte AWS nom, par exemple pour écrire des données de journal dans CloudWatch Logs. Pour plus d'informations sur le rôle de l'application, consultezDes autorisations SimSpace Weaver créées pour vous.

Rôles liés à un service pour SimSpace Weaver

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.

Exemples de politiques basées sur l'identité pour AWS SimSpace Weaver

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou modifier les ressources SimSpace Weaver . Ils ne peuvent pas non plus effectuer de tâches à l'aide de l'API AWS Management Console, AWS Command Line Interface (AWS CLI) ou de AWS l'API. 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. L’administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

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 SimSpace Weaver, y compris le format de ARNs pour chacun des types de ressources, voir Actions, ressources et clés de condition AWS SimSpace Weaver dans la référence d'autorisation de service.

Rubriques

- Bonnes pratiques en matière de politiques
- Utilisation de la console SimSpace Weaver
- Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations
- Permettre aux utilisateurs de créer et d'exécuter des simulations

Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer SimSpace Weaver des ressources 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 AWS 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 SimSpace Weaver

Pour accéder à la AWS SimSpace Weaver console, vous devez disposer d'un ensemble minimal d'autorisations. Ces autorisations doivent vous permettre de répertorier et d'afficher les détails

SimSpace Weaver des ressources de 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 SimSpace Weaver console, associez également la politique SimSpace Weaver *ConsoleAccess* ou la politique *ReadOnly* AWS 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:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
```

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

Permettre aux utilisateurs de créer et d'exécuter des simulations

Cet exemple de politique IAM fournit les autorisations de base requises pour créer et exécuter des simulations dans SimSpace Weaver.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CreateAndRunSimulations",
            "Effect": "Allow",
            "Action": [
                "simspaceweaver:*",
                "iam:GetRole",
                "iam:ListRoles",
                "iam:CreateRole",
                "iam:DeleteRole",
                "iam:UpdateRole",
                "iam:CreatePolicy",
                "iam:AttachRolePolicy",
                "iam:PutRolePolicy",
                "iam:GetRolePolicy",
                "iam:DeleteRolePolicy",
                "s3:PutObject",
                "s3:GetObject",
                "s3:ListAllMyBuckets",
                "s3:PutBucketPolicy",
                "s3:CreateBucket",
                "s3:ListBucket",
                "s3:PutEncryptionConfiguration",
                "s3:DeleteBucket",
```

```
                "cloudformation:CreateStack",
                "cloudformation:UpdateStack",
                "cloudformation:DescribeStacks"
            ],
            "Resource": "*"
        },
        {
            "Sid": "PassAppRoleToSimSpaceWeaver",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "simspaceweaver.amazonaws.com"
                }
            }
        }
    ]
}
```

Des autorisations SimSpace Weaver créées pour vous

Lorsque vous créez un SimSpace Weaver projet, le service crée un rôle AWS Identity and Access Management (IAM) avec le nom `weaver-`*`project-name`*`-app-role` et une politique de confiance IAM. La politique de confiance permet SimSpace Weaver d'assumer le rôle afin qu'il puisse effectuer des opérations pour vous.

Politique d'autorisation des rôles dans les applications

Le rôle d'application de simulation est soumis à la politique d'autorisation suivante.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:PutLogEvents",
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams",
                "logs:CreateLogGroup",
                "logs:CreateLogStream"
```

```
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": "*"
        }
    ]
}
```

Politique de confiance relative aux rôles des applications

SimSpace Weaver ajoute une relation de confiance au rôle de l'application de simulation en tant que politique de confiance. SimSpace Weaver crée une politique de confiance pour chaque simulation, comme dans l'exemple suivant.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "simspaceweaver.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn":
     "arn:aws:simspaceweaver:us-west-2:111122223333:simulation/MySimName*"
        }
      }
```

```
    }
  ]
}
```

> **Note**
>
> Dans cet exemple, le numéro de compte est `111122223333` et le nom de la simulation est`MySimName`. Ces valeurs sont différentes dans vos politiques de confiance.

Prévention du problème de l’adjoint confus entre services

Le problème des adjoints confus est un problème de sécurité dans lequel une entité qui n'est pas autorisée à effectuer une action peut inciter une entité plus privilégiée à effectuer l'action. En AWS, l'usurpation d'identité interservices peut entraîner la confusion des adjoints. L’usurpation d’identité entre services peut se produire lorsqu’un service (le service appelant) appelle un autre service (le service appelé). Le service appelant peut être manipulé pour utiliser ses autorisations afin d'agir sur les ressources d'un autre client de sorte qu'il n'y aurait pas accès autrement. Pour éviter cela, AWS fournit des outils qui vous aident à protéger vos données pour tous les services avec des principaux de service qui ont eu accès aux ressources de votre compte.

Nous recommandons d'utiliser les clés de contexte de condition `aws:SourceAccount`globale `aws:SourceArn`et les clés contextuelles dans les politiques de ressources afin de limiter les autorisations qui AWS SimSpace Weaver accordent un autre service à la ressource. Si la valeur `aws:SourceArn` ne contient pas l'ID de compte, tel que l'Amazon Resource Name (ARN) d'un compartiment Amazon S3, vous devez utiliser les deux clés de contexte de condition globale pour limiter les autorisations. Si vous utilisez les deux clés de contexte de condition globale et que la valeur `aws:SourceArn` contient l'ID de compte, la valeur `aws:SourceAccount` et le compte dans la valeur `aws:SourceArn` doivent utiliser le même ID de compte lorsqu'ils sont utilisés dans la même instruction de politique. Utilisez `aws:SourceArn` si vous souhaitez qu'une seule ressource soit associée à l'accès entre services. Utilisez `aws:SourceAccount` si vous souhaitez autoriser l’association d’une ressource de ce compte à l’utilisation interservices.

La valeur de `aws:SourceArn` doit utiliser l'ARN de l'extension.

Le moyen le plus efficace de se protéger contre le problème de député confus consiste à utiliser la clé de contexte de condition globale `aws:SourceArn` avec l’ARN complet de la ressource. Si vous ne connaissez pas l'ARN complet de l'extension ou si vous spécifiez plusieurs extensions, utilisez la

clé de contexte de condition globale `aws:SourceArn` avec des caractères génériques (*) pour les parties inconnues de l'ARN. Par exemple, `arn:aws:`*`simspaceweaver`*`:*:`*`111122223333`*`:*`.

L'exemple suivant montre comment vous pouvez utiliser les touches de contexte de condition `aws:SourceAccount` globale `aws:SourceArn` et globale SimSpace Weaver pour éviter le problème de confusion des adjoints. Cette politique ne permet d' SimSpace Weaver assumer le rôle que lorsque la demande provient du compte source spécifié et est fournie avec l'ARN spécifié. Dans ce cas, ne SimSpace Weaver peut assumer le rôle pour les demandes issues de simulations que dans le compte du demandeur (*111122223333*), et uniquement dans la région spécifiée (*us-west-2*).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "simspaceweaver.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        },
        "StringLike": {
          "aws:SourceArn": "arn:aws:simspaceweaver:us-west-2:111122223333:simulation/*"
        }
      }
    }
  ]
}
```

Une méthode plus sûre pour écrire cette politique consiste à inclure le nom de la simulation dans leaws:SourceArn, comme indiqué dans l'exemple suivant, ce qui limite la politique à une simulation nommée `MyProjectSimulation_22-10-04_22_10_15` :

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

```
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "simspaceweaver.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        },
        "StringLike": {
          "aws:SourceArn": "arn:aws:simspaceweaver:us-west-2:111122223333:simulation/
MyProjectSimulation_22-10-04_22_10_15"
        }
      }
    }
  ]
}
```

Lorsque vous `aws:SourceArn` incluez explicitement un numéro de compte, vous pouvez omettre le test d'`Condition`élément pour le `aws:SourceAccount` (voir le guide de l'utilisateur IAM pour plus d'informations), comme dans la politique simplifiée suivante :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "simspaceweaver.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringLike": {
          "aws:SourceArn": "arn:aws:simspaceweaver:us-west-2:111122223333:simulation/
MyProjectSimulation_22-10-04_22_10_15"
        }
      }
    }
```

```
  ]
}
```

Résolution des problèmes AWS SimSpace Weaver d'identité et d'accès

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

Rubriques

- Je ne suis pas autorisé à effectuer une action dans SimSpace Weaver
- Je ne suis pas autorisé à effectuer iam : PassRole
- Je veux afficher mes clés d'accès
- Je suis administrateur et je souhaite autoriser d'autres personnes à accéder SimSpace Weaver
- Je souhaite permettre à des personnes extérieures Compte AWS à moi d'accéder à mes SimSpace Weaver ressources

Je ne suis pas autorisé à effectuer une action dans SimSpace Weaver

S'il vous AWS Management Console indique que vous n'êtes pas autorisé à effectuer une action, vous devez contacter votre administrateur pour obtenir de l'aide. Votre administrateur est la personne qui vous a fourni votre nom d'utilisateur et votre mot de passe.

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

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

Dans ce cas, Mateo demande à son administrateur de mettre à jour ses politiques pour lui permettre d'accéder à la ressource *`my-example-widget`* à l'aide de l'action `simspaceweaver:`*`GetWidget`*.

Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez une erreur selon laquelle vous n'êtes pas autorisé à exécuter `iam:PassRole` l'action, vos stratégies doivent être mises à jour afin de vous permettre de transmettre un rôle à SimSpace Weaver.

Certains vous 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, un utilisateur doit 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 exécuter une action dans SimSpace Weaver. Toutefois, l'action nécessite que le service ait des autorisations accordées par une fonction de service. Mary ne dispose pas des autorisations nécessaires pour transférer 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 de connexion.

Je veux afficher mes clés d'accès

Une fois les clés d'accès utilisateur IAM créées, vous pouvez afficher votre ID de clé d'accès à tout moment. Toutefois, vous ne pouvez pas revoir votre clé d'accès secrète. Si vous perdez votre clé d'accès secrète, vous devez créer une nouvelle paire de clés.

Les clés d'accès se composent de deux parties : un ID de clé d'accès (par exemple, `AKIAIOSFODNN7EXAMPLE`) et une clé d'accès secrète (par exemple, `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`). À l'instar d'un nom d'utilisateur et un mot de passe, vous devez utiliser à la fois l'ID de clé d'accès et la clé d'accès secrète pour authentifier vos demandes. Gérez vos clés d'accès de manière aussi sécurisée que votre nom d'utilisateur et votre mot de passe.

> **Important**
>
> Ne communiquez pas vos clés d'accès à un tiers, même pour qu'il vous aide à trouver votre ID utilisateur canonique. Ce faisant, vous pourriez donner à quelqu'un un accès permanent à votre Compte AWS.

Lorsque vous créez une paire de clé d'accès, enregistrez l'ID de clé d'accès et la clé d'accès secrète dans un emplacement sécurisé. La clé d'accès secrète est accessible uniquement au moment de sa création. Si vous perdez votre clé d'accès secrète, vous devez ajouter de nouvelles clés d'accès

pour votre utilisateur IAM. Vous pouvez avoir un maximum de deux clés d'accès. Si vous en avez déjà deux, vous devez supprimer une paire de clés avant d'en créer une nouvelle. Pour afficher les instructions, consultez Gestion des clés d'accès dans le Guide de l'utilisateur IAM.

Je suis administrateur et je souhaite autoriser d'autres personnes à accéder SimSpace Weaver

Pour autoriser d'autres personnes à y accéder SimSpace Weaver, vous devez accorder l'autorisation aux personnes ou aux applications qui ont besoin d'y accéder. Si vous utilisez AWS IAM Identity Center pour gérer des personnes et des applications, vous attribuez des ensembles d'autorisations aux utilisateurs ou aux groupes afin de définir leur niveau d'accès. Les ensembles d'autorisations créent et attribuent automatiquement des politiques IAM aux rôles IAM associés à la personne ou à l'application. Pour plus d'informations, consultez la section Ensembles d'autorisations dans le guide de AWS IAM Identity Center l'utilisateur.

Si vous n'utilisez pas IAM Identity Center, vous devez créer des entités IAM (utilisateurs ou rôles) pour les personnes ou les applications qui ont besoin d'un accès. Vous devez ensuite associer une politique à l'entité qui leur accorde les autorisations appropriées dans SimSpace Weaver. Une fois les autorisations accordées, fournissez les informations d'identification à l'utilisateur ou au développeur de l'application. Ils utiliseront ces informations d'identification pour y accéder AWS. Pour en savoir plus sur la création d'utilisateurs, de groupes, de politiques et d'autorisations IAM, consultez la section Identités, politiques et autorisations IAM dans le guide de l'utilisateur IAM.

Je souhaite permettre à des personnes extérieures Compte AWS à moi d'accéder à mes SimSpace Weaver ressources

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 ces fonctionnalités sont prises SimSpace Weaver en charge, consultezComment AWS SimSpace Weaver 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.

Enregistrement et surveillance des événements de sécurité dans AWS SimSpace Weaver

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité SimSpace Weaver et des performances de vos AWS solutions. Vous devez collecter des données de surveillance provenant de toutes les parties de votre AWS solution afin de pouvoir corriger plus facilement une défaillance multipoint, le cas échéant.

AWS et SimSpace Weaver fournissez plusieurs outils pour surveiller vos ressources de simulation et répondre aux incidents potentiels.

Logs sur Amazon CloudWatch

SimSpace Weaver enregistre ses identifiants CloudWatch. Vous pouvez utiliser ces journaux pour surveiller les événements de votre simulation (tels que le démarrage et l'arrêt d'applications) ainsi que pour le débogage. Pour de plus amples informations, veuillez consulter SimSpace Weaver journaux dans Amazon CloudWatch Logs.

CloudWatch Alarmes Amazon

À l'aide des CloudWatch alarmes Amazon, vous observez une seule métrique sur une période que vous spécifiez. Si la métrique dépasse un seuil donné, une notification est envoyée à une rubrique Amazon SNS ou à une politique AWS Auto Scaling. CloudWatch les alarmes sont déclenchées lorsque leur état change et sont maintenues pendant un certain nombre de périodes, et non parce qu'elles sont dans un état particulier. Pour de plus amples informations, veuillez consulter Surveillance SimSpace Weaver avec Amazon CloudWatch.

AWS CloudTrail Journaux

CloudTrail fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans SimSpace Weaver. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été faite SimSpace Weaver, l'adresse IP à partir de laquelle la demande a été faite, qui a fait la demande, quand elle a été faite et des détails supplémentaires. Pour de plus amples informations, veuillez consulter Journalisation des appels AWS SimSpace Weaver d'API à l'aide AWS CloudTrail.

Validation de conformité pour AWS SimSpace Weaver

SimSpace Weaver ne fait l'objet d'aucun programme de AWS conformité.

Des auditeurs tiers évaluent la sécurité et la conformité des autres AWS services dans le cadre de multiples programmes de AWS conformité. Il s'agit notamment des certifications SOC, PCI, FedRAMP, HIPAA et d'autres.

Pour savoir si un programme Services AWS de conformité Service AWS s'inscrit dans le champ d'application de programmes de conformité spécifiques, consultez Services AWS la section de conformité et sélectionnez le programme de conformité qui vous intéresse. 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 .

Votre responsabilité en matière de conformité lors de l'utilisation Services AWS est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les lois et réglementations applicables. AWS fournit les ressources suivantes pour faciliter la mise en conformité :

- Conformité et gouvernance de la sécurité : ces guides de mise en œuvre de solutions traitent des considérations architecturales et fournissent les étapes à suivre afin de déployer des fonctionnalités de sécurité et de conformité.
- Référence des services éligibles HIPAA : liste les services éligibles HIPAA. Tous ne Services AWS sont pas éligibles à la loi HIPAA.
- 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.
- AWS Guides de conformité destinés aux clients — Comprenez le modèle de responsabilité partagée sous l'angle de la conformité. Les guides résument les meilleures pratiques en matière

de sécurisation Services AWS et décrivent les directives relatives aux contrôles de sécurité dans de nombreux cadres (notamment le National Institute of Standards and Technology (NIST), le Payment Card Industry Security Standards Council (PCI) et l'Organisation internationale de normalisation (ISO)).

