



Guía del usuario para la versión 1.17.0

AWS SimSpace Weaver



AWS SimSpace Weaver: Guía del usuario para la versión 1.17.0

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

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

¿Qué es SimSpace Weaver?	1
Conceptos clave	1
Cómo funciona SimSpace Weaver	2
Cómo se usa SimSpace Weaver	5
Esquema de simulación	6
Trabajadores y unidades de recursos	6
Reloj de simulación	7
Particiones	7
State Fabric	7
Entidades	8
móviles	8
Ejemplos de casos de uso de	11
Configuración	13
Configure su cuenta	13
Inscríbese en una Cuenta de AWS	13
Creación de un usuario con acceso administrativo	14
Agregar permisos para usar SimSpace Weaver	15
Configure su entorno local	17
AL2 en Docker	17
AL2 en WSL	19
Uso de software con licencia	23
Introducción	24
Tutorial de inicio rápido	24
Paso 1: Habilitar el registro (opcional)	25
Paso 2: Inicio rápido con el cliente de consola (opción 1)	25
Paso 2: Comience rápidamente con el cliente de Unreal Engine (opción 2)	26
Detenga y elimine su simulación	26
Resolución de problemas	26
Tutorial introductorio	27
Paso 1: Habilitar el registro (opcional)	28
Paso 2: Inicie la simulación	28
Paso 3: Compruebe los registros (opcional)	35
Paso 4: Ver la simulación	37
Paso 5: Detenga y elimine la simulación	38

Resolución de problemas	39
Trabajando con SimSpace Weaver	40
Configuración de su simulación	40
Parámetros de configuración de la simulación	41
Versión de SDK	42
Propiedades de simulación	42
Trabajadores	43
Reloj	44
Estrategias de partición	47
Dominios	48
Duración máxima	59
Valor máximo	59
Valor predeterminado	59
Valor mínimo	59
Iniciar una simulación mediante la consola	59
El estado de una simulación que alcanza su duración máxima	60
Desarrollo de aplicaciones	60
Aplicación espacial	61
Aplicaciones personalizadas	61
Desarrollo de aplicaciones de cliente	62
Obtenga la dirección IP y el número de puerto	63
Lanzamiento del cliente de visualización de Unreal Engine	66
Resolución de problemas	67
Desarrollo local	68
Paso 1: Lance su simulación local	68
Paso 2: Visualice su simulación local	70
Paso 3: Detenga la simulación local (opcional en Windows)	70
Solución de problemas de desarrollo local	71
SDK de la aplicación SimSpace Weaver	72
Los métodos de la API devuelven un Result	73
Interactuar con el SDK de aplicación de nivel superior	73
Gestión de la simulación	74
Suscripciones	77
Entidades	78
Eventos de entidad	90
Result y control de errores	97

Genéricos y tipos de dominio	99
Operaciones diversas del SDK de la aplicación	99
Marco de demostración de SimSpace Weaver	102
Trabajar con cuotas	103
Obtener los límites de una aplicación	103
Obtener la cantidad de recursos que utiliza una aplicación	104
Restablecimiento de métricas	105
Superar un límite	105
Agotamiento de la memoria	106
Prácticas recomendadas	106
Depuración de simulaciones	107
Utilice SimSpace Weaver Local y observe el resultado de la consola	107
Mira tus registros en Amazon CloudWatch Logs	107
Utilice llamadas a la API describe	107
Conectar un cliente	108
Depuración de simulaciones locales	108
Contenedores personalizados	109
Creación de un contenedor personalizado	110
Modificar un proyecto para usar un contenedor personalizado	111
Preguntas frecuentes	114
Resolución de problemas	115
Trabajo con Python	115
Crear un proyecto de Python	116
Inicio de una simulación de Python	118
Cliente Python de muestra	118
Preguntas frecuentes	119
Resolución de problemas	119
Compatibilidad con otros motores	121
Unity	121
Unreal Engine	122
Uso de software con licencia	122
Administrar los recursos con AWS CloudFormation	122
Instantáneas	125
Instantáneas	125
Casos de uso de instantáneas	126
SimSpace Weaver Consola	127

AWS CLI	129
Preguntas frecuentes	131
Mensajería	132
Casos de uso de mensajería	132
Uso de las API de mensajería	133
¿Cuándo usar la mensajería	140
Consejos para trabajar con la mensajería	144
Errores de mensajería y solución de problemas	145
Prácticas recomendadas	148
Configuración de alarmas de facturación	148
Uso SimSpace Weaver Local	148
Puede detener las simulaciones que no necesite.	149
Elimine los recursos que no necesite	149
Disposición de copias de seguridad	149
Seguridad	150
Protección de datos	151
Cifrado en reposo	152
Cifrado en tránsito	152
Privacidad del tráfico entre redes	153
Identity and Access Management	153
Público	153
Autenticación con identidades	154
Administración de acceso mediante políticas	158
¿Cómo AWS SimSpace Weaver funciona con IAM	160
Ejemplos de políticas basadas en identidades	167
Permisos que SimSpace Weaver crea para usted	172
Prevención de la sustitución confusa entre servicios	174
Resolución de problemas	176
Registro y supervisión de eventos de seguridad	180
Validación de la conformidad	180
Resiliencia	182
Seguridad de infraestructuras	182
Modelo de seguridad de la conectividad de red	183
Configuración y análisis de vulnerabilidades	184
Prácticas recomendadas de seguridad	184
Encripte las comunicaciones entre sus aplicaciones y sus clientes	185

Realice copias de seguridad periódicas del estado de la simulación	185
Mantenga sus aplicaciones y SDKs	185
Registro y monitorización	187
Inicia sesión CloudWatch	187
Acceder a sus registros de SimSpace Weaver	187
SimSpace Weaver registros	188
Monitorización con CloudWatch	190
SimSpace Weaver métricas a nivel de cuenta	191
CloudTrail registros	191
SimSpace Weaver información en CloudTrail	192
Descripción de las entradas de los archivos de SimSpace Weaver registro	193
Puntos de conexión y Service Quotas	195
Puntos de conexión de servicio	195
Service Quotas	196
Cuotas de mensajería	199
Velocidades del reloj	199
Service Quotas de SimSpace Weaver Local	199
Solución de problemas	201
AssumeRoleAccessDenied	201
InvalidBucketName	203
ServiceQuotaExceededException	204
TooManyBuckets	204
Permiso denegado durante el inicio de la simulación	205
Problemas relacionados con el tiempo de uso de Docker	205
El cliente de consola no se puede conectar	206
N hay <code>simspaceweaver</code> en el AWS CLI	208
Referencia del esquema	210
Ejemplo de un esquema completo	210
Formato de esquema	212
SDKversión	213
Propiedades de simulación	213
Procesos de trabajo	215
Reloj	216
Estrategias de partición	216
Dominios	218
Restricciones para la ubicación	229

Referencias de API	230
SimSpace Weaver versiones	231
Última versión	231
Cómo encontrar la versión actual	231
Descargue la versión más reciente	231
Solución de problemas de descargas del SDK de aplicaciones	232
Instale la versión más reciente	233
Versiones del servicio	233
1.17.0	249
Cambios importantes para la versión 1.17.0	249
Actualiza un proyecto a la versión 1.17.0	250
Preguntas frecuentes sobre la versión 1.17.0	251
1.15.1	252
Actualizar un proyecto de Python existente a 1.15.1	252
Solución de problemas con la versión 1.15.1	253
Preguntas frecuentes sobre la versión 1.15.1	253
Historial de documentos	255
Glosario	264
.....	cclxix

¿Qué es AWS SimSpace Weaver?

AWS SimSpace Weaver es un servicio que puede utilizar para crear y ejecutar simulaciones espaciales a gran escala en Nube de AWS. Por ejemplo, puede crear simulaciones de multitudes, grandes entornos del mundo real y experiencias inmersivas e interactivas.

Con SimSpace Weaver, puede distribuir cargas de trabajo de simulación entre varias instancias de Amazon Elastic Compute Cloud (Amazon EC2). SimSpace Weaver despliega la infraestructura de AWS subyacente y gestiona la simulación, la gestión de los datos y la comunicación de red entre las instancias de Amazon EC2 que ejecutan la simulación.

Conceptos clave de SimSpace Weaver

Una simulación o juego está limitado por el ordenador que lo ejecuta. A medida que aumenta el tamaño y la complejidad de su mundo virtual, el rendimiento del procesamiento comienza a degradarse. Los cálculos tardan más, los sistemas se quedan sin memoria y la velocidad de fotogramas de los clientes disminuye. En el caso de las simulaciones que no necesitan un rendimiento en tiempo real, esto solo puede resultar molesto. O bien, podría tratarse de una situación empresarial crítica en la que el aumento de las demoras en el procesamiento se traduzca en un aumento de los costos. Si su simulación o juego necesita un rendimiento en tiempo real, la degradación del rendimiento es sin duda un problema.

Una solución habitual para una simulación que alcanza un límite de rendimiento es simplificar la simulación. Los juegos online con muchos usuarios suelen solucionar los problemas de escalado haciendo copias de su mundo virtual en diferentes servidores y distribuyendo los usuarios entre ellos.

SimSpace Weaver resuelve el problema de escalamiento dividiendo el mundo virtual espacialmente y distribuyendo las piezas en un clúster de instancias informáticas que se ejecutan en el Nube de AWS. Las instancias de cómputo funcionan juntas para procesar todo el mundo de la simulación en paralelo. Su mundo de simulación aparece como un único espacio integrado para todo lo que contiene y para todos los clientes que se conectan a él. Ya no es necesario simplificar una simulación debido al límite de rendimiento del hardware. En su lugar, puede añadir más capacidad de cómputo en la nube.

Temas

- [Cómo funciona SimSpace Weaver](#)

- [Cómo se usa SimSpace Weaver](#)
- [Esquema de simulación](#)
- [Trabajadores y unidades de recursos](#)
- [Reloj de simulación](#)
- [Particiones](#)
- [State Fabric](#)
- [Entidades](#)
- [móviles](#)

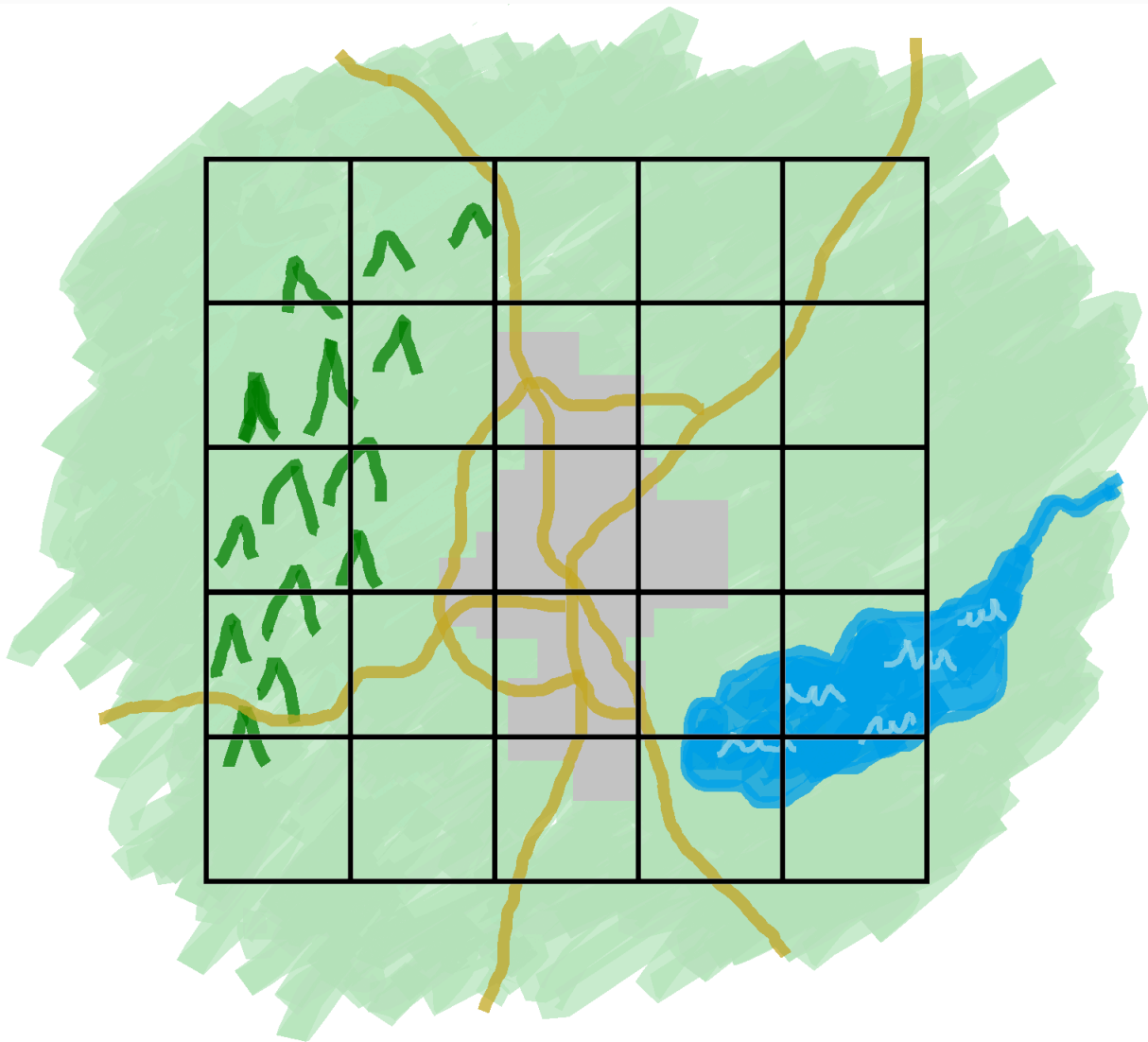
Cómo funciona SimSpace Weaver

La simulación consiste en un mundo con objetos en su interior. Algunos de los objetos (como personas y vehículos) se mueven y hacen cosas. Otros objetos (como árboles y edificios) son estáticos. En SimSpace Weaver, una entidad es un objeto del mundo de simulación.

Usted define los límites del mundo de simulación y lo divide en una cuadrícula. En lugar de crear una lógica de simulación que funcione en toda la cuadrícula, cree una lógica de simulación que funcione en una celda de la cuadrícula. En SimSpace Weaver, una aplicación espacial es un programa que se escribe y que implementa la lógica de simulación para una celda de la cuadrícula. Esto incluye la lógica de todas las entidades de esa celda. El área de propiedad de una aplicación espacial es la celda de la cuadrícula que controla la aplicación espacial.

Note

En SimSpace Weaver, el término «aplicación» puede hacer referencia al código de una aplicación o a una instancia en ejecución de ese código.



Su mundo de simulación dividido en una cuadrícula

Usted divide su mundo de simulación en una cuadrícula. Cada aplicación espacial implementa la lógica de simulación para una sola celda de la cuadrícula.

SimSpace Weaver ejecuta una instancia del código de la aplicación espacial para cada celda de la cuadrícula. Todas las instancias de aplicaciones espaciales se ejecutan en paralelo. Básicamente, SimSpace Weaver divide la simulación general en varias simulaciones más pequeñas. Cada una de las simulaciones más pequeñas gestiona una parte del mundo general de la simulación. SimSpace Weaver puede distribuir y ejecutar estas simulaciones más pequeñas en varias instancias de

Amazon Elastic Compute Cloud (Amazon EC2) (denominadas trabajadores) en el Nube de AWS. Un solo trabajador puede ejecutar varias aplicaciones espaciales.

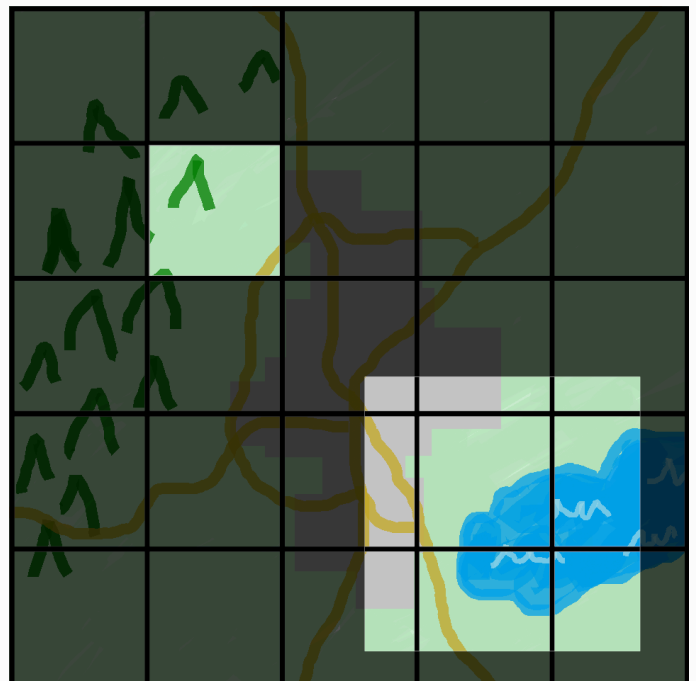
Las entidades pueden moverse por el mundo de la simulación. Si una entidad entra en el área de propiedad de otra aplicación espacial (otra celda de la cuadrícula), el propietario de la aplicación espacial de la nueva área asume el control de la entidad. Si la simulación se ejecuta en varios trabajadores, una entidad podría pasar del control de una aplicación espacial de un trabajador a una aplicación espacial de otro trabajador. Cuando una entidad se traslada a otro trabajador, SimSpace Weaver gestiona la comunicación de red subyacente.

Suscripciones

La visión del mundo de una aplicación espacial es su propia área de propiedad. Para saber qué sucede en otra parte del mundo de la simulación, la aplicación espacial crea una suscripción. El área de suscripción es un subconjunto del área global de simulación. Un área de suscripción puede incluir partes de varias áreas de propiedad, incluida la propia área de propiedad de la aplicación espacial. SimSpace Weaver notifica a la aplicación espacial todos los eventos de la entidad (por ejemplo, entrada, salida, creación, actualización y eliminación) que se produzcan en el área de suscripción.



La visión del mundo de una aplicación espacial



La vista de una aplicación espacial con un área de suscripción adicional

La visión del mundo de una aplicación espacial es su área de propiedad, que es una celda de la cuadrícula mundial.

Una aplicación espacial utiliza una suscripción para averiguar qué sucede en otra parte del mundo de la simulación. El área de suscripción puede contener varias celdas de cuadrícula y partes de celdas.

Por ejemplo, una aplicación que simula entidades que interactúan físicamente puede necesitar información sobre las entidades que se encuentran más allá de los límites espaciales de su área de propiedad. Para ello, la aplicación puede suscribirse a las áreas que bordean su área de propiedad. Tras crear la suscripción, la aplicación recibe notificaciones sobre los eventos de las entidades en esas áreas y puede leerlas. Otro ejemplo es un vehículo autónomo que necesita ver a todas las entidades situadas a 200 metros de distancia, independientemente de la aplicación que posea la zona. La aplicación para el vehículo puede crear una suscripción con un filtro en forma de cuadro delimitador alineado con los ejes que (AABB) cubre el área visible.

Puede crear una lógica de simulación que no se encargue de gestionar los aspectos espaciales de la simulación. Una aplicación personalizada es un programa ejecutable que se ejecuta en un único dispositivo de trabajo. Usted controla el ciclo de vida (inicio y finalización) de una aplicación personalizada. Los clientes de simulación pueden conectarse a una aplicación personalizada para ver la simulación o interactuar con ella. También puede crear una aplicación de servicio que se ejecute en todos los trabajadores. SimSpace Weaver inicia una instancia de la aplicación de servicio en cada trabajador que ejecuta la simulación.

Las aplicaciones personalizadas y las aplicaciones de servicio crean suscripciones para obtener información sobre los eventos de las entidades y leer las entidades. Estas aplicaciones no tienen áreas de propiedad porque no son espaciales. El uso de una suscripción es la única forma de que puedan enterarse de lo que sucede en el mundo de la simulación.

Cómo se usa SimSpace Weaver

Cuando usa SimSpace Weaver, estos son los pasos principales que debe seguir:

1. Escribir y crear aplicaciones de C++ que integren el SDK de la aplicación SimSpace Weaver.
 - a. Sus aplicaciones realizan llamadas a la API para interactuar con el estado de la simulación.
2. Cree clientes que vean su simulación e interactúen con ella a través de algunas aplicaciones.
3. Configure la simulación en un archivo de texto.
4. Cargue los paquetes de aplicaciones y la configuración de la simulación en el servicio.

5. Inicie la simulación.
6. Inicie y detenga sus aplicaciones personalizadas según sea necesario.
7. Conecte a los clientes con sus aplicaciones personalizadas o de servicio para ver la simulación o interactuar con ella.
8. Compruebe los registros de simulación en los registros de Amazon CloudWatch.
9. Detenga la simulación.
10. Limpie la simulación.

Esquema de simulación

El esquema de simulación (o esquema) es un archivo de texto con formato YAML que contiene información de configuración para la simulación. SimSpace Weaver utiliza el esquema al iniciar una simulación. El paquete distribuible del SDK de la aplicación SimSpace Weaver incluye un esquema para un proyecto de muestra. Puede utilizar este cliente como punto de partida para sus propias aplicaciones. Para obtener más información sobre el esquema de simulación, consulte [SimSpace Weaver referencia del esquema de simulación](#).

Trabajadores y unidades de recursos

Un trabajador es una instancia de Amazon EC2 que ejecuta la simulación. El tipo de trabajador se especifica en el esquema de simulación. SimSpace Weaver asigna su tipo de trabajador a un tipo de instancia de Amazon EC2 específico que utilice el servicio. SimSpace Weaver inicia y detiene a sus trabajadores por usted y gestiona la comunicación de red entre los trabajadores. SimSpace Weaver inicia un conjunto de trabajadores para cada simulación. Las diferentes simulaciones utilizan diferentes trabajadores.

La capacidad informática disponible (procesador y memoria) de un trabajador se divide en unidades lógicas denominadas unidades de recursos informáticos (o unidades de recursos). Una unidad de recursos representa una cantidad fija de capacidad de procesador y memoria.

Note

Anteriormente, denominábamos ranura a una unidad de recursos de cómputo. Es posible que aún vea este término anterior en nuestra documentación.

Reloj de simulación

Cada simulación tiene su propio reloj. El reloj se inicia y se detiene mediante llamadas a la API o la consola SimSpace Weaver. La simulación se actualiza solo cuando el reloj está en marcha. Todas las operaciones de la simulación se producen dentro de segmentos de tiempo denominados graduaciones. El reloj anuncia la hora de inicio de cada graduación a todos los trabajadores.

La frecuencia del reloj (o frecuencia de graduación) es el número de pulsaciones por segundo (hercios o Hz) que anuncia el reloj. La frecuencia de reloj deseada para una simulación forma parte del esquema de simulación. Todas las operaciones de una marca deben completarse antes de que comience la siguiente. Por este motivo, la frecuencia de reloj efectiva puede ser inferior a la frecuencia de reloj deseada. La frecuencia de reloj efectiva no será superior a la frecuencia de reloj deseada.

Particiones

Una partición es un segmento de la memoria compartida de un trabajador. Cada partición contiene parte de los datos de estado de la simulación.

Una partición de una aplicación espacial (también denominada partición de aplicación espacial o partición espacial) contiene todas las entidades del área de propiedad de una aplicación espacial. SimSpace Weaver coloca las entidades en las particiones de aplicaciones espaciales en función de la ubicación espacial de cada entidad. Esto significa que SimSpace Weaver intenta colocar las entidades que están espacialmente cerca unas de otras en el mismo elemento de trabajo. Esto minimiza la cantidad de conocimiento que una aplicación requiere sobre las entidades que no son de su propiedad para simular las entidades que sí son de su propiedad.

State Fabric

State Fabric es el sistema de memoria compartida (la colección de todas las particiones) entre todos los trabajadores. Contiene todos los datos de estado de la simulación.

State Fabric utiliza un formato binario personalizado que describe una entidad como un conjunto de datos iniciales y un registro de actualizaciones para cada campo de datos de esa entidad. Con este formato, puede acceder al estado de una entidad en un punto anterior del tiempo de simulación y volver a asignarlo a un punto en el tiempo real. El búfer tiene un tamaño finito y no es posible retroceder en el tiempo más allá de lo que hay en el búfer. SimSpace Weaver usa un puntero hacia el desplazamiento actual en el registro de actualizaciones de cada campo y actualiza un

puntero como parte de una actualización de campo. SimSpace Weaver asigna estos registros de actualización al espacio de procesos de una aplicación mediante la memoria compartida.

Este formato de objeto reduce los gastos generales y no supone costes de serialización. SimSpace Weaver también utiliza este formato de objeto para analizar e identificar los campos de índice (como la posición de la entidad).

Entidades

Una entidad es el bloque de datos más pequeño de la simulación. Algunos ejemplos de entidades son los actores (como personas y vehículos) y los objetos estáticos (como edificios y obstáculos). Las entidades tienen propiedades (como la posición y la orientación) que se pueden almacenar como datos persistentes en SimSpace Weaver. Las entidades existen dentro de las particiones.

móviles

Una aplicación de SimSpace Weaver es un software que se escribe y que contiene una lógica personalizada que ejecuta cada graduación de la simulación. El objetivo de la mayoría de las aplicaciones es actualizar las entidades a medida que se ejecuta la simulación. Sus aplicaciones utilizan las API del SDK de la aplicación SimSpace Weaver para realizar acciones (como leer y actualizar) en las entidades de la simulación.

Las aplicaciones y los recursos necesarios (como las bibliotecas) se empaquetan en archivos.zip y se cargan en SimSpace Weaver. Una aplicación se ejecuta en un contenedor de Docker en un dispositivo de trabajo. SimSpace Weaver asigna a cada aplicación un número fijo de unidades de recursos al trabajador.

SimSpace Weaver asigna la propiedad de una (y solo una) partición a cada aplicación. Una aplicación y su partición se encuentran en el mismo elemento de trabajo. Cada partición solo tiene un propietario de aplicación. Una aplicación puede crear, leer, actualizar y eliminar entidades en su partición. Una aplicación es propietaria de todas las entidades de su partición.

Hay tres tipos de aplicaciones: aplicaciones espaciales, aplicaciones personalizadas y aplicaciones de servicio. Se diferencian según los casos de uso y los ciclos de vida.

Note

En SimSpace Weaver, el término «aplicación» puede hacer referencia al código de una aplicación o a una instancia en ejecución de ese código.

Aplicaciones espaciales

Las aplicaciones espaciales actualizan el estado de las entidades que existen espacialmente en la simulación. Por ejemplo, puede definir una aplicación de `Physics` que se encargue de mover y colisionar entidades para cada tic en función de su velocidad, forma y tamaño. En este caso, SimSpace Weaver ejecuta varias instancias de la aplicación `Physics` en paralelo para gestionar el tamaño de la carga de trabajo.

SimSpace Weaver gestiona el ciclo de vida de las aplicaciones espaciales. Usted especifica una disposición de las particiones de las aplicaciones espaciales en el esquema de simulación. Al iniciar la simulación, SimSpace Weaver inicia una aplicación espacial para cada partición de la aplicación espacial. Al detener la simulación, SimSpace Weaver cierra las aplicaciones espaciales.

Otros tipos de aplicaciones pueden crear entidades, pero solo las aplicaciones espaciales pueden actualizar entidades. Otros tipos de aplicaciones deben transferir las entidades que crean a un dominio espacial. SimSpace Weaver usa la ubicación espacial de una entidad para moverla a la partición de una aplicación espacial. Esto transfiere la propiedad de la entidad a la aplicación espacial.

Aplicaciones personalizadas

Utiliza aplicaciones personalizadas para interactuar con la simulación. Una aplicación personalizada lee los datos de la entidad mediante suscripciones. Una aplicación personalizada puede crear entidades. Sin embargo, la aplicación debe transferir una entidad a una aplicación espacial para incluirla en la simulación y actualizarla. Puede asignar a SimSpace Weaver un punto de conexión de red a una aplicación personalizada. Los clientes de simulación pueden conectarse al punto de conexión de la red para interactuar con la simulación. Usted define sus aplicaciones personalizadas en su esquema de simulación, pero es responsable de iniciarlas y detenerlas (mediante llamadas a la API de SimSpace Weaver). Después de iniciar una instancia de aplicación personalizada en un trabajador, SimSpace Weaver no transfiere la instancia a otro trabajador.

Aplicaciones de servicio

Puede usar una aplicación de servicio cuando necesite ejecutar un proceso de solo lectura en todos los trabajadores. Por ejemplo, puede usar una aplicación de servicio si tiene una simulación grande y necesita un cliente de visualización que recorra la simulación y muestre solo las entidades visibles al usuario. En este caso, una sola instancia de aplicación personalizada no puede procesar todas las entidades de la simulación. Puede configurar una aplicación de servicio para que se ejecute en

todos los trabajadores. Luego, cada una de estas aplicaciones de servicio puede filtrar las entidades de su trabajador asignado y enviar solo las entidades relevantes a sus clientes conectados. Luego, su cliente de visualización puede conectarse a diferentes aplicaciones de servicio a medida que avanza por el espacio de simulación. Las aplicaciones de servicio se configuran en el esquema de simulación. SimSpace Weaver inicia y detiene las aplicaciones de servicio por usted.

Resumen de las API

En la tabla siguiente se resumen las características de los distintos tipos de aplicaciones SimSpace Weaver.

	Aplicaciones espaciales	Aplicaciones personalizadas	Aplicaciones de servicio
Entidades de lectura	Sí	Sí	Sí
Entidades de actualización	Sí	No	No
Creación de entidades	Sí	Sí*	Sí*
Ciclo de vida	Administrado (SimSpace Weaver lo controla).	No administrado (usted lo controla).	Administrado (SimSpace Weaver lo controla).
Método de inicio	SimSpace Weaver inicia una instancia de aplicación para cada partición espacial, tal y como se especifica en el esquema.	Usted inicia cada instancia de aplicación.	SimSpace Weaver inicia una o más instancias de aplicación en cada dispositivo de trabajo, tal y como se especifica en el esquema.
Los clientes pueden establecer conexión	No	Sí	Sí

* Cuando una aplicación o aplicación de servicio personalizada crea una entidad, la aplicación debe transferir la propiedad de la entidad a una aplicación espacial para que la aplicación espacial pueda actualizar el estado de la entidad.

Dominios

Un dominio de SimSpace Weaver es un conjunto de instancias de aplicación que ejecutan el mismo código de aplicación ejecutable y tienen las mismas opciones y comandos de inicio. Nos referimos a los dominios por los tipos de aplicaciones que contienen: dominios espaciales, dominios personalizados y dominios de servicio. Las aplicaciones se configuran dentro de los dominios.

Suscripciones y replicación

Una aplicación crea una suscripción a una región espacial para conocer los eventos de las entidades (por ejemplo, entrar, salir, crear, actualizar y eliminar) en esa región. Una aplicación procesa los eventos de entidades de una suscripción antes de leer los datos de las entidades de particiones que no son de su propiedad.

Una partición puede existir en el mismo servidor que la aplicación (esto se denomina partición local), pero otra aplicación puede ser propietaria de la partición. Una partición también puede existir en un equipo de trabajo diferente (esto se denomina partición remota). Si la suscripción es a una partición remota, el trabajador crea una copia local de la partición remota mediante un proceso denominado replicación. A continuación, el trabajador lee la copia local (partición remota replicada). Si otra aplicación del trabajador necesita leer desde esa partición con la misma marca, el trabajador lee la misma copia local.

Ejemplos de casos de uso de SimSpace Weaver

Puede utilizar SimSpace Weaver para modelos basados en agentes y simulaciones de intervalos de tiempo discretos con un componente espacial.

Crear simulaciones de grandes multitudes

Puede utilizar SimSpace Weaver para simular multitudes en entornos del mundo real. SimSpace Weaver le permite escalar sus simulaciones a millones de objetos dinámicos con sus propios comportamientos.

Cree entornos a escala urbana

Use SimSpace Weaver para crear un gemelo digital de una ciudad entera. Cree simulaciones para la planificación urbana, el diseño de las rutas de tráfico y la planificación de la respuesta a los peligros ambientales. Puede utilizar sus propios orígenes de datos geoespaciales como componentes básicos de sus entornos.

Crear experiencias inmersivas e interactivas

Cree experiencias de simulación en las que puedan participar e interactuar varios usuarios. Use herramientas de desarrollo populares, como Unreal Engine y Unity, para crear mundos virtuales tridimensionales (3D). Personalice su experiencia 3D con su propio contenido y comportamiento.

Configuración de SimSpace Weaver

Para configurar SimSpace Weaver para usarlo por primera vez, debe configurar su Cuenta de AWS y su entorno local. Cuando haya terminado estas tareas, podrá pasar a los [tutoriales de introducción](#).

Tareas de configuración

1. [Configura tu dispositivo Cuenta de AWS para usar SimSpace Weaver](#).
2. [Configuración del entorno de SimSpace Weaver](#).

Configura tu dispositivo Cuenta de AWS para usar SimSpace Weaver

Complete las siguientes tareas para configurar su Cuenta de AWS uso SimSpace Weaver.

Inscríbase en una Cuenta de AWS

Si no tiene una Cuenta de AWS, complete los siguientes pasos para crearlo.

Para suscribirte a una Cuenta de AWS

1. Abrir <https://portal.aws.amazon.com/billing/registro>.
2. Siga las instrucciones que se le indiquen.

Parte del procedimiento de registro consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

Cuando te registras en una Cuenta de AWS, Usuario raíz de la cuenta de AWS se crea un. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar [tareas que requieren acceso de usuario raíz](#).

AWS te envía un correo electrónico de confirmación una vez finalizado el proceso de registro. En cualquier momento, puede ver la actividad de su cuenta actual y administrarla accediendo a <https://aws.amazon.com/> y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de crear un usuario administrativo Cuenta de AWS, asegúrelo Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilite y cree un usuario administrativo para no usar el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión [AWS Management Console](#) como propietario de la cuenta seleccionando el usuario root e introduciendo su dirección de Cuenta de AWS correo electrónico. En la siguiente página, escriba su contraseña.

Para obtener ayuda para iniciar sesión con el usuario raíz, consulte [Iniciar sesión como usuario raíz](#) en la Guía del usuario de AWS Sign-In .

2. Activa la autenticación multifactorial (MFA) para tu usuario root.

Para obtener instrucciones, consulte [Habilitar un MFA dispositivo virtual para el usuario Cuenta de AWS root \(consola\)](#) en la Guía del IAM usuario.

Creación de un usuario con acceso administrativo

1. Habilite IAM Identity Center.

Consulte las instrucciones en [Activar AWS IAM Identity Center](#) en la Guía del usuario de AWS IAM Identity Center .

2. En IAM Identity Center, conceda acceso administrativo a un usuario.

Para ver un tutorial sobre cómo usar el Directorio de IAM Identity Center como fuente de identidad, consulte [Configurar el acceso de los usuarios con la configuración predeterminada Directorio de IAM Identity Center](#) en la Guía del AWS IAM Identity Center usuario.

Iniciar sesión como usuario con acceso de administrador

- Para iniciar sesión con su usuario de IAM Identity Center, utilice el inicio de sesión URL que se envió a su dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

Para obtener ayuda para iniciar sesión con un usuario de IAM Identity Center, consulte [Iniciar sesión en el portal de AWS acceso](#) en la Guía del AWS Sign-In usuario.

Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos con privilegios mínimos.

Para conocer las instrucciones, consulte [Create a permission set](#) en la Guía del usuario de AWS IAM Identity Center .

2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

Para conocer las instrucciones, consulte [Add groups](#) en la Guía del usuario de AWS IAM Identity Center .

Agregar permisos para usar SimSpace Weaver

Para dar acceso, agregue permisos a los usuarios, grupos o roles:

- Usuarios y grupos en AWS IAM Identity Center:

Cree un conjunto de permisos. Siga las instrucciones de [Creación de un conjunto de permisos](#) en la Guía del usuario de AWS IAM Identity Center .

- Usuarios gestionados IAM a través de un proveedor de identidad:

Cree un rol para la federación de identidades. Siga las instrucciones de la Guía del IAM usuario sobre cómo [crear un rol para un proveedor de identidades externo \(federación\)](#).

- IAMusuarios:

- Cree un rol que el usuario pueda aceptar. Siga las instrucciones de la Guía del [IAMusuario sobre cómo crear un rol para un](#) IAM usuario.
- (No recomendado) Adjunte una política directamente a un usuario o añada un usuario a un grupo de usuarios. Siga las instrucciones de [Añadir permisos a un usuario \(consola\)](#) de la Guía del IAM usuario.

Example IAMpolítica para conceder permisos de uso 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"
      }
    }
  }
]
}

```


Configuración del entorno de SimSpace Weaver

Las simulaciones SimSpace Weaver se ejecutan en entornos contenerizados de Amazon Linux 2 (AL2). Debe tener un entorno AL2 para compilar y vincular sus aplicaciones con el SDK de la aplicación SimSpace Weaver. El entorno de desarrollo local estándar es un contenedor AL2 en Docker. Si decide no usar Docker, le proporcionaremos instrucciones alternativas para ejecutar un entorno AL2 en Windows Subsystem for Linux (WSL). También puede usar su propio método para crear un entorno AL2 en la zona. Para ver otras formas de ejecutar AL2 localmente, consulte la [documentación de Amazon EC2](#).

Important

Docker en Microsoft Windows es el entorno de desarrollo estándar. Para su comodidad, le sugerimos otras formas de configurar su entorno de desarrollo local, pero no son estándares y no son compatibles.

Temas

- [Configure el paquete SimSpace Weaver de distribución para Amazon Linux 2 \(AL2\) en Docker](#)
- [Configure el paquete SimSpace Weaver de distribución para Amazon Linux 2 \(AL2\) en Windows Subsystem for Linux \(WSL\)](#)

Configure el paquete SimSpace Weaver de distribución para Amazon Linux 2 (AL2) en Docker

En esta sección se proporcionan instrucciones para configurar el código postal de SimSpace Weaver distribución local con un entorno AL2 incorporado. Docker Para obtener instrucciones de configuración con una entrada AL2Windows Subsystem for Linux (WSL), consulte. [Configure el paquete SimSpace Weaver de distribución para Amazon Linux 2 \(AL2\) en Windows Subsystem for Linux \(WSL\)](#)

Requisitos

- Microsoft Windows 10 o superior, o un sistema Linux compatible
- [Microsoft Visual Studio 2019](#) o superior, con la carga de trabajo de [Desktop development with C++](#) instalada
- [CMake3](#)

- [Git](#)
- [Docker Desktop](#)
- [AWS CLI](#)
- [Python 3.9](#)

Para configurar la SimSpace Weaver distribución, comprima con AL2 en Docker

1. Si aún no ha configurado sus AWS credenciales para el AWS CLI, siga estas instrucciones: [Configuración de la AWS CLI](#).
2. [Descargue el paquete distribuible del SDK de la SimSpace Weaver aplicación](#). Contiene lo siguiente:
 - Binarios y bibliotecas para SimSpace Weaver el desarrollo de aplicaciones
 - Scripts auxiliares que automatizan partes del flujo de trabajo de desarrollo
 - Ejemplos de aplicaciones que muestran conceptos SimSpace Weaver
3. Descomprima el archivo en una *sdk-folder* de su elección.
4. Acceda a *sdk-folder*.
5. Introduzca el siguiente comando para instalar los paquetes de Python necesarios:

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

6. Ingresa el siguiente comando para configurar la SimSpace Weaver distribución con una imagen de Docker.

```
python setup.py
```

Este comando hace lo siguiente:

- Crea una imagen de docker AL2 con todos los requisitos para proyectos de SimSpace Weaver construcción instalados.
- Crea los CloudFormation recursos necesarios para lanzar una simulación.
 - La plantilla de CloudFormation pila de muestras se encuentra en *sdk-folder*/
`PackagingTools/sample-stack-template.yaml`
- Configura los proyectos de muestra proporcionados con las rutas correctas para su sistema local.

Resolución de problemas

- Parece que Docker está atascado
 - Si la salida de la consola parece estar bloqueada después de ejecutar los comandos de Docker, intente reiniciar el motor de Docker. Si eso no funciona, reinicia el ordenador.

Configure el paquete SimSpace Weaver de distribución para Amazon Linux 2 (AL2) en Windows Subsystem for Linux (WSL)

En esta sección se proporcionan instrucciones para configurar el archivo zip SimSpace Weaver de distribución con un entorno AL2 incorporado. Windows Subsystem for Linux (WSL) Para obtener instrucciones para configurar AL2 en Docker, consulte [Configure el paquete SimSpace Weaver de distribución para Amazon Linux 2 \(AL2\) en Docker](#).

Important

En esta sección se describe una solución que utiliza una versión de AL2 que no es propiedad, desarrollada ni compatible con Amazon. Esta solución se proporciona únicamente para su comodidad, si decide no utilizar Docker. Amazon y no AWS asumimos ninguna responsabilidad si decides utilizar esta solución.

Requisitos

- [Hyper-V del Windows 10](#)
- [Windows Subsystem for Linux \(WSL\)](#)
- Distribución AL2 de código abierto de terceros para WSL ([descargue la versión 2.0.20200722.0-update.2](#)) (consulte las [instrucciones](#))

Important

Nuestras instrucciones de WSL utilizan la versión [2.0.20200722.0-update.2](#) de la distribución AL2 para WSL. Es posible que se produzcan errores si utiliza cualquier otra versión.

Para configurar el zip de SimSpace Weaver distribución, introduzca AL2 WSL

1. En una línea de comandos de Windows, inicie el entorno AL2 en WSL.

```
wsl -d Amazon2
```

 Important

Mientras esté ejecutando WSL, incluya la `--al2` opción cuando ejecute uno de los scripts auxiliares de `quick-start.py` Python que se encuentran en `ensdky-folder/Samples/sample-name/tools/cloud/quick-start.py`.

2. En un intérprete de comandos de Linux, actualice su administrador de paquetes yum.

```
yum update -y
```

 Important

Si se agota el tiempo de espera de este paso, puede que tenga que cambiar a WSL1 y volver a intentar estos procedimientos. Cierre la sesión de AL2 de WSL e introduzca lo siguiente en la línea de comandos de Windows:

```
wsl --set-version Amazon2 1
```

3. Instale la herramienta de descompresión.

```
yum install -y unzip
```

4. Elimine los AWS CLI que estén yum instalados. Pruebe los dos comandos siguientes si no está seguro de si yum ha instalado un AWS CLI.

```
yum remove awscli
```

```
yum remove aws-cli
```

5. Cree un directorio temporal y acceda a él.

```
mkdir ~/temp
```

```
cd ~/temp
```

6. Descargue e instale AWS CLI:

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

7. Podrá eliminar el directorio temporal.

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

8. Reinicie la sesión del intérprete de comandos para actualizar la ruta en el entorno.

```
exec
```

9. Configure sus AWS credenciales para su entorno AL2. AWS CLI Para obtener más información, consulte [Configuración de AWS CLI](#). Si las usa AWS IAM Identity Center, consulte [Configurar las AWS CLI que se van a usar AWS IAM Identity Center](#) en la Guía del AWS Command Line Interface usuario.

```
aws configure
```

10. Instale Git

```
yum install -y git
```

11. Instale wget.

```
yum install -y wget
```

12. Cree una carpeta para el SDK de la SimSpace Weaver aplicación.

```
mkdir sdk-folder
```

13. Acceda a su carpeta de SDK.

```
cd sdk-folder
```

14. Descarga el paquete distribuible del SDK de la SimSpace Weaver aplicación. Contiene lo siguiente:

- Binarios y bibliotecas para SimSpace Weaver el desarrollo de aplicaciones
- Scripts auxiliares que automatizan partes del flujo de trabajo de desarrollo
- Ejemplos de aplicaciones que muestran conceptos SimSpace Weaver

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

15. Descomprima el archivo.

```
unzip *.zip
```

16. Ejecute el script de configuración de WSL.

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

17. Introduzca el siguiente comando para instalar los paquetes de Python necesarios:

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

18. Ejecute el script de configuración zip de SimSpace Weaver distribución:

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

Este comando hace lo siguiente:

- Crea los CloudFormation recursos necesarios para lanzar una simulación.
 - La plantilla de CloudFormation pila de muestras se encuentra en *sdk-folder/PackagingTools/sample-stack-template.yaml*
- Configura los proyectos de muestra proporcionados con las rutas correctas para su sistema local.

Note

Solo tiene que hacerlo una vez para su entorno AL2 en WSL.

Uso de software con licencia con AWS SimSpace Weaver

AWS SimSpace Weaver le permite crear simulaciones con el motor de simulación y el contenido que elija. En relación con su uso de SimSpace Weaver, usted es responsable de obtener, mantener y cumplir los términos de la licencia de cualquier software o contenido que utilice en sus simulaciones. Compruebe que el acuerdo de licencia le permite implementar el software y el contenido en un entorno de alojamiento virtual.

Empezar con SimSpace Weaver

En esta sección se proporcionan tutoriales que le ayudarán a empezar SimSpace Weaver. Estos tutoriales le presentan el flujo de trabajo general para crear simulaciones SimSpace Weaver. Estos tutoriales muestran cómo crear, implementar y ejecutar simulaciones en SimSpace Weaver. Le recomendamos que comience con el tutorial de inicio rápido para ejecutar una simulación en cuestión de minutos. Consulte los otros tutoriales posteriores para obtener más información.

Estos tutoriales utilizan una aplicación (PathfindingSample) de muestra incluida en el archivo.zip del SDK de la SimSpace Weaver aplicación que descargaste durante los [procedimientos de configuración](#). La aplicación de ejemplo muestra los conceptos que comparten todas las SimSpace Weaver simulaciones, como la partición espacial, la transferencia de entidades entre particiones, las aplicaciones y las suscripciones.

En los tutoriales, creará una simulación con cuatro particiones espaciales. Una instancia independiente de la aplicación espacial de PathfindingSample gestiona cada partición individual. Las aplicaciones espaciales crean entidades en sus propias particiones. Las entidades se mueven a una posición determinada en el mundo de la simulación, evitando los obstáculos a medida que se mueven. Puede usar una aplicación cliente independiente (incluida en el SDK de la SimSpace Weaver aplicación) para ver la simulación.

Temas

- [Tutorial de inicio rápido para SimSpace Weaver](#)
- [Tutorial detallado: conozca los detalles mientras crea la aplicación de muestra](#)

Tutorial de inicio rápido para SimSpace Weaver

Este tutorial le guía a través del proceso para crear y ejecutar una simulación en cuestión de SimSpace Weaver minutos. Le recomendamos que comience con este tutorial y luego continúe con el [tutorial detallado](#).

Requisitos

Antes de comenzar, asegúrese de que ha realizado los pasos que se detallan en [Configuración de SimSpace Weaver](#).

Note

Los scripts que se utilizan aquí se proporcionan para su comodidad y NO son obligatorios. Consulte el [tutorial detallado](#) para saber cómo se pueden realizar estos pasos manualmente.

Paso 1: Habilitar el registro (opcional)

Para activar el registro

1. Navegue hasta:

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

2. Abra el archivo de esquema en un editor de texto:

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

3. Busque la sección `simulation_properties:` al principio del archivo:

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

4. Inserte las dos líneas siguientes después de la línea `simulation_properties::`

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

5. Confirme que su sección `simulation_properties:` es como sigue:

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

6. Guarde el archivo y salga del editor de texto.

Paso 2: Inicio rápido con el cliente de consola (opción 1)

Navegue hasta:

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

Ejecute uno de los siguientes comandos:

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

De forma predeterminada, se iniciará una simulación con una sola partición en un solo equipo de trabajo. Se pueden iniciar otras configuraciones transfiriéndolas `--schema {file name}.yaml` desde la `/Samples/PathfindingSample/tools/` carpeta.

Note

Consulte [Tutorial detallado: conozca los detalles mientras crea la aplicación de muestra](#) para obtener una explicación detallada de lo que hace este script.

Paso 2: Comience rápidamente con el cliente de Unreal Engine (opción 2)

Consulte [Lanzamiento del cliente de visualización de Unreal Engine](#).

Detenga y elimine su simulación

Navegue hasta:

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

Busque los nombres de sus simulaciones:

```
aws simspaceweaver list-simulations
```

Detenga y elimine la simulación

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

Resolución de problemas

- FileNotFoundError: cmake

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

- **Resolución:** el script no puede encontrar el comando `cmake`. Asegúrese de tener instalada la versión mínima recomendada de CMake y de que se pueda llamar a ella con el `cmake` comando de la ruta. Usa el comando `cmake -version` para verificarlo.
- **ImportError:** No se pudo cargar la DLL al importar `libweaver_app_sdk_python_v1`: no se encontró el módulo especificado.
 - **Solución:** este error se produce cuando Python 3.9 no se utiliza para iniciar el SDK de Python de Weaver. Asegúrese de que la versión de Python asociada al comando «python» sea Python 3.9. Puede comprobarlo ejecutando el `python --version` comando.
- El script de inicio rápido aparece bloqueado después de iniciar Docker Build.
 - **Solución:** a veces, Docker necesita unos minutos para calentarse. Si el problema persiste durante más de aproximadamente 5 minutos, reinicie Docker o el sistema.
- `target_compile_features` no tiene ninguna función conocida del compilador CXX «GNU»:
 - **Solución:** borre la memoria caché de Docker, elimine la imagen de Docker de `weaverappbuilder`, elimine los artefactos de creación de su proyecto y vuelva a ejecutarlo. `setup.py` Esto debería restablecer el entorno de Docker y resolver el error.

Tutorial detallado: conozca los detalles mientras crea la aplicación de muestra

El [tutorial de inicio rápido](#) explicaba cómo crear, iniciar, detener y eliminar un ejemplo de simulación mediante `quick-start.py` y `ystop-and-delete.py`. Este tutorial explicará en detalle cómo funcionan estos scripts y los parámetros adicionales que estos scripts pueden adoptar para maximizar la flexibilidad de las simulaciones personalizadas de Weaver.

Requisitos

Antes de comenzar, asegúrese de que ha realizado los pasos que se detallan en [Configuración de SimSpace Weaver](#).

Paso 1: Habilitar el registro (opcional)

Para activar el registro

1. Navegue hasta:

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

2. Abra el archivo de esquema en un editor de texto:

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

3. Busque la sección `simulation_properties:` al principio del archivo:

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

4. Inserte las dos líneas siguientes después de la línea `simulation_properties::`

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

5. Confirme que su sección `simulation_properties:` es como sigue:

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

6. Guarde el archivo y salga del editor de texto.

Paso 2: Inicie la simulación

Como se muestra en el [tutorial de inicio rápido](#), los pasos más básicos para lanzar un ejemplo de simulación son:

1. Navegue hasta:

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

2. Ejecute uno de los siguientes comandos:

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

Este script automatiza los comandos de terminal más comunes, todos los cuales se pueden ejecutar manualmente mediante el AWS CLI. Estos pasos son:

1. Cargue el esquema de Weaver en S3.
 - SimSpace Weaver utiliza un esquema para configurar la simulación. El esquema es un archivo de texto plano con formato YAML. Para obtener más información, consulte [Configuración de su simulación](#).
2. Crea y carga un contenedor personalizado (opcional).
 - Si su esquema define un contenedor personalizado, el script de inicio rápido compilará la imagen del docker y la cargará en Amazon ECR. Para obtener más información, consulte [Contenedores personalizados](#). Consulte el `PythonBubblesSample` esquema para ver un ejemplo de esta función.
3. Compilar el proyecto.
 - `quick-start.py` llama a la `build_project` función definida en `build.py`. Este paso variará en función del proyecto. Para ello `PathfindingSample`, se utiliza CMake. Los comandos CMake y Docker se encuentran en `build.py`
4. Sube los artefactos de construcción a S3.
 - Puedes comprobar tus depósitos de S3 para asegurarte de que todas las cargas se han realizado correctamente. Para obtener más información sobre cómo usar Amazon S3, consulte [Creación, configuración y trabajo con buckets de Amazon S3](#) en la Guía del usuario de la consola de Amazon Simple Storage Service.
 - Los archivos zip de la aplicación de muestra y el bucket de S3 utilizan el siguiente formato de nombre:
 - `weaver-sample-bucket-account-number-region`
 - Aplicación espacial: `ProjectNameSpatial.zip`
 - Ver aplicación (personalizada): `ProjectNameView.zip`
5. Inicie la simulación.

- Esto es un resumen de la `aws simspaceweaver start-simulation` AWS CLI llamada. Para obtener más información, consulte la [Referencia de AWS CLI comandos de SimSpace Weaver](#).
- El script se repetirá hasta que el estado de la simulación sea `STARTED` o `FAILED`. Puede que transcurran unos minutos hasta que la simulación se inicie.

6. Obtenga los detalles de la simulación.

- La API de `DescribeSimulation` proporciona detalles sobre la simulación, incluido su estado. Una simulación puede tener uno de los siguientes estados:

Estados del ciclo de vida de la simulación

1. **STARTING**: estado inicial después de llamar a `StartSimulation`
2. **STARTED**: todas las aplicaciones espaciales están iniciadas y en buen estado
3. **STOPPING**: estado inicial después de llamar a `StopSimulation`
4. **STOPPED**: todos los recursos de cómputo están detenidos
5. **DELETING**: estado inicial después de llamar a `DeleteSimulation`
6. **DELETED**: se eliminan todos los recursos asignados a la simulación
7. **FAILED**: la simulación ha experimentado un error o fallo crítico y se ha detenido
8. **SNAPSHOT_IN_PROGRESS**: hay una [instantánea](#) en curso

Para obtener los detalles de la simulación

1. Llame a la API de `ListSimulations`.

```
aws simspaceweaver list-simulations
```

El script debe mostrar detalles sobre cada una de sus simulaciones, como se muestra a continuación:

```
{
  "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. Llame a `DescribeSimulation` para obtener los detalles de su simulación. Reemplace *simulation-name* con el Name de su simulación del resultado del paso anterior.

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

El script debe mostrar más detalles sobre la simulación que ha especificado, como se muestra a continuación:

```
{
  "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. Inicie aplicaciones personalizadas.

- SimSpace Weaver no gestiona el ciclo de vida de las aplicaciones personalizadas. Debe iniciar sus aplicaciones personalizadas. Se recomienda iniciar las aplicaciones personalizadas antes de iniciar el reloj de simulación, pero puede iniciar las aplicaciones personalizadas después de encender el reloj.

Puedes llamar a la `StartApp` API para iniciar tus aplicaciones personalizadas.

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

La llamada a la API de `StartApp` creará e iniciará una nueva instancia de la aplicación personalizada con el nombre que proporcione. Si proporciona el nombre de una aplicación que ya existe, recibirá un error. Si quiere reiniciar una aplicación (instancia) en particular, primero debe detenerla y eliminarla.

Note

El estado de la simulación debe estar STARTED antes de que pueda iniciar aplicaciones personalizadas.

La aplicación de ejemplo proporciona la aplicación personalizada ViewApp para ver la simulación. Esta aplicación le proporciona una dirección IP estática y un número de puerto para conectar los clientes de simulación (lo hará en un paso posterior de este tutorial). Puede pensar en un domain como una clase de aplicaciones que tienen el mismo código ejecutable y las mismas opciones de inicio. app name identifica la instancia de la aplicación. Para obtener más información sobre SimSpace Weaver los conceptos, consulte [Conceptos clave de SimSpace Weaver](#).

Puede utilizar la API de DescribeApp para comprobar el estado de una aplicación personalizada después de iniciarla.

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

Para iniciar la aplicación de visualización en este tutorial

1. Llame StartApp para ViewApp.

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

2. Llame a DescribeApp para comprobar el estado de la aplicación personalizada.

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

Cuando el estado de la aplicación personalizada (instancia) sea STARTED, el resultado de DescribeApp incluirá la dirección IP y el número de puerto de esa aplicación personalizada (instancia). En el siguiente ejemplo de salida, la dirección IP es el valor del bloque Address y el número de puerto es el valor Actual del bloque EndpointInfo.


```
{
  "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


El valor de `Declared` es el número de puerto al que debe enlazarse el código de la aplicación. El valor de `Actual` es el número de puerto al que se SimSpace Weaver exponen los clientes para que se conecten a tu aplicación. SimSpace Weaver asigna el `Declared` puerto al `Actual` puerto.

Note

Puede usar el procedimiento descrito en [Obtenga la dirección IP y el número de puerto de una aplicación personalizada](#) para obtener la dirección IP y el número de puerto de cualquier aplicación personalizada iniciada.

8. Arranca el reloj.


- Cuando crea la simulación por primera vez, tiene un reloj, pero el reloj no está funcionando. Cuando el reloj no esté funcionando, la simulación no actualizará su estado. Cuando encienda el reloj, empezará a enviar graduaciones a sus aplicaciones. Con cada movimiento, tus aplicaciones espaciales recorren las entidades que poseen y con las que archivan los resultados SimSpace Weaver

 Note

El reloj puede tardar entre 30 y 60 segundos en ponerse en marcha.

Llame a la API de StartClock.

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

 Note

La API de StartClock usa su *simulation-name*, que puede encontrar en la API de ListSimulations:

```
aws simspaceweaver list-simulations
```

parámetros de inicio rápido

- -h, --help
 - Enumere estos parámetros.
- --limpiar
 - Elimine el contenido del directorio de compilación antes de compilar.
- --al2
 - Se basa directamente en la máquina nativa en lugar de en Docker. Úselo únicamente si se ejecuta en un entorno Amazon Linux 2, como WSL.
- --solo carga
 - Cargue únicamente el esquema y los zips de la aplicación en Amazon S3, no inicie la simulación.

- `--sin compilación`
 - Omite la reconstrucción del proyecto.
- `--sin contenedor`
 - Omite la reconstrucción del contenedor de simulación que aparece en el esquema.
- `--consoleclient`
 - Cree y conecte automáticamente el cliente de consola que aparece en `config.py`.
- `--esquema`
 - Qué esquema utilizará esta invocación. El valor predeterminado es 'SCHEMA' en `config.py`.
- `--name NAME`
 - Nombre que tendrá la simulación. El valor predeterminado es el de fecha y hora de «PROJECT_NAME» en `config.py`.

Paso 3: Compruebe los registros (opcional)

SimSpace Weaver escribe los mensajes de administración de la simulación y la salida de la consola de sus aplicaciones en Amazon CloudWatch Logs. Para obtener más información sobre cómo trabajar con registros, consulte [Trabajar con grupos de registros y flujos](#) de CloudWatch registros en la Guía del usuario de Amazon Logs.

Cada simulación que cree tiene su propio grupo de CloudWatch registros en Logs. El nombre del grupo de registro se especifica en el esquema de simulación. En el siguiente fragmento de esquema, el valor de `log_destination_service` es `logs`. Esto significa que el valor de `log_destination_resource_name` es el nombre de un grupo de registro. En este caso, el grupo de registro es `MySimulationLogs`.

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

También puede usar la API de `DescribeSimulation` para buscar el nombre del grupo de registro para la simulación después de iniciarla.

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

El siguiente ejemplo muestra la parte del resultado de DescribeSimulation que describe la configuración de registro. El nombre del grupo de registro se muestra al final de LogGroupArn.

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

Cada grupo de registro de simulación contiene varios flujos de registro:

- Flujo de registro de administración: mensajes de administración de simulaciones producidos por el SimSpace Weaver servicio.

```
/sim/management
```

- Flujo de registro de errores: mensajes de error producidos por el SimSpace Weaver servicio. Este flujo de registro solo existe si hay errores. SimSpace Weaver almacena los errores escritos por tus aplicaciones en sus propios flujos de registro de aplicaciones (consulta los siguientes flujos de registro).

```
/sim/errors
```

- Flujos de registro de aplicaciones espaciales (1 para cada aplicación espacial de cada trabajador): resultado de la consola generado por aplicaciones espaciales. Cada aplicación espacial escribe en su propio flujo de registro. *spatial-app-id* son todos los caracteres que aparecen después de la barra final del worker-id.

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

- Flujos de registro de aplicaciones personalizadas (1 para cada instancia de aplicación personalizada): resultado de la consola generado por aplicaciones personalizadas. Cada instancia de aplicación personalizada escribe en su propio flujo de registro.

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

- Flujos de registro de aplicaciones de servicio (1 para cada instancia de aplicación de servicio): resultado de la consola generado por las aplicaciones de servicio. Cada aplicación de servicio escribe en su propio flujo de registro. *service-app-id son todos los caracteres que aparecen después de la barra final del service-app-name.*

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

Note

La aplicación de ejemplo no tiene aplicaciones de servicio.

Paso 4: Ver la simulación

El SDK de la SimSpace Weaver aplicación ofrece diferentes opciones para ver la aplicación de muestra. Puedes usar el cliente de consola de ejemplo si no tienes soporte local para el desarrollo de Unreal Engine. En las instrucciones del cliente de Unreal Engine se presupone que estás utilizando Windows.

El cliente de la consola muestra una lista de los eventos de la entidad a medida que se producen. El cliente obtiene la información de los eventos de la entidad del ViewApp. Si el cliente de la consola muestra la lista de eventos, confirma la conectividad de la red con la actividad de ViewApp y de la simulación.

La simulación de PathfindingSample crea entidades estacionarias y móviles en un plano bidimensional. Las entidades móviles se mueven alrededor de las entidades estacionarias. El cliente Unreal Engine proporciona una visualización de los eventos de la entidad.

Ciente de consola

El cliente de consola se puede crear y conectar automáticamente al lanzar un ejemplo `quick-start.py` si se incluye la `--consoleclient` opción. Para crear y conectar el cliente de consola después `quick-start.py` de haberlo llamado, haga lo siguiente:

Navegue hasta:

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

Ejecute el script para crear y conectar el cliente:

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

El script hará lo siguiente:

1. Construye el cliente de consola con CMake.
2. Inicie el ejecutable creado con la dirección IP y el número de puerto proporcionados.

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

Ciente Unreal Engine

Consulte [Lanzamiento del cliente de visualización de Unreal Engine](#).

Paso 5: Detenga y elimine la simulación

Navegue hasta:

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

Busque los nombres de sus simulaciones:

```
aws simspaceweaver list-simulations
```

Detenga y elimine la simulación:

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

El script `stop-and-delete.py` hará lo siguiente:

1. Ejecute el AWS CLI comando para detener una simulación.
 - `aws simspaceweaver stop-simulation`
 - Para obtener más información, consulte la [Referencia de AWS CLI comandos](#) para SimSpace Weaver.
2. Ejecute el AWS CLI comando para eliminar una simulación.
 - `aws simpaceweaver delete-simulation`
 - Para obtener más información, consulte la [Referencia de AWS CLI comandos](#) para SimSpace Weaver.

stop-and-delete parámetros

- `-h, --help`
 - Enumere estos parámetros.
- `--SIMULACIÓN, SIMULACIÓN`
 - El nombre de la simulación para stop-and-delete
- `--parar`
 - Solo detenga la simulación. No lo borra.
- `--eliminar`
 - Elimine solo una simulación. Solo funcionará si la simulación es una de las dos STOPPEDFAILED.

Resolución de problemas

Consulte [Resolución de problemas](#) en el tutorial de inicio rápido.

Trabajando con SimSpace Weaver

Este capítulo proporciona información y orientación para ayudarle a crear sus propias aplicaciones en SimSpace Weaver.

Temas

- [Configuración de su simulación](#)
- [Duración máxima de una simulación](#)
- [Desarrollo de aplicaciones](#)
- [Desarrollo de aplicaciones de cliente](#)
- [Obtenga la dirección IP y el número de puerto de una aplicación personalizada](#)
- [Lanzamiento del cliente de visualización de Unreal Engine](#)
- [Desarrollo local en SimSpace Weaver](#)
- [SDK de la aplicación AWS SimSpace Weaver](#)
- [Marco de demostración de AWS SimSpace Weaver](#)
- [Trabajar con Service Quotas](#)
- [Depuración de simulaciones](#)
- [Contenedores personalizados](#)
- [Trabajo con Python](#)
- [Compatibilidad con otros motores](#)
- [Uso de software con licencia con AWS SimSpace Weaver](#)
- [Administrar sus recursos con AWS CloudFormation](#)
- [Instantáneas](#)
- [Mensajería](#)

Configuración de su simulación

Un esquema de simulación (o esquema) es un archivo de texto con formato YAML que especifica la configuración de una simulación. Puede utilizar el mismo esquema para iniciar varias simulaciones. El archivo de esquema se encuentra en la carpeta del proyecto de la simulación. Puede utilizar

cualquier editor de texto para editar el archivo. SimSpace Weaver solo lee el esquema cuando inicia la simulación. Las modificaciones que realice en un archivo de esquema solo afectarán a las simulaciones nuevas que se inicien después de las modificaciones.

Para configurar la simulación, edite el archivo de esquema de simulación (utilice el separador de rutas adecuado para su sistema operativo):

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

El esquema de simulación se carga al crear una nueva simulación. El script auxiliar de inicio rápido de su proyecto cargará el esquema como parte del proceso de creación de la simulación:

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

Para obtener más información sobre la ejecución del script de inicio rápido, consulte [Tutorial introductorio](#) el [Introducción](#) capítulo de esta guía.

Parámetros de configuración de la simulación

El esquema de simulación contiene información de arranque, que incluye:

- Propiedades de la simulación: [versión del SDK y configuración de procesamiento \(tipo y número de trabajadores\)](#)
- Relojes: velocidad de graduación y tolerancias
- Estrategias de partición espacial: topología espacial (como una cuadrícula), límites y grupos de ubicación (agrupación de particiones espaciales según los trabajadores)
- Dominios y sus aplicaciones: bucket de aplicaciones, ruta y comando(s) de lanzamiento

SimSpace Weaver utiliza la configuración del esquema para configurar y organizar las particiones espaciales, lanzar aplicaciones y hacer avanzar la simulación a la velocidad de aceleración especificada.

Note

El script de creación de proyectos del SDK de la SimSpace Weaver aplicación generará automáticamente un esquema de simulación basado en la aplicación de ejemplo.

En los siguientes temas se describen los parámetros del esquema de simulación. Para obtener una descripción completa del esquema de simulación, consulte [SimSpace Weaver referencia del esquema de simulación](#).

Temas

- [Versión de SDK](#)
- [Propiedades de simulación](#)
- [Trabajadores](#)
- [Reloj](#)
- [Estrategias de partición](#)
- [Dominios](#)

Versión de SDK

El `sdk_version` campo especifica la versión para la SimSpace Weaver que está formateado el esquema. Valores válidos: 1.17, 1.16, 1.15, 1.14, 1.13, 1.12

Important

El valor de `sdk_version` solo incluye el número de versión principal y el número de la primera versión secundaria. Por ejemplo, el valor 1.12 especifica todas las versiones, como 1.12.x, 1.12.0, 1.12.1 y 1.12.2.

Propiedades de simulación

La sección `simulation_properties` del esquema especifica la configuración de registro y el tipo de datos para el campo de índice (normalmente la ubicación espacial) de las entidades.

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

El valor de `log_destination_service` determina la interpretación del valor de `log_destination_resource_name`. Actualmente el único valor admitido es `logs`. Esto significa

que el valor de `log_destination_resource_name` es el nombre de un grupo de registros en los registros de Amazon CloudWatch.

Note

El registro es opcional. Si no configura las propiedades de destino de los registros, la simulación no generará registros.

Solo es obligatoria la propiedad `default_entity_index_key_type`. El único valor válido es `Vector3<f32>`.

Trabajadores

En la sección `workers` se especifica el tipo y el número de trabajadores que desea ejecutar para su simulación. SimSpace Weaver utiliza sus propios tipos de trabajadores que se asignan a los tipos de instancias de Amazon EC2.

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

Permitir simulaciones con varios trabajadores

Puede crear una simulación que utilice más de un trabajador. De forma predeterminada, las simulaciones utilizan 1 trabajador. Debe modificar el esquema de simulación antes de iniciar la simulación.

Note

No se puede cambiar una simulación que ya se ha iniciado. Si desea habilitar la funcionalidad de varios trabajadores para una simulación en ejecución, primero debe detener y eliminar la simulación.

Para usar más de un trabajador, defina el número `desired` de instancias de cómputo en un valor superior a 1. Hay un número máximo de aplicaciones para cada trabajador. Para obtener

más información, consulte [SimSpace Puntos finales y cuotas de Weaver](#). SimSpace Weaver solo utilizará más de un trabajador cuando el número de aplicaciones de un trabajador supere este límite. SimSpace Weaver puede colocar una aplicación en cualquiera de los trabajadores disponibles. No se garantiza la colocación de la aplicación en un trabajador específico.

El siguiente fragmento de esquema muestra una configuración para una simulación que solicita 2 trabajadores. SimSpace Weaver intentará asignar el segundo trabajador si el número de aplicaciones supera el número máximo de aplicaciones para 1 trabajador.

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

Reloj


La sección `clock` especifica las propiedades del reloj de simulación. Actualmente, solo se puede configurar la frecuencia de graduación (el número de graduaciones por segundo que el reloj envía a las aplicaciones). La frecuencia de graduación es la velocidad máxima. La frecuencia de graduación efectiva podría ser inferior, ya que todas las operaciones (como las actualizaciones de entidades) de una graduación deben finalizar antes de que comience la siguiente. La frecuencia de graduación también se denomina frecuencia de reloj.

Los valores válidos de `tick_rate` dependen del `sdk_version` especificado en el esquema.

Valores válidos para la frecuencia de graduación

- Versiones anteriores a "1.14":
 - 10
 - 15
 - 30
- Versión "1.14" o posterior:
 - "10"
 - "15"
 - "30"
 - "unlimited"

Para obtener más información, consulte [Frecuencia de graduación ilimitada](#).


 Important

- Para los esquemas con un valor `sdk_version` anterior a "1.14" el valor de `tick_rate` es un entero, como 30.
- Para los esquemas con un `sdk_version` de "1.14" o posterior, el valor de `tick_rate` es una cadena, como "30". El valor debe incluir comillas dobles.

Si convierte una versión "1.12" o un esquema "1.13" en una versión "1.14" o posterior, debe escribir el valor de `tick_rate` entre comillas dobles.

Frecuencia de graduación ilimitada

Puede configurar `tick_rate` para que "unlimited" permita que la simulación se ejecute tan rápido como el código. Con una tasa de tildes ilimitada, SimSpace Weaver envía la siguiente graduación inmediatamente después de que todas las aplicaciones finalicen las confirmaciones correspondientes a la graduación actual.

 Important

La frecuencia de graduación ilimitada no es compatible con las versiones anteriores a la 1.14.0 de SimSpace Weaver. El valor mínimo de `sdk_version` en el esquema es "1.14".

Frecuencia de graduación ilimitada en SimSpace Weaver Local

SimSpace Weaver Local implementa "unlimited" como si el esquema especificara una frecuencia de graduación de 10 kHz (10000). El efecto es el mismo que el de una frecuencia de graduación ilimitada en Nube de AWS. Aún debe especificar `tick_rate`: "unlimited" en su esquema. Para obtener más información acerca de SimSpace Weaver Local, consulte [Desarrollo local en SimSpace Weaver](#).

Preguntas frecuentes acerca del reloj

Q1 (P1). ¿Puedo cambiar una simulación iniciada para que utilice una frecuencia de graduación diferente?

No se puede cambiar la tasa de activación de una simulación que ya existe en el Nube de AWS en ninguna fase de su ciclo de vida. Tampoco puede cambiar la tasa de activación de una simulación en ejecución en SimSpace Weaver Local. Puede configurar `tick_rate` en el esquema e iniciar una nueva simulación a partir de ese esquema.

Q2 (P2). ¿Puedo ejecutar mi simulación con una frecuencia de graduación ilimitada en una versión anterior a la 1.14?

No, la frecuencia de graduación ilimitada no es compatible con las versiones anteriores a la 1.14.0.

Solución de errores del reloj

Si la simulación no se inicia, puede comprobar el valor de `StartError` en el resultado de la API `DescribeSimulation`. Un valor de `tick_rate` no válido en el esquema generará los siguientes errores.

Note

El resultado de error que se muestra aquí se muestra en varias líneas para mejorar la legibilidad. La salida de error real es una sola línea.

- El `sdk_version` es anterior "1.14" y el valor de `tick_rate` es un entero no válido. Valores válidos: 10, 15, 30

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

- El `sdk_version` es anterior "1.14" y el valor de `tick_rate` es una cadena. Valores válidos: 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\"}"]"
```

- El `sdk_version` es "1.14" o posterior y el valor de `tick_rate` es una cadena no válida. Valores válidos: "10", "15", "30", "unlimited"

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

- El `sdk_version` es "1.14" o posterior y el valor de `tick_rate` es un número entero. Valores válidos: "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\"}]"
```

Estrategias de partición

En la sección `partitioning_strategies` se especifican las propiedades de configuración de las particiones de las aplicaciones espaciales. Debe proporcionar su propio nombre para la estrategia de partición (un conjunto de propiedades en esta sección) y utilizarlo en la configuración de su aplicación espacial.

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

La propiedad `topology` especifica el tipo de sistema de coordenadas que utiliza la simulación. El valor `Grid` especifica una rejilla bidimensional (2D).

En el caso de una topología `Grid`, el espacio de simulación se modela como un cuadro delimitador alineado con los ejes (AABB). Los límites de coordenadas de cada eje de su AABB se especifican

en la propiedad de `aabb_bounds`. Todas las entidades que existen espacialmente en la simulación deben tener una posición dentro del AABB.

Grupos de ubicación de cuadrícula

Un grupo de ubicación es un conjunto de particiones de aplicaciones espaciales que desea que SimSpace Weaver coloque en el mismo elemento de trabajo. Debe especificar el número y la disposición de los grupos de ubicaciones (en una cuadrícula) en la propiedad `grid_placement_groups`. SimSpace Weaver intentará distribuir uniformemente las particiones entre los grupos de colocación. Las áreas de propiedad de las aplicaciones espaciales con particiones en el mismo grupo de ubicación estarán espacialmente adyacentes.

Recomendamos que x^* y sea igual al número deseado de trabajadores. Si no es igual, SimSpace Weaver intentará equilibrar tus grupos de colocación entre los trabajadores disponibles.

Si no especifica un grupo de colocación, SimSpace Weaver calculará uno de ellos.

Dominios

Debe proporcionar un nombre para un conjunto de propiedades de configuración de un dominio. La configuración de inicio de las aplicaciones de un dominio determina el tipo de dominio:

- **launch_apps_via_start_app_call**: dominio personalizado
- **launch_apps_by_partitioning_strategy**: dominio espacial
- **launch_apps_per_worker**(no incluido en la aplicación de muestra): dominio de servicio

Important

SimSpace Weaver admite hasta 5 dominios para cada simulación. Esto incluye todos los dominios espaciales, personalizados y de servicio.

```
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 Los proyectos de la SDK versión 1.12.x de la aplicación utilizan cubos separados para los archivos.zip de la aplicación y el esquema:

- tejedora-*lowercase-project-name-account-number-app-zips-region*
- tejedora-*lowercase-project-name-account-number-esquemas-region*

Temas

- [Configuración de la aplicación](#)
- [Configuración de dominios espaciales](#)
- [Puntos de conexión de red](#)
- [Configuración de dominios de servicio](#)

Configuración de la aplicación

La configuración de una aplicación (`app_config`) se especifica como parte de la configuración de su dominio. Todos los tipos de dominios utilizan estas mismas propiedades de configuración de la aplicación.

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

Note

SimSpace Weaver Los proyectos de SDK la versión 1.12.x de la aplicación utilizan cubos separados para los archivos.zip de la aplicación y el esquema:

- tejedora-*lowercase-project-name-account-number*-app-zips-*region*
- tejedora-*lowercase-project-name-account-number*-esquemas-*region*

La `package` propiedad especifica el S3 URI de un archivo zip en un bucket de S3. El archivo zip contiene el ejecutable de la aplicación (también denominado binario) y cualquier otro recurso que necesite (como bibliotecas). Cada instancia del ejecutable de la aplicación se ejecuta en un Docker contenedor en un trabajador.

La propiedad `launch_command` especifica el nombre del ejecutable y cualquier opción de línea de comandos para ejecutar la aplicación. El valor de `launch_command` es una matriz. Cada símbolo de toda la cadena de comandos de lanzamiento es un elemento de la matriz.

Ejemplo

- Para el comando de lanzamiento: `MyTestApp --option1 value1`
- Especifique: `launch_command: ["MyTestApp", "-option1", "value1"]`.

La `required_resource_units` propiedad especifica el número de unidades de recursos informáticos que se SimSpace Weaver deben asignar a esta aplicación. Una unidad de recursos informáticos es una cantidad fija de capacidad de procesamiento (vCPU) y memoria (RAM) sobre un trabajador. Puede aumentar este valor para aumentar la cantidad de potencia de cálculo disponible para la aplicación cuando se ejecuta en un trabajador. Hay un número limitado de unidades de recursos informáticos en cada trabajador. Para obtener más información, consulte [SimSpace Puntos finales y cuotas de Weaver](#).

Configuración de dominios espaciales

Para los dominios espaciales, debe especificar un `partitioning_strategy`. El valor de esta propiedad es el nombre que ha asignado a una estrategia de partición que ha definido en otra parte del esquema.

```
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 Los proyectos de la SDK versión 1.12.x de la aplicación utilizan cubos separados para los archivos.zip de la aplicación y el esquema:

- tejedora-*lowercase-project-name-account-number*-app-zips-*region*
- tejedora-*lowercase-project-name-account-number*-esquemas-*region*

Una estrategia de particionamiento con una Grid topología (la única topología compatible en esta versión) permite organizar las particiones de SimSpace Weaver aplicaciones espaciales de este dominio en una cuadrícula. La propiedad `grid_partition` especifica el número de filas y columnas de la cuadrícula de particiones.

SimSpace Weaver iniciará 1 instancia de la aplicación espacial para cada celda de la cuadrícula de particiones. Por ejemplo, si un dominio espacial tiene `grid_partition` valores `x: 2` y `y: 2` hay $2 * 2 = 4$ particiones en el dominio espacial. SimSpace Weaver iniciará 4 instancias de la aplicación configuradas en el dominio espacial y asignará 1 partición a cada instancia de la aplicación.

Temas

- [Requisitos de recursos para los dominios espaciales](#)
- [Múltiples dominios espaciales](#)
- [Preguntas frecuentes sobre dominios espaciales](#)
- [Solución de problemas de dominios espaciales](#)

Requisitos de recursos para los dominios espaciales

Puede asignar hasta 17 unidades de recursos de procesamiento para cada trabajador. Usted especifica el número de unidades de recursos informáticos que utiliza cada aplicación espacial en la sección `app_config` de su dominio espacial.

Example fragmento de esquema que muestra las unidades de recursos de cómputo de una aplicación espacial

```
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
```

Para calcular el número de unidades de recursos informáticos que requiere un dominio, multiplique el número de celdas de la cuadrícula (en su `grid_partition`, $x * y$) por el número de unidades de recursos informáticos asignadas a las aplicaciones espaciales.

En el ejemplo anterior, el dominio `MySpatialDomain` especifica:

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

La cuadrícula para `MySpatialDomain` tiene $2 * 2 = 4$ celdas. El dominio espacial requiere $4 * 1 = 4$ unidades de recursos informáticos.

El número total de unidades de recursos informáticos para todos los dominios especificados en el esquema debe ser inferior o igual al `desired` número de trabajadores multiplicado por el número máximo de unidades de recursos informáticos de cada trabajador (17).

Múltiples dominios espaciales

Puede configurar la simulación para que utilice más de un dominio espacial. Por ejemplo, puede usar un dominio espacial para controlar los actores principales de una simulación (como personas y automóviles) y un dominio espacial diferente para controlar el entorno.