- 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— Cela Service AWS fournit une vue complète de votre état de sécurité interne AWS. Security Hub utilise des contrôles de sécurité pour évaluer vos ressources AWS et vérifier votre conformité par rapport aux normes et aux bonnes pratiques du secteur de la sécurité. Pour obtenir la liste des services et des contrôles pris en charge, consultez Référence des contrôles Security Hub.
- Amazon GuardDuty — Cela Service AWS détecte les menaces potentielles qui pèsent sur vos charges de travail Comptes AWS, vos conteneurs et vos données en surveillant votre environnement pour détecter toute activité suspecte et malveillante. GuardDuty peut vous aider à répondre à diverses exigences de conformité, telles que la norme PCI DSS, en répondant aux exigences de détection des intrusions imposées par certains cadres de conformité.
- AWS Audit Manager— Cela vous Service AWS permet d'auditer en permanence votre AWS utilisation afin de simplifier la gestion des risques et la conformité aux réglementations et aux normes du secteur.

Résilience dans AWS SimSpace Weaver

L'infrastructure AWS mondiale est construite autour Régions AWS de zones de disponibilité. Régions AWS 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 les zones de disponibilité Régions AWS et les zones de disponibilité, consultez la section Infrastructure AWS globale.

Sécurité de l'infrastructure dans AWS SimSpace Weaver

En tant que service géré, AWS SimSpace Weaver 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 SimSpace Weaver 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.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser AWS Security Token Service (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Modèle de sécurité de connectivité réseau

Vos simulations s'exécutent sur des instances de calcul au sein d'un Amazon VPC situé dans une AWS région que vous sélectionnez. Un Amazon VPC est un réseau virtuel dans le AWS cloud qui isole l'infrastructure par charge de travail ou entité organisationnelle. Les communications entre les instances de calcul au sein d'Amazon VPC restent au sein du AWS réseau et ne transitent pas par Internet. Certaines communications internes du service transitent par Internet et sont cryptées. Les simulations pour tous les clients opérant dans la même AWS région partagent le même Amazon VPC. Les simulations pour différents clients utilisent des instances de calcul distinctes au sein du même Amazon VPC.

Communications entre vos clients de simulation et vos simulations exécutées dans le SimSpace Weaver cadre de voyages sur Internet. SimSpace Weaver ne gère pas ces connexions. Il est de votre responsabilité de sécuriser les connexions avec vos clients.

Vos connexions au SimSpace Weaver service transitent par Internet et sont cryptées. Cela inclut les connexions utilisant le AWS Management Console, AWS Command Line Interface (AWS CLI), les kits de développement AWS logiciel (SDK) et le SDK de l' SimSpace Weaver application.

Analyse de configuration et de vulnérabilité dans AWS SimSpace Weaver

La configuration et les contrôles informatiques sont une responsabilité partagée entre vous AWS et vous. Pour plus d'informations, voir le modèle de responsabilité AWS partagée Modèle . AWS gère les tâches de sécurité de base pour l'infrastructure sous-jacente, telles que l'application de correctifs au système d'exploitation sur les instances de calcul, la configuration du pare-feu et la reprise après sinistre de AWS l'infrastructure. Ces procédures ont été vérifiées et certifiées par les tiers appropriés. Pour plus de détails, consultez les meilleures pratiques en matière de sécurité, d'identité et de conformité.

Vous êtes responsable de la sécurité de votre logiciel de simulation :

- Conservez le code de votre application, y compris les mises à jour et les correctifs de sécurité.
- Authentifiez et chiffrez les communications entre vos clients de simulation et les applications auxquelles ils se connectent.
- Mettez à jour vos simulations pour utiliser les dernières versions du SDK, y compris le AWS SDK et le SDK de SimSpace Weaver l'application.

> **Note**
>
> SimSpace Weaver ne prend pas en charge les mises à jour des applications dans une simulation en cours d'exécution. Si vous devez mettre à jour vos applications, vous devez arrêter et supprimer la simulation, puis créer une nouvelle simulation avec le code d'application mis à jour. Nous vous recommandons d'enregistrer l'état de la simulation dans un magasin de données externe afin de pouvoir le restaurer si vous devez recréer la simulation.

Bonnes pratiques de sécurité pour SimSpace Weaver

Cette section décrit les meilleures pratiques de sécurité spécifiques à SimSpace Weaver. Pour en savoir plus sur les meilleures pratiques en matière de sécurité dans AWS, consultez la section Meilleures pratiques en matière de sécurité, d'identité et de conformité.

Rubriques

- Chiffrez les communications entre vos applications et leurs clients
- Sauvegardez régulièrement l'état de votre simulation
- Gérez vos applications et SDKs

Chiffrez les communications entre vos applications et leurs clients

SimSpace Weaver ne gère pas les communications entre vos applications et leurs clients. Vous devez implémenter une forme d'authentification et de chiffrement pour les sessions client.

Sauvegardez régulièrement l'état de votre simulation

SimSpace Weaver n'enregistre pas l'état de votre simulation. Les simulations qui sont arrêtées (suite à un appel d'API, à une option de console ou à un crash du système) n'enregistrent pas leur état et ne disposent d'aucun moyen intrinsèque de les restaurer. Les simulations arrêtées ne peuvent pas être redémarrées. La seule façon d'effectuer l'équivalent d'un redémarrage est de recréer votre simulation en utilisant la même configuration et les mêmes données. Vous pouvez utiliser des sauvegardes de l'état de votre simulation pour initialiser la nouvelle simulation. AWS propose des services de stockage et de base de données dans le cloud très fiables et disponibles que vous pouvez utiliser pour enregistrer l'état de votre simulation.

Gérez vos applications et SDKs

Gérez vos applications, les installations locales des kits de développement AWS logiciel (SDKs) et le SDK de SimSpace Weaver l'application. Vous pouvez télécharger et installer de nouvelles versions du AWS SDKs. Testez les nouvelles versions du SDK d' SimSpace Weaver applications avec des versions d'applications non destinées à la production pour vous assurer que vos applications continuent de fonctionner comme prévu. Vous ne pouvez pas mettre à jour vos applications dans une simulation en cours d'exécution. Pour mettre à jour vos applications :

1. Mettez à jour et testez le code de l'application localement (ou dans un environnement de test).

2. Arrêtez de modifier l'état de votre simulation et enregistrez-la (si nécessaire).
3. Arrêtez votre simulation (une fois arrêtée, elle ne peut pas être redémarrée).
4. Supprimez votre simulation (les simulations arrêtées qui ne sont pas supprimées sont prises en compte dans vos limites de service).
5. Recréez votre simulation avec la même configuration et le code d'application mis à jour.
6. Initialisez votre simulation à l'aide des données d'état enregistrées (si disponibles).
7. Démarrez votre nouvelle simulation.

> **Note**
>
> Une nouvelle simulation créée avec la même configuration est distincte de l'ancienne simulation. Il aura un nouvel identifiant de simulation et enverra les journaux vers un nouveau flux de journaux sur Amazon CloudWatch.

Connexion et surveillance SimSpace Weaver

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité SimSpace Weaver et des performances de vos autres AWS solutions. AWS fournit les outils de surveillance suivants pour surveiller SimSpace Weaver, signaler tout problème et prendre des mesures automatiques le cas échéant :

- Amazon CloudWatch surveille vos AWS ressources et les applications que vous utilisez AWS en temps réel. Vous pouvez collecter et suivre les métriques, créer des tableaux de bord personnalisés, et définir des alarmes qui vous informent ou prennent des mesures lorsqu'une métrique spécifique atteint un seuil que vous spécifiez. Pour plus d'informations, consultez le guide de CloudWatch l'utilisateur Amazon.
- Amazon CloudWatch Logs vous permet de surveiller, de stocker et d'accéder aux données de vos journaux provenant de vos SimSpace Weaver employés et d'autres sources. CloudTrail CloudWatch Les journaux peuvent surveiller les informations contenues dans les données du journal et vous avertir lorsque certains seuils sont atteints. Vous pouvez également archiver vos données de journaux dans une solution de stockage hautement durable. Pour plus d'informations, consultez le guide de l'utilisateur d'Amazon CloudWatch Logs.
- AWS CloudTrail capture les appels d'API et les événements associés créés par votre Compte AWS ou au nom de celui-ci et livre les fichiers journaux dans un compartiment Amazon S3 que vous spécifiez. Vous pouvez identifier les utilisateurs et les comptes qui ont appelé AWS, l'adresse IP source à partir de laquelle les appels ont été émis, ainsi que le moment où les appels ont eu lieu. Pour de plus amples informations, veuillez consulter le Guide de l'utilisateur AWS CloudTrail.

Rubriques

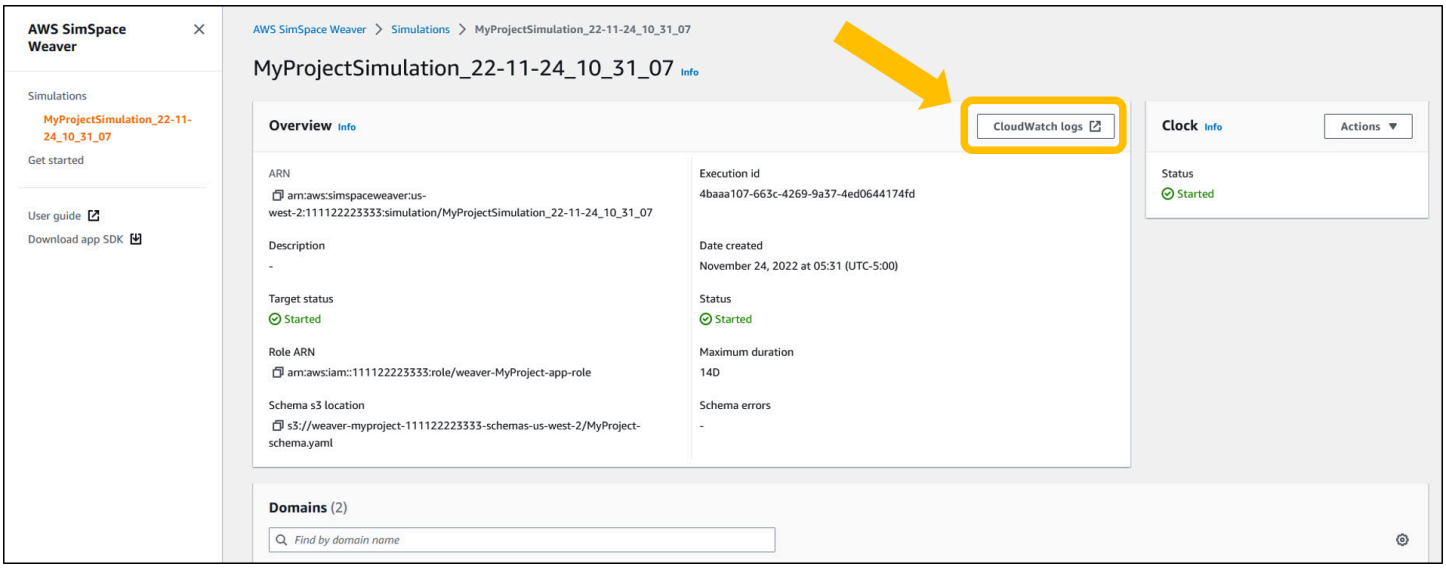
- SimSpace Weaver journaux dans Amazon CloudWatch Logs
- Surveillance SimSpace Weaver avec Amazon CloudWatch
- Journalisation des appels AWS SimSpace Weaver d'API à l'aide AWS CloudTrail

SimSpace Weaver journaux dans Amazon CloudWatch Logs

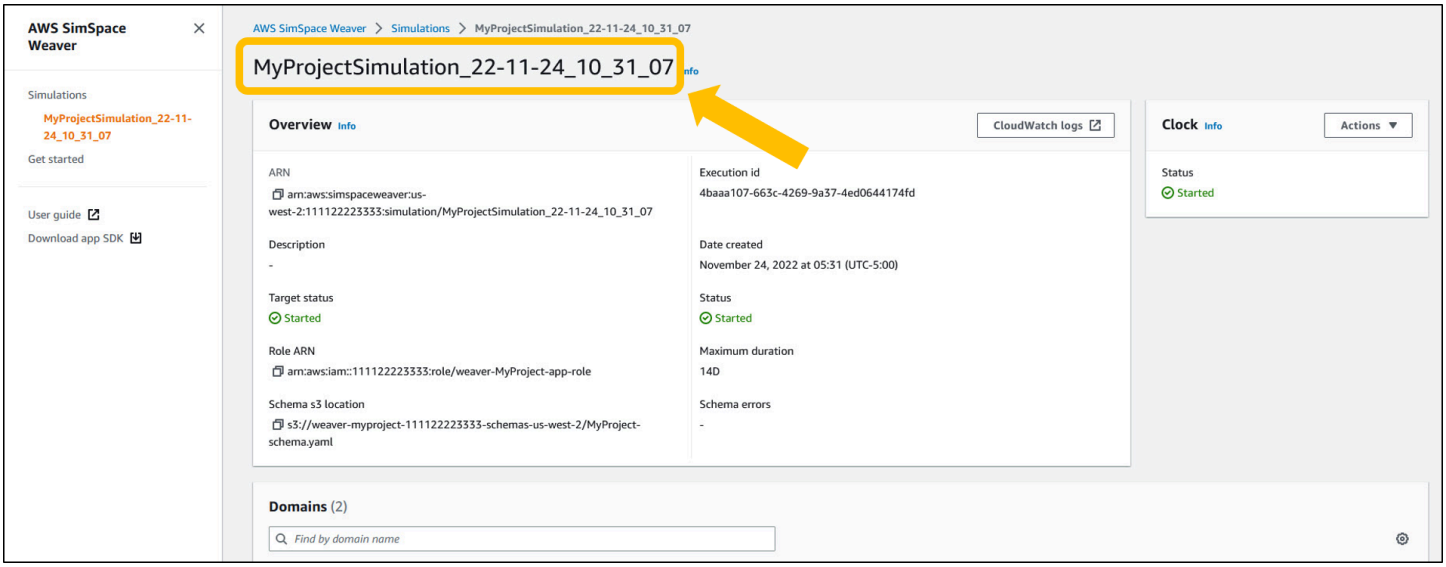
Accès à vos SimSpace Weaver journaux

Tous les journaux générés par vos SimSpace Weaver simulations sont stockés dans Amazon CloudWatch Logs. Pour accéder à vos journaux, vous pouvez utiliser le bouton CloudWatch journaux

dans le volet Vue d'ensemble de votre simulation dans la SimSpace Weaver console, qui vous permet d'accéder directement aux journaux de cette simulation spécifique.



Vous pouvez également accéder aux journaux via la CloudWatch console. Vous aurez besoin du nom de votre simulation pour pouvoir rechercher ses journaux.



SimSpace Weaver journaux

SimSpace Weaver écrit les messages de gestion des simulations et les résultats de console de vos applications sur Amazon CloudWatch Logs. Pour plus d'informations sur l'utilisation des journaux, consultez la section Utilisation des groupes de journaux et des flux de CloudWatch journaux dans le guide de l'utilisateur Amazon Logs.