También puede utilizar varios dominios espaciales para asignar distintos recursos a distintas partes de la simulación. Por ejemplo, si la simulación tiene un tipo de entidad que tiene 10 veces más instancias de entidad que otro tipo, puede crear diferentes dominios para gestionar cada tipo de entidad y asignar más recursos al dominio con más entidades.

Important

SimSpace Weaver las versiones anteriores a la 1.14.0 no admiten varios dominios espaciales.

Important

AWS SimSpace Weaver Local actualmente no admite varios dominios espaciales. Para obtener más información acerca de SimSpace Weaver Local, consulte [Desarrollo local en SimSpace Weaver](#).

Important

SimSpace Weaver admite hasta 5 dominios para cada simulación. Esto incluye todos los dominios espaciales, personalizados y de servicio.

Configurar varios dominios espaciales

Para configurar más de un dominio espacial, añade las demás definiciones de dominio espacial como secciones con nombres independientes en su esquema. Cada dominio debe especificar la clave `launch_apps_by_partitioning_strategy`. Consulte el siguiente esquema de ejemplo.

```
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
```

Colocar dominios espaciales juntos

En algunos casos, es posible que desee colocar las particiones de un dominio espacial en los trabajadores junto a las particiones de otro dominio. Esto puede mejorar las características de rendimiento si esas particiones se suscriben entre sí a varios dominios.

Añada la clave de nivel superior `placement_constraints` a su esquema para especificar qué dominios SimSpace Weaver deben colocarse juntos. La clave `on_workers` requerida debe hacer referencia a una configuración de `workers` con nombre en el esquema.

Example fragmento de esquema que muestra los dominios espaciales colocados juntos

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

Important

- Si utiliza grupos de ubicación:
 - Asegúrese de que x^* y sea múltiplo de la cantidad de trabajadores.
 - Asegúrese de que los valores de los grupos de ubicación sean divisores comunes para las dimensiones de la cuadrícula de los dominios que coloque juntos.
- Si no utiliza grupos de ubicación:
 - Asegúrese de que 1 eje de las cuadrículas de su dominio espacial tenga un divisor común que sea igual al número de trabajadores.

Para obtener más información acerca de los grupos de ubicación, consulte [Estrategias de partición](#).

Preguntas frecuentes sobre dominios espaciales

Q1 (P1). ¿Cómo puedo añadir otro dominio espacial a una simulación existente?

- Para una simulación en ejecución: no se puede cambiar la configuración de una simulación en ejecución. Cambie la configuración del dominio en el esquema, cargue el esquema y los archivos comprimidos de la aplicación e inicie una nueva simulación.
- Para una nueva simulación: añada la configuración del dominio en el esquema, cargue el esquema y los archivos comprimidos de la aplicación e inicie una nueva simulación.

Solución de problemas de dominios espaciales

Cuando intenta iniciar la simulación, pero la configuración de dominio no es válida, puede aparecer el siguiente mensaje de error:

```
"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  
  [\u0027grid_partition\u0027] values, or adjusting the  
  dimensions of your [\u0027grid_placement_groups\u0027].\"}]"
```

Causas posibles

- El esquema asigna más unidades de recursos de cómputo a las aplicaciones de las que están disponibles para los trabajadores.
- SimSpace Weaver no se puede determinar un acuerdo para agrupar los dominios en los trabajadores. Esto ocurre cuando se especifican varios dominios espaciales, pero no hay un divisor común o un múltiplo entre las cuadrículas de dominios (por ejemplo, entre una cuadrícula de 2x4 y una cuadrícula de 3x5).

Puntos de conexión de red

Las aplicaciones personalizadas y de servicio pueden tener puntos de conexión de red a los que se pueden conectar los clientes externos. Debe especificar una lista de números de puerto como valor `ingress_ports` en `endpoint_config`. Estos números de puerto son ambos TCP y UDP. La aplicación personalizada o de servicio debe enlazarse a los números de puerto que especifique `ingress_ports`. SimSpace Weaver asigna dinámicamente los números de puerto en

tiempo de ejecución y asigna estos puertos a los puertos dinámicos. Puede llamar al describe-app API una vez que sus aplicaciones hayan empezado a encontrar los números de puerto dinámicos (reales). Para obtener más información, consulte [Obtenga la dirección IP y el número de puerto de una aplicación personalizada](#) en el tutorial de inicio rápido.

```
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 Los proyectos de la SDK versión 1.12.x de la aplicación utilizan cubos separados para los archivos.zip de la aplicación y el esquema:

- tejedora-*lowercase-project-name-account-number-app-zips-region*
- tejedora-*lowercase-project-name-account-number-esquemas-region*

Note

endpoint_config es una propiedad opcional para aplicaciones personalizadas y aplicaciones de servicio. Si no especifica ningún endpoint_config, la aplicación no tendrá un punto de conexión de red.

Configuración de dominios de servicio

La presencia de launch_apps_per_worker: en una configuración de dominio indica que se trata de un dominio de servicio que tiene aplicaciones de servicio. SimSpace Weaver inicia y detiene las aplicaciones de servicio por usted. Cuando se SimSpace Weaver inicia y detiene una aplicación, se

considera que la aplicación tiene un ciclo de vida gestionado. SimSpace Weaver actualmente permite iniciar 1 o 2 aplicaciones de servicio en todos y cada uno de los trabajadores.

Example Ejemplo de un dominio configurado para lanzar una aplicación de servicio en cada trabajador

```
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 Ejemplo de un dominio configurado para lanzar dos aplicaciones de servicio en cada trabajador

```
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
```

Duración máxima de una simulación

Cada simulación AWS SimSpace Weaver tiene una configuración de duración máxima que especifica el tiempo máximo que se puede ejecutar la simulación. La duración máxima se proporciona como parámetro al iniciar una simulación. La [interfaz de programación de aplicaciones \(API\)](#) de `StartSimulation` tiene un parámetro opcional `MaximumDuration`. El valor del parámetro es un número de minutos (m o M), horas (h o H) o días (d o D). Por ejemplo, 1h o 1H significa 1 hora. SimSpace Weaver detiene la simulación cuando alcanza este límite.

Valor máximo

El valor válido más alto para `MaximumDuration` es 14D, o su equivalente en horas (336H) o minutos (20160M).

Valor predeterminado

El parámetro `MaximumDuration` es opcional. Si no proporciona un valor, SimSpace Weaver utiliza un valor de 14D.

Valor mínimo

El valor válido más bajo para `MaximumDuration` es un valor que es numéricamente equivalente a 0. Por ejemplo, los valores 0M, 0H y 0D, son todos numéricamente equivalentes a 0.

Si proporciona el valor mínimo para una duración máxima, la simulación pasa inmediatamente al STOPPING estado en cuanto lo alcanza. STARTED

Iniciar una simulación mediante la consola

Puede proporcionar un valor para la duración máxima al iniciar una simulación en la [consola de SimSpace Weaver](#). Introduzca el valor en el campo Duración máxima del formulario de configuración de simulación cuando seleccione Iniciar simulación.

Important

Si no proporciona un valor para Duración máxima, SimSpace Weaver utiliza el [valor por defecto](#) (14D).

El estado de una simulación que alcanza su duración máxima

Cuando detiene SimSpace Weaver automáticamente una simulación que alcanza su duración máxima, el estado de la simulación es STOPPING (si está en curso) o STOPPED. En la [consola de SimSpace Weaver](#), el estado objetivo de la simulación aún es STARTED, porque ese fue el último estado solicitado por un usuario.

Desarrollo de aplicaciones

SimSpace Weaver el desarrollo requiere un Amazon Linux 2 (AL2) entorno para crear aplicaciones porque las simulaciones se ejecutan Amazon Linux en AWS Cloud. Si lo estás usando Windows, puedes usar los scripts del SDK de la SimSpace Weaver aplicación para crear y lanzar un Docker contenedor que se ejecute AL2 con las dependencias que necesitas para crear SimSpace Weaver aplicaciones. También puede lanzar un entorno AL2 con Windows Subsystem for Linux (WSL) o usar un sistema nativo AL2. Para obtener más información, consulte [Configuración del entorno de SimSpace Weaver](#).

Note

Independientemente de cómo configure su entorno de desarrollo local, sus aplicaciones se ejecutan en contenedores Docker cuando las carga para ejecutarlas en Nube de AWS. Las aplicaciones no tienen acceso directo al sistema operativo de host.

Flujo general de una aplicación SimSpace Weaver

1. Crear una aplicación.
2. Bucle:
 - a. Comience la actualización creando un `Transaction`.
 - Salga del ciclo si la simulación se está cerrando.
 - b. Procese los eventos de la entidad de suscripción y propiedad.
 - c. Actualice la simulación.
 - d. Envíe `Transaction` para finalizar la actualización.
3. Destruya la aplicación.

Aplicación espacial

Cada aplicación espacial tiene un área de propiedad que es una región espacial del mundo de la simulación. Las entidades ubicadas en el área de propiedad de una aplicación espacial se almacenan en la partición asignada a la aplicación. La aplicación espacial única tiene la propiedad total (permisos de lectura y escritura) de todas las entidades de la partición asignada. Ninguna otra aplicación puede escribir en esas entidades. La aplicación espacial avanza en el estado de sus entidades. Cada aplicación espacial posee solo 1 partición. SimSpace Weaver usa la ubicación espacial de una entidad para indexarla y asignarla a una partición de aplicación espacial.

El SDK de la SimSpace Weaver aplicación proporciona una aplicación de ejemplo. Puede encontrar el código fuente de la aplicación espacial de la aplicación de ejemplo en la siguiente carpeta (utilice el separador de rutas correcto para su sistema operativo):

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

Aplicaciones personalizadas

Puede crear y utilizar aplicaciones personalizadas para interactuar con la simulación.

Las aplicaciones personalizadas pueden

- Crear entidades
- Suscribirse a otras particiones
- Confirmar cambios

Flujo general de una aplicación personalizada

1. Crear una aplicación.
2. Suscribirse a una región específica de la simulación:
 - a. Crear una `Transaction` para comenzar la primera actualización.
 - b. Crear una suscripción para la región específica.
 - c. Enviar `Transaction` para finalizar la primera actualización.
3. Bucle:
 - a. Cree una `Transaction` para comenzar la actualización.

- Salga del ciclo si la simulación se está cerrando.
 - b. El estado del proceso cambia.
 - c. Envíe `Transaction` para finalizar la actualización.
4. Destruya la aplicación.

Una vez que una aplicación personalizada crea una entidad, debe transferirla a un dominio espacial para que la entidad exista espacialmente dentro de la simulación. SimSpace Weaver usa la ubicación espacial de la entidad para colocarla en la partición de aplicación espacial adecuada. La aplicación personalizada que creó la entidad no puede actualizar ni eliminar la entidad después de transferirla a un dominio espacial.

El SDK de la SimSpace Weaver aplicación proporciona una aplicación de muestra. Puede usar las aplicaciones personalizadas incluidas en la aplicación de ejemplo como modelos para sus propias aplicaciones personalizadas. Puede encontrar el código fuente de la aplicación de visualización (una aplicación personalizada) de la aplicación de ejemplo en la siguiente carpeta (utilice el separador de rutas correcto para su sistema operativo):

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

Desarrollo de aplicaciones de cliente

Algunas de las razones por las que podría querer conectar un cliente a una simulación son las siguientes:

- Introducir información sobre el tráfico en tiempo real en una simulación a escala de ciudad.
- Crear simulaciones informales en las que un operador humano controle algún aspecto de la simulación.
- Hacer posible que los usuarios interactúen con la simulación, por ejemplo, en una simulación de entrenamiento.

Las aplicaciones personalizadas de estos ejemplos actúan como interfaz entre el estado de la simulación y el mundo exterior. Los clientes se conectan a las aplicaciones personalizadas para interactuar con la simulación.

SimSpace Weaver no gestiona las aplicaciones cliente ni su comunicación con tus aplicaciones personalizadas. Usted es responsable del diseño, la creación, el funcionamiento y la seguridad de las aplicaciones cliente y de su comunicación con las aplicaciones personalizadas. SimSpace Weaver solo muestra una dirección IP y un número de puerto para cada una de tus aplicaciones personalizadas para que los clientes puedan conectarse a ellas.

El SDK de la SimSpace Weaver aplicación proporciona clientes para su aplicación de muestra. Puede utilizar estos clientes como modelos para sus propias aplicaciones cliente. Puede encontrar el código fuente de los clientes de muestra en la siguiente carpeta:

Docker

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

WSL

Important

Proporcionamos estas instrucciones para su comodidad. Son para su uso con Windows Subsystem for Linux (WSL), y no son compatibles. Para obtener más información, consulte [Configuración del entorno de SimSpace Weaver](#).

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

Para obtener más información sobre la creación y el uso de los clientes de aplicaciones de muestra, consulte los tutoriales de [Empezar con SimSpace Weaver](#).

Obtenga la dirección IP y el número de puerto de una aplicación personalizada

Para ver la simulación, debe crear una aplicación personalizada y conectarse a ella con un cliente. Para obtener más información, consulte los tutoriales de [Empezar con SimSpace Weaver](#). Puede usar el siguiente procedimiento para obtener la dirección IP y el número de puerto de su aplicación personalizada. Utilice el separador de rutas adecuado para su sistema operativo (por ejemplo, \ en Windows y / Linux).

Para obtener su dirección IP y número de puerto

1. Utilice la `ListSimulationsAPI` para obtener el nombre de la simulación.

```
aws simspaceweaver list-simulations
```

Ejemplo de salida:

```
{
  "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. Utilice la `DescribeSimulationAPI` para obtener una lista de los dominios de la simulación.

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

Busque la sección `Domains` en la sección `LiveSimulationState` del resultado.

Ejemplo de salida:

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



```

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

```

- Use la `ListAppsAPI` para obtener una lista de aplicaciones personalizadas en un dominio. Por ejemplo, el nombre de dominio de la aplicación de visualización (personalizada) del proyecto de ejemplo es `MyViewDomain`. Busque el nombre de la aplicación en el resultado.

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

Ejemplo de salida:

```

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

```

- Use la `DescribeAppAPI` para obtener la dirección IP y el número de puerto. Para el proyecto de ejemplo, el nombre de dominio es `MyViewDomain` y el nombre de la aplicación es `ViewApp`.

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

La dirección IP y el número de puerto están en el bloque `EndpointInfo` del resultado. La dirección IP es el valor de `Address` y el número de puerto es el valor de `Actual`.

Ejemplo de salida:

```
{
  "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

El valor de `Declared` es el número de puerto al que debe enlazarse el código de la aplicación. El valor de `Actual` es el número de puerto al que se SimSpace Weaver exponen los clientes para que se conecten a tu aplicación. SimSpace Weaver asigna el `Declared` puerto al `Actual` puerto.

Lanzamiento del cliente de visualización de Unreal Engine

Navegue hasta:

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

1. Ejecute uno de los siguientes comandos:

- Docker: `python quick-start.py`
 - WSL: `python quick-start.py --a12`
2. Obtenga la dirección IP y el número de puerto «real». Estarán en la salida de la consola al ejecutar `quick-start.py`, o puede obtenerlos siguiendo los procedimientos que se indican en [Obtenga la dirección IP y el número de puerto de una aplicación personalizada](#).
 3. Navegue hasta:

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

4. Ejecute los siguientes comandos para crear la biblioteca NNG:

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

5. En un editor de texto, abra `view_app_url.txt`.
6. Actualice la URL con la dirección IP y el número de puerto de tu aplicación de visualización: `tcp://ip-address:actual-port-number` (debería tener el aspecto de `tcp://198.51.100.135:1234`).
7. En el editor Unreal, selecciona reproducir.

Resolución de problemas

- El paso de instalación de NNG CMake falla y dice «Quizás necesite privilegios administrativos»:

```
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)
```

- Solución: si `nng.lib` `nng.so` existe en el directorio `UnrealClient/lib`, este error se puede ignorar de forma segura. Si no es así, intente ejecutar los comandos `cmake build` en una terminal con privilegios de administrador.
- «CMake busca un archivo de configuración de paquetes proporcionado por 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.
```

- Solución: CMake tiene problemas para encontrar el archivo. Findnng.cmake Cuando construyas con CMake, agrega el argumento. `-DTHIRD_PARTY_LIB_PATH sdk-folder/ThirdParty` Asegúrese de que el Findnng.cmake archivo aún esté en el ThirdParty directorio antes de volver a ejecutar la compilación de CMake.

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

Desarrollo local en SimSpace Weaver

Puede implementar sus SimSpace Weaver aplicaciones localmente para realizar pruebas y depuraciones rápidas.

Requisitos

- Realice los pasos que se indican en [Configuración de SimSpace Weaver](#).

Temas

- [Paso 1: Lance su simulación local](#)
- [Paso 2: Visualice su simulación local](#)
- [Paso 3: Detenga la simulación local \(opcional en Windows\)](#)
- [Solución de problemas de desarrollo local en SimSpace Weaver](#)

Paso 1: Lance su simulación local

1. Navegue hasta

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

2. Ejecute el siguiente comando para crear e iniciar la simulación de forma local.

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

Este script hará lo siguiente:

1. Compilar el proyecto.

- `quick-start.py` llama a la `build_project` función definida en `build.py`. Este paso variará en función del proyecto. Para ello `PathfindingSample`, se utiliza `CMake`. Los comandos `CMake` y `Docker` se encuentran en `build.py`.

2. Lance su simulación local

- El script lanzará un proceso local para cada partición espacial definida en el esquema.
- El script lanzará un proceso para cada aplicación personalizada definida en el esquema.
- Primero se lanzarán las aplicaciones espaciales, seguidas de las aplicaciones personalizadas, cada una en el orden en que aparecen en el esquema.

Important

Cuando se inicie en un entorno que no sea compatible con la GUI, como una sesión SSH de consola, utilice la `--noappwindow` opción para redirigir todos los resultados al terminal actual.

Important

Para los usuarios de Linux, el script asume que el sistema tiene el `xterm` comando. Si su distribución de Linux no tiene el `xterm` comando, utilice la `--noappwindow` opción para redirigir todos los resultados a la terminal actual.

- `-h, --help`
 - Enumere estos parámetros.
- `--limpiar`
 - Elimine el contenido del directorio de compilación antes de compilar.
- `--nobuild`
 - Omite la reconstrucción del proyecto.

- --no hay ventana de aplicación
 - No abras una ventana nueva para cada aplicación. En su lugar, redirige la salida estándar a la terminal actual.
- --archivo de registro
 - Escribe el resultado de la consola en un archivo de registros.
- --consoleclient
 - Conecta automáticamente el cliente de consola que aparece en la configuración.
- --esquema de esquema
 - Qué esquema utilizará esta invocación. El valor predeterminado es 'SCHEMA' en config.py.

Paso 2: Visualice su simulación local

Para ver su simulación local, puede utilizar cualquiera de los clientes que se incluyen en el `SimSpaceWeaverAppSdkDistributable`. Para obtener más información sobre la creación y el uso de los clientes de muestra, consulte los tutoriales en [Empezar con SimSpace Weaver](#).

Debe actualizar la dirección IP y el número de puerto del cliente para conectarse a la aplicación de visualización para su simulación local. Utilice siempre los siguientes valores con SimSpace Weaver Local:

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

Según el cliente que seleccione, puede actualizar la dirección IP y el número de puerto de la siguiente manera:

- Unreal: cambie la URL en la línea 1 de `view_app_url.txt`
- Consola: inicie el cliente con la dirección IP y la URL del número de puerto como parámetro

Paso 3: Detenga la simulación local (opcional en Windows)

Note

Este paso es obligatorio en Linux, pero opcional en Windows.

1. Navegue hasta:

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

2. Ejecute el siguiente comando para detener la simulación local y eliminar cualquier recurso de memoria compartida.

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

Este script hará lo siguiente:

- Detenga los procesos locales.
- Elimine el objeto de memoria compartida (solo es necesario en Linux).

parámetros de stop-and-delete.py

- -h, --help
 - Enumere estos parámetros.
- --parar
 - Intente detener los procesos únicamente.
- --eliminar
 - Intente eliminar únicamente los recursos de memoria compartidos.
- --proceso
 - El nombre del proceso que se va a detener. Use esta opción si el nombre de su proceso no coincide con el nombre del paquete en el esquema.
- --esquema de esquema
 - Qué esquema utilizará esta invocación. El valor predeterminado es 'SCHEMA' en config.py.

Solución de problemas de desarrollo local en SimSpace Weaver

- Linux: no se encuentra el comando xterm o no se puede abrir
 - Los scripts locales asumen que el comando xterm existe cuando se ejecuta en Linux. Si no dispone del comando xterm o lo ejecuta en un entorno que no admite la interfaz gráfica de usuario, utilice la --noappwindow opción cuando ejecute el script de inicio rápido.
- ¡No se abre ninguna ventana de aplicaciones!

- Esto sucede cuando la simulación local se bloquea inmediatamente. Para ver el resultado de la consola tras el bloqueo, utilice las `--logfile` opciones `--noappwindow` o al ejecutar el script de inicio rápido.
- ¡La simulación no funciona una vez que se inicia la aplicación de visualización o se conecta el cliente de visualización!
- Ejecutar `--noappwindow` esta opción normalmente resuelve este tipo de problemas. De lo contrario, reiniciar varias veces también tiene éxito (aunque a un ritmo mucho menor).

SDK de la aplicación AWS SimSpace Weaver

El SDK de la aplicación SimSpace Weaver proporciona API que puede utilizar para controlar las entidades de la simulación y responder a los eventos de SimSpace Weaver. Incluye el espacio de nombres siguiente:

- API: definiciones básicas de la API y su uso

Enlace con la siguiente biblioteca:

- `libweaver_app_sdk_cxx_v1_full.so`

Important

La biblioteca está disponible para la vinculación dinámica cuando ejecuta sus aplicaciones en Nube de AWS. No es necesario cargarlo con las aplicaciones.

Note

Las API del SDK de aplicaciones SimSpace Weaver controlan los datos dentro de su simulación. Estas API están separadas de las API de servicio de SimSpace Weaver, que controlan sus recursos de servicio de SimSpace Weaver (como simulaciones, aplicaciones y relojes) en AWS. Para obtener más información, consulte [SimSpace Weaver Referencias de API](#).

Temas

- [Los métodos de la API devuelven un Result](#)
- [Interactuar con el SDK de aplicación de nivel superior](#)
- [Gestión de la simulación](#)
- [Suscripciones](#)
- [Entidades](#)
- [Eventos de entidad](#)
- [Result y control de errores](#)
- [Genéricos y tipos de dominio](#)
- [Operaciones diversas del SDK de la aplicación](#)

Los métodos de la API devuelven un Result

La mayoría de las funciones de la API SimSpace Weaver tienen un tipo de retorno `Aws::WeaverRuntime::Result<T>`. Si la función se ha ejecutado correctamente, `Result` contiene `T`. De lo contrario, `Result` contiene un código `Aws::WeaverRuntime::ErrorCode` que representa un código de error de Rust App SDK.

Example Ejemplo

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

Este método:

- Devuelve `Transaction` si se `BeginUpdate()` ejecuta correctamente.
- Regresa `Aws::WeaverRuntime::ErrorCode` si `BeginUpdate()` falla.

Interactuar con el SDK de aplicación de nivel superior

Ciclo de vida

- El SDK de la aplicación SimSpace Weaver gestiona el ciclo de vida de la aplicación. No es necesario leer ni escribir el estado del ciclo de vida de una aplicación.

Particiones

- Use `Result <PartitionSet> AssignedPartitions(Transaction& txn);` para obtener particiones propias.
- Use `Result <PartitionSet> AllPartitions(Transaction& txn);` para obtener todas las particiones de la simulación.

Gestión de la simulación

En esta sección se describen las soluciones para las tareas comunes de administración de simulaciones.

Temas

- [Iniciar una simulación](#)
- [Actualizar una simulación](#)
- [Finalizar una simulación](#)

Iniciar una simulación

Use `CreateApplication()` para crear una aplicación.

Example Ejemplo

```
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()));
```

Actualizar una simulación

Utilice las siguientes funciones de `BeginUpdate` para actualizar la aplicación:

- `Result<Transaction> BeginUpdate(Application& app)`
- `Result<bool> BeginUpdateWillBlock(Application& app)`: le indica si `BeginUpdate()` bloqueará o no bloqueará.

Use `Result<void> Commit(Transaction& txn)` para confirmar los cambios.

Example Ejemplo

```
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();
}
```

Finalizar una simulación

Use `Result<void> DestroyApplication(Application&& app)` para finalizar la aplicación y la simulación.

Otras aplicaciones descubren que la simulación se cierra cuando reciben llamadas `ErrorCode::ShuttingDown` a `BeginUpdateWillBlock()` o `BeginUpdate()`. Cuando una aplicación recibe `ErrorCode::ShuttingDown`, puede llamar a `Result<void> DestroyApplication(Application&& app)` para que se cierre automáticamente.

Example Ejemplo

```
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

Solo llama a `Result<void> DestroyApplication(Application&& app)` después de `Api::Commit()`. Destruir una aplicación durante una actualización puede provocar un comportamiento indefinido.

⚠ Important

Debe llamar a `DestroyApplication()` antes de que se cierre el programa para asegurarse de que la aplicación informa que se ha cerrado correctamente. Si no llama a `DestroyApplication()` cuando el programa se cierra, el estado se considerará como FATAL.

Suscripciones

Creará una suscripción con un área de suscripción y un ID de dominio. El ID de dominio representa el dominio propietario de esa área de suscripción. Un `BoundingBox2F32` describe el área de suscripción. Para crear una suscripción, utilice la siguiente función:

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

Example Ejemplo

```
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();
}
```

Puede utilizar los datos de `Api::SubscriptionHandle` devueltos por `CreateSubscriptionBoundingBox2F32()` para modificar la suscripción. Se pasa como argumento a las siguientes funciones:

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

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

Entidades

Puede llamar a las API Store y Load utilizando la `Api::Entity` del `Result<Api::Entity>` devuelto desde `CreateEntity()`, o desde un evento de cambio de propiedad cuando una entidad entra en el área de suscripción de la aplicación (para más información, consulte [Eventos de entidad](#)). Le recomendamos que realice un seguimiento de sus objetos de `Api::Entity` para poder usarlos con estas API.

Temas

- [Creación de entidades](#)
- [Transferir una entidad a un dominio espacial](#)
- [Escribir y leer los datos del campo de la entidad](#)
- [Almacenar la posición de una entidad](#)
- [Cargar la posición de una entidad](#)

Creación de entidades

Use `CreateEntity()` para crear una entidad. Usted define el significado de `Api::TypeId` que pasa a esta función.

```
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

Los valores del 0 al 511 para `Api::BuiltinTypeId` están reservados. Su entidad `TypeID` (`k_entityTypeId` en este ejemplo) debe tener un valor de 512 o superior.

Transferir una entidad a un dominio espacial

Después de que una aplicación o aplicación de servicio personalizada cree una entidad, la aplicación debe transferirla a un dominio espacial para que la entidad exista espacialmente en la simulación. Otras aplicaciones pueden leer las entidades de un dominio espacial y actualizarlas una aplicación espacial. Utilice la API `ModifyEntityDomain()` para transferir una entidad a un dominio espacial.

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

Si el valor de `DomainId` no coincide con el asignado a `Partition` de la aplicación que realiza la llamada, `DomainId` debe ser para un `DomainType::Spatial Domain`. La transferencia de propiedad a la nueva `Domain` se produce durante el `Commit(Transaction&&)`.

Parámetros

txn

La actual Transaction.

entity

El Entity objetivo para el cambio de Domain.

domainId

El DomainId de la Domain de destino para la Entity.

Esta API devuelve Success si el dominio de la entidad se ha modificado correctamente.

Escribir y leer los datos del campo de la entidad

Todos los campos de datos de entidades son de tipo blob. Puede escribir hasta 1024 bytes de datos en una entidad. Le recomendamos que mantenga los blobs lo más pequeños posible, ya que los tamaños más grandes reducirán el rendimiento. Al escribir en un blob, se pasa SimSpace Weaver un puntero a los datos y a su longitud. Cuando lee un blob, SimSpace Weaver le proporciona un puntero y una longitud para leer. Todas las lecturas deben estar completas antes de que la aplicación llame a `Commit()`. Los punteros devueltos por una llamada de lectura se invalidan cuando la aplicación llama a `Commit()`.

Important

- No se admite la lectura de un puntero blob almacenado en caché después de `Commit()` y puede provocar un error en la simulación.
- No se admite la escritura en un puntero de blob devuelto por una llamada de lectura y puede provocar un error en la simulación.

Temas

- [Almacenar los datos de campo de una entidad](#)
- [Cargar los datos de campo de una entidad](#)
- [Cargar los datos de campo de las entidades eliminadas](#)

Almacenar los datos de campo de una entidad

Los siguientes ejemplos muestran cómo se pueden almacenar (escribir en la estructura de estados) los datos de campo de una entidad propiedad de la aplicación. Estos ejemplos usan la siguiente función:

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

El parámetro `Api::TypeId keyTypeId` representa el tipo de datos de los datos transferidos.

El parámetro `Api::TypeId keyTypeId` debe recibir la información correspondiente `Api::TypeId` de `Api::BuiltinTypeId`. Si no hay una conversión adecuada, puede usar `Api::BuiltinTypeId::Dynamic`.

Para tipos de datos complejos, utilice `Api::BuiltinTypeId::Dynamic`.

Note

El valor de `FieldIndex index` debe ser mayor que cero. El valor 0 está reservado para la clave de índice (consulte `StoreEntityIndexKey()`).

Example Ejemplo de uso de tipos de datos primitivos

```
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 Ejemplo de uso de struct para almacenar los datos

```

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));
}

```

Cargar los datos de campo de una entidad

Los siguientes ejemplos muestran cómo se pueden cargar (leer desde la estructura de estados) los datos de campo de una entidad. Estos ejemplos usan la siguiente función:

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

El parámetro `Api::TypeId keyTypeId` debe recibir la información correspondiente `Api::TypeId` de `Api::BuiltinTypeId`. Si no hay una conversión adecuada, puede usar `Api::BuiltinTypeId::Dynamic`.

Note

El valor del índice `FieldIndex` debe ser mayor que 0. El valor 0 está reservado para la clave de índice (consulte `StoreEntityIndexKey()`).

Example Ejemplo de uso de tipos de datos primitivos

```
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 Ejemplo de uso de struct para almacenar los datos

```

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);
}

```

Cargar los datos de campo de las entidades eliminadas

No puede cargar (leer de la estructura estatal) los datos de campo de las entidades que se han eliminado de las áreas de propiedad y suscripción de la aplicación. En el siguiente ejemplo, se produce un error porque `Api::LoadIndexKey()` llama a una entidad como resultado de un `Api::ChangeListAction::Remove`. El segundo ejemplo muestra una forma correcta de almacenar y cargar los datos de la entidad directamente en la aplicación.

Example Ejemplo de código incorrecto

```
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 Ejemplo de una forma correcta de almacenar y cargar datos de entidades en la aplicación

```

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;
}
}
}

/* ... */
}

```

Almacenar la posición de una entidad

Puede almacenar (escribir en la estructura de estados) la posición de una entidad mediante una estructura de datos de enteros. Estos ejemplos usan la siguiente función:

```

Result<void> StoreEntityIndexKey(
    Transaction& txn,
    const Entity& entity,
    TypeId keyTypeId,
    std::int8_t* src,
    std::size_t length)

```

Note

Debe proporcionar `Api::BuiltinTypeId::Vector3F32` a `Api::StoreEntityIndexKey()`, como se muestra en los siguientes ejemplos.

Example Ejemplo: uso de una matriz para representar la posición

```
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 Ejemplo de uso de struct para representar la posición

```
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));
}

```

Cargar la posición de una entidad

Puede cargar (leer en la estructura de estados) la posición de una entidad mediante una estructura de datos de enteros. Estos ejemplos usan la siguiente función:

Note

Debe proporcionar `Api::BuiltinTypeId::Vector3F32` a `Api::LoadEntityIndexKey()`, como se muestra en los siguientes ejemplos.

Example Ejemplo: uso de una matriz para representar la posición

```

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 Ejemplo de uso de struct para representar la posición

```

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);
}
```

Eventos de entidad

Puede utilizar las siguientes funciones del SDK de la aplicación SimSpace Weaver para obtener todos los eventos de propiedad y suscripción:

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

Puede utilizar el marco de demostración de SimSpace Weaver si necesita un procesamiento de eventos de entidades basado en llamadas. Para obtener más información, consulte el siguiente archivo de encabezado:

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

También puede crear su propio procesamiento de eventos de entidad.

Temas

- [Repasar los eventos de las entidades de su propiedad](#)

- [Reparar los eventos de las entidades suscritas](#)
- [Reparar los eventos de las entidades de su propiedad](#)

Reparar los eventos de las entidades de su propiedad

Use `OwnershipChanges()` para obtener una lista de eventos para las entidades en propiedad (entidades del área de propiedad de la aplicación). La función tiene la siguiente firma:

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

A continuación, recorra en iteraciones las entidades con un bucle, como se muestra en el siguiente ejemplo.

Example Ejemplo

```
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;
  }
}
```

```
}
```

Tipos de eventos

- None: la entidad está en el área y sus datos de posición y campo no han sufrido modificación.
- Remove: la entidad se ha retirado del área.
- Add: la entidad se ha agregado al área.
- Update: la entidad se encuentra en el área y se ha modificado.
- Reject: la aplicación no ha podido eliminar la entidad del área.

Note

En el caso de un evento de Reject, la aplicación volverá a intentar la transferencia al marcar la casilla siguiente.

Repasar los eventos de las entidades suscritas

Use `AllSubscriptionEvents()` para obtener una lista de eventos para las entidades suscritas (entidades del área de suscripción de la aplicación). La función tiene la siguiente firma:

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

A continuación, recorra en iteraciones las entidades con un bucle, como se muestra en el siguiente ejemplo.

Example Ejemplo

```
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;
    }
}
```

Tipos de eventos

- **None:** la entidad está en el área y sus datos de posición y campo no han sufrido modificación.
- **Remove:** la entidad se ha retirado del área.
- **Add:** la entidad se ha agregado al área.
- **Update:** la entidad se encuentra en el área y se ha modificado.
- **Reject:** la aplicación no ha podido eliminar la entidad del área.

Note

En el caso de un evento de **Reject**, la aplicación volverá a intentar la transferencia al marcar la casilla siguiente.

Repasar los eventos de las entidades de su propiedad

Para ver los eventos en los que una entidad se mueve entre un área de propiedad y un área de suscripción, compare los cambios entre los eventos de suscripción y propiedad de la entidad actuales y anteriores.

Puede gestionar estos eventos leyendo:

- `Api::SubscriptionChangeList`

- `Api::OwnershipEvents`

A continuación, puede comparar los cambios con los datos almacenados anteriormente.

En el siguiente ejemplo, se muestra cómo se pueden gestionar los eventos de cambio de propiedad de una entidad. En este ejemplo se supone que, en el caso de las entidades que pasan de ser entidades suscritas a entidades de propiedad (en cualquier dirección), el evento de eliminación o adición de la propiedad se produce primero, seguido del evento de eliminación o adición de la suscripción en la siguiente casilla.

Example Ejemplo

```
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 entityIdByAction;

/**
 * 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:
            {
                entityIdByAction[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 y control de errores

La clase de `Aws::WeaverRuntime::Result<T>` usa una biblioteca `Outcome` de terceros. Puede usar el siguiente patrón para comprobar `Result` y atrapar los errores devueltos por las llamadas a la 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();
         */
    }
}
}

```

Macro de instrucciones de control de Result

Dentro de una función con un tipo de retorno `Aws::WeaverRuntime::Result<T>`, puede utilizar la macro `WEAVERRUNTIME_TRY` en lugar del patrón de código anterior. La macro ejecutará la función que se le haya transferido. Si la función pasada falla, la macro hará que la función adjunta devuelva

un error. Si la función pasada se ejecuta correctamente, la ejecución pasa a la siguiente línea. El siguiente ejemplo muestra una reescritura de la función de `DoBeginUpdate()` anterior. Esta versión utiliza la macro `WEAVERRUNTIME_TRY` en lugar de la estructura de control `if-else`. Tenga en cuenta que el tipo de retorno de la función es `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();
}
```

Si `BeginUpdate()` falla, la macro `DoBeginUpdate()` vuelve antes de tiempo y se produce un error. Puede utilizar la macro `WEAVERRUNTIME_EXPECT_ERROR` para obtener el `Aws::WeaverRuntime::ErrorCode` de `BeginUpdate()`. En el siguiente ejemplo, se muestra cómo la función `Update()` llama a `DoBeginUpdate()` y obtiene el código de error en caso de error.

```
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);

}
}
```

Puede hacer que el código de error `BeginUpdate()` esté disponible para una función que llama a `Update()` cambiando el tipo de retorno `Update()` a `Aws::WeaverRuntime::Result<void>`. Puede repetir este proceso para seguir enviando el código de error más abajo en la pila de llamadas.

Genéricos y tipos de dominio

El SDK de la aplicación SimSpace Weaver proporciona los tipos de datos de precisión simple `Api::Vector2F32` y `Api::BoundingBox2F32`, y los de doble precisión `Api::Vector2F64` y `Api::BoundingBox2F64`. Estos tipos de datos son estructuras de datos pasivas sin métodos prácticos. Tenga en cuenta que la API solo usa `Api::Vector2F32` y `Api::BoundingBox2F32`. Puede usar estos tipos de datos para crear y modificar suscripciones.

El marco de demostración de SimSpace Weaver proporciona una versión mínima de la biblioteca matemática `AzCore`, que contiene `Vector3` y `Aabb`. Para obtener más información, consulte los archivos de encabezado en:

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

Operaciones diversas del SDK de la aplicación

Temas

- [AllSubscriptionEvents y OwnershipChanges contienen eventos de la última llamada](#)
- [Liberar los bloqueos de lectura después del procesamiento de SubscriptionChangeList](#)
- [Creación de una instancia de aplicación independiente para realizar pruebas](#)

AllSubscriptionEvents y OwnershipChanges contienen eventos de la última llamada

Los valores devuelven las llamadas a la última llamada `Api::AllSubscriptionEvents()` y `Api::OwnershipChanges()` contienen los eventos de la última llamada, no los de la última graduación. En el ejemplo siguiente, `secondSubscriptionEvents` y

`secondOwnershipChangeList` están vacíos porque sus funciones se invocan inmediatamente después de las primeras llamadas.

Si espera 10 pasos y, a continuación, llama a `Api::AllSubscriptionEvents()` y `Api::OwnershipChanges()`, sus resultados contendrán tanto los eventos como los cambios de los últimos 10 dígitos (no del último).

Example Ejemplo

```
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 función `AllSubscriptionEvents()` está implementada pero la función `SubscriptionEvents()` no está implementada.

Liberar los bloqueos de lectura después del procesamiento de `SubscriptionChangeList`

Al iniciar una actualización, hay segmentos de memoria compartida para los datos confirmados en otras particiones correspondientes a la marca anterior. Es posible que los lectores bloqueen estos

segmentos de memoria compartida. Una aplicación no puede confirmarse por completo hasta que todos los lectores hayan liberado los bloqueos. Como optimización, una aplicación debería llamar a `Api::ReleaseReadLeases()` para liberar los bloqueos después de procesar los elementos de `Api::SubscriptionChangelist`. Esto reduce la contención en el momento de la confirmación. `Api::Commit()` publica las concesiones de lectura de forma predeterminada, pero se recomienda publicarlas manualmente después de procesar las actualizaciones de las suscripciones.