Chaque simulation que vous créez possède son propre groupe de CloudWatch journaux dans Logs. Le nom du groupe de journaux est spécifié dans le schéma de simulation. Dans l'extrait de schéma suivant, la valeur de `log_destination_service` est. `logs` Cela signifie que la valeur de `log_destination_resource_name` est le nom d'un groupe de journaux. Dans ce cas, le groupe de journaux est`MySimulationLogs`.

```
simulation_properties:
  log_destination_service: "logs"
  log_destination_resource_name: "MySimulationLogs"
  default_entity_index_key_type: "Vector3<f32>"
```

Vous pouvez également utiliser l'DescribeSimulationAPI pour trouver le nom du groupe de journaux pour la simulation une fois que vous l'avez démarrée.

```
aws simspaceweaver describe-simulation --simulation simulation-name
```

L'exemple suivant montre la partie de la sortie DescribeSimulation qui décrit la configuration de journalisation. Le nom du groupe de journaux est affiché à la fin du`LogGroupArn`.

```
    "LoggingConfiguration": {
        "Destinations": [
            {
                "CloudWatchLogsLogGroup": {
                    "LogGroupArn": "arn:aws:logs:us-west-2:111122223333:log-
group:MySimulationLogs"
                }
            }
        ]
    },
```

Chaque groupe de journaux de simulation contient plusieurs flux de journaux :

- Flux du journal de gestion : messages de gestion de simulation produits par le SimSpace Weaver service.

```
/sim/management
```

- Flux du journal des erreurs : messages d'erreur produits par le SimSpace Weaver service. Ce flux de journal n'existe qu'en cas d'erreur. SimSpace Weaver stocke les erreurs écrites par vos applications dans leurs propres flux de journaux d'applications (voir les flux de journaux suivants).

```
/sim/errors
```

- Flux de journaux d'applications spatiales (1 pour chaque application spatiale sur chaque travailleur) : sortie de console produite par les applications spatiales. Chaque application spatiale écrit dans son propre flux de log. Tous les caractères *spatial-app-id* se trouvent après la barre oblique finale à la fin du. *worker-id*

```
/domain/spatial-domain-name/app/worker-worker-id/spatial-app-id
```

- Flux de journaux d'applications personnalisés (1 pour chaque instance d'application personnalisée) : sortie de console produite par des applications personnalisées. Chaque instance d'application personnalisée écrit dans son propre flux de journal.

```
/domain/custom-domain-name/app/custom-app-name/random-id
```

- Flux de journal des applications de service (1 pour chaque instance d'application de service) : sortie de console produite par les applications de service. Chaque application de service écrit dans son propre flux de journal. Tous les caractères *service-app-id* se trouvent après la barre oblique finale à la fin du. *service-app-name*

```
/domain/service-domain-name/app/service-app-name/service-app-id
```

Surveillance SimSpace Weaver avec Amazon CloudWatch

Vous pouvez effectuer un suivi SimSpace Weaver à l'aide d'Amazon CloudWatch, qui collecte les données brutes et les transforme en indicateurs lisibles en temps quasi réel. Ces statistiques sont enregistrées pour une durée de 15 mois ; par conséquent, vous pouvez accéder aux informations historiques et acquérir un meilleur point de vue de la façon dont votre service ou application web s'exécute. Vous pouvez également définir des alarmes qui surveillent certains seuils et envoient des notifications ou prennent des mesures lorsque ces seuils sont atteints. Pour plus d'informations, consultez le guide de CloudWatch l'utilisateur Amazon.

Le SimSpace Weaver service indique les métriques suivantes dans l'espace de `AWS/simspaceweaver` noms.

SimSpace Weaver statistiques au niveau du compte

L'espace de SimSpace Weaver noms inclut les métriques suivantes relatives à l'activité au niveau du AWS compte.

Métrique	Description
`SimulationCount`	Le nombre de simulations pour le compte courant. Unités : nombre Dimensions : aucune Statistiques : moyenne, minimale, maximale

Journalisation des appels AWS SimSpace Weaver d'API à l'aide AWS CloudTrail

AWS SimSpace Weaver est intégré à AWS CloudTrail un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans SimSpace Weaver. CloudTrail capture tous les appels d'API SimSpace Weaver sous forme d'événements. Les appels capturés incluent des appels provenant de la SimSpace Weaver console et des appels de code vers les opérations de l' SimSpace Weaver API. Si vous créez un suivi, vous pouvez activer la diffusion continue d' CloudTrail événements vers un compartiment Amazon S3, y compris les événements pour SimSpace Weaver. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Event history. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été faite SimSpace Weaver, l'adresse IP à partir de laquelle la demande a été faite, qui a fait la demande, quand elle a été faite et des détails supplémentaires.

Pour en savoir plus CloudTrail, consultez le guide de AWS CloudTrail l'utilisateur.

SimSpace Weaver informations dans CloudTrail

CloudTrail est activé sur votre compte Compte AWS lorsque vous créez le compte. Lorsqu'une activité se produit dans SimSpace Weaver, cette activité est enregistrée dans un CloudTrail événement avec d'autres événements AWS de service dans Event history. Vous pouvez consulter, rechercher et télécharger les événements récents dans votre Compte AWS. Pour plus d'informations, consultez la section Affichage des événements à l'aide de l'historique des CloudTrail événements.

Pour un enregistrement continu des événements de votre région Compte AWS, y compris des événements pour SimSpace Weaver, créez un parcours. Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Par défaut, lorsque vous créez un journal d'activité dans la console, il s'applique à toutes les régions Régions AWS. Le journal enregistre les événements de toutes les régions de la AWS partition et transmet les fichiers journaux au compartiment Amazon S3 que vous spécifiez. En outre, vous pouvez configurer d'autres AWS services pour analyser plus en détail les données d'événements collectées dans les CloudTrail journaux et agir en conséquence. Pour plus d'informations, consultez les ressources suivantes :

- Présentation de la création d'un journal de suivi
- CloudTrail services et intégrations pris en charge
- Configuration des notifications Amazon SNS pour CloudTrail
- Réception de fichiers CloudTrail journaux de plusieurs régions et réception de fichiers CloudTrail journaux de plusieurs comptes

Toutes les SimSpace Weaver actions sont enregistrées CloudTrail et documentées dans la référence de l'AWS SimSpace Weaver API. Par exemple, les appels au `ListSimulations` `DescribeSimulation` et les `DeleteSimulation` actions génèrent des entrées dans les fichiers CloudTrail journaux.

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 les éléments suivants :

- Si la demande a été faite avec les informations d'identification de l'utilisateur root ou AWS Identity and Access Management (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 demande a été faite par un autre AWS service.

Pour de plus amples informations, veuillez consulter l'élément userIdentity CloudTrail .

Comprendre les entrées du fichier SimSpace Weaver journal

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3 que vous spécifiez. CloudTrail les fichiers journaux contiennent une ou plusieurs entrées de journal. Un événement représente une demande unique provenant de n'importe quelle source et inclut des informations sur l'action demandée, telles que la date et l'heure de l'action, les paramètres de la demande et d'autres détails. CloudTrail les fichiers journaux ne constituent pas une trace ordonnée des appels d'API publics, ils n'apparaissent donc pas dans un ordre spécifique.

L'exemple suivant montre une entrée de CloudTrail journal illustrant l'`ListSimulations`action.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE:aws-console-signin-utils",
        "arn": "arn:aws:sts::111122223333:assumed-role/ConsoleSigninRole/aws-console-
signin-utils",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/ConsoleSigninRole",
                "accountId": "111122223333",
                "userName": "ConsoleSigninRole"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2022-02-14T15:57:02Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2022-02-14T15:57:08Z",
    "eventSource": "simspaceweaver.amazonaws.com",
    "eventName": "ListSimulations",
```

```
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.10",
    "userAgent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
 Gecko) Chrome/86.0.4240.0 Safari/537.36",
    "requestParameters": null,
    "responseElements": null,
    "requestID": "1234abcd-1234-5678-abcd-12345abcd123",
    "eventID": "5678abcd-5678-1234-ab12-123abc123abc",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

SimSpace Points de terminaison et quotas Weaver

Les tableaux suivants décrivent les points de terminaison de service et les quotas de service pour SimSpace Weaver. Les quotas de service, également appelés limites, sont le nombre maximal de ressources ou d'opérations de service pour vous Compte AWS. Pour plus d'informations, veuillez consulter la rubrique Quotas du service AWS dans les Références générales AWS .

Points de terminaison de service

Nom de la région	Région	Point de terminaison	Protocole
USA Est (Virginie du Nord)	us-east-1	simspaceweaver.us-east-1.amazonaws.com	HTTPS
USA Est (Ohio)	us-east-2	simspaceweaver.us-east-2.amazonaws.com	HTTPS
USA Ouest (Oregon)	us-west-2	simspaceweaver.us-west-2.amazonaws.com	HTTPS
Asie-Pacifique (Singapour)	ap-southeast-1	simspaceweaver.ap-southeast-1.amazonaws.com	HTTPS
Asie-Pacifique (Sydney)	ap-southeast-2	simspaceweaver.ap-southeast-2.amazonaws.com	HTTPS
Europe (Stockholm)	eu-north-1	simspaceweaver.eu-north-1.amazonaws.com	HTTPS

Nom de la région	Région	Point de terminaison	Protocole
Europe (Francfort)	eu-central-1	simspaceweaver.eu-central-1.amazonaws.com	HTTPS
Europe (Irlande)	eu-west-1	simspaceweaver.eu-west-1.amazonaws.com	HTTPS
AWS GovCloud (USA Est)	us-gov-east-1	simspaceweaver.us-gov-east-1.amazonaws.com	HTTPS
AWS GovCloud (US-Ouest)	us-gov-west-1	simspaceweaver.us-gov-west-1.amazonaws.com	HTTPS

Quotas de service

Nom	Par défaut	Ajusta	Description
Calculer les unités de ressources pour chaque application	Chaque Région prise en charge : 4	Non	Le nombre maximal d'unités de ressources de calcul que vous pouvez allouer à chaque application.
Calculer les unités de ressources pour chaque travailleur	Chaque région prise en charge : 17	Non	Le nombre d'unités de ressources de calcul disponibles pour chaque travailleur.
Champs de données pour chaque entité	Chaque région prise en charge : 7	Non	Le nombre maximum de champs de données (autres que ceux d'index) qu'une entité peut avoir.

Nom	Par défaut	Ajusta	Description
Entités d'une partition	Chaque région prise en charge : 8 192	Non	Le nombre maximum d'entités dans une partition.
Taille du champ de données de l'entité	Chaque région prise en charge : 1 024 octets	Non	Taille maximale d'un champ de données (autre qu'un index) d'une entité.
Transferts d'entités entre travailleurs	Chaque région prise en charge : 25	Non	Le nombre maximum de transferts d'entités entre les travailleurs, pour chaque partition et chaque tick.
Transferts d'entités concernant le même travailleur	Chaque région prise en charge : 500	Non	Le nombre maximum de transferts d'entités sur le même worker, pour chaque partition et chaque tick.
Champs d'index pour chaque entité	Par région prise en charge : 1	Non	Le nombre maximum de champs d'index qu'une entité peut avoir.
Durée maximale maximale (en jours) pour une simulation	Chaque région prise en charge : 14	Non	Le plus grand nombre de jours que vous pouvez spécifier comme durée maximale d'une simulation. Toutes les simulations ont une durée maximale, même si vous n'en spécifiez pas la valeur. Une simulation s'arrête automatiquement lorsqu'elle atteint sa durée maximale.

Nom	Par défaut	Ajusta	Description
Mémoire pour chaque unité de ressource de calcul	Chaque région prise en charge : 1 gigaoctets	Non	La quantité de mémoire vive (RAM) qu'une application obtient pour chaque unité de ressource de calcul.
Abonnements à distance pour chaque travailleur	Chaque région prise en charge : 24	Non	Le nombre maximum d'abonnements à distance pour chaque collaborateur.
Nombre de simulations	Chaque région prise en charge : 2	Oui	Le nombre maximum de simulations dont le statut cible est STARTED sur votre compte. Vous pouvez demander une augmentation de quota jusqu'à 10.
Travailleurs pour une simulation	Chaque région prise en charge : 2	Oui	Le nombre maximum de travailleurs que vous pouvez affecter à une simulation. Vous pouvez demander une augmentation de quota jusqu'à 10.
v CPUs pour chaque unité de ressource de calcul	Chaque région prise en charge : 2	Non	Nombre d'unités centrales virtuelles (vCPUs) qu'une application obtient pour chaque unité de ressource informatique.

Quotas de messagerie

Les quotas suivants s'appliquent à la messagerie d'application à application pour SimSpace Weaver Local et dans le AWS Cloud.

Nom	Par défaut	Ajustable	Description
Taille maximale du message	Chaque région prise en charge : 256 octets	Non	Taille maximale de la charge utile d'un message.
Débit d'envoi de messages maximal	Chaque région prise en charge : 128	Non	Le nombre maximum de messages que chaque application peut envoyer par tick.

Tarifs des horloges

Le schéma de simulation indique la fréquence d'horloge (également appelée fréquence de tic-tac) d'une simulation. Le tableau suivant indique les fréquences d'horloge valides que vous pouvez utiliser.

Nom	Valeurs valides	Description
Fréquence de l'horloge	Chaque région prise en charge : « 10 », « 15 », « 30 », « illimité »	Les fréquences d'horloge valides pour une simulation.
Fréquence d'horloge (versions 1.13 et 1.12)	Chaque région prise en charge : 10, 15, 30	Les fréquences d'horloge valides pour une simulation.

Quotas de service pour SimSpace Weaver Local

Les quotas de service suivants s'appliquent à SimSpace Weaver Local uniquement. Tous les autres quotas s'appliquent également à SimSpace Weaver Local.

Nom	Par défaut	Ajustable	Description
Nombre maximal de partitions	SimSpace Weaver Local : 24	Non	Nombre maximal de partitions pour une simulation.
Nombre maximum d'applications	SimSpace Weaver Local : 24	Non	Le nombre total maximum d'applications (de tous types) pour une simulation.
Nombre maximum de domaines	SimSpace Weaver Local : 24	Non	Le nombre total maximum de domaines (tous types confondus) pour une simulation.
Entités par partition	SimSpace Weaver Local: 4 096	Non	Le nombre maximum d'entités dans chaque partition.
Champs par entité	SimSpace Weaver Local: 8	Non	Le nombre maximum de champs pour chaque entité.
Taille du champ	SimSpace Weaver Local: 1024 octets	Non	Taille maximale d'un champ d'entité.

Résolution des problèmes dans SimSpace Weaver

Rubriques

- AssumeRoleAccessDenied
- InvalidBucketName
- ServiceQuotaExceededException
- TooManyBuckets
- Autorisation refusée lors du démarrage de la simulation
- Problèmes liés au temps d'utilisation Docker
- PathfindingSample le client de console ne parvient pas à se connecter
- Ils AWS CLI ne reconnaissent pas simspaceweaver

AssumeRoleAccessDenied

Le message d'erreur suivant peut s'afficher si votre simulation ne démarre pas :

```
Unable to assume role arn:aws:iam::111122223333:role/weaver-project-name-app-role;
 verify the role exists and has trust policy on SimSpace Weaver
```

Vous pouvez recevoir cette erreur si l'une des conditions suivantes est vraie concernant le rôle AWS Identity and Access Management (IAM) de votre simulation :

- Le nom de la ressource Amazon (ARN) fait référence à un rôle IAM qui n'existe pas.
- La politique de confiance pour le rôle IAM qui n'autorise pas le nom de la nouvelle simulation à assumer le rôle.

Vérifiez que le rôle existe. Si le rôle existe, vérifiez votre politique de confiance pour le rôle. `aws:SourceArn`Dans l'exemple suivant, la politique de confiance autorise uniquement une simulation (dans le compte 111122223333) dont le nom commence par « `MySimulation` assumer le rôle ».

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

```
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "simspaceweaver.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:simspaceweaver:us-
west-2:111122223333:simulation/MySimulation*"
                }
            }
        }
    ]
}
```

Pour permettre à une autre simulation dont le nom commence par `MyOtherSimulation` d'assumer le rôle, la politique de confiance doit être modifiée comme dans l'exemple édité suivant :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "simspaceweaver.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": [
                        aws:SourceArn": "arn:aws:simspaceweaver:us-
west-2:111122223333:simulation/MySimulation*",
                        aws:SourceArn": "arn:aws:simspaceweaver:us-
west-2:111122223333:simulation/MyOtherSimulation*"
                        ]
                }
            }
        }
    ]
}
```

InvalidBucketName

Le message d'erreur suivant peut s'afficher lors de la création d'un projet :

```
An error occurred (InvalidBucketName) when calling the CreateBucket operation: The
 specified bucket is not valid.
```

Vous avez reçu cette erreur car le nom SimSpace Weaver transmis à Amazon Simple Storage Service (Amazon S3) ne respectait pas les règles de dénomination des compartiments (pour plus d'informations, consultez les règles de dénomination des compartiments dans le guide de l'utilisateur d'Amazon Simple Storage Service).

Le `create-project` script du SDK de l' SimSpace Weaver application crée des noms de bucket en utilisant le nom de projet que vous avez fourni au script. Les noms des compartiments utilisent les formats suivants :

- Version 1.13.x ou ultérieure
  - `weaver-`*`lowercase-project-name`*`-`*`account-number`*`-`*`region`*
- La version 1.12.x
  - `weaver-`*`lowercase-project-name`*`-`*`account-number`*`-app-zips-`*`region`*
  - `weaver-`*`lowercase-project-name`*`-`*`account-number`*`-schemas-`*`region`*

Par exemple, compte tenu des propriétés de projet suivantes :

- Nom du projet : `MyProject`
- Compte AWS numéro : `111122223333`
- Région AWS: `us-west-2`

Le projet comporterait les compartiments suivants :

- Version 1.13.x ou ultérieure
  - `weaver-myproject-111122223333-us-west-2`
- La version 1.12.x
  - `weaver-myproject-111122223333-app-zips-us-west-2`
  - `weaver-myproject-111122223333-schemas-us-west-2`

Le nom de votre projet ne doit pas enfreindre les règles de dénomination d'Amazon S3. Vous devez également utiliser un nom de projet suffisamment court pour que les noms de compartiment créés par le `create-project` script ne dépassent pas la limite de longueur des noms pour les compartiments Amazon S3.

ServiceQuotaExceededException

Le message d'erreur suivant peut s'afficher lorsque vous démarrez une simulation :

```
An error occurred (ServiceQuotaExceededException) when calling the StartSimulation
 operation: Failed to start simulation due to: simulation quota has already been
 reached.
```

Vous recevrez cette erreur si vous essayez de démarrer une nouvelle simulation alors que votre compte dispose actuellement du nombre maximum de simulations avec un statut cible de STARTED. Cela inclut l'exécution de simulations, les simulations échouées et les simulations qui se sont arrêtées parce qu'elles ont atteint leur durée maximale. Vous pouvez supprimer une simulation arrêtée ou ayant échoué pour démarrer une nouvelle simulation. Si toutes vos simulations sont en cours d'exécution, vous pouvez arrêter et supprimer une simulation en cours d'exécution. Vous pouvez également demander une augmentation de vos quotas de service si vous n'avez pas encore atteint la limite de demandes.

TooManyBuckets

Le message d'erreur suivant peut s'afficher lors de la création d'un projet :

```
An error occurred (TooManyBuckets) when calling the CreateBucket operation: You have
 attempted to create more buckets than allowed.
```

Amazon Simple Storage Service (Amazon S3) limite le nombre de compartiments que vous pouvez avoir dans AWS votre compte (pour plus d'informations, consultez la section Restrictions et limitations relatives aux compartiments dans le guide de l'utilisateur d'Amazon Simple Storage Service).

Vous devez effectuer l'une des opérations suivantes avant de pouvoir continuer :

- Supprimez au moins 2 compartiments Amazon S3 existants dont vous n'avez pas besoin.

- Demandez une augmentation de la limite Amazon S3 (pour plus d'informations, consultez la section Restrictions et limitations relatives aux compartiments dans le guide de l'utilisateur d'Amazon Simple Storage Service).
- Utilisez un autre AWS compte.

> **Note**
>
> L'`DeleteSimulation`API intégrée SimSpace Weaver ne supprime pas les ressources Amazon S3 associées à votre simulation. Nous vous recommandons de supprimer toutes les ressources associées à vos simulations lorsque vous n'en avez plus besoin.

Autorisation refusée lors du démarrage de la simulation

Lorsque vous lancez une simulation, un message d'erreur peut s'afficher indiquant que l'autorisation a été refusée ou qu'une erreur s'est produite lors de l'accès aux artefacts de votre application. Ce problème peut se produire lorsque vous spécifiez des compartiments Amazon S3 pour votre simulation qui SimSpace Weaver n'ont pas été créés pour vous (via la console ou les scripts du SDK de l' SimSpace Weaver application).

Les causes profondes les plus probables sont les suivantes :

- Le service n'est pas autorisé à accéder à un ou plusieurs des compartiments Amazon S3 que vous avez spécifiés dans votre schéma de simulation. Vérifiez la politique d'autorisation de votre rôle d'application, les politiques relatives aux compartiments Amazon S3 et les autorisations des compartiments Amazon S3 pour vous assurer que celui-ci `simspaceweaver.amazonaws.com` dispose des autorisations appropriées pour accéder à vos compartiments. Pour plus d'informations sur la politique d'autorisation des rôles d'application, consultezDes autorisations SimSpace Weaver créées pour vous.
- Votre compartiment Amazon S3 peut se trouver dans un emplacement différent Région AWS de celui de votre simulation. Vos compartiments Amazon S3 pour vos artefacts de simulation doivent être identiques à ceux Région AWS de votre simulation. Vérifiez votre console Amazon S3 pour voir dans quoi se trouve Région AWS votre compartiment. Si votre compartiment Amazon S3 se trouve dans un autre compartiment Région AWS, sélectionnez un compartiment Région AWS identique à celui de votre simulation.

Problèmes liés au temps d'utilisation Docker

Si vous utilisez Docker et vous recevez des erreurs liées au temps lors de l'exécution de scripts à partir du SDK de SimSpace Weaver l'application. Cela peut être dû au fait que votre Docker l'horloge de la machine virtuelle est incorrecte. Cela peut se produire si votre ordinateur était en cours d'exécution Docker puis sort du sommeil ou de l'hibernation.

Solutions à essayer

- Redémarrer Docker.
- Désactiver puis réactiver la synchronisation de l'heure dans Windows PowerShell:

```
Get-VMIntegrationService -VMName DockerDesktopVM -Name "Time Synchronization" |
 Disable-VMIntegrationService
Get-VMIntegrationService -VMName DockerDesktopVM -Name "Time Synchronization" |
 Enable-VMIntegrationService
```

PathfindingSample le client de console ne parvient pas à se connecter

Le client de console peut vous envoyer l'erreur suivante lorsque vous vous connectez à la `PathfindingSample` simulation décrite dans les didacticiels deCommencer avec SimSpace Weaver. Cette erreur se produit car le client ne peut pas ouvrir de connexion réseau à l'adresse IP et `ViewApp` au numéro de port combinés que vous avez fournis.

```
Fatal error in function nng_dial. Error code: 268435577. Error message: no link
```

Pour une simulation dans AWS Cloud

- Votre connexion réseau fonctionne-t-elle correctement ? Vérifiez que vous pouvez vous connecter à d'autres adresses IP ou sites Web qui devraient fonctionner. Assurez-vous que votre navigateur Web ne charge pas un site Web à partir de son cache.
- Votre simulation est-elle en cours d'exécution ? Vous pouvez utiliser l'ListSimulationsAPI pour connaître le statut de votre simulation. Pour de plus amples informations, veuillez consulter Obtenez l'adresse IP et le numéro de port d'une application personnalisée. Vous pouvez également utiliser la SimSpace Weaver console pour vérifier l'état de vos simulations.

- Vos applications sont-elles en cours d'exécution ? Vous pouvez utiliser l'DescribeAppAPI pour connaître le statut de vos applications. Pour de plus amples informations, veuillez consulter Obtenez l'adresse IP et le numéro de port d'une application personnalisée. Vous pouvez également utiliser la SimSpace Weaver console pour vérifier l'état de vos simulations.
- Vos applications sont-elles en cours d'exécution ? Vous pouvez utiliser l'DescribeAppAPI pour connaître le statut de vos applications. Pour de plus amples informations, veuillez consulter Obtenez l'adresse IP et le numéro de port d'une application personnalisée. Vous pouvez également utiliser la SimSpace Weaver console pour vérifier l'état de vos simulations.
- Avez-vous utilisé la bonne adresse IP et le bon numéro de port ? Lorsque vous vous connectez via Internet, vous devez utiliser l'adresse IP et le numéro de `Actual` port du`ViewApp`. Vous pouvez trouver l'adresse IP `Address` et le numéro de `Actual` port dans le `EndpointInfo` bloc de sortie de l'DescribeAppAPI. Vous pouvez également utiliser la SimSpace Weaver console pour trouver l'adresse IP (URI) et le numéro de port (port d'entrée) de votre **`ViewApp`** adresse IP (URI) sur la page `MyViewDomain` détaillée.
- Votre connexion réseau traverse-t-elle un pare-feu ? Votre pare-feu peut bloquer votre connexion à l'adresse IP ou au numéro de port (ou aux deux). Vérifiez les paramètres de votre pare-feu ou adressez-vous à votre administrateur de pare-feu.

Pour une simulation locale

- Pouvez-vous vous connecter à votre adresse de boucle (127.0.0.1) ? Si vous disposez de l'outil de ligne de `ping` commande sous Windows, vous pouvez ouvrir une fenêtre d'invite de commande et essayer d'envoyer un ping à 127.0.0.1. Appuyez sur Ctrl - C pour terminer le ping.

```
ping 127.0.0.1
```

Example sortie ping

```
C:\>ping 127.0.0.1

Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time=1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
```

```
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
Control-C
^C
C:\>
```

Si le ping indique qu'il a perdu des paquets, il se peut que d'autres logiciels (tels qu'un pare-feu local, des paramètres de sécurité ou des programmes anti-malware) bloquent votre connexion.

- Vos applications sont-elles en cours d'exécution ? Votre simulation locale s'exécute sous forme de fenêtres distinctes pour chaque application. Assurez-vous que les fenêtres de vos applications spatiales `ViewApp` sont ouvertes. Pour de plus amples informations, veuillez consulter Développement local en SimSpace Weaver.
- Avez-vous utilisé la bonne adresse IP et le bon numéro de port ? Vous devez l'utiliser `tcp://127.0.0.1:7000` lorsque vous vous connectez à une simulation locale. Pour de plus amples informations, veuillez consulter Développement local en SimSpace Weaver.
- Disposez-vous d'un logiciel de sécurité local susceptible de bloquer votre connexion ? Vérifiez vos paramètres de sécurité, votre pare-feu local ou vos programmes anti-malware pour voir s'ils bloquent votre connexion à `127.0.0.1` un port TCP. `7000`

Ils AWS CLI ne reconnaissent pas **`simspaceweaver`**

Si vous AWS CLI recevez des erreurs qui suggèrent qu'il n'est pas au courant SimSpace Weaver, exécutez la commande suivante.

```
aws simspaceweaver help
```

Si vous recevez une erreur commençant par les lignes suivantes et répertoriant tous les choix disponibles, il AWS CLI s'agit peut-être d'une ancienne version.

```
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:

  aws help
  aws <command> help
  aws <command> <subcommand> help

aws: error: argument command: Invalid choice, valid choices are:
```

Exécutez la commande suivante pour vérifier la version de votre AWS CLI.

```
aws --version
```

Si le numéro de version est antérieur à 2.9.19, vous devez mettre à jour votre. AWS CLI Notez que la version actuelle du AWS CLI est ultérieure à la version 2.9.19.

Pour mettre à jour votre AWS CLI, voir Installer ou mettre à jour la dernière version du AWS CLI dans le guide de AWS Command Line Interface l'utilisateur de la version 2.

SimSpace Weaver référence du schéma de simulation

SimSpace Weaver utilise un fichier YAML pour configurer les propriétés d'une simulation. Ce fichier est appelé schéma de simulation (ou simplement schéma). L'exemple de simulation inclus dans le SDK de SimSpace Weaver l'application inclut un schéma que vous pouvez copier et modifier pour votre propre simulation.

Rubriques

- Exemple de schéma complet
- Format du schéma

Exemple de schéma complet

L'exemple suivant montre YAML-fichier texte formaté qui décrit une SimSpace Weaver simulation. Cet exemple inclut des valeurs fictives pour les propriétés. Le format du fichier varie en fonction de la valeur `sdk_version` spécifiée dans celui-ci. Consultez Format du schéma pour une description complète des propriétés et de leurs valeurs valides.

```
sdk_version: "1.17"
simulation_properties:
  log_destination_resource_name: "MySimulationLogs"
  log_destination_service: "logs"
  default_entity_index_key_type: "Vector3<f32>"
  default_image: "111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-
repository:latest"
workers:
  MyComputeWorkers:
    type: "sim.c5.24xlarge"
    desired: 3
clock:
  tick_rate: "30"
partitioning_strategies:
  MyGridPartitioning:
    topology: "Grid"
    aabb_bounds:
      x: [-1000, 1000]
      y: [-1000, 1000]
    grid_placement_groups:
      x: 3
```

```
      y: 3
domains:
  MyCustomDomain:
    launch_apps_via_start_app_call: {}
    app_config:
      package: "s3://weaver-myproject-111122223333-us-west-2/MyViewApp.zip"
      launch_command: ["MyViewApp"]
      required_resource_units:
        compute: 1
      endpoint_config:
        ingress_ports: [9000, 9001]
  MyServiceDomain:
    launch_apps_per_worker:
      count: 1
    app_config:
      package: "s3://weaver-myproject-111122223333-us-west-2/
MyConnectionServiceApp.zip"
      launch_command: ["MyConnectionServiceApp"]
      required_resource_units:
        compute: 1
      endpoint_config:
        ingress_ports:
          - 9000
          - 9001
  MySpatialDomain:
    launch_apps_by_partitioning_strategy:
      partitioning_strategy: "MyGridPartitioning"
      grid_partition:
        x: 6
        y: 6
    app_config:
      package: "s3://weaver-myproject-111122223333-us-west-2/MySpatialApp.zip"
      launch_command: ["MySpatialApp"]
      required_resource_units:
        compute: 1
  MySpatialDomainWithCustomContainer:
    launch_apps_by_partitioning_strategy:
      partitioning_strategy: "MyGridPartitioning"
      grid_partition:
        x: 6
        y: 6
    app_config:
      package: "s3://weaver-myproject-111122223333-us-west-2/MySpatialApp2.zip"
      launch_command: ["MySpatialApp2"]
```

```
      required_resource_units:
        compute: 1
      image: "111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest"
placement_constraints:
  - placed_together: ["MySpatialDomain", "MySpatialDomainWithCustomContainer"]
    on_workers: ["MyComputeWorkers"]
```

Format du schéma

L'exemple suivant montre la structure globale d'un schéma. L'ordre des propriétés à chaque niveau du schéma n'a pas d'importance, tant que les relations parent-enfant sont les mêmes. L'ordre est important pour les éléments d'un tableau.