Example Ejemplo

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

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

    WEAVERRUNTIME_EXPECT(Api::ReleaseReadLeases(transaction));

    ...
}
```

Creación de una instancia de aplicación independiente para realizar pruebas

Puede utilizar `Api::CreateStandaloneApplication()` para crear una aplicación independiente para probar la lógica de la aplicación antes de ejecutar el código en una simulación real.

Example Ejemplo

```
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);

    ...
}
```

Marco de demostración de AWS SimSpace Weaver

El marco de demostración de AWS SimSpace Weaver (marco de demostración) es una biblioteca de utilidades que puede utilizar para desarrollar aplicaciones SimSpace Weaver.

El marco de demostración proporciona

- Ejemplos de código y patrones de programación para que los utilice y examine
- Abstracciones y funciones de utilidad que agilizan el desarrollo de aplicaciones sencillas
- Una forma más sencilla de probar las funciones experimentales del SDK de la aplicación SimSpace Weaver

Diseñamos el SDK de la aplicación SimSpace Weaver con un acceso de bajo nivel a las API de SimSpace Weaver para ofrecer un mayor rendimiento. Por el contrario, diseñamos el marco de demostración para proporcionar abstracciones de alto nivel y acceso a las API que faciliten el uso de SimSpace Weaver. El coste que supone la facilidad de uso es un nivel de rendimiento inferior en comparación con el uso directo del SDK de la aplicación SimSpace Weaver. Las simulaciones que toleran un rendimiento inferior (como las que no requieren requisitos de rendimiento en tiempo real) pueden ser buenas candidatas para utilizar el marco de demostración. Le recomendamos que utilice la funcionalidad nativa del SDK de la aplicación SimSpace Weaver para aplicaciones complejas, ya que el marco de demostración no es un conjunto de herramientas completo.

El marco de demostración incluye

- Ejemplos de código funcional que respaldan y demuestran lo siguiente:
 - Gestión del flujo de aplicaciones
 - Procesamiento de eventos de entidades basado en llamadas
- Un conjunto de bibliotecas de utilidades de terceros:
 - spdlog (una biblioteca de registro)
 - Una versión mínima de AZCore (una biblioteca de matemáticas) que contiene únicamente:
 - Vector3
 - Aabb
 - cxxopts (una biblioteca de analizadores de opciones de línea de comandos)
- Funciones de utilidad específicas de SimSpace Weaver

El marco de demostración consta de una biblioteca, archivos fuente y CMakeLists. Los archivos se incluyen en el paquete distribuible del SDK de la aplicación SimSpace Weaver.

Trabajar con Service Quotas

En esta sección se describen las Service Quotas de SimSpace Weaver. Las cuotas también se denominan límites. Para obtener una lista de Service Quotas, consulte [SimSpace Puntos finales y cuotas de Weaver](#). Las API de esta sección pertenecen al conjunto de API de aplicaciones. Las API de aplicaciones son diferentes de las API de servicio. Las API de la aplicación forman parte del SDK de la aplicación SimSpace Weaver. Puede encontrar la documentación de las API de la aplicación en la carpeta SDK de la aplicación de su sistema local:

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

Temas

- [Obtener los límites de una aplicación](#)
- [Obtener la cantidad de recursos que utiliza una aplicación](#)
- [Restablecimiento de métricas](#)
- [Superar un límite](#)
- [Agotamiento de la memoria](#)
- [Prácticas recomendadas](#)

Obtener los límites de una aplicación

Puede usar la API de la aplicación RuntimeLimits para consultar los límites de una aplicación.

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

Parámetros

Application y aplicación

Una referencia a la aplicación

LimitType type

Una enumeración con los siguientes tipos de límites:

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

En el siguiente ejemplo, se consulta el límite de recuento de entidades.

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

Obtener la cantidad de recursos que utiliza una aplicación

Puede llamar a la API de la aplicación `RuntimeMetrics` para obtener la cantidad de recursos que utiliza una aplicación:

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

Parámetros

Application y aplicación

Una referencia a la aplicación

La API devuelve una referencia a una struct que contiene las métricas. Una contramétrica contiene un valor total acumulado y solo aumenta. Una métrica de indicador contiene un valor que puede aumentar o disminuir. El tiempo de ejecución de la aplicación actualiza un contador cada vez que un evento aumenta el valor. El motor de ejecución solo actualiza los indicadores al llamar a la API. SimSpace Weaver garantiza que la referencia sea válida durante toda la vida útil de la aplicación. Las llamadas repetidas a la API no cambiarán la referencia.

```
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
}

```

Restablecimiento de métricas

La API de la aplicación `ResetRuntimeMetrics` restablece los valores de `AppRuntimeMetrics` struct.

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

En el siguiente ejemplo se muestra cómo hacer llamadas `ResetRuntimeMetrics` en la aplicación.

```

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));
}

```

Superar un límite

Una llamada a la API de una aplicación que supere un límite devolverá un `ErrorCode::CapacityExceeded`, excepto en el caso de las transferencias de entidades. SimSpace Weaver gestiona las transferencias de entidades de forma asíncrona como parte de las operaciones de envío y de la API de la aplicación `BeginUpdate`, por lo que no hay ninguna operación específica que devuelva un error si una transferencia falla debido al límite de transferencia de la entidad. Para detectar errores de transferencia, puedes comparar los valores actuales de

`rejected_incoming_transfer_counter` y `rejected_outgoing_transfer_counter` (en `AppRuntimeMetrics` struct) con sus valores anteriores. Las entidades rechazadas no estarán en la partición, pero la aplicación podrá simularlas.

Agotamiento de la memoria

SimSpace Weaver utiliza un proceso de recolección de basura para limpiar y liberar la memoria liberada. Es posible escribir datos más rápido de lo que el recolector de basura puede liberar memoria. Si esto ocurre, las operaciones de escritura podrían superar el límite de memoria reservada de la aplicación. SimSpace Weaver devolverá un error interno con un mensaje que contiene `OutOfMemory` (y detalles adicionales). Para obtener más información, consulte [Distribuya las escrituras en el tiempo](#).

Prácticas recomendadas

Las siguientes prácticas recomendadas son directrices generales para diseñar las aplicaciones de forma que no excedan los límites. Es posible que no se apliquen al diseño específico de su aplicación.

Supervise con frecuencia y reduzca la velocidad

Debe supervisar sus métricas con frecuencia y ralentizar las operaciones que estén cerca de alcanzar un límite.

Evite sobrepasar los límites de suscripción y transferencia

Si es posible, diseñe la simulación para reducir la cantidad de suscripciones remotas y transferencias de entidades. Puede usar grupos de ubicación para colocar varias particiones en el mismo trabajador y reducir la necesidad de transferencias de entidades remotas entre trabajadores.

Distribuya las escrituras en el tiempo

La cantidad y el tamaño de las actualizaciones de una graduación pueden tener un impacto significativo en el tiempo y la memoria necesarios para confirmar una transacción. Los requisitos de memoria elevados pueden provocar que el tiempo de ejecución de la aplicación se quede sin memoria. Puedes repartir las escrituras a lo largo del tiempo para reducir el tamaño total medio de las actualizaciones por tilde. Esto puede ayudar a mejorar el rendimiento y evitar sobrepasar los límites. Le recomendamos que no escriba más de una media de 12 MB por graduación o 1,5 KB por entidad.

Depuración de simulaciones

Puede utilizar los siguientes métodos de para obtener información sobre sus simulaciones.

Temas

- [Utilice SimSpace Weaver Local y observe el resultado de la consola](#)
- [Mira tus registros en Amazon CloudWatch Logs](#)
- [Utilice llamadas a la API describe](#)
- [Conectar un cliente](#)

Utilice SimSpace Weaver Local y observe el resultado de la consola

Le recomendamos que desarrolle primero las simulaciones de forma local y, a continuación, las ejecute en el Nube de AWS. Puede ver el resultado de la consola directamente cuando lo ejecute con SimSpace Weaver Local. Para obtener más información, consulte [Desarrollo local en SimSpace Weaver](#).

Mira tus registros en Amazon CloudWatch Logs

Cuando ejecutas la simulación en la consola, Nube de AWS la salida de tus aplicaciones se envía a las transmisiones de registro de Amazon CloudWatch Logs. La simulación también escribe otros datos de registro. Debe habilitar el registro en el esquema de simulación si desea que la simulación escriba datos de registro. Para obtener más información, consulte [SimSpace Weaver inicia sesión en Amazon CloudWatch Logs](#).

Warning

La simulación puede producir grandes cantidades de datos de registro. Los datos de registro pueden crecer muy rápidamente. Debe vigilar sus registros de cerca y detener las simulaciones cuando ya no necesite que se ejecuten. El registro puede generar grandes costes.

Utilice llamadas a la API describe

Puede utilizar las siguientes API de servicio para obtener información sobre sus simulaciones en Nube de AWS.

- `ListSimulations`— obtenga una lista de todas sus simulaciones en Nube de AWS.

Example Ejemplo

```
aws simspaceweaver list-simulations
```

- `DescribeSimulation`: obtener detalles sobre una simulación.

Example Ejemplo

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

- `DescribeApp`: obtener detalles sobre una aplicación.

Example Ejemplo

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

Para obtener más información sobre las SimSpace Weaver API, consulte [SimSpace Weaver Referencias de API](#).

Conectar un cliente

Puede conectar un cliente a una aplicación personalizada o de servicio en ejecución que haya definido `endpoint_config` en su esquema de simulación. El SDK de la SimSpace Weaver aplicación incluye clientes de muestra que puede usar para ver la aplicación de muestra. Puede consultar el código fuente de estos clientes de ejemplo y la aplicación de ejemplo para ver cómo puede crear sus propios clientes. Para obtener más información sobre cómo crear y ejecutar los clientes de muestra, consulta los tutoriales en [Empezar con SimSpace Weaver](#).

Puede encontrar el código fuente de los clientes de muestra en la siguiente carpeta:

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

Depuración de simulaciones locales

Puede depurar sus aplicaciones de SimSpace Weaver Local con Microsoft Visual Studio. Para obtener más información sobre cómo depurar con Visual Studio, consulte [Microsoft Visual Studio documentation](#).

Para depurar su simulación local

1. Asegúrese de que `schema.yaml` está en su directorio de trabajo.
2. En Visual Studio, abra el menú contextual de cada aplicación que quieras depurar (por ejemplo, `PathfindingSampleLocalSpatial` o `PathfindingSampleLocalView`) y establezca el directorio de trabajo en la sección de depuración.
3. Abra el menú contextual de la aplicación que quiera depurar y seleccione Establecer como proyecto de inicio.
4. Seleccione F5 para empezar a depurar la aplicación.

Los requisitos para depurar una simulación son los mismos que para ejecutar una simulación con normalidad. Debe iniciar el número de aplicaciones espaciales especificado en el esquema. Por ejemplo, si su esquema especifica una cuadrícula de 2 x 2 e inicia una aplicación espacial en modo de depuración, la simulación no se ejecutará hasta que inicie 3 aplicaciones espaciales más (en modo de depuración o no en modo de depuración).

Para depurar una aplicación personalizada, primero debe iniciar las aplicaciones espaciales y, a continuación, iniciar la aplicación personalizada en el depurador.

Tenga en cuenta que la simulación se ejecuta al mismo ritmo. En cuanto una aplicación alcance un punto de interrupción, todas las demás se pausarán. Cuando continúe desde ese punto de interrupción, las demás aplicaciones continuarán.

Contenedores personalizados

AWS SimSpace Weaver las aplicaciones se ejecutan en entornos contenerizados Amazon Linux 2 (AL2). En el Nube de AWS, SimSpace Weaver ejecuta sus simulaciones en contenedores Docker creados a partir de una `amazonlinux:2` imagen servida desde Amazon Elastic Container Registry (Amazon ECR). Puede crear una imagen de Docker personalizada, almacenarla en Amazon ECR y utilizarla para la simulación en lugar de la imagen de Docker predeterminada que le proporcionamos.

Puede usar un contenedor personalizado para administrar sus dependencias de software e incluir componentes de software adicionales que no estén en la imagen de Docker estándar. Por ejemplo,

puede añadir al contenedor las bibliotecas de software disponibles públicamente que usa su aplicación y solo colocar el código personalizado en el archivo zip de la aplicación.

Important

Solo admitimos imágenes de Docker AL2 alojadas en los repositorios de Amazon ECR, ya sea en la galería pública de Amazon ECR o en su registro privado de Amazon ECR. No admitimos las imágenes de Docker alojadas fuera de Amazon ECR. Para obtener más información sobre Amazon ECR, consulte la [Documentación de Amazon Elastic Container Registry](#).

Temas

- [Creación de un contenedor personalizado](#)
- [Modificar un proyecto para usar un contenedor personalizado](#)
- [Preguntas frecuentes sobre contenedores personalizados](#)
- [Solución de problemas con contenedores personalizados](#)

Creación de un contenedor personalizado

En estas instrucciones se presupone que sabe utilizar Docker y Amazon Elastic Container Registry (Amazon ECR). Para obtener más información sobre Amazon ECR, consulte la [Guía del usuario de Amazon ECR](#).

Requisitos previos

- La identidad de IAM (uso o rol) que utiliza para realizar estas acciones tiene los permisos correctos para usar Amazon ECR
- Docker está instalado en su sistema local.

Para crear un contenedor personalizado

1. Cree su `Dockerfile`.

R `Dockerfile` Para ejecutar AWS SimSpace Weaver aplicaciones comienza con la Amazon Linux 2 imagen en Amazon ECR.

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

2. Compile su Dockerfile.
3. Cargue su imagen de contenedor en Amazon ECR.
 - [Utilice la AWS Management Console.](#)
 - [Utilice la AWS Command Line Interface.](#)

Note

Si recibe un error de `AccessDeniedException` al intentar subir la imagen de su contenedor a Amazon ECR, es posible que su identidad de IAM (usuario o rol) no tenga los permisos necesarios para usar Amazon ECR. Puede adjuntar la política `AmazonEC2ContainerRegistryPowerUser` AWS gestionada a su identidad de IAM y volver a intentarlo. Para más información sobre cómo adjuntar una política, consulte [Adición y eliminación de permisos de identidad de IAM](#) en la Guía del usuario de AWS Identity and Access Management .

Modificar un proyecto para usar un contenedor personalizado

Estas instrucciones dan por sentado que ya sabes cómo utilizarla AWS SimSpace Weaver y quieres que tus flujos de trabajo de almacenamiento y desarrollo de aplicaciones sean Nube de AWS más eficientes.

Requisitos previos

- Dispone de un contenedor personalizado en Amazon Elastic Container Registry (Amazon ECR). Para obtener más información acerca de la creación de un contenedor personalizado, consulte [Creación de un contenedor personalizado](#).

Para modificar su proyecto para usar un contenedor personalizado

1. Añada permisos al rol de aplicación de simulación de su proyecto para utilizar Amazon ECR.
 - a. Si aún no dispone de una política de IAM con los siguientes permisos, cree la política. Le sugerimos `simspaceweaver-ecr` como nombre de política. Para obtener más información

sobre cómo crear una política de IAM, consulte [Creación de políticas de IAM](#) en la Guía del usuario de AWS Identity and Access Management .

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

b. Buscar el nombre del rol de la aplicación de simulación de su proyecto:

i. En un editor de texto, abra la AWS CloudFormation plantilla:

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

ii. Busque la propiedad `RoleName` que aparece debajo de `WeaverAppRole`. El valor es el nombre del rol de la aplicación de simulación de su proyecto.

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. Adjunte la política de `simspaceweaver-ecr` al rol de la aplicación de simulación del proyecto. Para más información sobre cómo adjuntar una política, consulte [Adición y eliminación de permisos de identidad de IAM](#) en la Guía del usuario de AWS Identity and Access Management .
- d. Navegue hasta el siguiente comando `sdk-folder` y ejecútelo para actualizar la SimSpace Weaver pila de muestras:

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

2. Especifique las imágenes del contenedor en el esquema de simulación del proyecto.
 - Puede añadir la propiedad `default_image` opcional bajo `simulation_properties` para especificar una imagen de contenedor personalizada predeterminada para todos los dominios.
 - Agregue la propiedad `image` a `app_config` para un dominio para el que desee utilizar una imagen contenedora personalizada. Especifique el URI del repositorio de Amazon ECR como valor. Puede especificar una imagen diferente para cada dominio.
 - Si `image` no se especifica para un dominio y `default_image` se especifica, las aplicaciones de ese dominio utilizan la imagen predeterminada.
 - Si `image` no se especifica para un dominio ni `default_image` se especifica, las aplicaciones de ese dominio se ejecutan en un SimSpace Weaver contenedor estándar.

Example Fragmento de esquema que incluye una configuración de contenedor personalizada

```
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. Cree y cargue su proyecto como de costumbre.

Preguntas frecuentes sobre contenedores personalizados

Q1 (P1). ¿Qué hago si quiero cambiar el contenido de mi contenedor?

- Para una simulación en ejecución: no se puede cambiar el contenedor de una simulación en ejecución. Debe crear un contenedor nuevo e iniciar una nueva simulación que utilice ese contenedor.
- Para una nueva simulación: cree un contenedor nuevo, cárguelo en Amazon Elastic Container Registry (Amazon ECR) e inicie una nueva simulación que utilice ese contenedor.

Q2 (P2). ¿Cómo puedo cambiar la imagen del contenedor para mi simulación?

- Para una simulación en ejecución: no se puede cambiar el contenedor de una simulación en ejecución. Debe iniciar una nueva simulación que utilice ese nuevo contenedor.
- Para una nueva simulación: especifique la nueva imagen del contenedor en el esquema de simulación del proyecto. Para obtener más información, consulte [Modificar un proyecto para usar un contenedor personalizado](#).

Solución de problemas con contenedores personalizados

Temas

- [AccessDeniedException al subir la imagen a Amazon Elastic Container Registry \(Amazon ECR\)](#)
- [No se puede iniciar una simulación que utiliza un contenedor personalizado](#)

AccessDeniedException al subir la imagen a Amazon Elastic Container Registry (Amazon ECR)

Si recibe un error de `AccessDeniedException` al intentar subir la imagen de su contenedor a Amazon ECR, es posible que su identidad de IAM (usuario o rol) no tenga los permisos necesarios para usar Amazon ECR. Puede adjuntar la política `AmazonEC2ContainerRegistryPowerUser` AWS gestionada a su identidad de IAM y volver a intentarlo. Para más información sobre cómo adjuntar una política, consulte [Adición y eliminación de permisos de identidad de IAM](#) en la Guía del usuario de AWS Identity and Access Management .

No se puede iniciar una simulación que utiliza un contenedor personalizado

Consejos para la solución de problemas

- Si el registro está activado para la simulación, compruebe los registros de errores.
- Pruebe la simulación sin un contenedor personalizado.
- Pruebe la simulación localmente. Para obtener más información, consulte [Desarrollo local en SimSpace Weaver](#).

Trabajo con Python

Puedes usar Python para tus SimSpace Weaver aplicaciones y tu cliente. El kit de desarrollo de software Python (SDK de Python) se incluye como parte del paquete distributable del SDK de SimSpace Weaver aplicaciones estándar. El desarrollo con Python funciona de forma similar al desarrollo en los demás lenguajes compatibles.

Important

SimSpace Weaver solo es compatible con Python versión 3.9.

⚠ Important

SimSpace Weaver la compatibilidad con Python requiere la SimSpace Weaver versión 1.15.0 o posterior.

Temas

- [Crear un proyecto de Python](#)
- [Inicio de una simulación de Python](#)
- [Cliente Python de muestra](#)
- [Preguntas frecuentes sobre el uso de Python](#)
- [Solución de problemas relacionados con Python](#)

Crear un proyecto de Python

Contenedor de Python personalizado

Para ejecutar su SimSpace Weaver simulación basada en Python en Nube de AWS, puede crear un contenedor personalizado que incluya las dependencias necesarias. Para obtener más información, consulte [Contenedores personalizados](#).

Un contenedor de Python personalizado debe incluir lo siguiente:

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

Si utiliza la plantilla `PythonBubblesSample` para crear su proyecto, puede ejecutar el script `quick-start.py` (ubicado en la carpeta `tools` de su proyecto) para crear una imagen de Docker

con las dependencias necesarias. El script carga la imagen en Amazon Elastic Container Registry (Amazon ECR).

El script usa `quick-start.py` para hacer el siguiente 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
```

Puede añadir sus propias dependencias a Dockerfile:

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

El archivo `requirements.txt` contiene una lista de los paquetes de Python necesarios para la simulación de ejemplo `PythonBubblesSample`:

```
Flask==2.1.1
```

Puede añadir sus propias dependencias de paquetes de Python a `requirements.txt`:

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

Las Dockerfile y `requirements.txt` están en la carpeta `tools` de su proyecto.

Important

Técnicamente, no tiene que usar un contenedor personalizado con su simulación de Python, pero le recomendamos encarecidamente que utilice un contenedor personalizado. El

contenedor de Amazon Linux 2 (AL2) estándar que proporcionamos no tiene Python. Por lo tanto, si no utilizas un contenedor personalizado que tenga Python, debes incluir Python y las dependencias necesarias en cada archivo zip de la aplicación en el que cargues. SimSpace Weaver

Inicio de una simulación de Python

Puede iniciar su simulación basada en Python de la misma manera que una SimSpace Weaver simulación normal, SimSpace Weaver Local tanto en como en. SimSpace Weaver Nube de AWS Para obtener más información, consulte los tutoriales de. [Empezar con SimSpace Weaver](#)

PythonBubblesSample incluye su propio cliente de muestra de Python. Para obtener más información, consulte [Cliente Python de muestra](#).

Cliente Python de muestra

Si utiliza la plantilla PythonBubblesSample para crear un proyecto, su proyecto contiene un cliente de muestra de Python. Puede usar el cliente de muestra para ver la simulación de PythonBubblesSample. También puede usar el cliente de ejemplo como punto de partida para crear su propio cliente de Python.

En el siguiente procedimiento, se da por sentado que ha creado un proyecto PythonBubblesSample y ha iniciado su simulación.

Para iniciar el cliente Python

1. En una ventana de línea de comandos, vaya a la carpeta del proyecto de PyBubbleClient ejemplo.

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

2. Ejecute el cliente Python.

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

Parámetros

host

La dirección IP de la simulación. Para iniciar una simulación en Nube de AWS, puede buscar la dirección IP de la simulación en la [SimSpace Weaver consola](#) o seguir el procedimiento [Obtenga la dirección IP y el número de puerto de una aplicación personalizada](#) del tutorial de inicio rápido. Para una simulación local, use 127.0.0.1 como dirección IP.

port

El número de puerto de la simulación. Para una simulación iniciada en Nube de AWS, este es el número de Actual puerto. Puede encontrar el número de puerto de la simulación en la [consola de SimSpace Weaver](#) o utilizar el procedimiento [Obtenga la dirección IP y el número de puerto de una aplicación personalizada](#) del tutorial de inicio rápido. Para una simulación local, use 7000 como número de puerto.

simszize

El número máximo de entidades que se mostrarán en el cliente.

Preguntas frecuentes sobre el uso de Python

Q1 (P1). ¿Qué versiones de Python son compatibles?

SimSpace Weaver solo es compatible con Python versión 3.9.

Solución de problemas relacionados con Python

Temas

- [Error durante la creación de un contenedor personalizado](#)
- [La simulación de Python no se inicia](#)
- [Un cliente de visualización o simulación de Python arroja un ModuleNotFoundError](#)

Error durante la creación de un contenedor personalizado

Si aparece un error no `basic auth credentials` después de ejecutar `quick-start.py`, es posible que haya un problema con las credenciales temporales de Amazon ECR. Ejecute el siguiente comando con su Región de AWS ID y número de AWS cuenta:

```
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

Asegúrese de que el Región de AWS que especifique es el mismo que utilizó para la simulación. Utilice uno de los Regiones de AWS que SimSpace Weaver sea compatible. Para obtener más información, consulte [SimSpace Puntos finales y cuotas de Weaver](#).

Después de ejecutar el comando `aws ecr`, vuelva a ejecutar `quick-start.py`.

Otros recursos de solución de problemas que comprobar

- [Solución de problemas con contenedores personalizados](#)
- [Solución de problemas de Amazon ECR](#) en la Guía del usuario de Amazon ECR
- [Configuración con Amazon ECR](#) en la Guía del usuario de Amazon ECR

La simulación de Python no se inicia

Es posible que aparezca un error de `Unable to start app` en el registro de administración de la simulación. Esto puede ocurrir si se produce un error al crear el contenedor personalizado. Para obtener más información, consulte [Error durante la creación de un contenedor personalizado](#). Para obtener más información acerca de los registros, consulte [SimSpace Weaver inicia sesión en Amazon CloudWatch Logs](#).

Si está seguro de que su contenedor no tiene ningún problema, compruebe el código fuente de Python de su aplicación. Puede usar SimSpace Weaver Local para probar su aplicación. Para obtener más información, consulte [Desarrollo local en SimSpace Weaver](#).

Un cliente de visualización o simulación de Python arroja un ModuleNotFound error

Python arroja un error ModuleNotFound cuando no puede encontrar un paquete de Python necesario.

Si su simulación está en Nube de AWS, asegúrese de que su contenedor personalizado tenga todas las dependencias necesarias enumeradas en `surequirements.txt`. Recuerde `quick-start.py` volver a ejecutar si edita `requirements.txt`.

Si aparece el error del cliente PythonBubblesSample, utilice `pip` para instalar el paquete indicado:

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

Compatibilidad con otros motores

Puedes usar tu propio C++ motor personalizado con SimSpace Weaver. Actualmente estamos desarrollando la compatibilidad para los siguientes motores. Hay documentación independiente para cada uno de estos motores.

Important

Las integraciones con los motores que se enumeran aquí son experimentales. Están disponibles para una vista previa.

Motores

- [Unity](#) (versión mínima 2022.3.19.F1)
- [Unreal Engine](#) (versión 5.0 como mínimo)

Unity

Debe tener el entorno de Unity desarrollo ya instalado antes de crear SimSpace Weaver simulaciones con Unity. Para obtener más información, consulte las instrucciones por separado:

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

Unreal Engine

Debe crear un servidor dedicado de Unreal Engine a partir del código fuente.

SimSpaceWeaverAppSdkDistributable Incluye una versión del PathfindingSample para Unreal Engine. Para obtener más información, consulte las instrucciones por separado:

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

Uso de software con licencia con AWS SimSpace Weaver

AWS SimSpace Weaver le permite crear simulaciones con el motor de simulación y el contenido que elija. En relación con su uso de SimSpace Weaver, usted es responsable de obtener, mantener y cumplir los términos de la licencia de cualquier software o contenido que utilice en sus simulaciones. Compruebe que el acuerdo de licencia le permite implementar el software y el contenido en un entorno de alojamiento virtual.

Administrar sus recursos con AWS CloudFormation

Puede utilizar AWS CloudFormation para gestionar sus recursos AWS SimSpace Weaver. AWS CloudFormation es un servicio independiente de AWS que le ayuda a especificar, aprovisionar y administrar su infraestructura de AWS como código. Con AWS CloudFormation, puede crear un archivo JSON o YAML, denominado [plantilla](#). La plantilla especifica los detalles de la infraestructura. AWS CloudFormation utiliza la plantilla para aprovisionar la infraestructura como una sola unidad, denominada [pila](#). Al eliminar su pila, puede hacer que AWS CloudFormation elimine todo lo que hay en la pila al mismo tiempo. Puede administrar su plantilla mediante procesos de administración de código fuente estándar (por ejemplo, rastrearla en un sistema de control de versiones como [Git](#)). Para obtener más información sobre AWS CloudFormation, consulte la [Guía del usuario de AWS CloudFormation](#).

Su recurso de simulación

En AWS, un recurso es una entidad con la que se puede trabajar. Entre los ejemplos se incluyen una instancia Amazon EC2, un bucket de Amazon S3 o un rol de IAM. Su simulación SimSpace Weaver es un recurso. En las configuraciones, normalmente se especifica un recurso de AWS en el formulario `AWS::service::resource`. Para SimSpace Weaver, especifique su recurso de simulación como `AWS::SimSpaceWeaver::Simulation`. Para obtener más información sobre el recurso de simulación en AWS CloudFormation, consulte la sección [SimSpace Weaver](#) de la Guía del usuario de AWS CloudFormation.

¿Cómo puedo usar AWS CloudFormation con SimSpace Weaver?

Puede crear una plantilla de AWS CloudFormation que especifique los recursos de AWS que desea aprovisionar. La plantilla puede especificar una arquitectura completa, parte de una arquitectura o una solución pequeña. Por ejemplo, puede especificar una arquitectura para su solución de SimSpace Weaver que incluya buckets de Amazon S3, permisos de IAM, una base de datos compatible en el servicio de base de datos relacional Amazon Relational Database Service o Amazon DynamoDB y su recurso de `Simulation`. A continuación, puede utilizar AWS CloudFormation para aprovisionar todos esos recursos como una unidad y al mismo tiempo.

Example plantilla que crea recursos de IAM e inicia una simulación

La siguiente plantilla de ejemplo crea un rol de IAM y los permisos que SimSpace Weaver necesita para realizar acciones en su cuenta. Los scripts del SDK de la aplicación SimSpace Weaver crean el rol y los permisos en un Región de AWS específico, pero puede usar una plantilla de AWS CloudFormation para implementar la simulación en otro Región de AWS sin tener que volver a ejecutar los scripts. Por ejemplo, puede hacerlo para configurar una simulación de copia de seguridad con fines de recuperación de desastres.

En este ejemplo, el nombre original de la simulación es `MySimulation`. Ya existe un bucket para el esquema en el Región de AWS donde AWS CloudFormation construirá la pila. El bucket contiene una versión del esquema que está correctamente configurada para ejecutar la simulación en Región de AWS. Recuerde que el esquema especifica la ubicación de los archivos zip de la aplicación, que es un bucket de Amazon S3 en el mismo Región de AWS que la simulación. El bucket comprimido de la aplicación y los archivos deben existir ya Región de AWS cuando AWS CloudFormation cree la pila; de lo contrario, la simulación no se iniciará. Tenga en cuenta que el nombre del bucket de este ejemplo incluye Región de AWS, pero eso no determina dónde se encuentra realmente el bucket. Debe asegurarse de que el bucket esté realmente en ese Región de AWS (puede comprobar las propiedades del bucket en la consola de Amazon S3, con las API de Amazon S3 o con los comandos de Amazon S3 en AWS CLI).

En este ejemplo, se utilizan algunas funciones y parámetros integrados en AWS CloudFormation para sustituir variables. Para obtener más información, consulte la [referencia sobre las funciones intrínsecas](#) y [la referencia sobre los pseudoparámetros](#) en la Guía del usuario de AWS CloudFormation.

```
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'
```

Uso de instantáneas con AWS CloudFormation

Una [instantánea](#) es una copia de seguridad de una simulación. En el siguiente ejemplo, se inicia una nueva simulación a partir de una instantánea en lugar de a partir de un esquema. La instantánea de este ejemplo se creó a partir de una simulación de proyecto del SDK de una aplicación de SimSpace Weaver. AWS CloudFormation crea el nuevo recurso de simulación y lo inicializa con los datos de la instantánea. La nueva simulación puede tener una `MaximumDuration` diferente a la original.

Le recomendamos que cree y utilice una copia de la función de aplicación de la simulación original. El rol de aplicación de la simulación original podría eliminarse si elimina la pila de esa simulación AWS CloudFormation.

```
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"
```

Instantáneas

Puede crear una instantánea para hacer una copia de seguridad de los datos de la entidad de simulación en cualquier momento. SimSpace Weaver crea un archivo `.zip` en un bucket de Amazon S3. Puede crear una nueva simulación con la instantánea. SimSpace Weaver inicializa el State Fabric de la nueva simulación con los datos de la entidad almacenados en la instantánea, inicia las aplicaciones espaciales y de servicio que estaban en ejecución cuando se creó la instantánea y ajusta el reloj a la hora correspondiente. SimSpace Weaver obtiene la configuración de la simulación a partir de la instantánea y no de un archivo de esquema. Los archivos `.zip` de la aplicación deben

estar en la misma ubicación en Amazon S3 que en la simulación original. Debe iniciar cualquier aplicación personalizada por separado.

Temas

- [Casos de uso de instantáneas](#)
- [Utilice la SimSpace Weaver consola para trabajar con instantáneas](#)
- [Utilice el AWS CLI para trabajar con instantáneas](#)
- [Uso de instantáneas con AWS CloudFormation](#)
- [Preguntas frecuentes sobre instantáneas](#)

Casos de uso de instantáneas

Regrese a un estado anterior y explore los escenarios de ramificación

Puede crear una instantánea de la simulación para guardarla en un estado específico. A continuación, puede crear varias simulaciones nuevas a partir de esa instantánea y explorar diferentes escenarios que podrían derivarse de ese estado.

Prácticas recomendadas de recuperación de desastres y seguridad

Le recomendamos que haga copias de seguridad de la simulación con regularidad, especialmente en el caso de simulaciones que se ejecuten durante más de 1 hora o en las que intervengan varios trabajadores. Las copias de seguridad pueden ayudarle a recuperarse de desastres e incidentes de seguridad. Las instantáneas permiten hacer copias de seguridad de la simulación. Las instantáneas requieren que los archivos.zip de la aplicación estén en la misma ubicación de Amazon S3 que antes. Si necesita poder mover los archivos.zip de la aplicación a otra ubicación, debe utilizar una solución de copia de seguridad personalizada.

Para obtener más información sobre otras prácticas recomendadas, consulte [Mejores prácticas al trabajar con SimSpace Weaver](#) y [Prácticas recomendadas de seguridad para SimSpace Weaver](#).

Ampliar la duración de la simulación

El recurso de simulación es la representación de la simulación en SimSpace Weaver. Todos los recursos de simulación tienen una configuración `MaximumDuration`. Una simulación se detiene automáticamente cuando alcanza su `MaximumDuration`. El valor máximo de `MaximumDuration` es 14D (14 días).

Si necesita que la simulación persista durante más tiempo que el `MaximumDuration` de su recurso de simulación, puede crear una instantánea antes de que el recurso de simulación llegue a su `MaximumDuration`. Puede iniciar una nueva simulación (crear un nuevo recurso de simulación) con la instantánea. SimSpace Weaver inicializa los datos de la entidad a partir de la instantánea, inicia las mismas aplicaciones espaciales y de servicio que se ejecutaban anteriormente y restaura el reloj. Puede iniciar sus aplicaciones personalizadas y realizar cualquier inicialización personalizada adicional. Puede establecer el nuevo recurso `MaximumDuration` de simulación en un valor diferente al iniciarlo.

Utilice la SimSpace Weaver consola para trabajar con instantáneas

Puede utilizar la SimSpace Weaver consola para crear una instantánea de la simulación.

Temas

- [Utilice la consola para crear una instantánea](#)
- [Utilice la consola para iniciar una simulación a partir de una instantánea](#)

Utilice la consola para crear una instantánea

Para crear una instantánea

1. Inicie sesión en la [SimSpace Weaver consola AWS Management Console y conéctese a ella](#).
2. En el panel de navegación, seleccione Simulaciones.
3. Seleccione el botón de opción situado junto al nombre de la simulación. El estado de la simulación debe ser Iniciado.
4. En la parte superior de la página, seleccione Crear instantánea.
5. En Configuración de instantáneas, en Destino de la instantánea, introduzca el URI de Amazon S3 de un depósito o un depósito y una carpeta en los que desee SimSpace Weaver crear la instantánea. Puede elegir Browse S3 si prefiere buscar entre los buckets disponibles y seleccionar una ubicación.

Important

El bucket de Amazon S3 debe estar en el mismo Región de AWS que la simulación.

 Note

SimSpace Weaver crea una snapshot carpeta dentro del destino de la instantánea seleccionada. SimSpace Weaver crea el archivo.zip de la instantánea en esa snapshot carpeta.


6. Seleccione Create snapshot (Crear instantánea).

Utilice la consola para iniciar una simulación a partir de una instantánea

Para iniciar una simulación a partir de una instantánea, el archivo .zip de la instantánea debe estar en un bucket de Amazon S3 al que pueda acceder la simulación. La simulación usa los permisos definidos en el rol de la aplicación que seleccionó al iniciar la simulación. Todos los archivos .zip de la aplicación de la simulación original deben estar en las mismas ubicaciones que cuando se creó la instantánea.

Iniciar una simulación a partir de una instantánea

1. Inicie sesión en la [SimSpace Weaver consola AWS Management Console](#) y conéctese a ella.
2. En el panel de navegación, seleccione Simulaciones.
3. En la parte superior de la página, seleccione Iniciar simulación.
4. En Configuración de simulación, escriba un nombre y una descripción opcional para la simulación. El nombre de la simulación debe ser exclusivo de su Cuenta de AWS.
5. Para el método de inicio de la simulación, seleccione Usar una instantánea en Amazon S3.
6. En el caso del URI de Amazon S3 para la instantánea, introduzca el URI de Amazon S3 del archivo de instantáneas o seleccione Browse S3 para buscar y seleccionar el archivo.

 Important

El bucket de Amazon S3 debe estar en el mismo Región de AWS que la simulación.

7. Para el rol de IAM, seleccione el rol de aplicación que usará la simulación.
8. En Duración máxima, introduzca la cantidad máxima de tiempo durante la que debe ejecutarse el recurso de simulación. El valor máximo es 14D. Para obtener más información acerca de la duración máxima, consulte [_](#).

9. En Etiquetas (opcional), elija Añadir nueva etiqueta si desea añadir una etiqueta.
10. Seleccione Iniciar simulación.

Utilice el AWS CLI para trabajar con instantáneas

Puede usarlo AWS CLI para llamar a las SimSpace Weaver API desde una línea de comandos. Debe tener las AWS CLI instaladas y configuradas correctamente. Para obtener más información, consulte [Instalación o actualización de la última versión de la AWS CLI](#) en la Guía del AWS Command Line Interface usuario de la versión 2.

Temas

- [Utilice el AWS CLI para crear una instantánea](#)
- [Utilice la AWS CLI para iniciar una simulación a partir de una instantánea](#)

Utilice el AWS CLI para crear una instantánea

Para crear una instantánea

- En una línea de comandos, llame a la API CreateSnapshot.

```
aws simspaceweaver create-snapshot --simulation simulation-name --destination s3-destination
```

Parámetros

Simulación

El nombre de una simulación iniciada. Se puede utilizar `aws simspaceweaver list-simulations` para ver los nombres y estados de las simulaciones.

destination

Una cadena que especifica el bucket de Amazon S3 de destino y el prefijo de clave de objeto opcional para el archivo de instantáneas. El prefijo de la clave de objeto suele ser una carpeta del depósito. SimSpace Weaver crea la instantánea dentro de una snapshot carpeta en este destino.

⚠ Important

El bucket de Amazon S3 debe estar en el mismo Región de AWS que la simulación.

Ejemplo

```
aws simspaceweaver create-snapshot --simulation
MyProjectSimulation_23-04-29_12_00_00 --destination BucketName=weaver-
myproject-111122223333-artifacts-us-west-2,ObjectKeyPrefix=myFolder
```

Para obtener más información sobre la CreateSnapshot API, consulte [CreateSnapshot](#) la referencia de la AWS SimSpace Weaver API.

Utilice la AWS CLI para iniciar una simulación a partir de una instantánea

Iniciar una simulación a partir de una instantánea

- En una línea de comandos, llame a la API StartSimulation.

```
aws simspaceweaver start-simulation --name simulation-name --role-arn role-arn --
snapshot-s3-location s3-location
```

Parámetros

name

El nombre de la nueva simulación. El nombre de la simulación debe ser único en su Cuenta de AWS. Se puede utilizar `aws simspaceweaver list-simulations` para ver los nombres de las simulaciones existentes.

role-arn

El nombre de recurso de Amazon (ARN) del rol de aplicación que usará la simulación.

snapshot-s3-location

Una cadena que especifica el bucket de Amazon S3 y la clave de objeto para el archivo de instantáneas.

⚠ Important

El bucket de Amazon S3 debe estar en el mismo Región de AWS que la simulación.

Ejemplo

```
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
```

Para obtener más información sobre la `StartSimulation` API, consulte [StartSimulation](#) la referencia de la AWS SimSpace Weaver API.

Preguntas frecuentes sobre instantáneas

¿Mi simulación sigue ejecutándose durante una instantánea?

Sus recursos de simulación seguirán funcionando durante una instantánea y seguirá recibiendo los cargos de facturación durante ese tiempo. El tiempo se tiene en cuenta para la duración máxima de la simulación. Sus aplicaciones no reciben marcas mientras la instantánea está en curso. Si el estado del reloj era `STARTED` cuando se inició la creación de la instantánea, el reloj seguirá indicando el estado `STARTED`. Sus aplicaciones vuelven a recibir marcas una vez finalizada la instantánea. Si el estado de tu reloj era `STOPPED`, el estado de tu reloj seguirá siendo `STOPPED`. Tenga en cuenta que se está ejecutando una simulación con un estado `STARTED` aunque el estado del reloj sea `STOPPED`.

¿Qué ocurre si hay una instantánea en curso y mi simulación alcanza su duración máxima?

La simulación finalizará la instantánea y, a continuación, se detendrá en cuanto finalice el proceso de la instantánea (con éxito o sin éxito). Le recomendamos que pruebe el proceso de captura de pantalla de antemano para saber cuánto tarda, el tamaño del archivo de instantáneas que puede esperar y si debe completarse correctamente.

¿Qué ocurre si detengo una simulación que tiene una instantánea en curso?

Una instantánea en curso se detiene inmediatamente cuando se detiene la simulación. No creará un archivo de instantáneas.

¿Cómo puedo detener una instantánea en curso?

La única forma de detener una instantánea en curso es detener la simulación. No puede reiniciar una simulación después de detenerla.

¿Cuánto tiempo llevará completar la instantánea?

El tiempo necesario para crear una instantánea depende de la simulación. Le recomendamos que pruebe el proceso de captura de imágenes de antemano para saber cuánto tardará en realizarse la simulación.

¿Qué tamaño tendrá mi archivo de instantáneas?

El tamaño de un archivo de instantánea depende de la simulación. Le recomendamos que pruebe el proceso de captura de imágenes de antemano para saber cuánto tardará en realizarse la simulación.

Mensajería

La API de mensajería simplifica la comunicación entre aplicaciones dentro de la simulación. Las API para enviar y recibir mensajes forman parte del SDK de la SimSpace Weaver aplicación. En la actualidad, la mensajería utiliza un enfoque de máximo esfuerzo para enviar y recibir mensajes. SimSpace Weaver intenta enviar o recibir mensajes en la siguiente casilla de simulación, pero no hay garantías de entrega, pedido o hora de llegada.

Temas

- [Casos de uso de mensajería](#)
- [Uso de las API de mensajería](#)
- [¿Cuándo usar la mensajería](#)
- [Consejos para trabajar con la mensajería](#)
- [Errores de mensajería y solución de problemas](#)

Casos de uso de mensajería

Comuníquese entre aplicaciones de simulación

Utilice la API de mensajería para comunicarse entre las aplicaciones de la simulación. Úsela para cambiar el estado de las entidades a distancia, cambiar el comportamiento de las entidades o transmitir información a toda la simulación.

Confirme la recepción de un mensaje

Los mensajes enviados contienen información sobre el remitente en el encabezado del mensaje. Use esta información para enviar una respuesta de acuse de recibo al recibir un mensaje.

Reenvía los datos recibidos por una aplicación personalizada a otras aplicaciones de la simulación

La mensajería no reemplaza la forma en que los clientes se conectan a las aplicaciones personalizadas que se ejecutan en ellas SimSpace Weaver. Sin embargo, la mensajería permite a los usuarios reenviar datos de aplicaciones personalizadas que reciben datos de clientes a otras aplicaciones que no tienen una conexión externa. El flujo de mensajes también puede funcionar a la inversa, lo que permite que las aplicaciones sin conexión externa reenvíen los datos a una aplicación personalizada y luego a un cliente.

Uso de las API de mensajería

Las API de mensajería están incluidas en el SDK de la SimSpace Weaver aplicación (versión mínima 1.16.0). La mensajería es compatible con C++, Python y nuestras integraciones con Unreal Engine 5 y Unity.

Hay dos funciones que gestionan las transacciones de mensajes: `SendMessage` y `ReceiveMessages`. Todos los mensajes enviados contienen un destino y una carga útil. La `ReceiveMessages` API devuelve una lista de los mensajes que se encuentran actualmente en la cola de mensajes entrantes de una aplicación.

C++

Enviar mensaje

```
AWS_WEAVERRUNTIME_API Result<void> SendMessage(  
    Transaction& txn,  
    const MessagePayload& payload,  
    const MessageEndpoint& destination,  
    MessageDeliveryType deliveryType = MessageDeliveryType::BestEffort  
    ) noexcept;
```

Recibir mensajes

```
AWS_WEAVERRUNTIME_API Result<MessageList> ReceiveMessages(  
    Transaction& txn) noexcept;
```

Python

Enviar mensaje

```
api.send_message(  
    txn, # Transaction  
    payload, # api.MessagePayload  
    destination, # api.MessageDestination  
    api.MessageDeliveryType.BestEffort # api.MessageDeliveryType  
)
```

Recibir mensajes

```
api.receive_messages(  
    txn, # Transaction  
) -> api.MessageList
```

Temas

- [Enviar mensajes](#)
- [Recepción de mensajes](#)
- [Responder al remitente](#)

Enviar mensajes

Los mensajes constan de una transacción (similar a otras llamadas a la API de Weaver), una carga útil y un destino.

Carga útil de mensaje

La carga útil del mensaje es una estructura de datos flexible de hasta 256 bytes. Te recomendamos lo siguiente como práctica recomendada para crear tus cargas útiles de mensajes.

Para crear la carga útil del mensaje

1. Cree una estructura de datos (por ejemplo, `struct` en C++) que defina el contenido del mensaje.
2. Cree la carga útil del mensaje que contenga los valores que desee enviar en el mensaje.
3. Crea el `MessagePayload` objeto.

Destino del mensaje

El destino de un mensaje lo define el `MessageEndpoint` objeto. Esto incluye un tipo de punto final y un identificador de punto final. El único tipo de punto final que se `Partition` admite actualmente es el que permite dirigir los mensajes a otras particiones de la simulación. El ID del punto final es el ID de la partición del destino de destino.

Solo puede proporcionar una dirección de destino en un mensaje. Cree y envíe varios mensajes si desea enviar mensajes a más de una partición al mismo tiempo.

Para obtener instrucciones sobre cómo resolver un punto final de un mensaje desde una posición, consulte [Consejos para trabajar con la mensajería](#).

Envíe el mensaje

Puedes usar la `SendMessage` API después de crear los objetos de destino y carga útil.

C++

```
Api::SendMessage(transaction, payload, destination,  
MessageDeliveryType::BestEffort);
```

Python

```
api.send_message(txn, payload, destination, api.MessageDeliveryType.BestEffort)
```

Ejemplo completo de envío de mensajes

El siguiente ejemplo demuestra cómo se puede crear y enviar un mensaje genérico. En este ejemplo se envían 16 mensajes individuales. Cada mensaje contiene una carga útil con un valor entre 0 y 15, y el tic de simulación actual.

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

```

Recepción de mensajes

SimSpace Weaver entrega los mensajes a la cola de mensajes entrantes de una partición. Usa la `ReceiveMessages` API para obtener un `MessageList` objeto que contenga los mensajes de la cola. Procesa cada mensaje con la `ExtractMessage` API para obtener los datos del mensaje.

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

```

Responder al remitente

Cada mensaje recibido contiene un encabezado con información sobre el remitente original del mensaje. Puedes usar `message.header.source_endpoint` para enviar una respuesta.

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
```

¿Cuándo usar la mensajería

La mensajería interna SimSpace Weaver ofrece otro patrón de intercambio de información entre aplicaciones de simulación. Las suscripciones proporcionan un mecanismo de extracción para leer datos de aplicaciones o áreas específicas de la simulación; los mensajes proporcionan un mecanismo de inserción para enviar datos a aplicaciones o áreas específicas de la simulación.

A continuación, se muestran dos casos de uso en los que resulta más útil enviar datos mediante mensajería en lugar de extraerlos o leerlos a través de una suscripción.

Example 1: Enviar un comando a otra aplicación para cambiar la posición de una entidad

```
// 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);
```

En el lado receptor, la aplicación actualiza la posición de la entidad y la escribe en 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: Enviar un mensaje de creación de entidad a una aplicación espacial

```

struct WeaverMessage
{
    const Aws::WeaverRuntime::Api::TypeId messageType;
};

const Aws::WeaverRuntime::Api::TypeId k_createEntityMessageType = { 1 };

struct CreateEntityMessage : WeaverMessage
{
    const Vector3 position;
    const Aws::WeaverRuntime::Api::TypeId typeId;
};

CreateEntityMessage messageData {
    k_createEntityMessageType,
    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);

```

Por el lado receptor, la aplicación crea una nueva entidad en el State Fabric y actualiza su posición.

```

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)));
}

```

Consejos para trabajar con la mensajería

Resolver un punto final a partir de una posición o el nombre de una aplicación

Puede usar la `AllPartitions` función para obtener los límites espaciales y el ID de dominio que necesita para determinar los ID de las particiones de los mensajes y los destinos de los mensajes. Sin embargo, si conoce la posición a la que desea enviar el mensaje, pero no su ID de partición, puede utilizar la `MessageEndpointResolver` función.

```

/**
 * 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);
};

```


Serializar y deserializar la carga útil del mensaje

Puede utilizar las siguientes funciones para crear y leer cargas útiles de mensajes. Para obtener más información, consulta `MessagingUtils .h` en la biblioteca del SDK de aplicaciones de tu sistema 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());
}
```

Errores de mensajería y solución de problemas

Es posible que se produzcan los siguientes errores al utilizar las API de mensajería.

Errores de resolución de terminales

Estos errores pueden producirse antes de que una aplicación envíe un mensaje.

Verificación del nombre de dominio

Al enviar un mensaje a un punto final no válido, se produce el siguiente error:

```
ManifoldError::InvalidArgument {"No DomainId found for the given domain name" }
```

Esto puede ocurrir cuando intentas enviar un mensaje a una aplicación personalizada y esa aplicación personalizada aún no se ha unido a la simulación. Usa la `DescribeSimulation` API para asegurarte de que la aplicación personalizada se haya lanzado antes de enviarle un mensaje. Este comportamiento es el mismo en SimSpace Weaver Local y en Nube de AWS.

Verificación de posición

Al intentar resolver un punto final con un nombre de dominio válido pero una posición no válida, se produce el siguiente error.

```
ManifoldError::InvalidArgument {"Could not resolve endpoint from domain : DomainId  
{ value: domain-id } and position: Vector2F32 { x: x-position, y: y-position"} }
```

Te sugerimos que utilices el que se encuentra `MessageEndpointResolver` en la `MessageUtils` biblioteca incluida en el SDK de la SimSpace Weaver aplicación.

Errores al enviar mensajes

Se pueden producir los siguientes errores cuando una aplicación envía un mensaje.

Se ha superado el límite de envío de mensajes por aplicación y por marca

El límite actual de mensajes que se pueden enviar por aplicación y por marca de simulación es de 128. Las llamadas subsiguientes con la misma marca fallarán y se mostrará el siguiente error:

```
ManifoldError::CapacityExceeded {"At Max Outgoing Message capacity: {}", 128}
```

SimSpace Weaver intenta enviar los mensajes no enviados en la siguiente casilla. Reduzca la frecuencia de envío para resolver este problema. Combine las cargas útiles de mensajes inferiores al límite de 256 bytes para reducir el número de mensajes salientes.

Este comportamiento es el mismo dentro SimSpace Weaver Local y dentro del. Nube de AWS

Se ha superado el límite de tamaño de carga útil de los mensajes

El límite actual para el tamaño de la carga útil de los mensajes es de 256 bytes tanto en el como SimSpace Weaver Local en el. Nube de AWS Al enviar un mensaje con una carga útil superior a 256 bytes, se produce el siguiente error:

```
ManifoldError::CapacityExceeded {"Message data too large! Max size: {}", 256}
```

SimSpace Weaver comprueba cada mensaje y solo rechaza los que superan el límite. Por ejemplo, si tu aplicación intenta enviar 10 mensajes y uno no supera la comprobación, solo se rechazará ese mensaje. SimSpace Weaver envía los 9 mensajes restantes.

Este comportamiento es el mismo en SimSpace Weaver Local y en Nube de AWS.

El destino es el mismo que el origen

Las aplicaciones no pueden enviar mensajes a las particiones de su propiedad. Si una aplicación envía un mensaje a una partición de su propiedad, aparece el siguiente error.

```
ManifoldError::InvalidArgument { "Destination is the same as source" }
```

Este comportamiento es el mismo en SimSpace Weaver Local y en Nube de AWS.

Mensajes de Best Esfuerzo

SimSpace Weaver no garantiza la entrega del mensaje. El servicio intentará completar la entrega de los mensajes al marcar la siguiente casilla de simulación, pero es posible que los mensajes se pierdan o se retrasen.

Mejores prácticas al trabajar con SimSpace Weaver

Recomendamos las siguientes prácticas recomendadas al trabajar con SimSpace Weaver.

Temas

- [Configuración de alarmas de facturación](#)
- [Uso SimSpace Weaver Local](#)
- [Puede detener las simulaciones que no necesite.](#)
- [Elimine los recursos que no necesite](#)
- [Disposición de copias de seguridad](#)

Configuración de alarmas de facturación

Es fácil aprovisionar los recursos AWS y dejarlos en funcionamiento todo el tiempo, incluso cuando ya no se necesitan. Esto puede generar costes desmesurados que pueden sorprenderle cuando reciba su factura. Puedes configurar una alarma en Amazon CloudWatch que se active y te notifique cuando tus costes superen el umbral que hayas establecido. Puede examinar sus costes mediante herramientas de administración de costes. Para obtener más información, consulte:

- [Crea una alarma de facturación para controlar tus AWS cargos estimados](#)
- [¿Qué es AWS Cost Management](#)

Uso SimSpace Weaver Local

Le recomendamos que utilice SimSpace Weaver Local para desarrollar y probar sus simulaciones antes de cargarlas al SimSpace Weaver servicio del Nube de AWS. Los beneficios de desarrollar con SimSpace Weaver Local incluyen:

- No es necesario esperar a que se suban archivos grandes.
- No hay límite en cuanto a la cantidad de simulaciones locales que puede crear.
- No se le cobrará por el tiempo de procesamiento en su equipo local.
- Acceso directo a la salida de la consola desde tus aplicaciones.
- Modifique, reconstruya y reinicie la simulación local sin tener que volver a crearla en el Nube de AWS

Puede detener las simulaciones que no necesite.

Podrá ver los cargos de facturación de una simulación mientras está en ejecución. Para dejar de recibir cargos por una simulación, deberá detenerla. La ejecución de simulaciones también se tiene en cuenta a la hora de calcular el número máximo de simulaciones. Una simulación en ejecución que tenga el registro configurado también puede generar grandes cantidades de registros, por lo que también se cobrarán gastos de facturación. Debe interrumpir cualquier simulación que no necesite para dejar de recibir cargos adicionales.

Important

Detener el reloj de simulación no detiene la simulación, sino que simplemente deja de publicar graduaciones en sus aplicaciones. No puede reiniciar una simulación después de detenerla.

Elimine los recursos que no necesite

Cada simulación que cree SimSpace Weaver también crea recursos en otros AWS servicios. Puede obtener los cargos de facturación de los recursos y los datos de estos otros servicios. Las simulaciones en ejecución y las fallidas se tienen en cuenta a la hora de calcular la cantidad máxima de simulaciones. Debe eliminar las simulaciones fallidas innecesarias para poder iniciar nuevas simulaciones. Al eliminar una simulación, es posible que no se eliminen los recursos de la simulación que existen en otros AWS servicios. Por ejemplo, los datos de registro de simulación de Amazon CloudWatch Logs permanecerán allí hasta que los elimine. Se le cobrarán los gastos de facturación por esos datos de registro. Debería limpiar todos los recursos asociados para sus simulaciones si ya no los necesita.

Disposición de copias de seguridad

Es una buena idea disponer de copias de seguridad y planes de respaldo para todo. No debes dar por sentado que, solo porque tus datos estén ahí, no tienes AWS que hacer una copia de seguridad de ellos. Debe crear su propio sistema si necesita hacer una copia de seguridad del estado de la simulación. Considere la posibilidad de utilizar varias Regiones de AWS y de tener un plan establecido para poder cambiar rápidamente su carga de trabajo de producción a otra Región de AWS si es necesario. Para obtener más información sobre Regiones de AWS ese soporte SimSpace Weaver, consulte [SimSpace Puntos finales y cuotas de Weaver](#).

Seguridad en AWS SimSpace Weaver

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de los centros de datos y las arquitecturas de red diseñados para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre AWS usted y usted. El [modelo de responsabilidad compartida](#) la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la Nube de AWS. AWS también le proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los [AWS programas](#) de de . Para obtener más información sobre los programas de cumplimiento aplicables AWS SimSpace Weaver, consulte [AWS Servicios incluidos en el ámbito de aplicación por programa de conformidad y AWS servicios incluidos](#) .
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. Usted también es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos aplicables.

Esta documentación le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza SimSpace Weaver. Los siguientes temas muestran cómo configurarlo SimSpace Weaver para cumplir sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que le ayudan a supervisar y proteger sus SimSpace Weaver recursos.

Temas

- [Protección de datos en AWS SimSpace Weaver](#)
- [Identity and Access Management para AWS SimSpace Weaver](#)
- [Registro y supervisión de eventos de seguridad AWS SimSpace Weaver](#)
- [Validación de conformidad para AWS SimSpace Weaver](#)
- [Resiliencia en AWS SimSpace Weaver](#)
- [Seguridad de la infraestructura en AWS SimSpace Weaver](#)
- [Análisis de configuración y vulnerabilidad en AWS SimSpace Weaver](#)
- [Prácticas recomendadas de seguridad para SimSpace Weaver](#)

Protección de datos en AWS SimSpace Weaver

El modelo de [responsabilidad AWS compartida modelo](#) se aplica a la protección de datos en AWS SimSpace Weaver. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global que ejecuta todos los Nube de AWS. Usted es responsable de mantener el control sobre el contenido alojado en esta infraestructura. Usted también es responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte la sección [Privacidad de datos FAQ](#). Para obtener información sobre la protección de datos en Europa, consulte el [modelo de responsabilidad AWS compartida y](#) la entrada del GDPR blog sobre AWS seguridad.

Con fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utilice la autenticación multifactorial (MFA) con cada cuenta.
- UseSSL/TLSpara comunicarse con AWS los recursos. Necesitamos TLS 1.2 y recomendamos TLS 1.3.
- Configure API y registre la actividad del usuario con AWS CloudTrail. Para obtener información sobre el uso de CloudTrail senderos para capturar AWS actividades, consulte [Cómo trabajar con CloudTrail senderos](#) en la Guía del AWS CloudTrail usuario.
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.
- Utilice servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita entre FIPS 140 y 3 módulos criptográficos validados para acceder a AWS través de una interfaz de línea de comandos o unaAPI, utilice un FIPS terminal. Para obtener más información sobre los FIPS puntos finales disponibles, consulte la [Norma federal de procesamiento de información \(\) FIPS 140-3](#).

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como, por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja con SimSpace Weaver o Servicios de AWS utiliza la consola, API AWS CLI, o. AWS SDKs Cualquier dato que ingrese en etiquetas o

campos de formato libre utilizados para nombres se puede emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, le recomendamos encarecidamente que no incluya información sobre las credenciales URL para validar la solicitud a ese servidor.

Cifrado en reposo

Los datos se consideran en reposo cuando se encuentran en un almacenamiento de datos no volátil (persistente), como un disco. Los datos ubicados en un almacenamiento de datos volátil, como la memoria y los registros, no se consideran en reposo.

Cuando lo utilices SimSpace Weaver, los únicos datos en reposo son:

- Aplicaciones y esquemas que cargue en Amazon Simple Storage Service (Amazon S3)
- Datos del registro de simulación almacenados en Amazon CloudWatch

Los demás datos que se SimSpace Weaver utilizan internamente no se conservan después de detener la simulación.

Para obtener información sobre cómo cifrar sus datos en reposo, consulte:

- [Cifrado de datos en Amazon S3](#)
- [Cifrar sus datos de registro](#)

Cifrado en tránsito

Las conexiones a SimSpace Weaver API través de AWS Command Line Interface (AWS CLI) y a la SimSpace Weaver aplicación SDK utilizan el TLS cifrado con el [proceso de firma Signature Version 4](#). AWS SDK AWS administra la autenticación mediante las políticas de acceso IAM definidas para las credenciales de seguridad que utiliza para conectarse.

Internamente, TLS se SimSpace Weaver utiliza para conectarse a otros AWS servicios que utiliza.

Important

Las comunicaciones entre sus aplicaciones y sus clientes no implican SimSpace Weaver. Es su responsabilidad cifrar las comunicaciones con los clientes de simulación, si es necesario.

Le recomendamos que cree una solución para cifrar todos los datos en tránsito entre las conexiones de los clientes.

Para obtener más información sobre AWS los servicios compatibles con sus soluciones de cifrado, consulte [el blog AWS de seguridad](#).

Privacidad del tráfico entre redes

SimSpace Weaver los recursos de cómputo residen en 1 Amazon VPC compartido por todos SimSpace Weaver los clientes. Todo el tráfico del SimSpace Weaver servicio interno permanece dentro de la AWS red y no viaja por Internet. La comunicación entre los clientes de simulación y sus aplicaciones se realiza a través de Internet.

Identity and Access Management para AWS SimSpace Weaver

AWS Identity and Access Management (IAM) es una Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a AWS los recursos. IAM los administradores controlan quién puede autenticarse (iniciar sesión) y quién está autorizado (tiene permisos) para usar SimSpace Weaver los recursos. IAM es una Servicio de AWS que puede utilizar sin coste adicional.

Temas

- [Público](#)
- [Autenticación con identidades](#)
- [Administración de acceso mediante políticas](#)
- [¿Cómo AWS SimSpace Weaver funciona con IAM](#)
- [Ejemplos de políticas basadas en la identidad para AWS SimSpace Weaver](#)
- [Permisos que SimSpace Weaver crea para usted](#)
- [Prevención de la sustitución confusa entre servicios](#)
- [Solución de problemas AWS SimSpace Weaver de identidad y acceso](#)

Público

La forma de usar AWS Identity and Access Management (IAM) varía según el trabajo en el que se realice SimSpace Weaver.

Usuario del servicio: si utiliza el SimSpace Weaver servicio para realizar su trabajo, el administrador le proporcionará las credenciales y los permisos que necesita. A medida que vaya utilizando más SimSpace Weaver funciones para realizar su trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarlo a solicitar los permisos correctos al administrador. Si no puede acceder a una característica en SimSpace Weaver, consulte [Solución de problemas AWS SimSpace Weaver de identidad y acceso](#).

Administrador de servicios: si estás a cargo de SimSpace Weaver los recursos de tu empresa, probablemente tengas acceso total a ellos SimSpace Weaver. Su trabajo consiste en determinar a qué SimSpace Weaver funciones y recursos deben acceder los usuarios del servicio. A continuación, debe enviar solicitudes a su IAM administrador para cambiar los permisos de los usuarios del servicio. Revise la información de esta página para comprender los conceptos básicos de IAM. Para obtener más información sobre cómo su empresa puede utilizar IAM con SimSpace Weaver, consulte [¿Cómo AWS SimSpace Weaver funciona con IAM](#).

IAM administrador: si es IAM administrador, puede que desee obtener más información sobre cómo puede redactar políticas para administrar el acceso a ellas SimSpace Weaver. Para ver ejemplos de políticas SimSpace Weaver basadas en la identidad que puede utilizar IAM, consulte [Ejemplos de políticas basadas en la identidad para AWS SimSpace Weaver](#)

Autenticación con identidades

La autenticación es la forma de iniciar sesión AWS con sus credenciales de identidad. Debe estar autenticado (con quien haya iniciado sesión AWS) como IAM usuario o asumiendo un IAM rol.

Usuario raíz de la cuenta de AWS

Puede iniciar sesión AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (IAM Identity Center), la autenticación de inicio de sesión único de su empresa y sus credenciales de Google o Facebook son ejemplos de identidades federadas. Al iniciar sesión como una identidad federada, el administrador configuró previamente la federación de identidades mediante roles. IAM Cuando accede AWS mediante la federación, asume indirectamente un rol.

Según el tipo de usuario que sea, puede iniciar sesión en el portal AWS Management Console o en el de AWS acceso. Para obtener más información sobre cómo iniciar sesión AWS, consulte [Cómo iniciar sesión Cuenta de AWS en su](#) Guía del AWS Sign-In usuario.

Si accede AWS mediante programación, AWS incluye un kit de desarrollo de software (SDK) y una interfaz de línea de comandos (CLI) para firmar criptográficamente sus solicitudes con sus

credenciales. Si no utilizas AWS herramientas, debes firmar las solicitudes tú mismo. Para obtener más información sobre cómo usar el método recomendado para firmar las solicitudes usted mismo, consulte [Firmar AWS API las solicitudes](#) en la Guía del IAM usuario.

Independientemente del método de autenticación que use, es posible que deba proporcionar información de seguridad adicional. Por ejemplo, le AWS recomienda que utilice la autenticación multifactorial (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte [Autenticación multifactorial](#) en la Guía del AWS IAM Identity Center usuario y [Uso de la autenticación multifactorial \(MFA\) AWS en](#) la Guía del IAM usuario.

Cuenta de AWS usuario root

Al crear una Cuenta de AWS, comienza con una identidad de inicio de sesión que tiene acceso completo a todos Servicios de AWS los recursos de la cuenta. Esta identidad se denomina usuario Cuenta de AWS raíz y se accede a ella iniciando sesión con la dirección de correo electrónico y la contraseña que utilizaste para crear la cuenta. Recomendamos encarecidamente que no utilice el usuario raíz para sus tareas diarias. Proteja las credenciales del usuario raíz y utilícelas solo para las tareas que solo el usuario raíz pueda realizar. Para ver la lista completa de tareas que requieren que inicie sesión como usuario root, consulte [Tareas que requieren credenciales de usuario root](#) en la Guía del IAM usuario.

Identidad federada

Como práctica recomendada, exija a los usuarios humanos, incluidos los que requieren acceso de administrador, que utilicen la federación con un proveedor de identidades para acceder Servicios de AWS mediante credenciales temporales.

Una identidad federada es un usuario del directorio de usuarios de su empresa, un proveedor de identidades web AWS Directory Service, el directorio del Centro de Identidad o cualquier usuario al que acceda Servicios de AWS mediante las credenciales proporcionadas a través de una fuente de identidad. Cuando las identidades federadas acceden Cuentas de AWS, asumen funciones y las funciones proporcionan credenciales temporales.

Para una administración de acceso centralizada, le recomendamos que utilice AWS IAM Identity Center. Puede crear usuarios y grupos en IAM Identity Center, o puede conectarse y sincronizarse con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todas sus aplicaciones Cuentas de AWS . Para obtener información sobre IAM Identity Center, consulte [¿Qué es IAM Identity Center?](#) en la Guía AWS IAM Identity Center del usuario.

Usuarios y grupos de IAM

Un [IAMusuario](#) es una identidad propia Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos utilizar credenciales temporales en lugar de crear IAM usuarios con credenciales de larga duración, como contraseñas y claves de acceso. Sin embargo, si tiene casos de uso específicos que requieren credenciales a largo plazo con IAM los usuarios, le recomendamos que rote las claves de acceso. Para obtener más información, consulte [Rotar las claves de acceso con regularidad para los casos de uso que requieran credenciales de larga duración](#) en la Guía del IAM usuario.

Un [IAMgrupo](#) es una identidad que especifica un conjunto de IAM usuarios. No puede iniciar sesión como grupo. Puede usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos para grandes conjuntos de usuarios. Por ejemplo, puede asignar un nombre a un grupo IAMAdminsy concederle permisos para administrar IAM los recursos.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales de larga duración permanentes; no obstante, los roles proporcionan credenciales temporales. Para obtener más información, consulte [Cuándo crear un IAM usuario \(en lugar de un rol\)](#) en la Guía del IAM usuario.

IAMroles

Un [IAMrol](#) es una identidad dentro de tu Cuenta de AWS que tiene permisos específicos. Es similar a un IAM usuario, pero no está asociado a una persona específica. Puede asumir temporalmente un IAM rol en el AWS Management Console [cambiando de rol](#). Puede asumir un rol llamando a una AWS API operación AWS CLI o utilizando una operación personalizadaURL. Para obtener más información sobre los métodos de uso de roles, consulte [Uso de IAM roles](#) en la Guía del IAM usuario.

IAMlos roles con credenciales temporales son útiles en las siguientes situaciones:

- Acceso de usuario federado: para asignar permisos a una identidad federada, puede crear un rol y definir sus permisos. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información sobre los roles para la federación, consulte [Creación de un rol para un proveedor de identidad externo](#) en la Guía del IAM usuario. Si usa IAM Identity Center, configura un conjunto de permisos. Para controlar a qué pueden acceder sus identidades después de autenticarse, IAM Identity Center correlaciona el

conjunto de permisos con un rol en IAM. Para obtener información acerca de los conjuntos de permisos, consulte [Conjuntos de permisos](#) en la Guía del usuario de AWS IAM Identity Center.

- **Permisos IAM de usuario temporales:** un IAM usuario o rol puede asumir un IAM rol para asumir temporalmente diferentes permisos para una tarea específica.
- **Acceso multicuenta:** puedes usar un IAM rol para permitir que alguien (un responsable de confianza) de una cuenta diferente acceda a los recursos de tu cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunos Servicios de AWS, puedes adjuntar una política directamente a un recurso (en lugar de usar un rol como proxy). Para conocer la diferencia entre las funciones y las políticas basadas en recursos para el acceso multicuenta, consulta el tema sobre el acceso a los [recursos entre cuentas IAM en](#) la Guía del IAM usuario.
- **Acceso entre servicios:** algunos Servicios de AWS utilizan funciones en otros. Servicios de AWS Por ejemplo, cuando realizas una llamada en un servicio, es habitual que ese servicio ejecute aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado al servicio.
- **Sesiones de acceso directo (FAS):** cuando utilizas un IAM usuario o un rol para realizar acciones en AWS ellas, se te considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos del principal que llama a un Servicio de AWS, junto con los que solicitan, Servicio de AWS para realizar solicitudes a los servicios descendentes. FAS las solicitudes solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información detallada sobre la política a la hora de realizar FAS solicitudes, consulte [Reenviar las sesiones de acceso](#).
- **Función de servicio:** una función de servicio es una [IAM función](#) que un servicio asume para realizar acciones en su nombre. Un IAM administrador puede crear, modificar y eliminar un rol de servicio desde dentro IAM. Para obtener más información, consulte [Crear un rol para delegar permisos Servicio de AWS en un rol en el IAM Manual del usuario](#).
- **Función vinculada a un servicio:** una función vinculada a un servicio es un tipo de función de servicio que está vinculada a un Servicio de AWS. El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en su Cuenta de AWS y son propiedad del servicio. Un IAM administrador puede ver los permisos de los roles vinculados al servicio, pero no editarlos.
- **Aplicaciones que se ejecutan en Amazon EC2:** puedes usar un IAM rol para administrar las credenciales temporales de las aplicaciones que se ejecutan en una EC2 instancia y que realizan

AWS CLI o AWS API solicitan. Esto es preferible a almacenar las claves de acceso en la EC2 instancia. Para asignar un AWS rol a una EC2 instancia y ponerlo a disposición de todas sus aplicaciones, debe crear un perfil de instancia adjunto a la instancia. Un perfil de instancia contiene el rol y permite que los programas que se ejecutan en la EC2 instancia obtengan credenciales temporales. Para obtener más información, consulte [Uso de un IAM rol para conceder permisos a aplicaciones que se ejecutan en EC2 instancias de Amazon](#) en la Guía del IAM usuario.

Para saber si se deben usar IAM roles o IAM usuarios, consulte [Cuándo crear un IAM rol \(en lugar de un usuario\)](#) en la Guía del IAM usuario.

Administración de acceso mediante políticas

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política es un objeto AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando un director (usuario, usuario raíz o sesión de rol) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan AWS como JSON documentos. Para obtener más información sobre la estructura y el contenido de los documentos de JSON políticas, consulte [Descripción general de JSON las políticas](#) en la Guía del IAM usuario.

Los administradores pueden usar AWS JSON las políticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Para conceder a los usuarios permiso para realizar acciones en los recursos que necesitan, un IAM administrador puede crear IAM políticas. A continuación, el administrador puede añadir las IAM políticas a las funciones y los usuarios pueden asumir las funciones.

IAM las políticas definen los permisos para una acción independientemente del método que se utilice para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción `iam:GetRole`. Un usuario con esa política puede obtener información sobre el rol de AWS Management Console AWS CLI, el o el AWS API.

Políticas basadas en identidad

Las políticas basadas en la identidad son documentos de política de JSON permisos que se pueden adjuntar a una identidad, como un IAM usuario, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en

qué condiciones. Para obtener información sobre cómo crear una política basada en la identidad, consulte [Creación de IAM políticas](#) en la Guía del usuario. IAM

Las políticas basadas en identidades pueden clasificarse además como políticas insertadas o políticas administradas. Las políticas insertadas se integran directamente en un único usuario, grupo o rol. Las políticas administradas son políticas independientes que puede adjuntar a varios usuarios, grupos y funciones de su empresa. Cuenta de AWS Las políticas administradas incluyen políticas AWS administradas y políticas administradas por el cliente. Para saber cómo elegir entre una política gestionada o una política integrada, consulte [Elegir entre políticas gestionadas y políticas integradas en la Guía del IAM](#) usuario.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de JSON política que se adjuntan a un recurso. Algunos ejemplos de políticas basadas en recursos son las políticas de confianza de IAM roles y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe [especificar una entidad principal](#) en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puede usar políticas AWS administradas desde una política IAM basada en recursos.

Listas de control de acceso () ACLs

Las listas de control de acceso (ACLs) controlan qué responsables (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLs son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de JSON políticas.

Amazon S3 AWS WAF y Amazon VPC son ejemplos de servicios compatibles ACLs. Para obtener más información ACLs, consulte la [descripción general de la lista de control de acceso \(ACL\)](#) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite tipos de políticas adicionales y menos comunes. Estos tipos de políticas pueden establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- **Límites de permisos:** un límite de permisos es una función avanzada en la que se establecen los permisos máximos que una política basada en la identidad puede conceder a una IAM entidad (IAMusuario o rol). Puede establecer un límite de permisos para una entidad. Los permisos resultantes son la intersección de las políticas basadas en la identidad de la entidad y los límites de permisos. Las políticas basadas en recursos que especifiquen el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites de los permisos, consulte los [límites de los permisos para IAM las entidades](#) en la Guía del IAMusuario.
- **Políticas de control de servicios (SCPs):** SCPs son JSON políticas que especifican los permisos máximos para una organización o unidad organizativa (OU) en AWS Organizations. AWS Organizations es un servicio para agrupar y administrar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilitas todas las funciones de una organización, puedes aplicar políticas de control de servicios (SCPs) a una o a todas tus cuentas. SCPLimita los permisos de las entidades en las cuentas de los miembros, incluidas las de cada una Usuario raíz de la cuenta de AWS. Para obtener más información sobre OrganizationsSCPs, consulte las [políticas de control de servicios](#) en la Guía del AWS Organizations usuario.
- **Políticas de sesión:** las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades del rol y las políticas de la sesión. Los permisos también pueden proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información, consulte [las políticas de sesión](#) en la Guía del IAM usuario.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo se AWS determina si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la [lógica de evaluación de políticas](#) en la Guía del IAM usuario.

¿Cómo AWS SimSpace Weaver funciona con IAM

Antes de administrar el IAM acceso a SimSpace Weaver, infórmese sobre IAM las funciones disponibles para su uso SimSpace Weaver.

IAM funciones que puedes usar con AWS SimSpace Weaver

IAM característica	SimSpace Weaver apoyo
Políticas basadas en identidades	Sí
Políticas basadas en recursos	No
Acciones de políticas	Sí
Recursos de políticas	Sí
Claves de condición de política (específicas del servicio)	Sí
ACLs	No
ABAC(etiquetas en las políticas)	Sí
Credenciales temporales	Sí
Permisos de entidades principales	Sí
Roles de servicio	Sí
Roles vinculados al servicio	No

Para obtener una visión general de cómo SimSpace Weaver funcionan otros AWS servicios con la mayoría de las IAM funciones, consulte [AWS los servicios con los que funcionan IAM](#) en la Guía del IAM usuario.

Políticas basadas en la identidad para SimSpace Weaver

Compatibilidad con las políticas basadas en identidad: sí

Las políticas basadas en la identidad son documentos de política de JSON permisos que se pueden adjuntar a una identidad, como un IAM usuario, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener información sobre cómo crear una política basada en la identidad, consulte [Creación de IAM políticas](#) en la Guía del usuario. IAM

Con las políticas IAM basadas en la identidad, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está adjunto. Para obtener más información sobre todos los elementos que puede utilizar en una JSON política, consulte la [referencia sobre los elementos de la IAM JSON política](#) en la Guía del IAM usuario.

Ejemplos de políticas basadas en la identidad para SimSpace Weaver

Para ver ejemplos de políticas SimSpace Weaver basadas en la identidad, consulte. [Ejemplos de políticas basadas en la identidad para AWS SimSpace Weaver](#)

Políticas basadas en recursos incluidas SimSpace Weaver

Admite políticas basadas en recursos: no

Las políticas basadas en recursos son documentos de JSON política que se adjuntan a un recurso. Algunos ejemplos de políticas basadas en recursos son las políticas de confianza de IAM roles y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe [especificar una entidad principal](#) en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Para habilitar el acceso entre cuentas, puede especificar una cuenta completa o IAM entidades de otra cuenta como principales en una política basada en recursos. Añadir a una política en función de recursos una entidad principal entre cuentas es solo una parte del establecimiento de una relación de confianza. Cuando el principal y el recurso son diferentes Cuentas de AWS, el IAM administrador de la cuenta de confianza también debe conceder permiso a la entidad principal (usuario o rol) para acceder al recurso. Para conceder el permiso, adjunte la entidad a una política basada en identidad. Sin embargo, si la política en función de recursos concede el acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para obtener más información, consulte el [tema Acceso a recursos entre cuentas IAM en](#) la Guía del IAM usuario.

Acciones políticas para SimSpace Weaver

Compatibilidad con las acciones de política: sí

Los administradores pueden usar AWS JSON políticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El `Action` elemento de una JSON política describe las acciones que puede utilizar para permitir o denegar el acceso en una política. Las acciones de política suelen tener el mismo nombre que la AWS API operación asociada. Hay algunas excepciones, como las acciones que solo permiten permisos y que no tienen una operación coincidente. API También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de SimSpace Weaver acciones, consulte las [acciones definidas AWS SimSpace Weaver en la Referencia de](#) autorización de servicios.