```
sdk_version: "sdk-version-number"
simulation_properties:
  simulation-properties
workers:
  worker-group-configurations
clock:
  tick_rate: tick-rate
partitioning_strategies:
  partitioning-strategy-configurations
domains:
  domain-configurations
placement_constraints:
  placement-constraints-configuration
```

Sections

- Version de SDK
- Propriétés de simulation
- Workers
- Horloge
- Stratégies de partitionnement
- Domaines
- Contraintes de placement

Version de SDK

La `sdk_version` section (obligatoire) identifie la version du SDK de l' SimSpace Weaver application qui prend en charge ce schéma. Valeurs valides: `1.17`, `1.16`, `1.15`, `1.14`, `1.13`, `1.12`

> **Important**
>
> La valeur de inclut `sdk_version` uniquement le numéro de version majeure et le numéro de première version secondaire. Par exemple, la valeur `1.12` spécifie toutes les versions`1.12.x`, telles que `1.12.0`1.12.1, et`1.12.2`.

```
sdk_version: "1.17"
```

Propriétés de simulation

La `simulation_properties` section (obligatoire) spécifie les différentes propriétés de votre simulation. Utilisez cette section pour configurer la journalisation et spécifier une image de conteneur par défaut. Cette section est obligatoire même si vous ne configurez pas la journalisation ou si vous choisissez de spécifier une image de conteneur par défaut.

```
simulation_properties:
  log_destination_resource_name: "log-destination-resource-name"
  log_destination_service: "log-destination-service"
  default_entity_index_key_type: "Vector3<f32>"
  default_image: "ecr-repository-uri"
```

Propriétés

`log_destination_resource_name`

Spécifie la ressource dans laquelle les journaux SimSpace Weaver seront écrits.

Nécessaire : Non Si cette propriété n'est pas incluse, SimSpace Weaver aucun journal n'est écrit pour la simulation.

Type : chaîne

Valeurs valides:

- Le nom d'un groupe de CloudWatch journaux Logs (par exemple,`MySimulationLogs`)
- Le nom de ressource Amazon (ARN) d'un groupe de CloudWatch journaux Logs (par exemple,`arn:aws:logs:us-west-2:111122223333:log-group/MySimulationLogs`)

> **Note**
>
> SimSpace Weaver ne prend en charge qu'une destination de journal dans le même compte et Région AWS que la simulation.

`log_destination_service`

Indique le type de ressource de destination de journalisation lorsque vous spécifiez une ressource `logging_destination resource_name` qui n'est pas un ARN.

Obligatoire : vous devez spécifier cette propriété si elle `log_destination_resource_name` est spécifiée et qu'il ne s'agit pas d'un ARN. Vous ne pouvez pas spécifier cette propriété si elle `log_destination_resource_name` n'est pas spécifiée ou s'il s'agit d'un ARN.

Type : chaîne

Valeurs valides:

- `logs`: La ressource de destination du journal est un groupe de journaux.

`default_entity_index_key_type`

Spécifie le type de données pour le champ clé d'index des entités de simulation.

Obligatoire : oui

Type : chaîne

Valeurs valides : `Vector3<f32>`

`default_image`

Spécifie l'image du conteneur par défaut pour votre simulation (non prise en charge pour les versions `1.13` et`1.12`). Si cette propriété est spécifiée, les domaines qui n'en spécifient aucun `image` utilisent le`default_image`.

Obligatoire : non

Type : chaîne

Valeurs valides :

- L'URI d'un référentiel dans Amazon Elastic Container Registry (Amazon ECR) (par exemple,) `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`

Workers

La `workers` section (obligatoire) spécifie les configurations pour les groupes de travail (groupes de travailleurs). SimSpace Weaver utilise ces informations conjointement `placement_constraints` pour configurer l'infrastructure sous-jacente de votre simulation. Un seul groupe de travail est actuellement pris en charge.

Pour spécifier les propriétés d'un groupe de travail, remplacez-le *`worker-group-name`* par le nom de votre choix. Le nom doit comporter de 3 à 64 caractères et peut contenir les caractères A à Z, a à z, 0 à 9 et _ - (tiret). Spécifiez les propriétés du groupe de travail après le nom.

```
workers:
  worker-group-name:
    type: "sim.c5.24xlarge"
    desired: number-of-workers
```

Propriétés

`type`

Spécifie le type de travailleur.

Obligatoire : oui

Type : chaîne

Valeurs valides : `sim.c5.24xlarge`

`desired`

Spécifie le nombre de travailleurs souhaité pour ce groupe de travail.

Obligatoire : oui

Type : entier

Valeurs valides : 1 -3. Votre quota de service (limite) pour le nombre de travailleurs pour vos simulations détermine la valeur maximale de cette propriété. Par exemple, si votre quota de service est égal 2 à, la valeur maximale de cette propriété est2. Vous pouvez demander une augmentation de votre quota de service. Pour de plus amples informations, veuillez consulter SimSpace Points de terminaison et quotas Weaver.

Horloge

La `clock` section (obligatoire) spécifie les propriétés de l'horloge de simulation.

```
clock:
  tick_rate: tick-rate
```

Propriétés

`tick_rate`

Spécifie le nombre de ticks publiés par seconde dans les applications par l'horloge.

Obligatoire : oui

Type :

- Version *`1.14`* et *`1.15`* : String
- Version *`1.13`* et *`1.12`* : entier

Valeurs valides:

- Version *`1.14`* et *`1.15`* : `"10"` | `"15"` | `"30"` | `"unlimited"`
  - `"unlimited"`: l'horloge envoie le prochain tick dès que toutes les applications ont terminé leurs opérations de validation pour le tick en cours.
- Version *`1.13`* et *`1.12`* : `10` | `15` | `30`

Stratégies de partitionnement

La `partitioning_strategies` section (obligatoire) indique l'organisation des partitions pour un domaine spatial.

> **Note**
>
> SimSpace Weaver ne prend en charge qu'une seule stratégie de partitionnement.

Pour spécifier les propriétés d'une stratégie de partitionnement, remplacez-la *`partitioning-strategy-name`* par le nom de votre choix. Le nom doit comporter de 3 à 64 caractères et peut contenir les caractères A à Z, a à z, 0 à 9 et _ - (tiret). Spécifiez les propriétés de la stratégie de partitionnement après le nom.

```
partitioning_strategies:
  partitioning-strategy-name:
    topology: "Grid"
    aabb_bounds:
      x: [aabb-min-x, aabb-max-x]
      y: [aabb-min-y, aabb-max-y]
    grid_placement_groups:
      x: number-of-placement-groups-along-x-axis
      y: number-of-placement-groups-along-y-axis
```

Propriétés

`topology`

Spécifie la topologie (schéma d'arrangement des partitions) pour cette stratégie de partitionnement.

Obligatoire : oui

Type : chaîne

Valeurs valides : `"Grid"`

`aabb_bounds`

Spécifie les limites du cadre de délimitation aligné sur l'axe principal (AABB) pour votre simulation. Vous spécifiez les limites sous forme de tableaux ordonnés à 2 éléments qui décrivent les valeurs minimale et maximale (dans cet ordre) pour chaque axe (x and y).

Obligatoire : selon les conditions. Cette propriété est obligatoire (et ne peut être spécifiée) que si la topologie est définie sur. `"Grid"`

Type : `Float` tableau (pour chaque axe)

Valeurs valides : `-3.4028235e38` - `3.4028235e38`

`grid_placement_groups`

Spécifie le nombre de groupes de placement le long de chaque axe (x et y) dans une topologie de grille. Un groupe de placement est un ensemble de partitions (dans le même domaine) adjacentes dans l'espace.

Obligatoire : selon les conditions. Cette propriété est obligatoire (et ne peut être spécifiée) que si la topologie est définie sur. `"Grid"` Si vous ne spécifiez pas de configuration de groupes de placement, SimSpace Weaver nous en calculerons une pour vous. Tout domaine utilisant une stratégie de partitionnement sans configuration de groupes de placement doit spécifier un `grid_partition` (voirStratégie de partitionnement du domaine spatial).

Type : entier (pour chaque axe)

Valeurs valides : `1` -`20`. Nous recommandons que x * y soit égal au nombre de travailleurs souhaité. Dans le cas contraire, SimSpace Weaver vous essaierez d'équilibrer vos groupes de placement entre les travailleurs disponibles.

Domaines

La `domains` section (obligatoire) indique les propriétés de chacun de vos domaines. Toutes les simulations doivent comporter au moins une section pour un domaine spatial. Vous pouvez créer plusieurs sections pour des domaines supplémentaires. Chaque type de domaine possède son propre format de configuration.

> **Important**
>
> Les versions `1.13` 1 et `1.12` 2 ne prennent pas en charge plusieurs domaines spatiaux.

> **Important**
>
> SimSpace Weaver prend en charge jusqu'à 5 domaines pour chaque simulation. Cela inclut tous les domaines spatiaux, personnalisés et de service.

```
domains:
```

```
  domain-name:
    domain-configuration
  domain-name:
    domain-configuration
  ...
```

Configuration du domaine

- Configuration du domaine spatial
- Configuration de domaine personnalisée
- Configuration du domaine de service

Configuration du domaine spatial

Pour spécifier les propriétés d'un domaine spatial, remplacez-le *spatial-domain-name* par le nom de votre choix. Le nom doit comporter de 3 à 64 caractères et peut contenir les caractères A à Z, a à z, 0 à 9 et _ - (tiret). Spécifiez les propriétés du domaine spatial après le nom.

```
spatial-domain-name:
  launch_apps_by_partitioning_strategy:
    partitioning_strategy: "partitioning-strategy-name"
    grid_partition:
      x: number-of-partitions-along-x-axis
      y: number-of-partitions-along-y-axis
  app_config:
    package: "app-package-s3-uri"
    launch_command: ["app-launch-command", "parameter1", ...]
    required_resource_units:
      compute: app-resource-units
  image: "ecr-repository-uri"
```

Stratégie de partitionnement du domaine spatial

La `launch_apps_by_partitioning_strategy` section (obligatoire) spécifie la stratégie de partitionnement et les dimensions (en nombre de partitions) de l'espace de simulation.

```
  launch_apps_by_partitioning_strategy:
    partitioning_strategy: "partitioning-strategy-name"
    grid_partition:
      x: number-of-partitions-along-x-axis
      y: number-of-partitions-along-y-axis
```

Propriétés

`partitioning_strategy`

Spécifie la stratégie de partitionnement pour ce domaine spatial.

Obligatoire : oui

Type : chaîne

Valeurs valides : La valeur de cette propriété doit correspondre au nom d'une stratégie de partitionnement définie dans la `partitioning_strategies` section. Pour de plus amples informations, veuillez consulter Stratégies de partitionnement.

`grid_partition`

Spécifie le nombre de partitions le long de chaque axe (x et y) dans une topologie de grille. Ces dimensions décrivent l'espace de simulation total pour ce domaine.

Obligatoire : selon les conditions. Cette propriété ne peut être spécifiée que si la topologie est définie sur. `"Grid"` Cette propriété dépend de la `grid_placement_groups` propriété de la stratégie de partitionnement spécifiée pour ce domaine :

- Cette propriété est obligatoire si la stratégie de partitionnement de ce domaine ne spécifie pas de `grid_placement_groups` configuration.
- S'il existe une `grid_placement_groups` configuration mais que vous ne spécifiez pas`grid_partition`, elle SimSpace Weaver utilisera les mêmes dimensions que la `grid_placment_groups` configuration.
- Si vous spécifiez les deux `grid_placement_groups` et`grid_partition`, les dimensions de `grid_partition` doivent être des multiples des dimensions de `grid_placement_groups` (par exemple, si vos `grid_placement_groups` dimensions sont 2x2, certaines dimensions valides pour `grid_partition` sont 2x2, 4x4, 6x6, 8x8, 10x10).

Type : entier (pour chaque axe)

Valeurs valides : `1 - 20`

Configuration spatiale de l'application

La `app_config` section (obligatoire) spécifie le package, la configuration de lancement et les ressources requises pour les applications de ce domaine.

```
    app_config:
      package: "app-package-s3-uri"
      launch_command: ["app-launch-command", "parameter1", ...]
      required_resource_units:
        compute: app-resource-units
```

Propriétés

`package`

Spécifie le package (fichier zip) qui contient le fichier exécutable/binaire de l'application. Le package doit être stocké dans un compartiment Amazon S3. Seul le format de fichier zip est pris en charge.

Obligatoire : oui

Type : chaîne

Valeurs valides : URI Amazon S3 du package dans un compartiment Amazon S3. Par exemple, `s3://weaver-myproject-111122223333-app-zips-us-west-2/MySpatialApp.zip`.

`launch_command`

Spécifie le nom du fichier exécutable/binaire et les paramètres de ligne de commande pour lancer l'application. Chaque jeton de chaîne de ligne de commande est un élément du tableau.

Obligatoire : oui

Type : tableau de chaînes

`required_resource_units`

Spécifie le nombre d'unités de ressources qui SimSpace Weaver doivent être allouées à chaque instance de cette application. Une unité de ressource est un nombre fixe d'unités centrales virtuelles (vCPUs) et mémoire à accès aléatoire (RAM) sur un travailleur. Pour plus d'informations sur les unités de ressources, consultezPoints de terminaison et quotas de service. La `compute` propriété spécifie une allocation d'unités de ressources pour la `compute` famille de travailleurs et constitue actuellement le seul type d'allocation valide.

Obligatoire : oui

Type : entier

Valeurs valides : 1 - 4

Image de conteneur personnalisée

La `image` propriété (facultative) indique l'emplacement d'une image de conteneur SimSpace Weaver utilisée pour exécuter des applications dans ce domaine (non prise en charge dans les versions `1.13` et`1.12`). Fournissez l'URI d'un référentiel dans Amazon Elastic Container Registry (Amazon ECR) contenant l'image. Si cette propriété n'est pas spécifiée mais que `default_image` c'est le cas dans la `simulation_properties` section de niveau supérieur, les applications de ce domaine utilisent le`default_image`. Pour de plus amples informations, veuillez consulter Conteneurs personnalisés.

```
  image: "ecr-repository-uri"
```

Propriétés

`image`

Spécifie l'emplacement d'une image de conteneur pour exécuter des applications dans ce domaine.

Obligatoire : non

Type : chaîne

Valeurs valides :

- L'URI d'un référentiel dans Amazon Elastic Container Registry (Amazon ECR) (par exemple,) `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`

Configuration de domaine personnalisée

Pour spécifier les propriétés d'un domaine personnalisé, remplacez-le *`custom-domain-name`* par le nom de votre choix. Le nom doit comporter de 3 à 64 caractères et peut contenir les caractères A à Z, a à z, 0 à 9 et _ - (tiret). Spécifiez les propriétés du domaine personnalisé après le nom. Répétez ce processus pour chaque domaine personnalisé.

```
custom-domain-name:
  launch_apps_via_start_app_call: {}
  app_config:
    package: "app-package-s3-uri"
```

```
      launch_command: ["app-launch-command", "parameter1", ...]
      required_resource_units:
        compute: app-resource-units
      endpoint_config:
        ingress_ports: [port1, port2, ...]
    image: "ecr-repository-uri"
```

Propriétés

`launch_apps_via_start_app_call`

Cette propriété est requise pour lancer vos applications personnalisées à l'aide du StartApp API.

Obligatoire : oui

Type : N/A

Valeurs valides : `{}`

Configuration personnalisée de l'application

Le `app_config section` (obligatoire) spécifie le package, la configuration de lancement, les besoins en ressources et les ports réseau pour les applications de ce domaine personnalisé.