Las acciones políticas SimSpace Weaver utilizan el siguiente prefijo antes de la acción:

```
simspaceweaver
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

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

Para ver ejemplos de políticas SimSpace Weaver basadas en la identidad, consulte. [Ejemplos de políticas basadas en la identidad para AWS SimSpace Weaver](#)

Recursos de políticas para SimSpace Weaver

Compatibilidad con los recursos de políticas: sí

Los administradores pueden usar AWS JSON políticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` JSON de política especifica el objeto o los objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso mediante su [nombre de recurso de Amazon \(ARN\)](#). Puede hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

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

Para ver una lista de los tipos de SimSpace Weaver recursos y sus respectivos tiposARNs, consulte [los recursos definidos AWS SimSpace Weaver](#) en la Referencia de autorización de servicio. Para saber con qué acciones puede especificar cada recurso, consulte [Acciones definidas por AWS SimSpace Weaver](#). ARN

Para ver ejemplos de políticas SimSpace Weaver basadas en la identidad, consulte. [Ejemplos de políticas basadas en la identidad para AWS SimSpace Weaver](#)

Claves de condición de la política para SimSpace Weaver

Compatibilidad con claves de condición de políticas específicas del servicio: sí

Los administradores pueden usar AWS JSON políticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puede crear expresiones condicionales que utilicen [operadores de condición](#), tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación AND lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición mediante una OR operación lógica. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puede utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puede conceder a un IAM usuario permiso para acceder a un recurso solo si está etiquetado con su nombre de IAM usuario. Para obtener más información, consulte [los elementos IAM de la política: variables y etiquetas](#) en la Guía del IAM usuario.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición AWS globales, consulte las claves de [contexto de condición AWS globales](#) en la Guía del IAM usuario.

Para ver una lista de claves de SimSpace Weaver condición, consulte las [claves de condición AWS SimSpace Weaver](#) en la Referencia de autorización de servicio. Para saber con qué acciones y recursos puede utilizar una clave de condición, consulte [Acciones definidas por AWS SimSpace Weaver](#).

Para ver ejemplos de políticas SimSpace Weaver basadas en la identidad, consulte. [Ejemplos de políticas basadas en la identidad para AWS SimSpace Weaver](#)

Listas de control de acceso () ACLs en SimSpace Weaver

SoportaACLs: No

Las listas de control de acceso (ACLs) controlan qué directores (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLs son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de JSON políticas.

Control de acceso basado en atributos () con ABAC SimSpace Weaver

Soportes ABAC (etiquetas en las políticas): Sí

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define los permisos en función de los atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a IAM entidades (usuarios o roles) y a muchos AWS recursos. Etiquetar entidades y recursos es el primer paso de ABAC. Luego, diseñe ABAC políticas para permitir las operaciones cuando la etiqueta del principal coincida con la etiqueta del recurso al que está intentando acceder.

ABAC es útil en entornos de rápido crecimiento y ayuda en situaciones en las que la administración de políticas se vuelve engorrosa.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el [elemento de condición](#) de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información al respecto ABAC, consulte [¿Qué es? ABAC](#) en la Guía IAM del usuario. Para ver un tutorial con los pasos de configuración ABAC, consulte [Usar el control de acceso basado en atributos \(ABAC\)](#) en la Guía del IAM usuario.

Usar credenciales temporales con SimSpace Weaver

Compatibilidad con credenciales temporales: sí

Algunos Servicios de AWS no funcionan cuando inicias sesión con credenciales temporales. Para obtener información adicional, incluida la información sobre cuáles Servicios de AWS funcionan con credenciales temporales, consulta la sección [Servicios de AWS Cómo trabajar con credenciales temporales IAM](#) en la Guía del IAM usuario.

Está utilizando credenciales temporales si inicia sesión AWS Management Console con cualquier método excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accedes AWS mediante el enlace de inicio de sesión único (SSO) de tu empresa, ese proceso crea automáticamente credenciales temporales. También crea credenciales temporales de forma automática cuando inicia sesión en la consola como usuario y luego cambia de rol. Para obtener más información sobre el cambio de rol, consulte [Cambiar a un rol \(consola\)](#) en la Guía del IAM usuario.

Puede crear credenciales temporales manualmente con la tecla AWS CLI o AWS API. A continuación, puede utilizar esas credenciales temporales para acceder AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para obtener más información, consulte [Credenciales de seguridad temporales en IAM](#).

Permisos principales entre servicios para SimSpace Weaver

Admite sesiones de acceso directo (FAS): Sí

Cuando utilizas un IAM usuario o un rol para realizar acciones en AWS, se te considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos del principal que llama a un Servicio de AWS, junto con los que solicita, Servicio de AWS para realizar solicitudes a los servicios descendentes. FAS las solicitudes solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información detallada sobre la política a la hora de realizar FAS solicitudes, consulte [Reenviar las sesiones de acceso](#).

Roles de servicio para SimSpace Weaver

Compatibilidad con roles de servicio: sí

Una función de servicio es una [IAM función](#) que un servicio asume para realizar acciones en su nombre. Un IAM administrador puede crear, modificar y eliminar un rol de servicio desde dentro IAM.

Para obtener más información, consulte [Crear un rol para delegar permisos Servicio de AWS en un rol](#) en el IAMManual del usuario.

Warning

Cambiar los permisos de un rol de servicio podría afectar a SimSpace Weaver la funcionalidad. Edite las funciones de servicio solo cuando se SimSpace Weaver proporcionen instrucciones para hacerlo.

Los SDK scripts de la SimSpace Weaver aplicación utilizan una AWS CloudFormation plantilla para crear recursos en otros AWS servicios que sirvan de apoyo a la simulación. Uno de estos recursos es la función de la aplicación para la simulación. SimSpace Weaver asume la función de aplicación para realizar acciones Cuenta de AWS en su nombre, como escribir datos de registro en los CloudWatch registros. Para obtener más información acerca de los roles de aplicación, consulte [Permisos que SimSpace Weaver crea para usted](#).

Funciones vinculadas al servicio para SimSpace Weaver

Compatibilidad con roles vinculados al servicio: no

Un rol vinculado a un servicio es un tipo de rol de servicio que está vinculado a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un IAM administrador puede ver los permisos de los roles vinculados al servicio, pero no editarlos.

Para obtener más información sobre la creación o la administración de funciones vinculadas a un servicio, consulte los [AWS servicios](#) que funcionan con. IAM Busque un servicio en la tabla que incluya Yes en la columna Rol vinculado a un servicio. Seleccione el vínculo Sí para ver la documentación acerca del rol vinculado a servicios para ese servicio.

Ejemplos de políticas basadas en la identidad para AWS SimSpace Weaver

De forma predeterminada, los usuarios y roles no tienen permiso para crear, ver ni modificar recursos de SimSpace Weaver . Tampoco pueden realizar tareas con AWS Management Console, AWS Command Line Interface (AWS CLI) o. AWS API Para conceder a los usuarios permiso para realizar acciones en los recursos que necesitan, un IAM administrador puede crear IAM políticas. A continuación, el administrador puede añadir las IAM políticas a las funciones y los usuarios pueden asumir las funciones.

Para obtener información sobre cómo crear una política IAM basada en la identidad mediante estos documentos de JSON política de ejemplo, consulte [Creación de IAM políticas](#) en la Guía del IAM usuario.

Para obtener más información sobre las acciones y los tipos de recursos definidos SimSpace Weaver, incluido el formato de cada uno de los tipos de recursos, consulte [las claves de condición, recursos y acciones de la Referencia AWS SimSpace Weaver](#) de autorización de servicios. ARNs

Temas

- [Prácticas recomendadas sobre las políticas](#)
- [Mediante la consola de SimSpace Weaver](#)
- [Cómo permitir a los usuarios consultar sus propios permisos](#)
- [Permiso a los usuarios para crear y ejecutar simulaciones](#)

Prácticas recomendadas sobre las políticas

Las políticas basadas en la identidad determinan si alguien puede crear SimSpace Weaver recursos de tu cuenta, acceder a ellos o eliminarlos. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están disponibles en su Cuenta de AWS. Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Para obtener más información, consulte [las políticas AWS gestionadas](#) o [las políticas AWS gestionadas para las funciones laborales](#) en la Guía del IAM usuario.
- Aplique permisos con privilegios mínimos: cuando establezca permisos con IAM políticas, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Para obtener más información sobre cómo IAM aplicar permisos, consulte [Políticas y permisos IAM en](#) la IAM Guía del usuario.
- Utilice las condiciones en IAM las políticas para restringir aún más el acceso: puede añadir una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de política para especificar que todas las solicitudes deben enviarse

mediante SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo AWS CloudFormation. Para obtener más información, consulte [los elementos IAM JSON de la política: Condición](#) en la Guía del IAM usuario.

- Utilice IAM Access Analyzer para validar sus IAM políticas y garantizar permisos seguros y funcionales: IAM Access Analyzer valida las políticas nuevas y existentes para que se ajusten al lenguaje de las políticas (JSON) y IAM a las IAM mejores prácticas. IAM Access Analyzer proporciona más de 100 comprobaciones de políticas y recomendaciones prácticas para ayudarlo a crear políticas seguras y funcionales. Para obtener más información, consulte la [validación de políticas de IAM Access Analyzer](#) en la Guía del IAM usuario.
- Requerir autenticación multifactorial (MFA): si se encuentra en una situación en la que se requieren IAM usuarios o un usuario raíz Cuenta de AWS, actívela MFA para aumentar la seguridad. Para solicitarlo MFA cuando se convoque a API las operaciones, añada MFA condiciones a sus políticas. Para obtener más información, consulte [Configuración del API acceso MFA protegido](#) en la Guía del IAM usuario.

Para obtener más información sobre las prácticas recomendadas IAM, consulte las [prácticas recomendadas de seguridad IAM en](#) la Guía del IAM usuario.

Mediante la consola de SimSpace Weaver

Para acceder a la AWS SimSpace Weaver consola, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los SimSpace Weaver recursos de su cuenta Cuenta de AWS. Si crea una política basada en identidades que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles) que tengan esa política.

No es necesario que concedas permisos mínimos de consola a los usuarios que realicen llamadas únicamente al AWS CLI o al AWS API. En su lugar, permita el acceso únicamente a las acciones que coincidan con la API operación que están intentando realizar.

Para garantizar que los usuarios y los roles puedan seguir utilizando la SimSpace Weaver consola, adjunte también la política *ReadOnly* AWS gestionada SimSpace Weaver *ConsoleAccess* o la política gestionada a las entidades. Para obtener más información, consulte [Añadir permisos a un usuario](#) en la Guía del IAM usuario.

Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo se muestra cómo se puede crear una política que permita a IAM los usuarios ver las políticas integradas y administradas asociadas a su identidad de usuario. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la tecla o. AWS CLI AWS API

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

Permiso a los usuarios para crear y ejecutar simulaciones

Este ejemplo IAM de política proporciona los permisos básicos necesarios para crear y ejecutar simulaciones en 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"
        }
    }
}
]
```

Permisos que SimSpace Weaver crea para usted

Al crear un SimSpace Weaver proyecto, el servicio creará un rol AWS Identity and Access Management (IAM) con el nombre `weaver-project-name-app-role` y una política de IAM confianza. La política de confianza SimSpace Weaver permite asumir el rol para que pueda realizar operaciones por usted.

Política de permisos del rol de aplicación

El rol de la aplicación de simulación utiliza la siguiente política de permisos.

```
{
  "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": "*"
    }
]
}

```

Política de confianza del rol de aplicación

SimSpace Weaver añade una relación de confianza al rol de la aplicación de simulación como [política de confianza](#). SimSpace Weaver crea una política de confianza para cada simulación, similar al ejemplo siguiente.

```

{
  "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

En este ejemplo, 111122223333 es el número de cuenta y MySimName es el nombre de simulación. Estos valores son diferentes en sus políticas de confianza.

Prevención de la sustitución confusa entre servicios

El [problema de la sustitución confusa](#) es una cuestión de seguridad en la que una entidad que no tiene permiso para realizar una acción puede engañar a una entidad con más privilegios para realizar la acción. En AWS, la suplantación de identidad entre servicios puede provocar el confuso problema de un diputado. La suplantación entre servicios puede producirse cuando un servicio (el servicio que lleva a cabo las llamadas) llama a otro servicio (el servicio al que se llama). El servicio que lleva a cabo las llamadas se puede manipular para utilizar sus permisos a fin de actuar en función de los recursos de otro cliente de una manera en la que no debe tener permiso para acceder. Para evitarlo, AWS proporciona herramientas que le ayudan a proteger sus datos para todos los servicios con entidades principales de servicio a las que se les ha dado acceso a los recursos de su cuenta.

Se recomienda utilizar las claves de contexto de condición [aws:SourceAccount](#) global [aws:SourceArn](#) las claves de contexto en las políticas de recursos para limitar los permisos que se AWS SimSpace Weaver otorgan a otro servicio al recurso. Si el `aws:SourceArn` valor no contiene el ID de la cuenta, como un bucket de Amazon S3 Amazon Resource Name (ARN), debe utilizar ambas claves de contexto de condición global para limitar los permisos. Si utiliza claves de contexto de condición global y el valor de `aws:SourceArn` contiene el ID de cuenta, el valor de `aws:SourceAccount` y la cuenta en el valor de `aws:SourceArn` deben utilizar el mismo ID de cuenta cuando se utiliza en la misma instrucción de política. Utilice `aws:SourceArn` si desea que solo se asocie un recurso al acceso entre servicios. Utilice `aws:SourceAccount` si quiere permitir que cualquier recurso de esa cuenta se asocie al uso entre servicios.

El valor de `aws:SourceArn` debe usar el de la extensiónARN.

La forma más eficaz de protegerse contra el confuso problema de los diputados es utilizar la clave del contexto ARN de la condición `aws:SourceArn` global con todo el recurso. Si no conoce la extensión completa ARN o si está especificando varias extensiones, utilice la clave de condición del contexto `aws:SourceArn` global con caracteres comodín (*) para las partes desconocidas de la ARN extensión. Por ejemplo, `arn:aws:simspaceweaver:*:111122223333:*`.

En el siguiente ejemplo, se muestra cómo utilizar las claves de contexto de condición `aws:SourceAccount` global `aws:SourceArn` y las claves contextuales SimSpace Weaver para evitar el confuso problema de los adjuntos. Esta política solo permitirá SimSpace Weaver asumir el rol cuando la solicitud provenga de la cuenta de origen especificada y se suministre con la especificadaARN. En este caso, solo SimSpace Weaver puede asumir la función para las solicitudes de simulaciones en la propia cuenta del solicitante (`111122223333`) y solo en la región especificada (`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/*"
        }
      }
    }
  ]
}
```

Una forma más segura de escribir esta política es incluir el nombre de la simulación en el `aws:SourceArn`, como se muestra en el siguiente ejemplo, lo que restringe la política a una simulación denominada `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"
    }
  }
}
]
}

```

Si incluye `aws:SourceArn` explícitamente un número de cuenta, puede omitir la prueba de `Condition` elementos `aws:SourceAccount` (consulte la [Guía del IAM usuario](#) para obtener más información), por ejemplo, en la siguiente política simplificada:

```

{
  "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"
        }
      }
    }
  ]
}

```

Solución de problemas AWS SimSpace Weaver de identidad y acceso

Utilice la siguiente información como ayuda para diagnosticar y solucionar los problemas más comunes que pueden surgir al trabajar con SimSpace Weaver y IAM.

Temas

- [No estoy autorizado a realizar ninguna acción en SimSpace Weaver](#)

- [No estoy autorizado a realizar iam: PassRole](#)
- [Quiero ver mis claves de acceso](#)
- [Soy administrador y quiero permitir el acceso de otras personas SimSpace Weaver](#)
- [Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis SimSpace Weaver recursos](#)

No estoy autorizado a realizar ninguna acción en SimSpace Weaver

Si AWS Management Console le indica que no está autorizado a realizar una acción, debe ponerse en contacto con su administrador para obtener ayuda. Su administrador es la persona que le facilitó su nombre de usuario y contraseña.

El siguiente ejemplo de error se produce cuando el mateojackson IAM usuario intenta usar la consola para ver los detalles de un *my-example-widget* recurso ficticio, pero no tiene los `simspaceweaver:GetWidget` permisos ficticios.

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

En este caso, Mateo pide a su administrador que actualice sus políticas de forma que pueda obtener acceso al recurso *my-example-widget* mediante la acción `simspaceweaver:GetWidget`.

No estoy autorizado a realizar iam: PassRole

Si recibe un error que indica que no tiene autorización para realizar la acción `iam:PassRole`, las políticas deben actualizarse a fin de permitirle pasar un rol a SimSpace Weaver.

Algunos Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada a un servicio. Para ello, debe tener permisos para transferir el rol al servicio.

El siguiente ejemplo de error se produce cuando un IAM usuario denominado marymajor intenta utilizar la consola para realizar una acción en ella. SimSpace Weaver Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

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

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con AWS el administrador. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero ver mis claves de acceso

Después de crear las claves IAM de acceso de usuario, podrá ver su ID de clave de acceso en cualquier momento. Sin embargo, no puede volver a ver su clave de acceso secreta. Si pierde la clave de acceso secreta, debe crear un nuevo par de claves de acceso.

Las claves de acceso se componen de dos partes: un ID de clave de acceso (por ejemplo, AKIAIOSFODNN7EXAMPLE) y una clave de acceso secreta (por ejemplo, wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY). El ID de clave de acceso y la clave de acceso secreta se utilizan juntos, como un nombre de usuario y contraseña, para autenticar sus solicitudes. Administre sus claves de acceso con el mismo nivel de seguridad que para el nombre de usuario y la contraseña.

Important

No proporcione las claves de acceso a terceros, ni siquiera para que lo ayuden a [buscar el ID de usuario canónico](#). De este modo, podrías dar a alguien acceso permanente a tu Cuenta de AWS.

Cuando crea un par de claves de acceso, se le pide que guarde el ID de clave de acceso y la clave de acceso secreta en un lugar seguro. La clave de acceso secreta solo está disponible en el momento de su creación. Si pierdes tu clave de acceso secreta, debes añadir nuevas claves de acceso a tu IAM usuario. Puede tener un máximo de dos claves de acceso. Si ya cuenta con dos, debe eliminar un par de claves antes de crear una nueva. Para ver las instrucciones, consulte [Administrar las claves de acceso](#) en la Guía del IAM usuario.

Soy administrador y quiero permitir el acceso de otras personas SimSpace Weaver

Para permitir el acceso de otras personas SimSpace Weaver, debes conceder permiso a las personas o aplicaciones que necesitan acceso. Si lo utiliza AWS IAM Identity Center para administrar personas y aplicaciones, debe asignar conjuntos de permisos a los usuarios o grupos para definir

su nivel de acceso. Los conjuntos de permisos crean y asignan automáticamente IAM políticas a las IAM funciones asociadas a la persona o la aplicación. Para obtener más información, consulte los [conjuntos de permisos](#) en la Guía del AWS IAM Identity Center usuario.

Si no utiliza IAM Identity Center, debe crear IAM entidades (usuarios o roles) para las personas o aplicaciones a las que necesitan acceso. A continuación, debe asociar una política a la entidad que le conceda los permisos correctos en SimSpace Weaver. Una vez concedidos los permisos, proporcione las credenciales al usuario o al desarrollador de la aplicación. Utilizarán esas credenciales para acceder AWS. Para obtener más información sobre la creación de IAM usuarios, grupos, políticas y permisos, consulte [IAM identidades](#) y [políticas y permisos IAM en](#) la Guía del IAM usuario.

Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis SimSpace Weaver recursos

Puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que respaldan las políticas basadas en recursos o las listas de control de acceso (ACLs), puedes usar esas políticas para permitir que las personas accedan a tus recursos.

Para más información, consulte lo siguiente:

- Para saber si SimSpace Weaver es compatible con estas funciones, consulte. [¿Cómo AWS SimSpace Weaver funciona con IAM](#)
- Para obtener información sobre cómo proporcionar acceso a los recursos de su propiedad, consulte [Proporcionar acceso a un IAM usuario en otro Cuenta de AWS de su propiedad](#) en la Guía del IAM usuario. Cuentas de AWS
- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta [Cómo permitir el acceso a recursos que Cuentas de AWS son propiedad de terceros](#) en la Guía del IAM usuario.
- Para obtener información sobre cómo proporcionar acceso mediante la federación de identidades, consulte [Proporcionar acceso a usuarios autenticados externamente \(federación de identidades\)](#) en la Guía del IAM usuario.
- Para saber la diferencia entre el uso de roles y políticas basadas en recursos para el acceso entre cuentas, consulte el acceso a [recursos entre cuentas IAM en la Guía](#) del usuario. IAM

Registro y supervisión de eventos de seguridad AWS SimSpace Weaver

El monitoreo es una parte importante del mantenimiento de la confiabilidad, la disponibilidad y el rendimiento de SimSpace Weaver sus AWS soluciones. Debe recopilar los datos de supervisión de todas las partes de la AWS solución para poder depurar más fácilmente un error multipunto en caso de que se produzca.

AWS y SimSpace Weaver proporcione varias herramientas para supervisar los recursos de simulación y responder a posibles incidentes.

Inicia sesión en Amazon CloudWatch

SimSpace Weaver almacena sus inicios de sesión CloudWatch. Puede utilizar estos registros para supervisar los eventos de la simulación (como el inicio y la detención de aplicaciones), así como para realizar tareas de depuración. Para obtener más información, consulte [SimSpace Weaver inicia sesión en Amazon CloudWatch Logs](#).

CloudWatch Alarmas Amazon

Con CloudWatch las alarmas de Amazon, observas una única métrica durante un período de tiempo que especifiques. Si la métrica supera un umbral determinado, se envía una notificación a un SNS tema de Amazon o a una política de AWS Auto Scaling. CloudWatch las alarmas se activan cuando su estado cambia y se mantienen durante un número específico de períodos, no por estar en un estado en particular. Para obtener más información, consulte [Monitorización SimSpace Weaver con Amazon CloudWatch](#).

AWS CloudTrail Registros

CloudTrail proporciona un registro de las acciones realizadas por un usuario, un rol o un AWS servicio en SimSpace Weaver. Con la información recopilada CloudTrail, puede determinar el destinatario de la solicitud SimSpace Weaver, la dirección IP desde la que se realizó la solicitud, quién la realizó, cuándo se realizó y detalles adicionales. Para obtener más información, consulte [Registrar llamadas mediante AWS SimSpace Weaver API AWS CloudTrail](#).

Validación de conformidad para AWS SimSpace Weaver

SimSpace Weaver no está incluido en el ámbito de ningún programa de AWS cumplimiento.


Los auditores externos evalúan la seguridad y el cumplimiento de otros AWS servicios como parte de varios programas de AWS cumplimiento. Estos incluyen SOC PCI la Reserva Federal RAMP HIPAA y otros.

Para saber si un programa de cumplimiento Servicio de AWS está dentro del ámbito de aplicación de programas de cumplimiento específicos, consulte [Servicios de AWS Alcance por programa](#) de de cumplimiento y elija el programa de cumplimiento que le interese. Para obtener información general, consulte Programas de [AWS cumplimiento > Programas AWS](#) .

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte [Descarga de informes en AWS Artifact](#) .

Su responsabilidad de cumplimiento al Servicios de AWS utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

- [Guías de inicio rápido sobre seguridad y cumplimiento](#): estas guías de implementación analizan las consideraciones arquitectónicas y proporcionan los pasos para implementar entornos básicos centrados en AWS la seguridad y el cumplimiento.
- [Diseñando una arquitectura basada en la HIPAA seguridad y el cumplimiento en Amazon Web Services](#): en este documento técnico se describe cómo pueden utilizar las empresas AWS para crear HIPAA aplicaciones aptas.

 Note

No todos son aptos. Servicios de AWS HIPAA Para obtener más información, consulta la [Referencia de servicios HIPAA aptos](#).

- [AWS Recursos](#) de de cumplimiento: esta colección de libros de trabajo y guías puede aplicarse a su industria y ubicación.
- [AWS Guías de cumplimiento para clientes](#): comprenda el modelo de responsabilidad compartida desde la perspectiva del cumplimiento. En las guías se resumen las mejores prácticas para garantizar la seguridad Servicios de AWS y se orientan a los controles de seguridad en varios marcos (incluidos el Instituto Nacional de Estándares y Tecnología (NIST), el Consejo de Normas de Seguridad de la Industria de Tarjetas de Pago (PCI) y la Organización Internacional de Normalización (ISO)).

- [Evaluación de los recursos con reglas](#) en la guía para AWS Config desarrolladores: el AWS Config servicio evalúa en qué medida las configuraciones de los recursos cumplen con las prácticas internas, las directrices del sector y las normas.
- [AWS Security Hub](#)— Este Servicio de AWS proporciona una visión completa del estado de su seguridad interior AWS. Security Hub utiliza controles de seguridad para evaluar sus recursos de AWS y comprobar su cumplimiento con los estándares y las prácticas recomendadas del sector de la seguridad. Para obtener una lista de los servicios y controles compatibles, consulte la [Referencia de controles de Security Hub](#).
- [Amazon GuardDuty](#): Servicio de AWS detecta posibles amenazas para sus cargas de trabajo Cuentas de AWS, contenedores y datos mediante la supervisión de su entorno para detectar actividades sospechosas y maliciosas. GuardDuty puede ayudarlo a cumplir con varios requisitos de conformidad, por ejemplo PCIDSS, cumpliendo con los requisitos de detección de intrusiones exigidos por ciertos marcos de cumplimiento.
- [AWS Audit Manager](#)— Esto le Servicio de AWS ayuda a auditar continuamente su AWS consumo para simplificar la gestión del riesgo y el cumplimiento de las normativas y los estándares del sector.

Resiliencia en AWS SimSpace Weaver

La infraestructura AWS global se basa en zonas Regiones de AWS de disponibilidad. Regiones de AWS proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están conectadas mediante redes de baja latencia, alto rendimiento y alta redundancia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos.

[Para obtener más información sobre las zonas de disponibilidad Regiones de AWS y las zonas de disponibilidad, consulte Infraestructura global.AWS](#)

Seguridad de la infraestructura en AWS SimSpace Weaver

Como servicio gestionado, AWS SimSpace Weaver está protegido por la seguridad de la red AWS global. Para obtener información sobre los servicios AWS de seguridad y cómo se AWS protege la infraestructura, consulte [Seguridad AWS en la nube](#). Para diseñar su AWS entorno utilizando las

mejores prácticas de seguridad de la infraestructura, consulte [Protección de infraestructuras en un marco](#) de buena AWS arquitectura basado en el pilar de la seguridad.

Utiliza las API llamadas AWS publicadas para acceder a SimSpace Weaver través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Necesitamos TLS 1.2 y recomendamos TLS 1.3.
- Cifre suites con perfecto secreto (PFS), como (Ephemeral Diffie-Hellman) o DHE ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben firmarse con un identificador de clave de acceso y una clave de acceso secreta asociada a un director. IAM También puede utilizar [AWS Security Token Service](#) (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Modelo de seguridad de la conectividad de red

Sus simulaciones se ejecutan en instancias de cómputo dentro de un Amazon VPC ubicado dentro de una AWS región que usted seleccione. Amazon VPC es una red virtual en la AWS nube que aísla la infraestructura por carga de trabajo o entidad organizativa. Las comunicaciones entre las instancias informáticas de Amazon VPC permanecen dentro de la AWS red y no viajan por Internet. Algunas comunicaciones de los servicios internos se transmiten por Internet y están cifradas. Las simulaciones para todos los clientes que operan en la misma AWS región comparten el mismo AmazonVPC. Las simulaciones para distintos clientes utilizan instancias informáticas independientes dentro del mismo AmazonVPC.

Las comunicaciones entre sus clientes de simulación y las simulaciones que se están ejecutando SimSpace Weaver viajan a través de Internet. SimSpace Weaver no gestiona estas conexiones. Es su responsabilidad proteger las conexiones de sus clientes.

Sus conexiones al SimSpace Weaver servicio atraviesan Internet y están cifradas. Esto incluye las conexiones mediante AWS Management Console, AWS Command Line Interface (AWS CLI), los kits de desarrollo de AWS software (SDK) y la SimSpace Weaver aplicaciónSDK.

Análisis de configuración y vulnerabilidad en AWS SimSpace Weaver

La configuración y los controles de TI son una responsabilidad compartida entre usted AWS y usted. [Para obtener más información, consulte el AWS de](#) . AWS gestiona las tareas de seguridad básicas de la infraestructura subyacente, como la aplicación de parches al sistema operativo en las instancias informáticas, la configuración del firewall y la recuperación ante desastres de la AWS infraestructura. Estos procedimientos han sido revisados y certificados por los terceros pertinentes. Para obtener más información, consulte las [Prácticas recomendadas sobre seguridad, identidad y conformidad](#).

Usted es responsable de la seguridad de su software de simulación:

- Conserve el código de su aplicación, incluidas las actualizaciones y los parches de seguridad.
- Autentique y cifre la comunicación entre sus clientes de simulación y las aplicaciones a las que se conectan.
- Actualice las simulaciones para utilizar las SDK versiones más recientes, incluida la SimSpace Weaver aplicación AWS SDK SDK y.

Note

SimSpace Weaver no admite actualizaciones de aplicaciones en una simulación en ejecución. Si necesita actualizar sus aplicaciones, debe detener y eliminar la simulación y, a continuación, crear una nueva simulación con el código de la aplicación actualizado. Le recomendamos que guarde el estado de la simulación en un banco de datos externo para poder restaurarlo si necesita volver a crear la simulación.

Prácticas recomendadas de seguridad para SimSpace Weaver

En esta sección se describen las mejores prácticas de seguridad específicas de SimSpace Weaver. Para obtener más información sobre las prácticas recomendadas de seguridad AWS, consulte [Prácticas recomendadas en materia de seguridad, identidad y conformidad](#).

Temas

- [Encripte las comunicaciones entre sus aplicaciones y sus clientes](#)

- [Realice copias de seguridad periódicas del estado de la simulación](#)
- [Mantenga sus aplicaciones y SDKs](#)

Encripte las comunicaciones entre sus aplicaciones y sus clientes

SimSpace Weaver no gestiona las comunicaciones entre sus aplicaciones y sus clientes. Debería implementar algún tipo de autenticación y cifrado para las sesiones de los clientes.


Realice copias de seguridad periódicas del estado de la simulación

SimSpace Weaver no guarda el estado de la simulación. Las simulaciones que se detienen (como resultado de una API llamada, de una opción de consola o de un fallo del sistema) no guardan su estado y no tienen una forma inherente de recuperarlas. Las simulaciones detenidas no se pueden reiniciar. La única forma de realizar el equivalente a un reinicio es volver a crear la simulación con la misma configuración y los mismos datos. Puede utilizar copias de seguridad del estado de la simulación para inicializar la nueva simulación. AWS ofrece servicios de [almacenamiento](#) y [bases de datos](#) en la nube altamente fiables y disponibles que puede utilizar para guardar el estado de la simulación.

Mantenga sus aplicaciones y SDKs

Mantenga sus aplicaciones, las instalaciones locales de los kits de desarrollo de AWS software (SDKs) y la SimSpace Weaver aplicación SDK. Puede descargar e instalar nuevas versiones de AWS SDKs. Pruebe nuevas versiones de la SimSpace Weaver aplicación SDK con compilaciones de aplicaciones que no sean de producción para asegurarse de que sus aplicaciones continúen funcionando según lo esperado. No puede actualizar sus aplicaciones en una simulación en ejecución. Para actualizar sus aplicaciones:

1. Actualice y pruebe el código de la aplicación localmente (o en un entorno de prueba).
2. Deje de cambiar el estado de la simulación y guárdela (si es necesario).
3. Detenga la simulación (una vez detenida, no se puede reiniciar).
4. Elimine la simulación (las simulaciones detenidas que no se eliminen se tendrán en cuenta para los límites de servicio).
5. Recree su simulación con la misma configuración y el código de aplicación actualizado.
6. Inicialice la simulación con los datos de estado guardados (si están disponibles).
7. Inicie la nueva simulación.

 **Note**

Una nueva simulación creada con la misma configuración es independiente de la simulación anterior. Tendrá un nuevo ID de simulación y enviará los registros a un nuevo flujo de registros en Amazon CloudWatch.

Inicio de sesión y supervisión SimSpace Weaver

La supervisión es una parte importante del mantenimiento de la confiabilidad, la disponibilidad y el rendimiento de SimSpace Weaver AWS las demás soluciones. AWS proporciona las siguientes herramientas de monitoreo para observar SimSpace Weaver, informar cuando algo anda mal y tomar medidas automáticas cuando sea apropiado:

- Amazon CloudWatch monitorea tus AWS recursos y las aplicaciones en las que AWS ejecutas en tiempo real. Puede recopilar métricas y realizar un seguimiento de las métricas, crear paneles personalizados y definir alarmas que le advierten o que toman medidas cuando una métrica determinada alcanza el umbral que se especifique. Para obtener más información, consulta la [Guía del CloudWatch usuario de Amazon](#).
- Amazon CloudWatch Logs le permite supervisar, almacenar y acceder a los datos de registro de sus SimSpace Weaver trabajadores y de otras fuentes. CloudTrail CloudWatch Los registros pueden monitorear la información de los datos de registro y notificarle cuando se alcanzan ciertos umbrales. También se pueden archivar los datos del registro en un almacenamiento de larga duración. Para obtener más información, consulta la [Guía del usuario CloudWatch de Amazon Logs](#).
- AWS CloudTrail captura API las llamadas y los eventos relacionados realizados por usted o en su nombre Cuenta de AWS y entrega los archivos de registro a un bucket de Amazon S3 que especifique. También puede identificar qué usuarios y cuentas llamaron a AWS, la dirección IP de origen de las llamadas y el momento en que se hicieron. Para obtener más información, consulte la [AWS CloudTrail Guía del usuario de](#) .

Temas

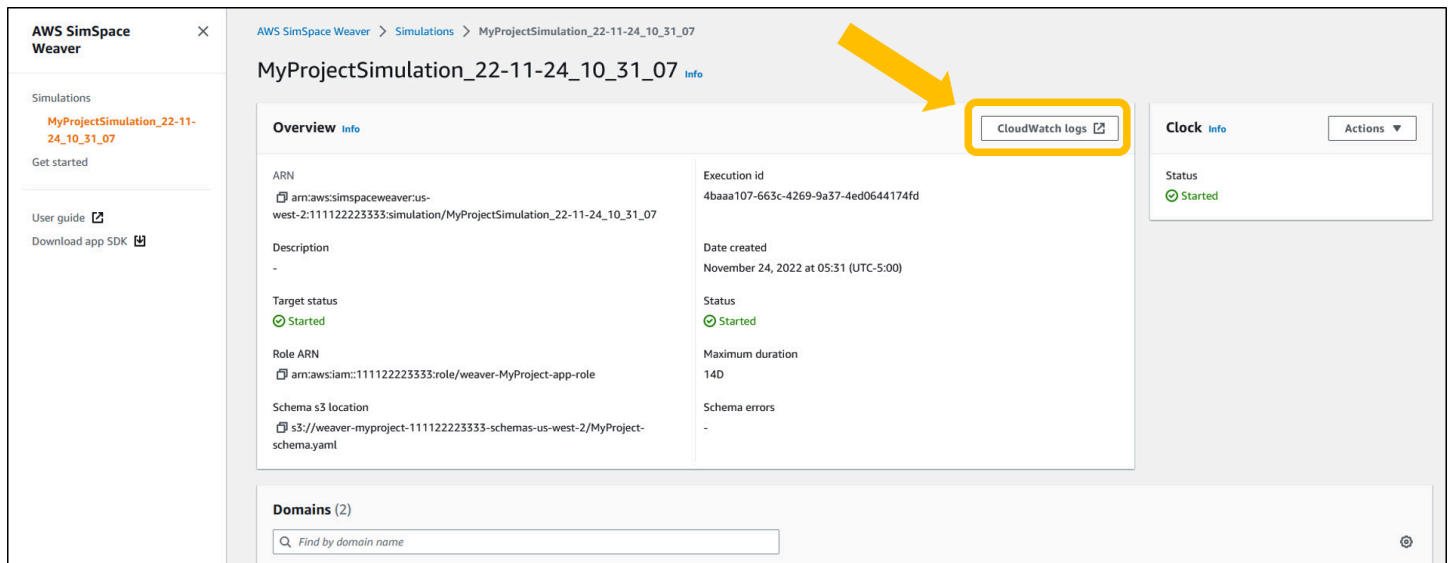
- [SimSpace Weaver inicia sesión en Amazon CloudWatch Logs](#)
- [Monitorización SimSpace Weaver con Amazon CloudWatch](#)
- [Registrar llamadas mediante AWS SimSpace Weaver API AWS CloudTrail](#)

SimSpace Weaver inicia sesión en Amazon CloudWatch Logs

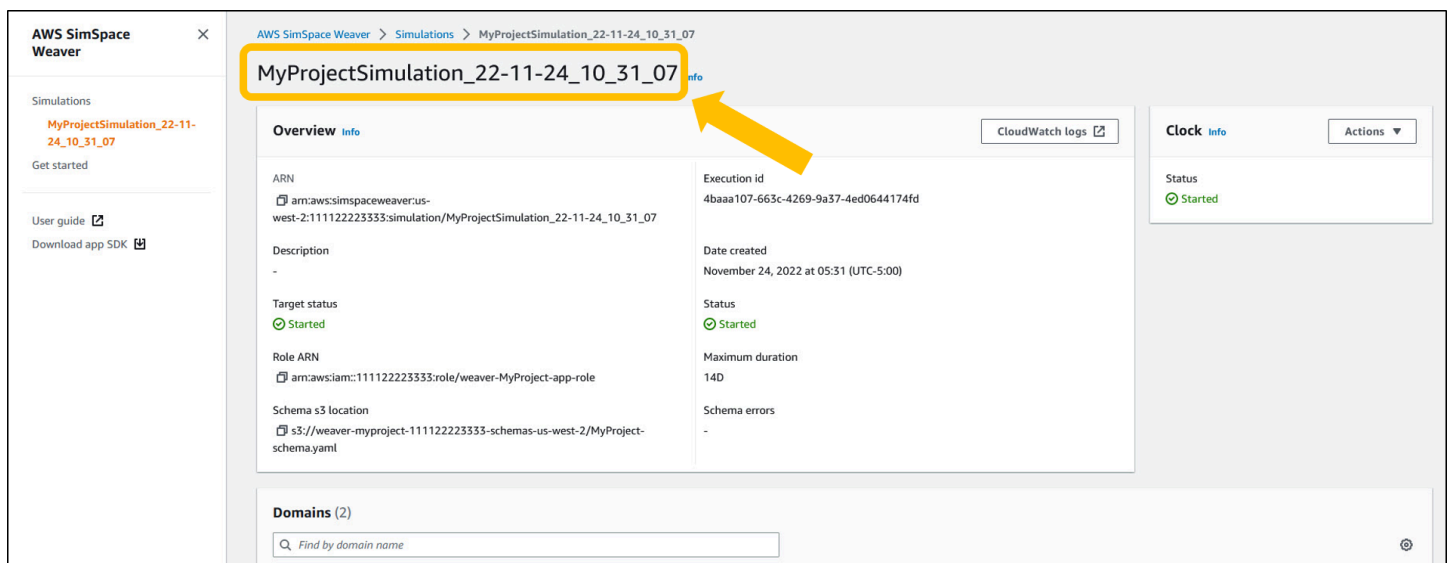
Acceder a sus registros de SimSpace Weaver

Todos los registros generados a partir de SimSpace Weaver las simulaciones se almacenan en Amazon CloudWatch Logs. Para acceder a sus registros, puede utilizar el botón de CloudWatch

registros del panel de información general de la simulación en la SimSpace Weaver consola, que le llevará directamente a los registros de esa simulación específica.



También puede acceder a los registros a través de la CloudWatch consola. Necesitará el nombre de la simulación para buscar sus registros.



SimSpace Weaver registros

SimSpace Weaver escribe los mensajes de administración de la simulación y la salida de la consola de sus aplicaciones en Amazon CloudWatch Logs. Para obtener más información sobre cómo trabajar con registros, consulte [Trabajar con grupos de registros y flujos](#) de CloudWatch registros en la Guía del usuario de Amazon Logs.

Cada simulación que cree tiene su propio grupo de CloudWatch registros en Logs. El nombre del grupo de registro se especifica en el esquema de simulación. En el siguiente fragmento de esquema, el valor de `log_destination_service` es `logs`. Esto significa que el valor de `log_destination_resource_name` es el nombre de un grupo de registro. En este caso, el grupo de registro es `MySimulationLogs`.

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

También puede utilizar el `DescribeSimulation` API para buscar el nombre del grupo de registros para la simulación después de iniciarla.

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

El siguiente ejemplo muestra la parte de la salida de `DescribeSimulation` que describe la configuración de registro. El nombre del grupo de registro se muestra al final de `LogGroupArn`.

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

Cada grupo de registro de simulación contiene varios flujos de registro:

- Flujo de registro de administración: mensajes de administración de simulaciones producidos por el SimSpace Weaver servicio.