```
    app_config:
      package: "app-package-s3-uri"
      launch_command: ["app-launch-command", "parameter1", ...]
      required_resource_units:
        compute: app-resource-units
      endpoint_config:
        ingress_ports: [port1, port2, ...]
```

Propriétés

`package`

Spécifie le package (fichier zip) qui contient le fichier exécutable/binaire de l'application. Le package doit être stocké dans un compartiment Amazon S3. Seul le format de fichier zip est pris en charge.

Obligatoire : oui

Type : chaîne

Valeurs valides : URI Amazon S3 du package dans un compartiment Amazon S3. Par exemple, `s3://weaver-myproject-111122223333-app-zips-us-west-2/MyCustomApp.zip`.

`launch_command`

Spécifie le nom du fichier exécutable/binaire et les paramètres de ligne de commande pour lancer l'application. Chaque jeton de chaîne de ligne de commande est un élément du tableau.

Obligatoire : oui

Type : tableau de chaînes

`required_resource_units`

Spécifie le nombre d'unités de ressources qui SimSpace Weaver doivent être allouées à chaque instance de cette application. Une unité de ressource est un nombre fixe d'unités centrales virtuelles (vCPUs) et mémoire à accès aléatoire (RAM) sur un travailleur. Pour plus d'informations sur les unités de ressources, consultezPoints de terminaison et quotas de service. La `compute` propriété spécifie une allocation d'unités de ressources pour la `compute` famille de travailleurs et constitue actuellement le seul type d'allocation valide.

Obligatoire : oui

Type : entier

Valeurs valides : `1 - 4`

`endpoint_config`

Spécifie les points de terminaison réseau pour les applications de ce domaine. La valeur de `ingress_ports` indique les ports auxquels vos applications personnalisées se lient pour les connexions client entrantes. SimSpace Weaver mappe les ports alloués dynamiquement aux ports d'entrée que vous avez spécifiés. Les ports d'entrée sont à la fois TCP et UDP. Utilisez la commande DescribeApp API pour trouver le numéro de port réel pour connecter vos clients.

Nécessaire : Non Si vous ne spécifiez pas de configuration de point de terminaison, vos applications personnalisées dans ce domaine ne disposeront pas de points de terminaison réseau.

Type : tableau d'entiers

Valeurs valides : `1024 -49152`. Les valeurs doivent être uniques.

Image de conteneur personnalisée

La `image` propriété (facultative) indique l'emplacement d'une image de conteneur SimSpace Weaver utilisée pour exécuter des applications dans ce domaine (non prise en charge dans les versions `1.13` et`1.12`). Fournissez l'URI d'un référentiel dans Amazon Elastic Container Registry (Amazon ECR) contenant l'image. Si cette propriété n'est pas spécifiée mais que `default_image` c'est le cas dans la `simulation_properties` section de niveau supérieur, les applications de ce domaine utilisent le`default_image`. Pour de plus amples informations, veuillez consulter Conteneurs personnalisés.

```
    image: "ecr-repository-uri"
```

Propriétés

`image`

Spécifie l'emplacement d'une image de conteneur pour exécuter des applications dans ce domaine.

Obligatoire : non

Type : chaîne

Valeurs valides :

- L'URI d'un référentiel dans Amazon Elastic Container Registry (Amazon ECR) (par exemple,) `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`

Configuration du domaine de service

Pour spécifier les propriétés d'un domaine de service, remplacez-le *`service-domain-name`* par le nom de votre choix. Le nom doit comporter de 3 à 64 caractères et peut contenir les caractères A à Z, a à z, 0 à 9 et _ - (tiret). Spécifiez les propriétés du domaine de service après le nom. Répétez cette procédure pour chaque domaine de service.

```
service-domain-name:
  launch_apps_per_worker:
    count: number-of-apps-to-launch
  app_config:
    package: "app-package-s3-uri"
    launch_command: ["app-launch-command", "parameter1", ...]
```

```
      required_resource_units:
        compute: app-resource-units
      endpoint_config:
        ingress_ports: [port1, port2, ...]
    image: "ecr-repository-uri"
```

Lancer des applications par collaborateur

La `launch_apps_per_worker` section (obligatoire) indique qu'il s'agit d'une configuration de domaine de service et indique le nombre d'applications de service à lancer par travailleur.

```
    launch_apps_per_worker:
      count: number-of-apps-to-launch
```

Propriétés

`count`

Cette propriété indique le nombre d'applications de service à lancer par travailleur.

Obligatoire : oui

Type : entier

Valeurs valides : `{}` | 1 |2. La valeur de `{}` indique la valeur par défaut de1.

Configuration de l'application de service

Le `app_config section` (obligatoire) spécifie le package, la configuration de lancement, les besoins en ressources et les ports réseau pour les applications de ce domaine de service.

```
    app_config:
      package: "app-package-s3-uri"
      launch_command: ["app-launch-command", "parameter1", ...]
      required_resource_units:
        compute: app-resource-units
      endpoint_config:
        ingress_ports: [port1, port2, ...]
```

Propriétés

`package`

Spécifie le package (fichier zip) qui contient le fichier exécutable/binaire de l'application. Le package doit être stocké dans un compartiment Amazon S3. Seul le format de fichier zip est pris en charge.

Obligatoire : oui

Type : chaîne

Valeurs valides : URI Amazon S3 du package dans un compartiment Amazon S3. Par exemple, `s3://weaver-myproject-111122223333-app-zips-us-west-2/MyServiceApp.zip`.

`launch_command`

Spécifie le nom du fichier exécutable/binaire et les paramètres de ligne de commande pour lancer l'application. Chaque jeton de chaîne de ligne de commande est un élément du tableau.

Obligatoire : oui

Type : tableau de chaînes

`required_resource_units`

Spécifie le nombre d'unités de ressources qui SimSpace Weaver doivent être allouées à chaque instance de cette application. Une unité de ressource est un nombre fixe d'unités centrales virtuelles (vCPUs) et mémoire à accès aléatoire (RAM) sur un travailleur. Pour plus d'informations sur les unités de ressources, consultezPoints de terminaison et quotas de service. La `compute` propriété spécifie une allocation d'unités de ressources pour la `compute` famille de travailleurs et constitue actuellement le seul type d'allocation valide.

Obligatoire : oui

Type : entier

Valeurs valides : `1 - 4`

`endpoint_config`

Spécifie les points de terminaison réseau pour les applications de ce domaine. La valeur de `ingress_ports` indique les ports auxquels vos applications de service se lient pour les connexions client entrantes. SimSpace Weaver mappe les ports alloués dynamiquement aux ports d'entrée que vous avez spécifiés. Les ports d'entrée sont à la fois TCP et UDP. Utilisez la commande DescribeApp API pour trouver le numéro de port réel pour connecter vos clients.

Nécessaire : Non Si vous ne spécifiez pas de configuration de point de terminaison, vos applications de service dans ce domaine ne disposeront pas de points de terminaison réseau.

Type : tableau d'entiers

Valeurs valides : `1024` -`49152`. Les valeurs doivent être uniques.

Image de conteneur personnalisée

La `image` propriété (facultative) indique l'emplacement d'une image de conteneur SimSpace Weaver utilisée pour exécuter des applications dans ce domaine (non prise en charge dans les versions `1.13` et`1.12`). Fournissez l'URI d'un référentiel dans Amazon Elastic Container Registry (Amazon ECR) contenant l'image. Si cette propriété n'est pas spécifiée mais que `default_image` c'est le cas dans la `simulation_properties` section de niveau supérieur, les applications de ce domaine utilisent le`default_image`. Pour de plus amples informations, veuillez consulter Conteneurs personnalisés.

```
    image: "ecr-repository-uri"
```

Propriétés

`image`

Spécifie l'emplacement d'une image de conteneur pour exécuter des applications dans ce domaine.

Obligatoire : non

Type : chaîne

Valeurs valides :

- L'URI d'un référentiel dans Amazon Elastic Container Registry (Amazon ECR) (par exemple,) `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`

Contraintes de placement

La `placement_constraints` section (facultative) indique quels domaines spatiaux SimSpace Weaver doivent être placés ensemble sur les mêmes travailleurs. Pour de plus amples informations, veuillez consulter Configuration de domaines spatiaux.

> **Important**
>
> Versions `1.13` et `1.12` non prises en charge`placement_constraints`.