```
/sim/management
```

- Flujo de registro de errores: mensajes de error producidos por el SimSpace Weaver servicio. Este flujo de registro solo existe si hay errores. SimSpace Weaver almacena los errores escritos por tus aplicaciones en sus propios flujos de registro de aplicaciones (consulta los siguientes flujos de registro).

```
/sim/errors
```

- Flujos de registro de aplicaciones espaciales (1 para cada aplicación espacial de cada trabajador): resultado de la consola generado por aplicaciones espaciales. Cada aplicación espacial escribe en su propio flujo de registro. La *spatial-app-id* son todos los caracteres que aparecen después de la barra final del *worker-id*.

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

- Flujos de registro de aplicaciones personalizadas (1 para cada instancia de aplicación personalizada): resultado de la consola generado por aplicaciones personalizadas. Cada instancia de aplicación personalizada escribe en su propio flujo de registro.

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

- Flujos de registro de aplicaciones de servicio (1 para cada instancia de aplicación de servicio): resultado de la consola generado por las aplicaciones de servicio. Cada aplicación de servicio escribe en su propio flujo de registro. La *service-app-id* son todos los caracteres que aparecen después de la barra final del *service-app-name*.

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

Monitorización SimSpace Weaver con Amazon CloudWatch

Puedes monitorizar SimSpace Weaver con Amazon CloudWatch, que recopila datos sin procesar y los procesa para convertirlos en métricas legibles prácticamente en tiempo real. Estas estadísticas se mantienen durante 15 meses, de forma que pueda obtener acceso a información histórica y disponer de una mejor perspectiva sobre el desempeño de su aplicación web o servicio. También puede establecer alarmas que vigilen determinados umbrales y enviar notificaciones o realizar

acciones cuando se cumplan dichos umbrales. Para obtener más información, consulta la [Guía del CloudWatch usuario de Amazon](#).

El SimSpace Weaver servicio informa de las siguientes métricas en el espacio de AWS/simspaceweaver nombres.

SimSpace Weaver métricas a nivel de cuenta

El espacio de SimSpace Weaver nombres incluye las siguientes métricas relacionadas con la actividad a nivel de AWS cuenta.

Métrica	Descripción
SimulationCount	El número de simulaciones de la cuenta actual. Unidades: recuento Dimensiones: ninguna Estadísticas: promedio, mínimo, máximo

Registrar llamadas mediante AWS SimSpace Weaver API AWS CloudTrail

AWS SimSpace Weaver está integrado con AWS CloudTrail un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un AWS servicio en SimSpace Weaver. CloudTrail captura todas API las llamadas SimSpace Weaver como eventos. Las llamadas capturadas incluyen las llamadas desde la SimSpace Weaver consola y las llamadas en código a las SimSpace Weaver API operaciones. Si crea una ruta, puede habilitar la entrega continua de CloudTrail eventos a un bucket de Amazon S3, incluidos los eventos para SimSpace Weaver. Si no configura una ruta, podrá seguir viendo los eventos más recientes en la CloudTrail consola en Event history. Con la información recopilada CloudTrail, puede determinar el destinatario de la solicitud SimSpace Weaver, la dirección IP desde la que se realizó la solicitud, quién la realizó, cuándo se realizó y detalles adicionales.

Para obtener más información CloudTrail, consulte la [Guía AWS CloudTrail del usuario](#).

SimSpace Weaver información en CloudTrail

CloudTrail está habilitada en tu cuenta Cuenta de AWS al crear la cuenta. Cuando se produce una actividad en SimSpace Weaver, esa actividad se registra en un CloudTrail evento junto con otros eventos AWS de servicio en Event history. Puede ver, buscar y descargar eventos recientes en su Cuenta de AWS. Para obtener más información, consulte [Visualización de eventos con el historial de CloudTrail eventos](#).

Para tener un registro continuo de tus eventos Cuenta de AWS, incluidos los eventos para SimSpace Weaver ti, crea una ruta. Un rastro permite CloudTrail entregar archivos de registro a un bucket de Amazon S3. De forma predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las Regiones de AWS. La ruta registra los eventos de todas las regiones de la AWS partición y envía los archivos de registro al bucket de Amazon S3 que especifique. Además, puede configurar otros AWS servicios para analizar más a fondo los datos de eventos recopilados en los CloudTrail registros y actuar en función de ellos. Para más información, consulte los siguientes temas:

- [Introducción a la creación de registros de seguimiento](#)
- [CloudTrail servicios e integraciones compatibles](#)
- [Configuración de SNS las notificaciones de Amazon para CloudTrail](#)
- [Recibir archivos de CloudTrail registro de varias regiones](#) y [recibir archivos de CloudTrail registro de varias cuentas](#)

Todas SimSpace Weaver las acciones se registran CloudTrail y se documentan en la [AWS SimSpace Weaver APIReferencia](#). Por ejemplo, las llamadas a los ListSimulations archivos de registro DescribeSimulation y DeleteSimulation las acciones generan entradas en los archivos de CloudTrail registro.

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario lo ayuda a determinar lo siguiente:

- Si la solicitud se realizó con credenciales de usuario root o AWS Identity and Access Management (IAM).
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro AWS servicio.

Para obtener más información, consulte el [CloudTrail userIdentityelemento](#).

Descripción de las entradas de los archivos de SimSpace Weaver registro

Un rastro es una configuración que permite la entrega de eventos como archivos de registro a un bucket de Amazon S3 que usted especifique. CloudTrail Los archivos de registro contienen una o más entradas de registro. Un evento representa una solicitud única de cualquier fuente e incluye información sobre la acción solicitada, como la fecha y la hora de la acción, los parámetros de la solicitud y otros detalles. CloudTrail Los archivos de registro no son un registro ordenado de las API llamadas públicas, por lo que no aparecen en ningún orden específico.

El siguiente ejemplo muestra una entrada de CloudTrail registro que demuestra la `ListSimulations` acción.

```
{
  "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 Puntos finales y cuotas de Weaver

En las siguientes tablas se describen los puntos finales de servicio y las cuotas de servicio de Weaver. SimSpace Service Quotas, también denominadas límites, establecen el número máximo de recursos u operaciones de servicio para su Cuenta de AWS. Para obtener más información, consulte [AWS service quotas](#) en la Referencia general de AWS.

Puntos de conexión de servicio

Nombre de la región	Región	Punto de conexión	Protocolo
Este de EE. UU. (Norte de Virginia)	us-east-1	simspaceweaver.us-east-1.amazonaws.com	HTTPS
Este de EE. UU. (Ohio)	us-east-2	simspaceweaver.us-east-2.amazonaws.com	HTTPS
Oeste de EE. UU. (Oregón)	us-west-2	simspaceweaver.us-west-2.amazonaws.com	HTTPS
Asia-Pacífico (Singapur)	ap-southeast-1	simspaceweaver.ap-southeast-1.amazonaws.com	HTTPS
Asia-Pacífico (Sídney)	ap-southeast-2	simspaceweaver.ap-southeast-2.amazonaws.com	HTTPS
Europa (Estocolmo)	eu-north-1	simspaceweaver.eu-north-1.amazonaws.com	HTTPS

Nombre de la región	Región	Punto de conexión	Protocolo
Europa (Fráncfort)	eu-central-1	simspaceweaver.eu-central-1.amazonaws.com	HTTPS
Europa (Irlanda)	eu-west-1	simspaceweaver.eu-west-1.amazonaws.com	HTTPS
AWS GovCloud (Este de EE. UU.)	us-gov-east-1	simspaceweaver.us-gov-east-1.amazonaws.com	HTTPS
AWS GovCloud (Estados Unidos-Oeste)	us-gov-west-1	simspaceweaver.us-gov-west-1.amazonaws.com	HTTPS

Service Quotas

Nombre	Valor predeterminado	Ajuste	Descripción
Unidades de recursos de computación para cada aplicación	Cada región admitida: 4	No	Número máximo de unidades de recursos de computación que puede asignar a cada aplicación.
Unidades de recursos de computación para cada trabajador	Cada región admitida: 17	No	El número de unidades de recursos de computación disponibles para cada trabajador.
Campos de datos para cada entidad	Cada región admitida: 7	No	Número máximo de campos de datos (no indexados) que puede tener una entidad.

Nombre	Valor predeterminado	Ajuste	Descripción
Entidades de una partición	Cada región admitida: 8192	No	Número máximo de entidades en una partición.
Tamaño del campo de datos de la entidad	Cada región admitida: 1024 bytes	No	Tamaño máximo de un campo de datos (no indexado) de una entidad.
Transferencias de entidades entre trabajadores	Cada región admitida: 25	No	Número máximo de transferencias de entidades entre trabajadores, para cada partición y cada tic.
Transferencias de entidades en el mismo trabajador	Cada región admitida: 500	No	Número máximo de transferencias de entidades en el mismo trabajador, para cada partición y cada tic.
Campos de índice para cada entidad	Cada región admitida: 1	No	Número máximo de campos de índice que puede tener una entidad.
Duración máxima (en días) de una simulación	Cada región admitida: 14	No	Mayor número de días que puede especificar como duración máxima de una simulación. Todas las simulaciones tienen una duración máxima, incluso si no se especifica el valor. Una simulación se detiene automáticamente cuando alcanza su duración máxima.

Nombre	Valor predeterminado	Ajuste	Descripción
Memoria de cada unidad de recursos de computación	Cada región admitida: 1 gigabyte	No	Cantidad de memoria de acceso aleatorio (RAM) que obtiene una aplicación por cada unidad de recursos de computación.
Suscripciones remotas para cada trabajador	Cada región admitida: 24	No	Número máximo de suscripciones remotas para cada trabajador.
Recuento de simulaciones	Cada región admitida: 2	Sí	Número máximo de simulaciones con un estado de destino INICIADA en su cuenta. Puede solicitar un aumento de cuota de hasta 10.
Trabajadores para una simulación	Cada región admitida: 2	Sí	Número máximo de trabajadores que se puede asignar a 1 simulación. Puede solicitar un aumento de cuota de hasta 10.
CPU virtuales de cada unidad de recursos de computación	Cada región admitida: 2	No	Número de unidades de procesamiento central (CPU) virtuales que obtiene una aplicación por cada unidad de recursos de computación.

Cuotas de mensajería

Las siguientes cuotas se aplican a la mensajería de una aplicación a otra para SimSpace Weaver Local y dentro deNube de AWS.

Nombre	Valor predeterminado	Ajustable	Descripción
Tamaño máximo de mensaje	Cada región compatible: 256 bytes	No	El tamaño máximo de la carga útil de un mensaje.
Velocidad máxima de envío de mensajes	Cada región admitida: 128	No	El número máximo de mensajes que cada aplicación puede enviar por marca.

Velocidades del reloj

El esquema de simulación especifica la frecuencia de reloj (también denominada frecuencia de graduación) de una simulación. En la siguiente tabla se especifican las velocidades del reloj válidas que se pueden utilizar.

Nombre	Valores válidos	Descripción
Frecuencia del reloj	Cada región compatible: «10", «15", «30", «ilimitada»	Las frecuencias de reloj válidas para una simulación.
Frecuencia de reloj (versiones 1.13 y 1.12)	Cada región compatible: 10, 15, 30	Velocidades del reloj válidas para una simulación.

Service Quotas de SimSpace Weaver Local

Las siguientes Service Quotas se aplican únicamente a SimSpace Weaver Local. El resto de las cuotas también se aplican a SimSpace Weaver Local.

Nombre	Valor predeterminado	Ajustable	Descripción
Particiones máximas	SimSpace Weaver Local: 24	No	El número máximo de particiones para una simulación.
Aplicaciones máximas	SimSpace Weaver Local: 24	No	El número total máximo de aplicaciones (de cualquier tipo) para una simulación.
Dominios máximos	SimSpace Weaver Local: 24	No	El número total máximo de dominios (de cualquier tipo) para una simulación.
Entidades por partición	SimSpace Weaver Local: 4096	No	Número máximo de entidades en cada partición.
Campos por entidad	SimSpace Weaver Local: 8	No	El número máximo de campos para cada entidad.
Tamaño del campo	SimSpace Weaver Local: 1024 bytes	No	Tamaño máximo de un campo de entidad.

Solución de problemas en SimSpace Weaver

Temas

- [AssumeRoleAccessDenied](#)
- [InvalidBucketName](#)
- [ServiceQuotaExceededException](#)
- [TooManyBuckets](#)
- [Permiso denegado durante el inicio de la simulación](#)
- [Problemas relacionados con el tiempo de uso de Docker](#)
- [PathfindingSample el cliente de consola no se puede conectar](#)
- [La AWS CLI no reconoce simspaceweaver](#)

AssumeRoleAccessDenied

Es posible que reciba el siguiente error si la simulación no se inicia:

```
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
```

Puede recibir este error si se cumple una de las siguientes condiciones con respecto al rol de (IAM) de AWS Identity and Access Management de la simulación:

- El nombre del recurso de Amazon (ARN) hace referencia a un rol de IAM que no existe.
- La política de confianza del rol de IAM que no permite que el nombre de la nueva simulación asuma el rol.

Asegúrese de que el rol exista. Si el rol existe, compruebe su política de confianza para dicho rol. El `aws:SourceArn` en el siguiente ejemplo, la política de confianza solo permite que una simulación (en la cuenta 111122223333) cuyo nombre comience por `MySimulation` asuma el rol.

```
{
  "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*"
        }
      }
    }
  ]
}

```

Para permitir que otra simulación cuyo nombre comience por `MyOtherSimulation` «asuma el rol», la política de confianza debe modificarse como se muestra en el siguiente ejemplo editado:

```

{
  "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

Es posible que reciba el siguiente error al crear un proyecto:

```
An error occurred (InvalidBucketName) when calling the CreateBucket operation: The specified bucket is not valid.
```

Ha recibido este error porque el nombre que SimSpace Weaver ha transmitido a Amazon Simple Storage Service (Amazon S3) infringe las reglas de nomenclatura de los buckets (para obtener más información, consulte las [Reglas de nomenclatura de los buckets](#) en la Guía del usuario de Amazon Simple Storage Service).

El script `create-project` del SDK de la aplicación SimSpace Weaver crea los nombres de los buckets utilizando el nombre del proyecto que usted proporciona al script. Los nombres de bucket usan los siguientes formatos:

- Versión 1.13.x o posterior.
 - `weaver-lowercase-project-name-account-number-region`
- Versión 1.12.x
 - `weaver-lowercase-project-name-account-number-app-zips-region`
 - `weaver-lowercase-project-name-account-number-schemas-region`

Por ejemplo, dadas las siguientes propiedades del proyecto:

- Nombre del proyecto: `MyProject`
- Número de Cuenta de AWS: `111122223333`
- Región de AWS: `us-west-2`

El proyecto tendría los siguientes buckets:

- Versión 1.13.x o posterior.
 - `weaver-myproject-111122223333-us-west-2`
- Versión 1.12.x
 - `weaver-myproject-111122223333-app-zips-us-west-2`
 - `weaver-myproject-111122223333-schemas-us-west-2`

El nombre de su proyecto no debe infringir las normas de nomenclatura de Amazon S3. También debe usar un nombre de proyecto que sea lo suficientemente corto como para que los nombres de los buckets creados por el script de `create-project` no superen el límite de longitud de los nombres de los buckets de Amazon S3.

ServiceQuotaExceededException

Es posible que reciba el siguiente error cuando inicie una simulación:

```
An error occurred (ServiceQuotaExceededException) when calling the StartSimulation operation: Failed to start simulation due to: simulation quota has already been reached.
```

Recibirá este error si intenta iniciar una nueva simulación, pero su cuenta tiene actualmente el número máximo de simulaciones cuyo estado objetivo es INICIADO. Esto incluye la ejecución de simulaciones, las simulaciones fallidas y las simulaciones que se detuvieron porque alcanzaron su duración máxima. Puede eliminar una simulación detenida o fallida para poder iniciar una nueva simulación. Si todas las simulaciones se están ejecutando, puede detener y eliminar una simulación en ejecución. También puede solicitar un aumento de sus Service Quotas si aún no ha alcanzado el límite de solicitudes.

TooManyBuckets

Es posible que reciba el siguiente error al crear un proyecto:

```
An error occurred (TooManyBuckets) when calling the CreateBucket operation: You have attempted to create more buckets than allowed.
```

Amazon Simple Storage Service (Amazon S3) limita el número de depósitos que puede tener en AWS su cuenta (para obtener más información, [consulte Restricciones y limitaciones de los buckets en la Guía](#) del usuario de Amazon Simple Storage Service).

Para poder continuar, debe hacer una de las siguientes acciones:

- Elimine 2 o más buckets de Amazon S3 existentes de que no necesite.
- Solicite un aumento del límite de Amazon S3 (para obtener más información, consulte [Restricciones y limitaciones de los buckets](#) en la Guía del usuario de Amazon Simple Storage Service).

- Utilice una AWS cuenta diferente.

Note

La `DeleteSimulation` API in SimSpace Weaver no elimina los recursos de Amazon S3 asociados a la simulación. Le recomendamos eliminar todos los recursos asociados a sus simulaciones cuando ya no los necesite.

Permiso denegado durante el inicio de la simulación

Al iniciar una simulación, es posible que aparezca un mensaje de error que indique que se ha denegado el permiso o que se ha producido un error al acceder a los artefactos de la aplicación. Este problema puede producirse cuando especifica buckets de Amazon S3 para la simulación que SimSpace Weaver no creó para usted (ya sea a través de la consola o de los scripts del SDK de la aplicación SimSpace Weaver).

Las siguientes situaciones son las causas principales más probables:

- El servicio no tiene permiso para acceder a uno o más de los buckets de Amazon S3 que especificó en su esquema de simulación. Consulte la política de permisos de los roles de la aplicación, las políticas de los buckets de Amazon S3 y los permisos de los buckets de Amazon S3 para asegurarse de que `simspaceweaver.amazonaws.com` tiene los permisos correctos para acceder a los buckets. Para obtener más información sobre la política de permisos de roles de las aplicaciones, consulte [Permisos que SimSpace Weaver crea para usted](#).
- Su bucket de Amazon S3 podría estar en un Región de AWS diferente al de su simulación: los depósitos de Amazon S3 para sus artefactos de simulación deben estar en el mismo Región de AWS que su simulación. Compruebe su consola Amazon S3 para ver en qué Región de AWS se encuentra su bucket. Si su bucket de Amazon S3 está en otro Región de AWS, seleccione un bucket que esté en el mismo Región de AWS que su simulación.

Problemas relacionados con el tiempo de uso de Docker

Si está utilizando Docker y recibe errores relacionados con el tiempo al ejecutar scripts del SDK de la aplicación SimSpace Weaver, la causa podría ser que el reloj de su máquina virtual Docker no sea

correcto. Esto puede ocurrir si el ordenador estaba ejecutando Docker y, a continuación, se reanuda tras el modo de suspensión o hibernación.

Soluciones que puede probar

- Reinicie Docker.
- Desactive y, a continuación, vuelva a activar la sincronización horaria en Windows PowerShell:

```
Get-VMIntegrationService -VMName DockerDesktopVM -Name "Time Synchronization" |  
  Disable-VMIntegrationService  
Get-VMIntegrationService -VMName DockerDesktopVM -Name "Time Synchronization" |  
  Enable-VMIntegrationService
```

PathfindingSample el cliente de consola no se puede conectar

Es posible que reciba el siguiente error del cliente de la consola al conectarse a la PathfindingSample simulación descrita en los tutoriales de [Empezar con SimSpace Weaver](#). Este error se produce porque el cliente no puede abrir una conexión de red con la ViewApp dirección IP y el número de puerto combinados que has proporcionado.

```
Fatal error in function nng_dial. Error code: 268435577. Error message: no link
```

Para una simulación en el Nube de AWS

- ¿Su conexión de red funciona correctamente? Compruebe que puede conectarse a otras direcciones IP o sitios web que deberían funcionar. Asegúrese de que su navegador web no esté cargando un sitio web desde su caché.
- ¿Se está ejecutando la simulación? Puede utilizar la ListSimulationsAPI para obtener el estado de la simulación. Para obtener más información, consulte [Obtenga la dirección IP y el número de puerto de una aplicación personalizada](#). También puede utilizar la [consola de SimSpace Weaver](#) para verificar el estado de las simulaciones.
- ¿Están funcionando sus aplicaciones? Puede usar la DescribeAppAPI para obtener el estado de sus aplicaciones. Para obtener más información, consulte [Obtenga la dirección IP y el número de puerto de una aplicación personalizada](#). También puede utilizar la [consola de SimSpace Weaver](#) para verificar el estado de las simulaciones.
- ¿Están funcionando sus aplicaciones? Puedes usar la DescribeAppAPI para obtener el estado de tus aplicaciones. Para obtener más información, consulte [Obtenga la dirección IP y el número de](#)

[puerto de una aplicación personalizada](#). También puede utilizar la [consola de SimSpace Weaver](#) para verificar el estado de las simulaciones.

- ¿Utilizó la dirección IP y el número de puerto correctos? Cuando se conecte a través de Internet, debe utilizar la dirección IP y el número de puerto de Actual deViewApp. Puedes encontrar la IP Address y el número de Actual puerto en el EndpointInfo bloque del resultado de la DescribeAppAPI. También puede usar la [consola de SimSpace Weaver](#) para buscar su dirección IP (URI) y el número de puerto (puerto de entrada) para su ViewApp en la página de detalles de MyViewDomain.
- ¿Su conexión de red pasa por un firewall? Es posible que el firewall bloquee la conexión a la dirección IP o al número de puerto (o a ambos). Compruebe la configuración del firewall o contacte con el administrador del firewall.

Para una simulación local

- ¿Puede conectarse a su dirección de bucle invertido (127.0.0.1)? Si tiene la herramienta de línea de comandos ping en Windows, puede abrir una ventana de línea de comandos e intentar conectar con la dirección 127.0.0.1. Pulse Ctrl - C para finalizar la conexión.

```
ping 127.0.0.1
```

Example salida de la conexión

```
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 la conexión indica que se han perdido paquetes, es posible que haya otro software (como un firewall local, una configuración de seguridad o programas antimalware) que esté bloqueando la conexión.

- ¿Están funcionando sus aplicaciones? La simulación local se ejecuta como ventanas independientes para cada aplicación. Asegúrese de que las ventanas de sus aplicaciones espaciales y ViewApp estén abiertas. Para obtener más información, consulte [Desarrollo local en SimSpace Weaver](#).
- ¿Utilizó la dirección IP y el número de puerto correctos? Debe utilizar `tcp://127.0.0.1:7000` cuando se conecte a una simulación local. Para obtener más información, consulte [Desarrollo local en SimSpace Weaver](#).
- ¿Dispone de un software de seguridad local que pueda bloquear la conexión? Compruebe la configuración de seguridad, el firewall local o los programas antimalware para ver si bloquean su conexión a `127.0.0.1` en un puerto TCP de `7000`.

La AWS CLI no reconoce **simspaceweaver**

Si AWS CLI muestra algún error que sugiera que no conoce SimSpace Weaver, ejecute el siguiente comando.

```
aws simspaceweaver help
```

Si aparece un error que comienza con las siguientes líneas y muestra todas las opciones disponibles, es posible que se trate de una versión anterior de su AWS CLI.

```
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:
```

Ejecute el siguiente comando para comprobar la versión de su AWS CLI:

```
aws --version
```


Si su versión es anterior a 2.9.19, debe actualizar su AWS CLI. Tenga en cuenta que la versión actual de AWS CLI es posterior a la 2.9.19.

Para actualizar su AWS CLI, consulte [Instalación o actualización de la última versión del AWS CLI](#) en la Guía del usuario de AWS Command Line Interface, versión 2.

SimSpace Weaver referencia del esquema de simulación

SimSpace Weaver utiliza un YAML archivo para configurar las propiedades de una simulación. Este archivo se denomina esquema de simulación (o simplemente esquema). El ejemplo de simulación incluido en la SimSpace Weaver aplicación SDK incluye un esquema que puede copiar y editar para su propia simulación.

Temas

- [Ejemplo de un esquema completo](#)
- [Formato de esquema](#)

Ejemplo de un esquema completo

En el siguiente ejemplo se muestra el YAML Archivo de texto con formato -que describe una SimSpace Weaver simulación. En este ejemplo se incluyen valores ficticios para las propiedades. El formato del archivo varía en función del valor `sdk_version` especificado en él. Consulte [Formato de esquema](#) para obtener una descripción completa de las propiedades y sus valores válidos.

```
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"]
```

Formato de esquema

La estructura general de un esquema se muestra en el ejemplo siguiente. El orden de las propiedades en cada nivel del esquema no importa, siempre que las relaciones principal-secundario sean las mismas. El orden es importante para los elementos de una matriz.

```
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
```

Secciones

- [SDKversión](#)
- [Propiedades de simulación](#)
- [Procesos de trabajo](#)
- [Reloj](#)
- [Estrategias de partición](#)
- [Dominios](#)
- [Restricciones para la ubicación](#)

SDKversión

La `sdk_version` sección (obligatoria) identifica la versión de la SimSpace Weaver aplicación SDK que admite este esquema. Valores válidos: 1.17, 1.16, 1.15, 1.14, 1.13, 1.12

Important

El valor de `sdk_version` solo incluye el número de versión principal y el número de la primera versión secundaria. Por ejemplo, el valor 1.12 especifica todas las versiones, como 1.12.x, 1.12.0, 1.12.1 y 1.12.2.

```
sdk_version: "1.17"
```

Propiedades de simulación

La sección `simulation_properties` (obligatoria) especifica varias propiedades de la simulación. Utilice esta sección para configurar el registro y especificar una imagen de contenedor predeterminada. Esta sección es obligatoria incluso si no configura el registro ni decide especificar una imagen de contenedor predeterminada.

```
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"
```

Propiedades

`log_destination_resource_name`

Especifica el recurso en el que se SimSpace Weaver escribirán los registros.

Necesario: No. Si esta propiedad no está incluida, SimSpace Weaver no escribirá registros para la simulación.

Tipo: cadena

Valores válidos:

- El nombre de un grupo de CloudWatch registros (por ejemplo, MySimulationLogs)
- El nombre de recurso de Amazon (ARN) de un grupo de CloudWatch registros (por ejemplo, arn:aws:logs:us-west-2:111122223333:log-group/MySimulationLogs)

Note

SimSpace Weaver solo admite un destino de registro en la misma cuenta y Región de AWS que en la simulación.

log_destination_service

Indica el tipo de recurso de destino de registro cuando se especifica uno `logging_destination_resource_name` que no es un ARN.

Obligatorio: debe especificar esta propiedad si `log_destination_resource_name` está especificada y no es un ARN. No puede especificar esta propiedad si `log_destination_resource_name` no está especificada o es un ARN.

Tipo: cadena

Valores válidos:

- `logs`: El recurso de destino del registro es un grupo de registro.

default_entity_index_key_type

Especifica el tipo de datos del campo de clave de índice de las entidades de simulación.

Obligatorio: sí

Tipo: cadena

Valores válidos: `Vector3<f32>`

default_image

Especifica la imagen contenedora predeterminada para la simulación (no se admite en la versión 1.13 y 1.12). Si se especifica esta propiedad, los dominios que no especifiquen un `image` utilizarán el `default_image`.

Obligatorio: no

Tipo: cadena

Valores válidos:

- El URI de un repositorio en Amazon Elastic Container Registry (Amazon ECR) (por ejemplo, `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`)

Procesos de trabajo

La sección `workers` (obligatoria) especifica las configuraciones de los grupos de trabajadores (grupos de trabajadores). SimSpace Weaver utiliza esta información junto con `placement_constraints` para configurar la infraestructura subyacente de la simulación. En este momento solo es compatible un grupo de trabajo.

Para especificar las propiedades de un grupo de trabajo, sustituya *worker-group-name* por el nombre que elija. El nombre debe tener entre 3 y 64 caracteres y puede contener los caracteres A-Z, a-z, 0-9 y _ (guion). Especifique las propiedades del grupo de trabajo después del nombre.

```
workers:  
  worker-group-name:  
    type: "sim.c5.24xlarge"  
    desired: number-of-workers
```

Propiedades

type

Especifica el tipo de trabajador.

Obligatorio: sí

Tipo: cadena

Valores válidos: `sim.c5.24xlarge`

desired

Especifica el número de trabajadores deseado para este grupo de trabajadores.

Obligatorio: sí

Tipo: entero

Valores válidos: 1-3. Su Service Quota (límite) para el número de trabajadores de las simulaciones determina el valor máximo de esta propiedad. Por ejemplo, si su cuota de servicio es 2, el valor máximo de esta propiedad es 2. También puede solicitar un aumento de Service Quotas. Para obtener más información, consulte [SimSpace Puntos finales y cuotas de Weaver](#).

Reloj

La sección `clock` (obligatoria) especifica las propiedades del reloj de simulación.

```
clock:  
  tick_rate: tick-rate
```

Propiedades

`tick_rate`

Especifica el número de graduaciones por segundo que el reloj publica en las aplicaciones.

Obligatorio: sí

Type:

- Versión 1.14 y 1.15: cadena
- Versión 1.13 y 1.12: entero

Valores válidos:

- Versión 1.14 y 1.15: "10" | "15" | "30" | "unlimited"
 - "unlimited": el reloj envía la siguiente graduación tan pronto como todas las aplicaciones finalicen sus operaciones de confirmación de la graduación actual.
- Versión 1.13 y 1.12: 10 | 15 | 30

Estrategias de partición

La sección `partitioning_strategies` (obligatoria) especifica la organización de las particiones de un dominio espacial.

Note

SimSpace Weaver solo admite 1 estrategia de particionamiento.

Para especificar las propiedades de una estrategia de particionamiento, sustituya *partitioning-strategy-name* por el nombre que elija. El nombre debe tener entre 3 y 64 caracteres y puede contener los caracteres A-Z, a-z, 0-9 y _ (guion). Especifique las propiedades de la estrategia de partición después del nombre.

```
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
```

Propiedades

topology

Especifica la topología (esquema de disposición de particiones) de esta estrategia de partición.

Obligatorio: sí

Tipo: cadena

Valores válidos: "Grid"

aabb_bounds

Especifica los límites del cuadro delimitador alineado con el eje principal (AABB) para su simulación. Los límites se especifican como matrices ordenadas de 2 elementos que describen los valores mínimo y máximo (en ese orden) de cada eje (x y y).

Obligatorio: condicional. Esta propiedad es obligatoria (y solo se puede especificar) si la topología está establecida como "Grid".

Tipo: Float matriz (para cada eje)

Valores válidos: $-3.4028235e38$ - $3.4028235e38$

grid_placement_groups

Especifica el número de grupos de ubicación a lo largo de cada eje (x e y) en una topología de cuadro. Un grupo de ubicación es un conjunto de particiones (en el mismo dominio) que son espacialmente adyacentes.

Obligatorio: condicional. Esta propiedad es obligatoria (y solo se puede especificar) si la topología está establecida como "Grid". Si no especifica una configuración de grupos de ubicación, SimSpace Weaver calculará uno. Cualquier dominio que utilice una estrategia de particionamiento sin una configuración de grupos de ubicación debe especificar una `grid_partition` (consulte [Estrategia de particionamiento del dominio espacial](#)).

Tipo: entero (para cada eje)

Valores válidos: 1-20. Se recomienda que x^* y sea igual al número deseado de trabajadores. De lo contrario, SimSpace Weaver intentará equilibrar sus grupos de colocación entre los trabajadores disponibles.

Dominios

La sección `domains` (obligatoria) especifica las propiedades de cada uno de sus dominios. Todas las simulaciones deben tener al menos una sección para un dominio espacial. Puede crear distintas secciones para dominios adicionales. Cada tipo de dominio tiene su propio formato de configuración.

Important

Las versiones 1.13 y 1.12 no admiten varios dominios espaciales.

Important

SimSpace Weaver admite hasta 5 dominios para cada simulación. Esto incluye todos los dominios espaciales, personalizados y de servicio.

```
domains:  
  domain-name:
```

```

domain-configuration
domain-name:
  domain-configuration
  ...

```

Configuración del dominio

- [Configuración de dominio espacial](#)
- [Configuración de dominio personalizada](#)
- [Configuración de dominio de servicio](#)

Configuración de dominio espacial

Para especificar las propiedades de un dominio espacial, sustituya *spatial-domain-name* por el nombre que elija. El nombre debe tener entre 3 y 64 caracteres y puede contener los caracteres A-Z, a-z, 0-9 y _ (guion). Especifique las propiedades del dominio espacial después del nombre.

```

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"

```

Estrategia de particionamiento del dominio espacial

La sección `launch_apps_by_partitioning_strategy` (obligatoria) especifica la estrategia de partición y las dimensiones (en número de particiones) del espacio de simulación.

```

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

```

Propiedades

partitioning_strategy

Especifica la estrategia de partición para este dominio espacial.

Obligatorio: sí

Tipo: cadena

Valores válidos: el valor de esta propiedad debe coincidir con el nombre de una estrategia de partición definida en la sección `partitioning_strategies`. Para obtener más información, consulte [Estrategias de partición](#).

grid_partition

Especifica el número de particiones a lo largo de cada eje (x e y) en una topología de cuadro. Estas dimensiones describen el espacio de simulación total de este dominio.

Obligatorio: condicional. Únicamente puede especificarse si se ha establecido la topología como "Grid". Esta propiedad depende de la propiedad `grid_placement_groups` de la estrategia de partición especificada para este dominio:

- Esta propiedad es obligatoria si la estrategia de particionamiento de este dominio no especifica ninguna configuración de `grid_placement_groups`.
- Si hay una configuración de `grid_placement_groups` pero no especifica `grid_partition`, SimSpace Weaver utilizará las mismas dimensiones que la configuración `grid_placment_groups`.
- Si especifica `grid_placement_groups` y `grid_partition`, las dimensiones de `grid_partition` deben ser múltiplos de las dimensiones de `grid_placement_groups` (por ejemplo, si las dimensiones de `grid_placement_groups` son 2x2, algunas dimensiones válidas para `grid_partition` son 2x2, 4x4, 6x6, 8x8, 10x10).

Tipo: entero (para cada eje)

Valores válidos: 1-20

Configuración de aplicaciones espaciales

La sección `app_config` (obligatoria) especifica el paquete, la configuración de inicio y los requisitos de recursos de las aplicaciones de este dominio.

```
app_config:  
  package: "app-package-s3-uri"  
  launch_command: ["app-launch-command", "parameter1", ...]  
  required_resource_units:  
    compute: app-resource-units
```

Propiedades

package

Especifica el paquete (archivo zip) que contiene el archivo ejecutable o binario de la aplicación. Los datos tienen que estar almacenados en un bucket de Amazon S3. Solo se admite el formato de archivo ZIP.

Obligatorio: sí

Tipo: cadena

Valores válidos: Amazon S3 URI del paquete en un bucket de Amazon S3. Por ejemplo, `s3://weaver-myproject-111122223333-app-zips-us-west-2/MySpatialApp.zip`.

launch_command

Especifica el nombre del archivo ejecutable/binario y los parámetros de la línea de comandos para lanzar la aplicación. Cada token de cadena de línea de comandos es un elemento de la matriz.

Obligatorio: sí

Tipo: matriz de cadenas

required_resource_units

Especifica el número de unidades de recursos que SimSpace Weaver debe asignar a cada instancia de esta aplicación. Una unidad de recursos es una cantidad fija de unidades centrales de procesamiento virtuales (vCPUs) y memoria de acceso aleatorio (RAM) sobre un trabajador. Para obtener más información acerca de recursos, consulte [Puntos de conexión y Service Quotas](#). La propiedad de `compute` especifica una asignación de unidades de recursos para la familia de trabajadores `compute` y, actualmente, es el único tipo de asignación válido.

Obligatorio: sí

Tipo: entero

Valores válidos: 1-4

Imagen de contenedor personalizada

La `image` propiedad (opcional) especifica la ubicación de una imagen de contenedor que se SimSpace Weaver utiliza para ejecutar aplicaciones en este dominio (no es compatible con las versiones 1.13 y 1.12). Proporcione la imagen URI a un repositorio de Amazon Elastic Container Registry (Amazon ECR) que contenga la imagen. Si no se especifica esta propiedad, pero la `default_image` se especifica en la sección `simulation_properties` de nivel superior, las aplicaciones de este dominio utilizan la `default_image`. Para obtener más información, consulte [Contenedores personalizados](#).

```
image: "ecr-repository-uri"
```

Propiedades

`image`

Especifica la ubicación de una imagen de contenedor para ejecutar aplicaciones en este dominio.

Obligatorio: no

Tipo: cadena

Valores válidos:

- El URI de un repositorio en Amazon Elastic Container Registry (Amazon ECR) (por ejemplo, `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`)

Configuración de dominio personalizada

Para especificar las propiedades de un dominio personalizado, sustituya `custom-domain-name` por el nombre que elija. El nombre debe tener entre 3 y 64 caracteres y puede contener los caracteres A-Z, a-z, 0-9 y `_` (guion). Especifique las propiedades del dominio personalizado después del nombre. Repita este proceso para cada dominio personalizado.

```
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"
```

Propiedades

launch_apps_via_start_app_call

Esta propiedad es necesaria para iniciar las aplicaciones personalizadas mediante el StartApp API.

Obligatorio: sí

Tipo: N/A

Valores válidos: {}

Configuración de aplicación personalizada

El `app_config` section (obligatorio) especifica el paquete, la configuración de inicio, los requisitos de recursos y los puertos de red de las aplicaciones de este dominio personalizado.

```
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, ...]
```

Propiedades

package

Especifica el paquete (archivo zip) que contiene el archivo ejecutable o binario de la aplicación. Los datos tienen que estar almacenados en un bucket de Amazon S3. Solo se admite el formato de archivo ZIP.

Obligatorio: sí

Tipo: cadena

Valores válidos: Amazon S3 URI del paquete en un bucket de Amazon S3. Por ejemplo, `s3://weaver-myproject-111122223333-app-zips-us-west-2/MyCustomApp.zip`.

`launch_command`

Especifica el nombre del archivo ejecutable/binario y los parámetros de la línea de comandos para lanzar la aplicación. Cada token de cadena de línea de comandos es un elemento de la matriz.

Obligatorio: sí

Tipo: matriz de cadenas

`required_resource_units`

Especifica el número de unidades de recursos que SimSpace Weaver debe asignar a cada instancia de esta aplicación. Una unidad de recursos es una cantidad fija de unidades centrales de procesamiento virtuales (vCPUs) y memoria de acceso aleatorio (RAM) sobre un trabajador. Para obtener más información acerca de recursos, consulte [Puntos de conexión y Service Quotas](#). La propiedad de `compute` especifica una asignación de unidades de recursos para la familia de trabajadores `compute` y, actualmente, es el único tipo de asignación válido.

Obligatorio: sí

Tipo: entero

Valores válidos: 1-4

`endpoint_config`

Especifica los puntos de conexión de la red para las aplicaciones de este dominio. El valor de `ingress_ports` especifica los puertos a los que se enlazan las aplicaciones personalizadas para las conexiones entrantes de los clientes. SimSpace Weaver asigna los puertos asignados dinámicamente a los puertos de entrada especificados. Los puertos de entrada son a la vez TCP y UDP Use la `DescribeApp` API para encontrar el número de puerto real para conectar sus clientes.

Obligatorio: No. Si no especifica la configuración de los puntos de conexión, sus aplicaciones personalizadas de este dominio no tendrán puntos de conexión de red.

Tipo: matriz de enteros

Valores válidos: 1024-49152. Los valores deben ser únicos.