```
placement_constraints:
  - placed_together: ["spatial-domain-name", "spatial-domain-name", ...]
    on_workers: ["worker-group-name"]
```

Propriétés

`placed_together`

Spécifie les domaines spatiaux qui SimSpace Weaver doivent être placés ensemble.

Obligatoire : oui

Type : tableau de chaînes

Valeurs valides : noms des domaines spatiaux spécifiés dans le schéma

`on_workers`

Spécifie le groupe de travail sur lequel les domaines SimSpace Weaver doivent être placés.

Obligatoire : oui

Type : tableau de chaînes à un élément

Valeurs valides : nom d'un groupe de travail spécifié dans le schéma

SimSpace Weaver Références d'API

SimSpace Weaver possède 2 ensembles différents d'interfaces de programmation d'applications (APIs) :

- Service APIs : ils APIs contrôlent le service et les ressources du service, telles que vos simulations, vos horloges et vos applications. Ils font partie du kit de développement AWS logiciel (SDK) principal et vous pouvez utiliser l'interface de ligne de AWS commande (CLI) pour les appeler. Pour plus d'informations sur le service APIs, consultez la référence de l'SimSpace Weaver API.
- SDK d'application APIs : ils APIs contrôlent les données de votre simulation. Vous les utilisez dans le code de votre application pour lire et écrire des données de champs d'entités, gérer des abonnements et surveiller les événements de la simulation. Pour plus d'informations, consultez la documentation du SDK de SimSpace Weaver l'application dans le dossier du SDK de l'application : *sdk-folder*\SimSpaceWeaverAppSdk\documentation

> **Note**
>
> *sdk-folder*est le dossier dans lequel vous avez décompressé le SimSpaceWeaverAppSdkDistributable package.

AWS SimSpace Weaver versions

Nous nous améliorons continuellement AWS SimSpace Weaver. Vous devez télécharger le dernier SDK de SimSpace Weaver l'application lorsque nous publions une nouvelle version si vous souhaitez profiter des nouvelles fonctionnalités et des mises à jour des fonctionnalités. Pour exécuter une simulation existante avec une version plus récente, vous devrez peut-être mettre à jour son schéma et son code, puis démarrer une nouvelle instance de la simulation. Vous n'avez pas besoin de procéder à une mise à niveau et vous pouvez continuer à exécuter des simulations existantes avec les versions précédentes. Vous pouvez consulter cette page pour voir quelles sont les différences entre les versions. Toutes les versions sont actuellement prises en charge.

> **Important**
>
> La dernière version du guide de AWS SimSpace Weaver l'utilisateur ne couvre que la dernière version du service. Vous trouverez la documentation des versions précédentes dans le catalogue des AWS SimSpace Weaver guides, disponible sur la page d'accueil principale de la documentation.

Dernière version

La dernière version est : 1.17.0

Comment trouver votre version actuelle

Si vous avez créé une simulation avec le SDK de SimSpace Weaver l'application, le `create-project` script a téléchargé une version des bibliothèques du SDK dans un sous-répertoire de votre. *`sdk-folder`* Le sous-répertoire qui contient les bibliothèques du SDK a un nom qui inclut le numéro de version du SDK :. `SimSpaceWeaverAppSdk-`*`sdk-version`* Par exemple, les bibliothèques de la version 1.16.0 sont disponibles. `SimSpaceWeaverAppSdk-1.16.0`

Vous pouvez également trouver la version du package distribuable du SDK de SimSpace Weaver l'application dans le fichier `app_sdk_distributable_version.txt` texte de votre. *`sdk-folder`*

Téléchargez la dernière version

Utilisez l'un des liens suivants pour télécharger la dernière version.

- Package complet du SDK d'applications distribuable
- Uniquement les bibliothèques du SDK de l'application

Vous pouvez également télécharger le package distribuable complet du SDK de l' SimSpace Weaver application depuis la SimSpace Weaver console du. AWS Management Console Choisissez Télécharger le SDK de l'application dans le volet de navigation.

> **Warning**
>
> N'utilisez pas le AWS CLI pour télécharger tout ce qui semble être le package distribuable du SDK de SimSpace Weaver l'application. Utilisez uniquement les liens de téléchargement figurant sur cette page ou le lien de téléchargement dans la console. Toute autre méthode ou tout autre emplacement de téléchargement n'est pas pris en charge et peut contenir du code obsolète, incorrect ou malveillant.

Résolution des problèmes liés aux téléchargements du SDK d'applications

Nous utilisons Amazon CloudFront (CloudFront) pour distribuer les fichiers .zip du SDK de l'application. Il se peut que vous rencontriez certaines des situations suivantes.

- Le package téléchargé n'est pas la dernière version
  - Si le fichier .zip que vous avez téléchargé ne contient pas la dernière version, il est possible que le cache de votre emplacement CloudFront périphérique n'ait pas encore été mis à jour. Téléchargez à nouveau après 24 heures.
- Vous obtenez une erreur HTTP 4xx ou 5xx en utilisant un lien de téléchargement
  - Réessayez au bout de 24 heures. Si le même message d'erreur s'affiche, utilisez le lien Feedback en bas de la SimSpace Weaver console pour signaler le problème. Sélectionnez Signaler un problème comme type de commentaire.
- Votre navigateur indique qu'il ne peut pas charger la page
  - Il se peut que vous rencontriez un problème de configuration du réseau local ou du navigateur. Vérifiez que vous pouvez charger d'autres pages. Videz le cache de votre navigateur et réessayez. Assurez-vous que vous n'avez pas de règles de pare-feu susceptibles de bloquer l'URL de téléchargement.
- Un message d'erreur s'affiche lorsque vous essayez d'enregistrer le fichier

  - Vérifiez les autorisations de votre système de fichiers local afin de vous assurer que vous disposez des autorisations appropriées pour enregistrer le fichier.
- Votre navigateur s'affiche AccessDenied
  - Si vous avez saisi manuellement l'URL dans votre navigateur, vérifiez qu'elle est correcte. Si vous avez utilisé un lien de téléchargement, assurez-vous qu'aucun élément n'interférait avec l'URL de votre navigateur ; réutilisez-le.

Installez la dernière version

Pour installer la dernière version

1. Téléchargez la dernière version.
2. Décompressez le SimSpaceWeaverAppSdkDistributable fichier .zip dans un dossier.
3. Exécutez `python setup.py` à partir du dossier du SDK de l' SimSpace Weaver application décompressé de la dernière version.
4. Utilisez le dossier du SDK de l' SimSpace Weaver application décompressé de la dernière version au lieu de la version précédente.

Versions du service

Version	Remarques	Date de publication	Documentation	Téléchargement du SDK de l'application
1.17.0	Modifications majeures apportées au package distribuable du SDK de SimSpace Weaver l'application • Nous avons remplacé les scripts Windows	17 avril 2024	Ce guide	• Package complet • Bibliothèques uniquement Consultez la section Résolution des problèmes.

Version	Remarques	Date de publication	Documentation	Téléchargement du SDK de l'application
	batch et Linux Bash par des scripts basés sur Python. Python 3.9 est donc désormais requis pour utiliser les scripts et les exemples, même si vous n'utilisez pas (ou n'avez pas l'intention d'utiliser) le SDK Python. • Cette version augmente la prise en charge d'Amazon Linux 2. • Nous avons corrigé plusieurs bugs dans SimSpace Weaver Local. Pour plus d'informations, veuillez consulter les notes de mise à jour.			

Version	Remarques	Date de publication	Documentation	Téléchargement du SDK de l'application
	Correctif de bogue • Nous avons corrigé un bogue qui empêchait les entités de devenir propriétaires si elles n'effectuaient pas leur transfert entre télétrava illeurs.			

Version	Remarques	Date de publication	Documentation	Téléchargement du SDK de l'application
1.16.0	Nouvelle fonctionnalité : • Vous pouvez désormais utiliser la messagerie APIs dans le SDK de SimSpace Weaver l'application pour envoyer et recevoir des messages entre vos applications. Cette fonctionnalité est disponible pour les intégrations C++, Python et Unity et Unreal Engine 5.	12 février 2024	Consultez le catalogue AWS SimSpace Weaver des guides.	• Package complet • Bibliothèques uniquement Consultez la section Résolution des problèmes.

Version	Remarques	Date de publication	Documentation	Téléchargement du SDK de l'application
1.15.3	SimSpace Weaver Local mise à jour : • Nous avons changé SimSpace Weaver Local pour l'aligner plus étroitement sur le développement du AWS Cloud. Ces modifications ont un impact sur les projets et les flux de travail C++, Python, Unity et Unreal Engine pour SimSpace Weaver Local.	4 décembre 2023	Consultez le catalogue AWS SimSpace Weaver des guides.	Non disponible au téléchargement

Version	Remarques	Date de publication	Documentation	Téléchargement du SDK de l'application
1.15.2	Mise à jour du package distribuable du SDK de l'application : • Nous avons mis `Dockerfile` à jour le pour utiliser la version spécifique requise de cmake. Les builds de conteneurs Docker risquent d'échouer sans cette modification.	2 novembre 2023	Consultez le catalogue AWS SimSpace Weaver des guides.	Non disponible au téléchargement
1.15.1	Mise à jour des fonctionnalités : • SDK Python : cette version corrige un problème qui provoquait l'échec des simulations basées sur Python dans le. AWS Cloud	22 septembre 2023	Consultez le catalogue AWS SimSpace Weaver des guides.	Non disponible au téléchargement

Version	Remarques	Date de publication	Documentation	Téléchargement du SDK de l'application
1.15.0	Nouvelle fonctionnalité : • SDK Python : vous pouvez désormais développer vos simulations en Python. Le package distribuable du SDK de SimSpace Weaver l'application inclut un modèle pour un exemple de projet Python et son client de vue Python.	31 août 2023	Consultez le catalogue AWS SimSpace Weaver des guides.	Non disponible au téléchargement

Version	Remarques	Date de publication	Documentation	Téléchargement du SDK de l'application
1.14.0	Nouvelles fonctions : • Conteneurs personnalisés : créez votre propre image de conteneur basée sur Amazon Linux 2 (AL2), stockez-la dans Amazon Elastic Container Registry (Amazon ECR) et utilisez-la pour exécuter SimSpace Weaver vos applications dans le. AWS Cloud • Domaines spatiaux multiples : créez plusieurs domaines spatiaux dans une simulation. Séparez la logique de simulation au lieu de tout	26 juillet 2023	Consultez le catalogue AWS SimSpace Weaver des guides.	Non disponible au téléchargement

Version	Remarques	Date de publication	Documentation	Téléchargement du SDK de l'application
	combiner dans une seule application spatiale. Allouez différentes ressources aux domaines spatiaux en fonction de leurs besoins. • Taux de clics illimité : faites en sorte que votre simulation s'exécute aussi vite que votre code peut s'exécuter. Réglez l'horloge de votre simulation de manière à ce qu'elle envoie le prochain tick dès que toutes les applications auront terminé leurs opérations de validation pour le tick en cours.			

Version	Remarques	Date de publication	Documentation	Téléchargement du SDK de l'application
	SimSpace Weaver SDK de l'application : • La valeur de `tick_rate` est désormais une chaîne. La valeur doit inclure des guillemets ('). Le taux de réponse pour les versions antérieures est toujours un entier.			
1.13.1	SimSpace Weaver SDK de l'application : • Mise à jour des fonctionnalités : la création de projets fonctionne désormais correctement avec le `PathfindingSampleUnreal` modèle.	7 juin 2023	Consultez le catalogue AWS SimSpace Weaver des guides.	Non disponible au téléchargement

Version	Remarques	Date de publication	Documentation	Téléchargement du SDK de l'application
1.13.0	SimSpace Weaver service APIs : • • Nouvelle CreateSnapshot action • Modifications apportées à l'StartSimulation action : • Ajout d'un SnapshotS3Location paramètre permettant de démarrer à partir d'un instantané. • Le SchemaS3Location paramètre est désormais facultatif. • Modifications apportées à DescribeSimulation la sortie :	29 avril 2023	Consultez le catalogue AWS SimSpace Weaver des guides.	Non disponible au téléchargement

Version	Remarques	Date de publication	Documentation	Téléchargement du SDK de l'application
	• `SchemaError` est obsolète • Ajout d'un `StartError` champ. • Ajout d'un `SnapshotS3Location` champ. • Ajout d'un statut `SNAPSHOT_IN_PROGRESS` de simulation. • Nouveau type `S3Destination` de données SimSpace Weaver console : • Nouvelle fonctionnalité pour créer des instantanés. • Nouvelle fonctionnalité permettant			

Version	Remarques	Date de publication	Documentation	Téléchargement du SDK de l'application
	de démarrer une simulation à partir d'un instantané. SimSpace Weaver SDK de l'application : • Nouveaux scripts pour prendre en charge les instantanés • `create-snapshot- project-name .bat` • `start-from-snapshot- project-name .bat` • `quick-start-from-snapshot- project-name -cli.bat` • `list-snapshots- project-name .bat` • Les projets utilisent désormais un seul compartim			

Version	Remarques	Date de publication	Documentation	Téléchargement du SDK de l'application
	ent Amazon S3 par projet : weaver-*lowercase-project-name* --*account-number region*			

Version	Remarques	Date de publication	Documentation	Téléchargement du SDK de l'application
1.12.3	SimSpace Weaver SDK de l'application : • Les scripts suivants prennent désormais en charge le `--maximum-duration` paramètre : • `quick-start-`*`project-name`*`-cli.bat` • `quick-start-`*`project-name`*`-cli.sh` • `start-simulation-`*`pro`*`*name*`.bat` • `start-simulation-`*`pro`*`*name*`.sh` • `run-`*`project-name`*`.bat` • `run-`*`project-name`*`.sh`	27 mars 2023	Consultez le catalogue AWS SimSpace Weaver des guides.	Non disponible au téléchargement

Version	Remarques	Date de publication	Documentation	Téléchargement du SDK de l'application
1.12.2	SimSpace Weaver SDK de l'application : • Correction d'un bug : fonctionne `docker-create-image.bat` désormais correctement.	1er mars 2023	Consultez le catalogue AWS SimSpace Weaver des guides.	Non disponible au téléchargement

Version	Remarques	Date de publication	Documentation	Téléchargement du SDK de l'application
1.12.1	SimSpace Weaver SDK de l'application : • Les scripts acceptent désormais un AWS CLI profil à utiliser pour AWS l'authentification. • Les scripts prennent désormais en AWS IAM Identity Center charge AWS l'authentification. SimSpace Weaver Local: • Correction de bogue : renvoie `Api::BeginUpdateWillBlock` désormais correctement `true` si toutes les applications spatiales n'ont	28 février 2023	Consultez le catalogue AWS SimSpace Weaver des guides.	Non disponible au téléchargement

Version	Remarques	Date de publication	Documentation	Téléchargement du SDK de l'application
	pas rejoint la simulation.			
1.12.0	Version pour mise à disposition générale (GA)	29 novembre 2022	Consultez le catalogue AWS SimSpace Weaver des guides.	Non disponible au téléchargement

AWS SimSpace Weaver version 1.17.0

Cette version est une refonte du package distribuable du SDK de SimSpace Weaver l'application. Nous avons remplacé les scripts Windows Batch et Linux Bash obsolètes par des scripts basés sur Python.

> **Important**
>
> Python 3.9 est désormais obligatoire pour utiliser les scripts et les exemples, et pas seulement pour le SDK Python.

Table des matières

- Changements majeurs pour la version 1.17.0
- Mettre à jour un projet vers la version 1.17.0
- Questions fréquemment posées sur la version 1.17.0

Changements majeurs pour la version 1.17.0

- Création de projets simplifiée
  - Après l'exécution`setup.py`, vous pouvez créer votre propre projet en copiant et collant simplement un échantillon.
- Échantillons en 1 clic

  - Le fichier zip de distribution contient désormais ready-to-use des exemples qui fonctionnent une fois la distribution configurée.
- Chaque SDK existe désormais dans son propre répertoire :`cpp`, `pythonunreal`, et`unity`. Vous devrez peut-être mettre à jour vos chemins en fonction du SDK que vous utilisez.
- Améliorations apportées aux scripts d'assistance.
  - Les scripts contiennent désormais plusieurs AWS CLI options pour optimiser leur flexibilité.
  - Lancement et connexion du client de console intégrés dans le cadre du démarrage rapide.
  - Sortie console améliorée.
  - La création d'échantillons Unreal et Unity fonctionne désormais avec`quick-start`, plus aucune étape manuelle n'est requise.
  - SimSpace Weaver Local fonctionne désormais en appelant simplement`quick-start`, plus besoin de créer et de lancer manuellement.
  - SimSpace Weaver Local `quick-start`dispose d'un support intégré pour la journalisation des sorties des applications.
  - SimSpace Weaver Local peut désormais être lancé dans un environnement sans interface graphique, par exemple dans une session SSH.
  - La fonctionnalité « conteneur personnalisé » est désormais intégrée au `quick-start` script.
  - Support accru d'Amazon Linux 2 (AL2) : les flux de travail de script pour Windows AL2 sont désormais comparables. Auparavant, les AL2 projets nécessitaient davantage d'étapes manuelles et SimSpace Weaver Local n'était pas pris en charge pour AL2.
- Les plugins Unreal Engine et Unity sont désormais inclus dans le package distribuable du SDK de SimSpace Weaver l'application.
- Corrections de bugs pour SimSpace Weaver Local
  - Correction d'un bug qui permettait d'attribuer le même ID d'entité aux entités.
  - Correction d'un bug qui permettait d'attribuer le même ID de partition à deux partitions.
  - Correction d'un bogue lié aux tentatives d'écriture des applications dans des entités qui ne leur appartenaient pas.
  - Un problème de fuite de mémoire a été résolu.

Mettre à jour un projet vers la version 1.17.0

1. Configuration de la distribution 1.17.0 : Reprenez les procédures de configuration car nous les avons modifiées pour la version 1.17.0. Pour de plus amples informations, veuillez consulter Configuration pour SimSpace Weaver.
2. Chaque SDK de l'application Weaver existe désormais dans son propre répertoire. Mettez à jour vos chemins de construction en conséquence.

   a. Répertoire C++ : `SimSpaceWeaverAppSdk/cpp`

      - Le SDK de l'application SimSpace Weaver C++ utilise désormais un `FindSimSpaceWeaverAppSdk.cmake` fichier. Ce fichier définit une `weaver` cible à laquelle est liée, et inclut des corrections de bogues importantes lors de la création pour Weaver dans le AWS Cloud. Vous devriez l'utiliser au lieu de créer un lien direct vers les fichiers binaires.

   b. Répertoire Python : `SimSpaceWeaverAppSdk/python`

   c. Plug-in Unity : `SimSpaceWeaverAppSdk/unity`

   d. Plug-in Unreal Engine : `SimSpaceWeaverAppSdk/unreal`

3. Les `tools` scripts précédents ne fonctionneront pas avec la nouvelle SimSpace Weaver distribution. Pour utiliser les nouveaux `tools` scripts avec votre projet :

   a. Supprimez vos anciens `tools/local` répertoires `tools/windowstools/linux`, et.

   b. Copiez le `tools` répertoire d'un exemple de projet qui utilise le même SDK d' SimSpace Weaver application que votre projet. Assurez-vous d'avoir exécuté `setup.py` avant de copier ce répertoire.

> **Important**
>
> Le fonctionnement des scripts d'outils est garanti uniquement avec les exemples de projets. Vous devrez peut-être modifier ces scripts, en particulier le `build.py` script, pour qu'ils fonctionnent avec votre projet. Toutes les modifications seront propres à votre projet, nous ne pouvons donc pas vous fournir de conseils.

Questions fréquemment posées sur la version 1.17.0

Dois-je passer à la version 1.17.0 ?

Cette mise à jour n'est pas obligatoire car aucune modification n'a été apportée à l' SimSpace Weaver API ou au SDK de SimSpace Weaver l'application. Vous devez passer à la version 1.17.0 si vous souhaitez utiliser la version 1.17.0 SimSpace Weaver Local, qui contient plusieurs corrections de bogues.

Quelle est la version minimale de Python requise ?

Python 3.9 est la version minimale.

Quelle est la CMake version minimale requise ?

CMake la version 3.13 est la version minimale.

Quelle est la version minimale d'Unreal Engine requise ?

Unreal Engine 5.0 est le minimum.

Quelle est la version minimale de Unity requise ?

La version 2022.3.19.F1 de Unity est la version minimale.

AWS SimSpace Weaver version 1.15.1

Cette version est une mise à jour requise pour le SDK Python initialement publié dans la SimSpace Weaver version 1.15.0. Il corrige un problème d'incompatibilité de version qui entraînait l'échec des simulations basées sur Python dans le. AWS Cloud Utilisez cette version au lieu de la version 1.15.0.

Mettre à jour un projet Python existant vers la version 1.15.1

Si vous avez créé un projet Python avec la version 1.15.0 du SDK Python, vous devez effectuer les étapes suivantes pour le mettre à jour vers la version 1.15.1 afin qu'il puisse s'exécuter dans le. AWS Cloud

Au lieu de suivre cette procédure, vous pouvez également créer un nouveau projet Python avec le SDK Python 1.15.1 et déplacer votre code personnalisé vers le nouveau projet.

Pour mettre à jour un projet Python 1.15.0 vers la version 1.15.1

1. Accédez au dossier de votre projet Python.

2. `src/PythonBubblesSample/bin/run-python`Modifiez la ligne suivante :

```
export PYTHONPATH=$PYTHONPATH:/roapp/lib
```

Jusqu'à ce qui suit :

```
export PYTHONPATH=$PYTHONPATH:$LD_LIBRARY_PATH:/roapp/lib
```

3. `CMakeLists.txt`Supprimez les lignes suivantes :

- ```
  file(COPY "${SDK_PATH}/libweaver_app_sdk_python_v1_$ENV{PYTHON_VERSION}.so"
   DESTINATION "${ZIP_FILES_DIR}/lib/weaver_app_sdk_v1")
  ```
- ```
  file(RENAME "${ZIP_FILES_DIR}/lib/weaver_app_sdk_v1/libweaver_app_sdk_python_v1_
  $ENV{PYTHON_VERSION}.so" "${ZIP_FILES_DIR}/lib/weaver_app_sdk_v1/
  libweaver_app_sdk_python_v1.so")
  ```
- ```
  message("    * COPYING WEAVER PYTHON SDK TO BUILD DIR ${ZIP_FILES_DIR}....")
  ```
- ```
  file(COPY ${SDK_DIR} DESTINATION ${ZIP_FILES_DIR}/lib/weaver_app_sdk_v1)
  ```

Résolution des problèmes liés à la version 1.15.1

Après la mise à jour d'une simulation Python 1.15.0, elle ne démarre pas dans AWS Cloud

Symptômes : environ 5 à 10 minutes après le début de la simulation, le journal de gestion de la simulation signale un `internal error` et l'état de la simulation est`FAILED`.

Cela peut se produire si un fichier de bibliothèque du SDK Python 1.15.0 est inclus dans un fichier zip d'application. Assurez-vous d'avoir suivi les étapes de mise à jour de votre projet, et assurez-vous que cela ne figure pas dans vos fichiers zip ou qu'il `libweaver_app_sdk_python_v1.so` n'y est pas référencé de quelque manière que ce soit.

Questions fréquemment posées sur la version 1.15.1

Cette version affecte-t-elle autre chose que le SDK Python ?

Non.

Dois-je passer à la version 1.15.1 ?

Il n'est pas nécessaire de passer à la version 1.15.1 si vous n'avez pas l'intention d'utiliser Python pour vos applications spatiales. Si vous avez effectué la mise à jour vers la version 1.15.0, vos simulations basées sur Python ne s'exécuteront pas dans le. AWS Cloud Nous vous recommandons de passer à la version 1.15.1 si vous utilisez la version 1.15.0.

Qu’est-ce qu’**$LD_LIBRARY_PATH** ?

Il s'agit de l'emplacement du SDK Python lorsque votre simulation s'exécute dans le AWS Cloud. C'est nouveau pour la version 1.15.1. Nous avons apporté cette modification pour éviter les problèmes de version de Python à l'avenir. La création de liens vers ce répertoire est fonctionnellement identique à celle de la version `libweaver_app_sdk_python_v1.so` 1.15.0.

Historique du document pour AWS SimSpace Weaver

Le tableau suivant décrit les modifications importantes apportées à la SimSpace Weaver documentation.

Date	Modification	Mises à jour de la documentation	Versions d'API mises à jour
20 mai 2025	Avis de fin de support	Avis de fin de support : le 20 mai 2026, AWS le support de AWS SimSpace Weaver. Après le 20 mai 2026, vous ne pourrez plus accéder à la SimSpace Weaver console ni aux SimSpace Weaver ressources. Pour plus d'informations, voir AWS SimSpace Weaver fin du support.	N/A
17 avril 2024	Contenu mis à jour	Mise à jour complète du guide de l'utilisateur pour la version 1.17.0. Changements majeurs apportés au Configura tion chapitre et aux Premiers pas didactici els. Consultez les notes de publication pour plus d'informations.	N/A
12 février 20	Contenu mis à jour	Le AWS SimSpace Weaver versions chapitre a été mis à jour pour la version 1.16.0.	N/A

Date	Modification	Mises à jour de la documentation	Versions d'API mises à jour
12 février 202	Nouveau contenu	La Messagerie section a été ajoutée dans le cadre de la sortie de la version 1.16.0. Cette section décrit la messagerie APIs ajoutée au SDK de l' SimSpace Weaver application. Vous pouvez les utiliser APIs pour envoyer et recevoir des messages entre vos applications.	N/A
12 février 202	Contenu mis à jour	Le SimSpace Weaver référence du schéma de simulation chapitre a été mis à jour pour la version 1.16.0.	N/A
12 février 202	Contenu mis à jour	Des quotas de service pour la messagerie ont été ajoutés au SimSpace Points de terminaison et quotas Weaver chapitre.	N/A

Date	Modification	Mises à jour de la documentation	Versions d'API mises à jour
12 février 20	Nouveaux guides	Répartissez le contenu des versions antérieures à la version 1.16.0 dans un guide distinct. Ajout du catalogue de AWS SimSpace Weaver guides (disponible sur la page d'accueil principale de la documentation) pour accéder aux guides des versions précédentes.	N/A
4 décembre 2	Contenu mis à jour	Le AWS SimSpace Weaver versions chapitre a été mis à jour pour la sortie de la version 1.15.3.	N/A
4 décembre 2	Contenu mis à jour	Le AWS SimSpace Weaver versions chapitre a été mis à jour pour inclure les instructi ons d'installation de la dernière version.	N/A
4 décembre 2	Contenu mis à jour	A mis à jour leQuotas de service pour SimSpace Weaver Local.	N/A

Date	Modification	Mises à jour de la documentation	Versions d'API mises à jour
4 décembre 2	Contenu nouveau et mis à jour	Restructuration de la Développement local en SimSpace Weaver section et ajout d'une nouvelle page qui décrit les différences par rapport à celles SimSpace Weaver Local introduites dans la version 1.15.3.	N/A
7 novembre 2	Contenu mis à jour	Mise à jour des instructions pour configurer Docker et WSL afin qu'ils utilisent le lien/URL de téléchargement direct du SDK de l'application. Pour de plus amples informations, veuillez consulter Configurez votre environnement local pour SimSpace Weaver.	N/A
2 novembre 2	Contenu mis à jour	Mise à jour de la page des versions de service pour la version 1.15.2. Pour de plus amples informations, veuillez consulter Versions du service.	N/A

Date	Modification	Mises à jour de la documentation	Versions d'API mises à jour
23 octobre 2(	Contenu mis à jour	Mise à jour de la page des versions de service avec de nouvelles instructions pour télécharger le package distribuable du SDK de l'application. Les clients doivent désormais utiliser uniquement l'un de nos liens de téléchargement direct approuvés et ne pas l'utiliser AWS CLI pour télécharger le package distribuable du SDK de l'application. Pour de plus amples informations, veuillez consulter Téléchargez la dernière version.	N/A
22 septembre	Nouveau contenu	Ajout d'une page de notes de version avec des instructions de mise à jour pour la version 1.15.1. Pour de plus amples informations, veuillez consulter AWS SimSpace Weaver version 1.15.1.	N/A

Date	Modification	Mises à jour de la documentation	Versions d'API mises à jour
10 septembre 2023	Nouveau contenu	Ajout d'une section de résolution des problèmes pour les situations où le AWS CLI ne reconnaît pas SimSpace Weaver. Pour de plus amples informations, veuillez consulter Ils AWS CLI ne reconnaissent pas `simspaceweaver` .	N/A
10 septembre 2023	Contenu mis à jour	Instructions d'installation mises à jour pour le AWS CLI dans WSL. Pour de plus amples informati ons, veuillez consulter Configurez le package SimSpace Weaver de distribution pour Amazon Linux 2 (AL2) dans Windows Subsystem for Linux (WSL).	N/A
7 septembre 2(	Mise à jour d'API	BucketName et ObjectKey sont désormais obligatoires pour le type de données S3Location. BucketName est désormais obligatoire pour le type de données S3Destination.	AWS SDK : 2023-09-07

Date	Modification	Mises à jour de la documentation	Versions d'API mises à jour
31 août 2023	Nouveau contenu	Ajout d'une nouvelle section pour la version 1.15.0 :. Travail avec Python	N/A
15 août 2023	Contenu mis à jour	Instructions de téléchargement mises à jour AWS SimSpace Weaver versions pour ne répertorier que les buckets SimSpace Weaver Amazon S3 officiels. Les autres emplacements de téléchargement ne sont pas contrôlés par du code malveillant AWS et peuvent contenir du code malveillant.	N/A
26 juillet 202	Contenu mis à jour	Mis à jour Horloge.	N/A
26 juillet 202	Contenu mis à jour	Mis à jour Configuration de domaines spatiaux.	N/A
26 juillet 202	Nouveau contenu	Une section a été ajoutée : Conteneurs personnalisés.	N/A
26 juillet 202	Contenu mis à jour	Mise à jour AWS SimSpace Weaver versions pour la version 1.14.0.	N/A

Date	Modification	Mises à jour de la documentation	Versions d'API mises à jour
6 juillet 2023	Nouveau contenu	Une section a été ajoutée : Pathfinding Sample le client de console ne parvient pas à se connecter.	N/A
7 juin 2023	Contenu mis à jour	Mise à jour AWS SimSpace Weaver versions pour la version 1.13.1.	N/A
15 mai 2023	Nouveau contenu	Une section a été ajoutée : Utilisation d'instantanés avec AWS CloudFormation.	N/A
29 avril 2023	Nouveau contenu	Du contenu a été ajouté pour la version 1.13.0. Pour de plus amples informations, veuillez consulter AWS SimSpace Weaver versions.	AWS SDK : 28/04/2023
27 mars 202	Nouveau contenu	Ajout d'une section qui décrit la durée maximale des simulations. Ajout de notes dans les didacticiels de la version 1.12.3, qui ont ajouté la prise en charge du `--maximum-duration` paramètre aux scripts du SDK des SimSpace Weaver applications.	N/A

Date	Modification	Mises à jour de la documentation	Versions d'API mises à jour
9 mars 2023	Contenu modifié	Il a été précisé que nous ne fournissons que des instructions pour Docker Windows et pourWindows Subsystem for Linux (WSL), et que WSL (et tout autre environnement Linux) n'est pas pris en charge.	N/A
28 février 202	Nouveau contenu	Ajout d'un chapitre qui décrit SimSpace Weaver les versions.	N/A
28 février 202	Contenu modifié	Modification du contenu relatif à l'authentification pour inclure l'utilisation de () AWS IAM Identity Center et les profils nommés pour le AWS Command Line Interface (AWS CLI).	N/A
17 février 202	Nouveau contenu	Ajout d'une section sur la gestion de vos ressources avec AWS CloudFormation.	N/A
23 janvier 20	Nouveau contenu	Ajout d'instructions pour le débogage des simulations locales.	N/A

Date	Modification	Mises à jour de la documentation	Versions d'API mises à jour
29 novembre	Lancement de service	Publication du guide de l'utilisateur et de la référence d'API pour SimSpace Weaver.	AWS SDK : 29/11

Glossaire

Ce glossaire définit les termes spécifiques à AWS SimSpace Weaver.

Pour la AWS terminologie la plus récente, consultez le AWS glossaire dans le manuel de référence AWS général.

A

appli	Code exécutable (également appelé fichiers binaires) que vous créez. Le terme app peut faire référence au code ou à une instance en cours d'exécution de ce code. Une application encapsule le comportement de simulation. Les applications créent, suppriment, lisent et mettent à jour des entités.
SDK de l'application	Kit de développement logiciel (SDK) que vous utilisez pour intégrer une application. SimSpace Weaver Le SDK permet de lire et APIs d'écrire les données des entités et de suivre le temps de simulation. Pour de plus amples informations, veuillez consulter SimSpace Weaver SDK de l'application.

C

client	Processus (ou leurs définitions) qui existent en dehors de la simulation SimSpace Weaver et interagissent avec celle-ci par le biais d'une application personnalisée ou d'une application de service. Vous pouvez utiliser un client pour visualiser ou modifier l'état de la simulation.
horloge	Une abstraction des processus SimSpace Weaver de planification internes. L'horloge publie des ticks sur les applications pour maintenir la synchronisation de l'heure. Chaque simulation possède sa propre horloge.
fréquence d'horloge	Le nombre de ticks publiés par seconde par l'horloge sur les applications. Pour plus d'informations sur les fréquences d'horloge prises en charge, consultezSimSpace Points de terminaison et quotas Weaver.
taux de tic-tac	Voir la fréquence d'horloge.

unité de ressource de calcul	Unité de ressources informatiques (processeur et mémoire) d'un travailleur. Une seule instance d'une application se voit normalement attribuer une unité de ressource de calcul. Vous pouvez allouer plus d'une unité de ressource de calcul pour chaque application.
application personnalisée	Type d'application que vous utilisez pour lire l'état de la simulation et interagir avec celui-ci. Les applications personnalisées peuvent créer des entités dans la simulation, mais elles ne leur appartiennent pas. Lorsqu'une application personnalisée crée une entité, elle doit la transférer dans le domaine spatial. Vous contrôlez le cycle de vie d'une application personnalisée à l'aide de l'application APIs. Pour plus d'informations sur le SimSpace Weaver APIs, consultezSimSpace Weaver Références d'API.
domaine personnalisé	Un domaine qui contient des applications personnalisées.
partition personnalisée	Partition d'une application personnalisée.

D

date limite	Heure réelle à laquelle une opération (telle que le traitement d'une tique) doit être terminée.
domaine	Groupe d'instances d'applications qui exécutent le même code exécutable (binaire d'application) et disposent des mêmes options de lancement.

E

point de terminaison (service)	Nom de domaine complet (FQDN) que les programmes (tels que le AWS Command Line Interface) utilisent pour se connecter au SimSpace Weaver service.
point de terminaison (simulation)	Adresse IP et numéro de port que les clients utilisent pour se connecter à une simulation. Vous pouvez configurer les points de terminaison sur des applications personnalisées et des applications de service.
entité	Objets de données client (ou leurs définitions). Les entités peuvent être statiques (rester au même endroit) ou dynamiques (se déplacer dans l'espace de simulation). Par exemple, les personnes et les bâtiments dans une simulation.

I

indice (simulation)	Description des propriétés spatiales d'une simulation, y compris ses limites spatiales et son système de coordonnées.

L

cycle de vie (d'une application)	Description des étapes logiques attendues par une application au cours d'une simulation. Les cycles de vie sont gérés (SimSpace Weaver démarrage et arrêt de l'application) ou non gérés (vous démarrez et arrêtez l'application).
charge (données du champ d'entité)	Lisez les données des champs d'entité à partir du State Fabric.

P

cloison	Segment de mémoire partagée d'un travailleur. Chaque partition contient un sous-ensemble discret d'entités au sein d'un domaine. Une partition est attribuée à chaque application. Une application possède toutes les entités de sa partition. Lorsqu'une application crée une entité, elle la crée dans sa partition. Lorsque des entités se déplacent d'une partition à une autre, la propriété est transférée de l'application de la partition source à l'application de la partition de destination.

R

unité de ressources	Consultez ???.

S

schéma	Document YAML ou JSON qui décrit la configuration d'une simulation. SimSpace Weaver utilise un schéma pour créer une ressource de simulation.
application de service	Type d'application que vous utilisez pour lire l'état de la simulation et interagir avec celui-ci. Les applications de service peuvent créer des

	entités dans la simulation, mais elles doivent les transférer dans le domaine spatial. SimSpace Weaver gère le cycle de vie d'une application de service et en lance un (ou plusieurs, comme indiqué dans votre schéma de simulation) sur chaque collaborateur de votre simulation.
domaine de service	Un domaine qui contient des applications de service.
cloison de service	La partition d'une application de service.
simulation (ressource)	Abstraction d'un cluster de calcul qui exécute un espace virtuel simulé. Vous pouvez effectuer plusieurs simulations. Vous configurez une simulation à l'aide d'un schéma.
application spatiale	Type d'application qui encapsule la logique de simulation de base. Chaque application spatiale possède une (et une seule) partition.
domaine spatial	Un domaine qui contient des applications spatiales.
partition spatiale	Partition d'une application spatiale.
State Fabric	SimSpace Weaver est une base de données en mémoire. State FabricStocke l'état des simulations, y compris les entités et les SimSpace Weaver données internes.
store (données de champ d'entité)	Écrivez les données du champ d'entité dans le State Fabric.
abonnement	Demande de longue durée visant à ce qu'une instance d'application spécifique reçoive des données depuis une zone d'abonnement. L'application d'abonnement utilise un abonnement pour découvrir les modifications apportées aux entités dans la zone d'abonnement.
zone d'abonnement	Région bidimensionnelle de l'espace de simulation. Un abonnement fait référence à un espace d'abonnement. Une zone d'abonnement peut s'étendre sur plusieurs partitions et inclure également des parties de partitions. Une zone d'abonnement est continue dans les limites définies.

T

cocher	Une valeur discrète pour le temps (heure de l'horloge murale ou heure de simulation). Les applications peuvent itérer plus rapidement que la durée

	du tick, mais elles sont censées écrire des ticks spécifiés dans des délais spécifiques. Toutes les opérations relatives à toutes les applications pour une coche donnée doivent être terminées avant que la prochaine case puisse commencer.
taux de tic-tac	Voir la fréquence d'horloge.
heure (réelle)	L'époque actuelle du point de vue de la réalité. SimSpace Weaver utilise un horodatage POSIX de 64 bits qui est le nombre de nanosecondes écoulées depuis l'époque Unix (January 1, 1970, 00:00:00 UTC).
heure (simulation)	L'heure actuelle du point de vue de la simulation. SimSpace Weaver utilise un compteur logique de nombres entiers de 64 bits, qui peut ne pas correspondre directement à l'heure réelle.

W

travailleur	Une instance Amazon Elastic Compute Cloud (Amazon EC2) qui exécute du code de simulation.