Imagen de contenedor personalizada

La `image` propiedad (opcional) especifica la ubicación de una imagen de contenedor que se SimSpace Weaver utiliza para ejecutar aplicaciones en este dominio (no es compatible con las versiones 1.13 y 1.12). Proporcione la imagen URI a un repositorio de Amazon Elastic Container Registry (Amazon ECR) que contenga la imagen. Si no se especifica esta propiedad, pero la `default_image` se especifica en la sección `simulation_properties` de nivel superior, las aplicaciones de este dominio utilizan la `default_image`. Para obtener más información, consulte [Contenedores personalizados](#).

```
image: "ecr-repository-uri"
```

Propiedades

`image`

Especifica la ubicación de una imagen de contenedor para ejecutar aplicaciones en este dominio.

Obligatorio: no

Tipo: cadena

Valores válidos:

- El URI de un repositorio en Amazon Elastic Container Registry (Amazon ECR) (por ejemplo, `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`)

Configuración de dominio de servicio

Para especificar las propiedades de un dominio de servicio, sustituya `service-domain-name` por el nombre que elija. El nombre debe tener entre 3 y 64 caracteres y puede contener los caracteres A-Z, a-z, 0-9 y `_` (guion). Especifique las propiedades del dominio de servicio después del nombre. Repita este proceso para cada dominio de servicio.

```
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"
```

Lanzar aplicaciones por trabajador

La sección `launch_apps_per_worker` (obligatoria) indica que se trata de una configuración de dominio de servicio y especifica la cantidad de aplicaciones de servicio que se van a lanzar por trabajador.

```
launch_apps_per_worker:  
  count: number-of-apps-to-launch
```

Propiedades

count

Esta propiedad especifica el número de aplicaciones de servicio que se van a lanzar por trabajador.

Obligatorio: sí

Tipo: entero

Valores válidos: {} | 1 | 2. Un valor de {} especifica el valor de 1 predeterminado.

Configuración de aplicación de servicio

El `app_config` section (obligatorio) especifica el paquete, la configuración de inicio, los requisitos de recursos y los puertos de red de las aplicaciones de este dominio de servicio.

```
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, ...]
```

Propiedades

package

Especifica el paquete (archivo zip) que contiene el archivo ejecutable o binario de la aplicación. Los datos tienen que estar almacenados en un bucket de Amazon S3. Solo se admite el formato de archivo ZIP.

Obligatorio: sí

Tipo: cadena

Valores válidos: Amazon S3 URI del paquete en un bucket de Amazon S3. Por ejemplo, `s3://weaver-myproject-111122223333-app-zips-us-west-2/MyServiceApp.zip`.

launch_command

Especifica el nombre del archivo ejecutable/binario y los parámetros de la línea de comandos para lanzar la aplicación. Cada token de cadena de línea de comandos es un elemento de la matriz.

Obligatorio: sí

Tipo: matriz de cadenas

required_resource_units

Especifica el número de unidades de recursos que SimSpace Weaver debe asignar a cada instancia de esta aplicación. Una unidad de recursos es una cantidad fija de unidades centrales de procesamiento virtuales (vCPUs) y memoria de acceso aleatorio (RAM) sobre un trabajador. Para obtener más información acerca de recursos, consulte [Puntos de conexión y Service Quotas](#). La propiedad de `compute` especifica una asignación de unidades de recursos para la familia de trabajadores `compute` y, actualmente, es el único tipo de asignación válido.

Obligatorio: sí

Tipo: entero

Valores válidos: 1-4

endpoint_config

Especifica los puntos de conexión de la red para las aplicaciones de este dominio. El valor de `ingress_ports` especifica los puertos a los que se enlazan las aplicaciones de servicio para las conexiones entrantes de los clientes. SimSpace Weaver asigna los puertos asignados dinámicamente a los puertos de entrada especificados. Los puertos de entrada son a la vez TCP y UDP. Use la `DescribeApp` API para encontrar el número de puerto real para conectar sus clientes.

Obligatorio: No. Si no especifica la configuración de los puntos de conexión, sus aplicaciones de servicio de este dominio no tendrán puntos de conexión de red.

Tipo: matriz de enteros

Valores válidos: 1024-49152. Los valores deben ser únicos.

Imagen de contenedor personalizada

La `image` propiedad (opcional) especifica la ubicación de una imagen de contenedor que se utiliza SimSpace Weaver para ejecutar aplicaciones en este dominio (no es compatible con las versiones 1.13 y 1.12). Proporcione la imagen URI a un repositorio de Amazon Elastic Container Registry (Amazon ECR) que contenga la imagen. Si no se especifica esta propiedad, pero la `default_image` se especifica en la sección `simulation_properties` de nivel superior, las aplicaciones de este dominio utilizan la `default_image`. Para obtener más información, consulte [Contenedores personalizados](#).

```
image: "ecr-repository-uri"
```

Propiedades

image

Especifica la ubicación de una imagen de contenedor para ejecutar aplicaciones en este dominio.

Obligatorio: no

Tipo: cadena

Valores válidos:

- El URI de un repositorio en Amazon Elastic Container Registry (Amazon ECR) (por ejemplo, `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`)

Restricciones para la ubicación

La sección `placement_constraints` (opcional) especifica qué dominios espaciales de SimSpace Weaver deben colocarse juntos en los mismos elementos de trabajo. Para obtener más información, consulte [Configuración de dominios espaciales](#).

Important

Las versiones 1.13 y 1.12 no son compatibles con `placement_constraints`.

```
placement_constraints:  
  - placed_together: ["spatial-domain-name", "spatial-domain-name", ...]  
    on_workers: ["worker-group-name"]
```

Propiedades

`placed_together`

Especifica los dominios espaciales que SimSpace Weaver deben colocarse juntos.

Obligatorio: sí

Tipo: matriz de cadenas

Valores válidos: nombres de los dominios espaciales especificados en el esquema

`on_workers`

Especifica el grupo de trabajo en el que SimSpace Weaver debe colocar los dominios.

Obligatorio: sí

Tipo: matriz de cadenas de 1 elemento

Valores válidos: nombre del grupo de trabajo especificado en el esquema

SimSpace Weaver Referencias de API

SimSpace Weaver tiene 2 conjuntos diferentes de interfaces de programación de aplicaciones (API):

- API de servicio: estas API controlan el servicio y los recursos del servicio, como las simulaciones, los relojes y las aplicaciones. Forman parte del kit de desarrollo de AWS software (SDK) principal y puede utilizar la interfaz de línea de AWS comandos (CLI) para llamarlos. Para obtener más información sobre las API de servicio, consulte la [referencia de la API de SimSpace Weaver](#).
- API de SDK de aplicaciones: estas API controlan los datos de la simulación. Se utilizan en el código de la aplicación para realizar tareas como leer y escribir datos de campos de entidades, trabajar con suscripciones y supervisar los eventos de la simulación. Para obtener más información, consulta la documentación del SDK de la SimSpace Weaver aplicación en la carpeta del SDK de la aplicación: `sdk-folder\SimSpaceWeaverAppSdk\documentation`

Note

sdk-folder es la carpeta en la que descomprimiste el paquete.
SimSpaceWeaverAppSdkDistributable

AWS SimSpace Weaver versiones

Mejoramos AWS SimSpace Weaver continuamente. Debes descargar el último SDK de la SimSpace Weaver aplicación cuando publiquemos una nueva versión si quieres aprovechar las nuevas funciones y actualizaciones de funciones. Para ejecutar una simulación existente con una versión más reciente, es posible que tenga que actualizar su esquema y código y, a continuación, iniciar una nueva instancia de la simulación. No tiene que actualizar y puede seguir ejecutando las simulaciones existentes con las versiones anteriores. Puede consultar esta página para ver las diferencias entre las versiones. Actualmente, todas las versiones son compatibles.

Important

La última versión de la [Guía del AWS SimSpace Weaver usuario](#) solo incluye la versión más reciente del servicio. Puede encontrar la documentación de las versiones anteriores en el [catálogo de AWS SimSpace Weaver guías](#), disponible en la [página principal de la documentación](#).

Última versión

La última versión es: 1.17.0

Cómo encontrar la versión actual

Si creó una simulación con el SDK de la SimSpace Weaver aplicación, el `create-project` script descargó una versión de las bibliotecas del SDK en un subdirectorio del suyo. *sdk-folder* El subdirectorio que contiene las bibliotecas del SDK tiene un nombre que incluye el número de versión del SDK: `SimSpaceWeaverAppSdk-sdk-version`. Por ejemplo, las bibliotecas de la versión 1.16.0 están disponibles. `SimSpaceWeaverAppSdk-1.16.0`

También puedes encontrar la versión del paquete distribuible del SDK de la SimSpace Weaver aplicación en el archivo `app_sdk_distributable_version.txt` de texto de tu *sdk-folder*

Descargue la versión más reciente

Utilice uno de los siguientes enlaces para descargar la versión más reciente.

- [Paquete distribuible completo del SDK de la aplicación](#)
- [Solo las bibliotecas del SDK de aplicaciones](#)

También puedes descargar el paquete distribuible completo del SDK de la SimSpace Weaver aplicación desde la [SimSpace Weaver consola](#) del. AWS Management Console En el panel de navegación, seleccione Descargar el SDK de la aplicación en el panel de navegación.

Warning

No lo utilices AWS CLI para descargar nada que parezca ser el paquete distribuible del SDK de la SimSpace Weaver aplicación. Utilice únicamente los enlaces de descarga de esta página o el enlace de descarga de la consola. No se admite ningún otro método o ubicación de descarga y puede contener código obsoleto, incorrecto o malintencionado.

Solución de problemas de descargas del SDK de aplicaciones

Usamos Amazon CloudFront (CloudFront) para distribuir los archivos.zip del SDK de la aplicación. Es posible que experimente algunas de las siguientes situaciones.

- El paquete descargado no es la versión más reciente
 - Si el archivo.zip que descargaste no contiene la última versión, es posible que la memoria caché de tu ubicación CloudFront perimetral aún no se haya actualizado. Descárguelo de nuevo tras 24 horas.
- Aparece un error HTTP 4xx o 5xx al utilizar un enlace de descarga
 - Inténtelo de nuevo en 24 horas. Si aparece el mismo error, utilice el enlace Comentarios situado en la parte inferior de la consola de [SimSpace Weaver](#) para informar del problema. Seleccione Informar de un problema como tipo de comentario.
- Su navegador informa que no puede cargar la página
 - Es posible que tenga un problema de configuración de la red local o del navegador. Compruebe que puede cargar otras páginas Borre la memoria caché del navegador e inténtelo de nuevo. Asegúrese de que no cuenta con reglas de firewall que puedan bloquear la dirección URL de descarga.
- Aparece un error al intentar guardar el archivo
 - Compruebe los permisos del sistema de archivos local para asegurarse de que cuenta con los permisos correctos para guardar el archivo.

- Su navegador muestra AccessDenied
 - Si ha introducido la dirección URL de forma manual en el navegador, compruebe que es correcta. Si ha utilizado un enlace de descarga, asegúrese de que algo no interfiera con la URL de su navegador y vuelva a utilizar el enlace.

Instale la versión más reciente

Para instalar la versión más reciente

1. [Descargue la versión más reciente.](#)
2. Descomprime el SimSpaceWeaverAppSdkDistributable archivo.zip en una carpeta.
3. Ejecuta `python setup.py` desde la carpeta descomprimida del SDK de la aplicación de la última versión SimSpace Weaver .
4. Usa la carpeta descomprimida del SDK de la SimSpace Weaver aplicación de la última versión en lugar de la versión anterior.

Versiones del servicio

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
1.17.0	<p>Cambios importantes en el paquete distribuible del SDK SimSpace Weaver de la aplicación</p> <ul style="list-style-type: none"> • Sustituimos los scripts por lotes de Windows y Bash de Linux por scripts basados en Python. Por lo 	17 de abril de 2024	Esta guía	<ul style="list-style-type: none"> • Paquete completo • Solo bibliotecas <p>Consulte Solución de problemas</p>

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
	<p>tanto, ahora se requiere que Python 3.9 use los scripts y los ejemplos, incluso si no usa (o tiene la intención de usar) el SDK de Python.</p> <ul style="list-style-type: none">• Esta versión aumenta el soporte para Amazon Linux 2.• Hemos corregido varios errores en SimSpace Weaver Local. <p>Para más información, consulte las notas de la versión.</p> <p>Corrección de errores</p> <ul style="list-style-type: none">• Hemos corregido un error que provocaba que las entidades perdieran su propiedad si no completaban su transferencia			

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
	entre trabajos remotos.			
1.16.0	<p>Nueva característica:</p> <ul style="list-style-type: none">Ahora puedes usar las API de mensajería del SDK de la SimSpace Weaver aplicación para enviar y recibir mensajes entre tus aplicaciones. Esta función está disponible para C++, Python y las integraciones de Unity y Unreal Engine 5.	12 de febrero de 2024	Consulta el catálogo de AWS SimSpace Weaver guías .	<ul style="list-style-type: none">Paquete completoSolo bibliotecas <p>Consulte Solución de problemas</p>

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
1.15.3	<p>SimSpace Weaver Local update:</p> <ul style="list-style-type: none">• Cambiamos SimSpace Weaver Local para conseguir una mejor correspondencia con el desarrollo de Nube de AWS. Esta versión afecta a los proyectos y flujos de trabajo de C++, Python, Unity y Unreal Engine para SimSpace Weaver Local.	4 de diciembre de 2023	Consulte el catálogo de AWS SimSpace Weaver guías .	No está disponible para su descarga

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
1.15.2	<p>Actualización del paquete distribuido del SDK de la aplicación:</p> <ul style="list-style-type: none">Hemos actualizado Dockerfile para usar la versión específica requerida de cmake. Las compilaciones de contenedores de Docker podrían fallar sin este cambio.	2 de noviembre de 2023	Consulte el catálogo AWS SimSpace Weaver de guías .	No está disponible para su descarga
1.15.1	<p>Actualización de una característica:</p> <ul style="list-style-type: none">SDK de Python: esta versión corrige un problema que provocaba que las simulaciones basadas en Python fallaran en Nube de AWS.	22 de septiembre de 2023	Consulte el catálogo AWS SimSpace Weaver de guías .	No está disponible para su descarga

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
1.15.0	<p data-bbox="272 275 505 352">Nueva característica:</p> <ul data-bbox="272 401 548 1192" style="list-style-type: none"><li data-bbox="272 401 548 1192">• SDK de Python: Ahora puede desarrollar sus simulaciones en Python. El paquete distributable del SDK de la aplicación SimSpace Weaver incluye una plantilla para un proyecto de Python de muestra y su cliente de visualización de Python.	31 de agosto de 2023	Consulte el catálogo AWS SimSpace Weaver de guías .	No está disponible para su descarga

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
1.14.0	<p>Nuevas características:</p> <ul style="list-style-type: none">• Contenido personalizado: cree su propia imagen de contenedor basada en Amazon Linux 2 (AL2), guárdela en Amazon Elastic Container Registry (Amazon ECR) y utilícela para ejecutar sus aplicaciones de SimSpace Weaver en Nube de AWS.• Varios dominios espaciales: cree más de un dominio espacial en una simulación. Separe la lógica de simulación en lugar de combinarla toda en una sola aplicación espacial.	26 de julio de 2023	Consulte el catálogo AWS SimSpace Weaver de guías .	No está disponible para su descarga

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
	<p>Asigne diferentes recursos a los dominios espaciales en función de sus requisitos.</p> <ul style="list-style-type: none"> • Frecuencia de graduación ilimitada: permite que la simulación se ejecute tan rápido como se ejecute el código. Configure el reloj de la simulación para que envíe la siguiente graduación tan pronto como todas las aplicaciones terminen sus operaciones de confirmación de la graduación actual. <p>SimSpace Weaver SDK de aplicaciones:</p> <ul style="list-style-type: none"> • El valor de <code>tick_rate</code> 			

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
	<p>ahora es una cadena. El valor debe incluir comillas dobles ("). La frecuencia de graduación de las versiones anteriores sigue siendo un número entero.</p>			
1.13.1	<p>SimSpace Weaver SDK de aplicaciones:</p> <ul style="list-style-type: none">Actualización de una característica: la creación de proyectos ahora funciona correctamente con la plantilla PathfindingSampleUnreal .	7 de junio de 2023	Consulte el catálogo AWS SimSpace Weaver de guías .	No está disponible para su descarga

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
1.13.0	<p>SimSpace Weaver API de servicio:</p> <ul style="list-style-type: none"> • Nueva acción de CreateSnapshot • Cambios en la acción de StartSimulation : <ul style="list-style-type: none"> • Se ha añadido un parámetro de SnapshotS3Location para comenzar desde una instantánea. • El parámetro SchemaS3Location ahora es opcional. • Cambios en el resultado de DescribeSimulation : 	29 de abril de 2023	<p>Consulte el catálogo AWS SimSpace Weaver de guías.</p>	No está disponible para su descarga

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
	<ul style="list-style-type: none"> • SchemaError está obsoleto. • Se ha añadido el campo StartError . • Se ha añadido el campo SnapshotS3Location . • Se ha añadido un estado de simulación SNAPSHOT_IN_PROGRESS . • Nuevo tipo de datos de S3Destination <p>SimSpace Weaver consola:</p> <ul style="list-style-type: none"> • Nueva funcionalidad para crear instantáneas. 			

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
	<ul style="list-style-type: none"> Nueva funcionalidad para iniciar una simulación a partir de una instantánea. <p>SimSpace Weaver SDK de aplicaciones:</p> <ul style="list-style-type: none"> Nuevos scripts para admitir instantáneas <ul style="list-style-type: none"> <code>create-snapshot- <i>project-name</i> .bat</code> <code>start-from-snapshot- <i>project-name</i> .bat</code> <code>quick-start-from-snapshot- <i>project-name</i> -cli.bat</code> <code>list-snapshots- <i>project-name</i> .bat</code> Los proyectos ahora utilizan un único bucket de Amazon S3 por proyecto: 			

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
	<code>weaver-<i>lowercas</i> <i>-project-</i> <i>name</i> <i>-account-</i> <i>n</i> <i>umber</i> <i>-region</i></code>			

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
1.12.3	<p>SimSpace Weaver SDK de aplicaciones:</p> <ul style="list-style-type: none"> • Los siguientes scripts ahora admiten el parámetro <code>--maximum-duration</code> : • <code>quick-start-<i>project-name</i> - cli.bat</code> • <code>quick-start-<i>project-name</i> - cli.sh</code> • <code>start-simulation-<i>project-name</i> .bat</code> • <code>start-simulation-<i>project-name</i> .sh</code> • <code>run-<i>project-name</i> .bat</code> • <code>run-<i>project-name</i> .sh</code> 	27 de marzo de 2023	<p>Consulte el catálogo AWS SimSpace Weaver de guías.</p>	No está disponible para su descarga

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
1.12.2	<p>SimSpace Weaver SDK de aplicaciones:</p> <ul style="list-style-type: none">• Corrección de error: <code>docker-create-image.bat</code> ahora funciona correctamente.	1 de marzo de 2023	Consulte el catálogo AWS SimSpace Weaver de guías .	No está disponible para su descarga

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
1.12.1	<p>SimSpace Weaver SDK de aplicaciones:</p> <ul style="list-style-type: none"> • Los scripts ahora aceptan un AWS CLI perfil para usarlo en la AWS autenticación. • Los scripts ahora admiten AWS IAM Identity Center la AWS autenticación. <p>SimSpace Weaver Local:</p> <ul style="list-style-type: none"> • Corrección de error: <code>Api::BeginUpdateWillBlock</code> ahora vuelve correctamente a <code>true</code> si todas las aplicaciones espaciales no se han unido a la simulación. 	28 de febrero de 2023	<p>Consulte el catálogo AWS SimSpace Weaver de guías.</p>	No está disponible para su descarga

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
1.12.0	Lanzamiento de disponibilidad general	29 de noviembre de 2022	Consulte el catálogo AWS SimSpace Weaver de guías .	No está disponible para su descarga

AWS SimSpace Weaver versión 1.17.0

Esta versión es una revisión del paquete distribuible del SDK de la SimSpace Weaver aplicación. Sustituimos los scripts obsoletos por lotes de Windows y Bash de Linux por scripts basados en Python.

Important

Python 3.9 ahora es un requisito para usar los scripts y las muestras, no solo para el SDK de Python.

Contenido

- [Cambios importantes para la versión 1.17.0](#)
- [Actualiza un proyecto a la versión 1.17.0](#)
- [Preguntas frecuentes sobre la versión 1.17.0](#)

Cambios importantes para la versión 1.17.0

- Creación de proyectos simplificada
 - Después de `setup.py` ejecutarlo, puedes crear tu propio proyecto simplemente copiando y pegando una muestra.
- Muestras con un solo clic
 - El archivo zip de distribución ahora contiene ready-to-use ejemplos que funcionan después de configurar la distribución.
- Cada SDK ahora existe en su propio directorio: `cpppython`, `unreal`, `unity`. Es posible que tengas que actualizar tus rutas en función del SDK que utilices.

- Mejoras en los scripts auxiliares.
 - Los scripts ahora contienen múltiples AWS CLI opciones para maximizar su flexibilidad.
 - El inicio y la conexión del cliente de consola están integrados como parte del inicio rápido.
 - Salida de consola mejorada.
 - La creación de muestras con `quick-start` Unreal y Unity ahora funciona sin necesidad de realizar más pasos manuales.
 - SimSpace Weaver Local ahora funciona con solo llamar `quick-start`, ya no es necesario crear ni lanzar archivos manualmente.
 - SimSpace Weaver Local `quick-start` tiene soporte integrado para registrar los resultados de las aplicaciones.
 - SimSpace Weaver Local ahora se puede iniciar en un entorno que no sea de interfaz gráfica de usuario, como en una sesión ssh.
 - La función de «contenedor personalizado» ahora está integrada en el `quick-start` script.
 - Mayor compatibilidad con Amazon Linux 2 (AL2): los flujos de trabajo de scripts para Windows y AL2 ahora son comparables. Anteriormente, los proyectos de AL2 requerían más pasos manuales y SimSpace Weaver Local no eran compatibles con AL2.
- Los complementos Unreal Engine y Unity ahora se incluyen como parte del paquete distribuible del SDK de la SimSpace Weaver aplicación.
- Correcciones de errores para SimSpace Weaver Local
 - Se ha corregido un error por el que se podía asignar a las entidades el mismo ID de entidad.
 - Se ha corregido un error por el que se podía asignar a dos particiones el mismo ID de partición.
 - Se ha corregido un error relacionado con el intento de las aplicaciones de escribir en entidades que no eran de su propiedad.
 - Se ha resuelto un problema de pérdida de memoria.

Actualiza un proyecto a la versión 1.17.0

1. Configurar la distribución 1.17.0: Vuelva a realizar los procedimientos de configuración, ya que los cambiamos por la versión 1.17.0. Para obtener más información, consulte [Configuración de SimSpace Weaver](#).
2. Cada SDK de la aplicación Weaver ahora existe en su propio directorio. Actualice sus rutas de compilación para reflejar esto.

- a. Directorio C++: `SimSpaceWeaverAppSdk/cpp`
 - El SDK de la aplicación SimSpace Weaver C++ ahora usa un `FindSimSpaceWeaverAppSdk.cmake` archivo. Este archivo configura un `weaver` objetivo al que se enlaza e incluye importantes correcciones de errores al compilar para Weaver en. Nube de AWS Deberías usar esto en lugar de vincular directamente a los binarios.
 - b. Directorio de Python: `SimSpaceWeaverAppSdk/python`
 - c. Complemento de Unity: `SimSpaceWeaverAppSdk/unity`
 - d. Complemento Unreal Engine: `SimSpaceWeaverAppSdk/unreal`
3. Los `tools` scripts anteriores no funcionarán con la nueva SimSpace Weaver distribución. Para usar los nuevos `tools` scripts en su proyecto:
- a. Elimine los `tools/local` directorios `tools/linux` y directorios `antiguotools/windows`.
 - b. Copia el `tools` directorio de un proyecto de ejemplo que utilice el mismo SDK de SimSpace Weaver aplicaciones que tu proyecto. Asegúrese de haberlo ejecutado `setup.py` antes de copiar este directorio.

Important

Se garantiza que los scripts de las herramientas solo funcionen con los proyectos de muestra. Puede que tenga que editar estos scripts, especialmente el `build.py` script, para trabajar con su proyecto. Todas las modificaciones serán exclusivas de tu proyecto, por lo que no podemos proporcionarte ninguna orientación.

Preguntas frecuentes sobre la versión 1.17.0

¿Tengo que actualizar a la versión 1.17.0?

No es una actualización obligatoria porque no hay cambios en la SimSpace Weaver API ni en el SDK de la SimSpace Weaver aplicación. Debes actualizar a la versión 1.17.0 si quieres usar la versión 1.17.0 SimSpace Weaver Local, que contiene varias correcciones de errores.

¿Cuál es la versión mínima de Python requerida?

Python 3.9 es la versión mínima.

¿Cuál es la versión mínima de CMake requerida?

La versión 3.13 de CMake es la mínima.

¿Cuál es la versión mínima requerida de Unreal Engine?

Unreal Engine 5.0 es la mínima.

¿Cuál es la versión mínima de Unity requerida?

La versión 2022.3.19.F1 de Unity es la mínima.

AWS SimSpace Weaver versión 1.15.1

Esta versión es una actualización obligatoria para el SDK de Python que se publicó originalmente en la SimSpace Weaver versión 1.15.0. Corrige un problema de desajuste de versiones que provocaba que las simulaciones basadas en Python fallaran en el Nube de AWS. Utilice esta versión en lugar de la 1.15.0.

Actualizar un proyecto de Python existente a 1.15.1

Si tiene un proyecto de Python existente que creó con la versión 1.15.0 del SDK de Python, debe realizar los siguientes pasos para actualizarlo a la 1.15.1 para que pueda ejecutarse en Nube de AWS.

En lugar de seguir este procedimiento, también puede crear un nuevo proyecto de Python con el SDK de Python 1.15.1 y mover el código personalizado al nuevo proyecto.

Para actualizar un proyecto de Python de la versión 1.15.0 a la versión 1.15.1

1. Acceda a la carpeta de su proyecto de Python.
2. En `src/PythonBubblesSample/bin/run-python` cambie la siguiente línea:

```
export PYTHONPATH=$PYTHONPATH:/roapp/lib
```

A lo siguiente:

```
export PYTHONPATH=$PYTHONPATH:$LD_LIBRARY_PATH:/roapp/lib
```

3. En `CMakeLists.txt` borre las siguientes líneas:

- ```
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)
```

Solución de problemas con la versión 1.15.1

Tras actualizar una simulación de Python de la versión 1.15.0, no se puede iniciar en el Nube de AWS

Síntomas: Transcurridos aproximadamente entre 5 y 10 minutos desde el inicio de la simulación, el registro de gestión de la simulación muestra un `internal error` y el estado de la simulación es `FAILED`.

Esto puede suceder si se incluye un archivo de biblioteca del SDK de Python 1.15.0 en el archivo zip de una aplicación. Asegúrese de haber completado los pasos para actualizar tu proyecto y de que `libweaver_app_sdk_python_v1.so` no esté en sus archivos zip ni se haga referencia a él de ninguna manera.

Preguntas frecuentes sobre la versión 1.15.1

¿Afecta esta versión a algo más que al SDK de Python?

No.

¿Tengo que actualizar a la versión 1.15.1?

No tiene que actualizar a la versión 1.15.1 si no quiere usar Python para sus aplicaciones espaciales. Si actualizó a la versión 1.15.0, sus simulaciones basadas en Python no se ejecutarán en Nube de AWS. Le recomendamos que actualice a la versión 1.15.1 si utiliza la versión 1.15.0.

¿Qué es `$LD_LIBRARY_PATH`?

Es la ubicación del SDK de Python cuando la simulación se ejecuta en Nube de AWS. Esta es una novedad de la versión 1.15.1. Hicimos este cambio para evitar problemas con las versiones de Python en el futuro. La vinculación a ese directorio es funcionalmente igual a la de la versión `libweaver_app_sdk_python_v1.so` 1.15.0.

Historial de documentos para AWS SimSpace Weaver

En la siguiente tabla se describen los cambios importantes en la SimSpace Weaver documentación.

Date	Cambio	Actualizaciones de la documentación	Versiones de API actualizadas
17 de abril de 2024	Contenido actualizado	Se actualizó en toda la guía del usuario de la versión 1.17.0. Cambios importantes en el Configuración capítulo y en los tutoriales. Introducción Consulte las notas de la versión para obtener más información.	N/A
12 de febrero de 2024	Contenido actualizado	Se actualizó el AWS SimSpace Weaver versiones capítulo para la versión 1.16.0.	N/A
12 de febrero de 2024	Nuevo contenido	Se agregó la Mensajería sección como parte de la versión 1.16.0. En esta sección, se describen las API de mensajería agregadas al SDK de la SimSpace Weaver aplicación. Puedes usar estas API para enviar y recibir mensajes entre tus aplicaciones.	N/A
12 de febrero de 2024	Contenido actualizado	Se actualizó el SimSpace Weaver referencia del esquema de simulación	N/A

Date	Cambio	Actualizaciones de la documentación	Versiones de API actualizadas
		capítulo para la versión 1.16.0.	
12 de febrero de 2024	Contenido actualizado	Se agregaron cuotas de servicio para la mensajería a al SimSpace Puntos finales y cuotas de Weaver capítulo.	N/A
12 de febrero de 2024	Nuevas guías	Divida el contenido de las versiones anteriores a la 1.16.0 en una guía independiente. Se agregó el catálogo de AWS SimSpace Weaver guías (disponible en la página de inicio de la documentación principal) para acceder a las guías de las versiones anteriores.	N/A
4 de diciembre de 2023	Contenido actualizado	Se ha actualizado el capítulo AWS SimSpace Weaver versiones para la versión 1.15.3.	N/A
4 de diciembre de 2023	Contenido actualizado	Se actualizó el AWS SimSpace Weaver versiones capítulo para incluir las instrucciones de instalación de la última versión.	N/A

Date	Cambio	Actualizaciones de la documentación	Versiones de API actualizadas
4 de diciembre de 2023	Contenido actualizado	Se han actualizado los Service Quotas de SimSpace Weaver Local .	N/A
4 de diciembre de 2023	Contenido nuevo y actualizado	Se reestructuró la sección Desarrollo local en SimSpace Weaver y se agregó una nueva página en la que se describen las diferencias para SimSpace Weaver Local introducidas en la versión 1.15.3.	N/A
7 de noviembre de 2023	Contenido actualizado	Se han actualizado las instrucciones para configurar Docker y WSL para usar el enlace/URL de descarga directa del SDK de la aplicación. Para obtener más información, consulte Configuración del entorno de SimSpace Weaver .	N/A
2 de noviembre de 2023	Contenido actualizado	Se ha actualizado la página de versiones del servicio para la versión 1.15.2. Para obtener más información, consulte Versiones del servicio .	N/A

Date	Cambio	Actualizaciones de la documentación	Versiones de API actualizadas
23 de octubre de 2023	Contenido actualizado	Se ha actualizado la página de versiones del servicio con nuevas instrucciones para descargar el paquete distribuible del SDK de la aplicación. Los clientes ahora solo deben usar uno de nuestros enlaces de descarga directa aprobados y no usarlo AWS CLI para descargar el paquete distribuible del SDK de la aplicación. Para obtener más información, consulte Descargue la versión más reciente .	N/A
22 de septiembre de 2023	Nuevo contenido	Se ha añadido una página de notas de la versión con instrucciones de actualización para la versión 1.15.1. Para obtener más información, consulte AWS SimSpace Weaver versión 1.15.1 .	N/A

Date	Cambio	Actualizaciones de la documentación	Versiones de API actualizadas
10 de septiembre de 2023	Nuevo contenido	Se ha añadido una sección de solución de problemas para las situaciones en las AWS CLI que el no reconoce SimSpace Weaver. Para obtener más información, consulte La AWS CLI no reconoce simspaceweaver .	N/A
10 de septiembre de 2023	Contenido actualizado	Se actualizaron las instrucciones de instalación para AWS CLI el WSL. Para obtener más información, consulte Configure el paquete SimSpace Weaver de distribución para Amazon Linux 2 (AL2) en Windows Subsystem for Linux (WSL) .	N/A
7 de septiembre de 2023	Actualización de API	BucketName y ahora ObjectKey son necesarios para el tipo de datos S3Location . BucketName y ahora son necesarios para el tipo de datos S3Destination .	AWS SDK: 7 de septiembre de 2020

Date	Cambio	Actualizaciones de la documentación	Versiones de API actualizadas
31 de agosto de 2023	Nuevo contenido	Se ha añadido una nueva sección para la versión 1.15.0: Trabajo con Python .	N/A
15 de agosto de 2023	Contenido actualizado	Se han actualizado las instrucciones de descarga de AWS SimSpace Weaver versiones para que solo aparezcan los buckets de S3 oficiales de Amazon de SimSpace Weaver . Otras ubicaciones de descarga no están controladas por código malintencionado AWS y pueden contener código malintencionado.	N/A
26 de julio de 2023	Contenido actualizado	Actualizado Reloj .	N/A
26 de julio de 2023	Contenido actualizado	Actualizado Configuración de dominios espaciales .	N/A
26 de julio de 2023	Nuevo contenido	Se ha añadido una nueva sección: Contenedores personalizados .	N/A
26 de julio de 2023	Contenido actualizado	Se ha actualizado AWS SimSpace Weaver versiones a la versión 1.14.0.	N/A

Date	Cambio	Actualizaciones de la documentación	Versiones de API actualizadas
6 de julio de 2023	Nuevo contenido	Se ha añadido una nueva sección: PathfindingSample el cliente de consola no se puede conectar.	N/A
7 de junio de 2023	Contenido actualizado	Se ha actualizado AWS SimSpace Weaver versiones a la versión 1.13.1.	N/A
15 de mayo de 2023	Nuevo contenido	Se ha añadido una nueva sección: Uso de instantáneas con AWS CloudFormation.	N/A
29 de abril de 2023	Nuevo contenido	Se ha añadido contenido para la versión 1.13.0. Para obtener más información, consulte AWS SimSpace Weaver versiones.	AWS SDK: 28 de abril de 2020

Date	Cambio	Actualizaciones de la documentación	Versiones de API actualizadas
27 de marzo de 2023	Nuevo contenido	Se ha añadido una sección que describe la duración máxima de las simulaciones. Se han añadido notas en los tutoriales de la versión 1.12.3, que añaden compatibilidad con el parámetro de <code>--maximum-duration</code> a los scripts del SDK de las aplicaciones SimSpace Weaver .	N/A
9 de marzo de 2023	Cambio del contenido	Se ha añadido una aclaración sobre que solo proporcionamos instrucciones para activar Docker en Windows y Windows Subsystem for Linux (WSL) y que WSL (y cualquier otro entorno Linux) no es compatible.	N/A
28 de febrero de 2023	Nuevo contenido	Se agregó un capítulo que describe las versiones. SimSpace Weaver	N/A

Date	Cambio	Actualizaciones de la documentación	Versiones de API actualizadas
28 de febrero de 2023	Cambio del contenido	Se modificó el contenido sobre la autenticación para incluir el uso de los perfiles AWS IAM Identity Center y los nombres de los AWS Command Line Interface (AWS CLI).	N/A
17 de febrero de 2023	Nuevo contenido	Se agregó una sección sobre cómo administrar tus recursos con AWS CloudFormation.	N/A
23 de enero de 2023	Nuevo contenido	Se han añadido instrucciones para depurar simulaciones locales.	N/A
29 de noviembre de 2022	Lanzamiento del servicio	Se han publicado la Guía del usuario y la Referencia de la API para SimSpace Weaver.	AWS SDK: 29 de noviembre de 2022

Glosario

Este glosario define los términos que son específicos de AWS SimSpace Weaver

Para obtener la AWS terminología más reciente, consulte el [AWS glosario](#) en la Referencia general.AWS

A

Aplicación Código ejecutable (también denominado binario) que usted cree. El término aplicación puede hacer referencia al código o a una instancia en ejecución de ese código. Una aplicación encapsula el comportamiento de la simulación. Las aplicaciones crean, eliminan, leen y actualizan [entidades](#).

SDK de la aplicación Un kit de desarrollo de software (SDK) que se utiliza para integrar una aplicación con SimSpace Weaver. El SDK proporciona API para leer y escribir datos de [entidades](#) y realizar un seguimiento del tiempo de simulación. Para obtener más información, consulte [SDK de la aplicación SimSpace Weaver](#).

C

Cliente Procesos (o sus definiciones) que existen fuera de la simulación SimSpace Weaver e interactúan con ella a través de una [aplicación personalizada o una aplicación de servicio](#). Puede usar un cliente para ver o cambiar el estado de la simulación.

Reloj Una abstracción de sus procesos SimSpace Weaver de programación interna. El reloj publica las [graduaciones](#) en las [aplicaciones](#) para mantener la sincronización horaria. Cada simulación tiene su propio reloj.

Frecuencia del reloj El número de [graduaciones](#) por segundo que el [reloj](#) publica en las [aplicaciones](#). Para obtener más información acerca de las frecuencias del reloj, consulte [SimSpace Puntos finales y cuotas de Weaver](#).

Frecuencia de la graduación del reloj Consulte la [frecuencia del reloj](#).

Unidad de recursos de cómputo	Unidad de recursos de cómputo (procesador y memoria) de un trabajador . A una sola instancia de una aplicación se le asigna normalmente 1 unidad de recursos de cómputo. Puede asignar más de una unidad de recursos informáticos a cada aplicación.
Aplicación personalizada	Tipo de aplicación que se utiliza para leer el estado de la simulación e interactuar con él. Las aplicaciones personalizadas pueden crear entidades en la simulación, pero no son de su propiedad. Cuando una aplicación personalizada crea una entidad, debe transferirla al dominio espacial . Usted controla el ciclo de vida de una aplicación personalizada mediante las API de la aplicación. Para obtener más información sobre las SimSpace Weaver API, consulte SimSpace Weaver Referencias de API .
Dominio personalizado	Un dominio que contiene aplicaciones personalizadas .
Partición personalizada	La partición de una aplicación personalizada .

D

Fecha límite	El tiempo real en el que debe completarse una operación (como el procesamiento de una graduación).
Dominio	Grupo de instancias de aplicaciones que ejecutan el mismo código ejecutable (aplicación binaria) y tienen las mismas opciones de inicio.

E

Punto de conexión (servicio)	Un nombre de dominio completo (FQDN) que los programas (como el AWS Command Line Interface) utilizan para conectarse al SimSpace Weaver servicio.
Punto de conexión (simulación)	Una dirección IP y un número de puerto que los clientes utilizan para conectarse a una simulación. Puede configurar puntos de conexión en aplicaciones personalizadas y aplicaciones de servicio .
Entidad	Objetos de datos de clientes (o sus definiciones). Las entidades pueden ser estáticas (permanecen en una ubicación) o dinámicas (se mueven por el espacio de simulación). Por ejemplo, personas y edificios en una simulación.

I

Índice (simulación) Descripción de las propiedades espaciales de una simulación, incluidos sus límites espaciales y su sistema de coordenadas.

L

Ciclo de vida (de una aplicación) Una descripción de los pasos lógicos esperados que debe seguir una [aplicación](#) durante una simulación. Los ciclos de vida se administran (se inicia y se detiene la aplicación) o no se administran (se inicia y detiene la aplicación).

Cargar (datos del campo de la entidad) Lea los datos del campo de la [entidad](#) del [State Fabric](#).

P

Partición Un segmento de memoria compartida de un [trabajador](#). Cada partición contiene un subconjunto discreto de [entidades](#) dentro de un [dominio](#). Cada [aplicación](#) tiene una partición asignada. Una aplicación es propietaria de todas las entidades de su partición. Cuando una aplicación crea una entidad, la crea en su partición. Cuando las entidades se mueven de una partición a otra, la propiedad se transfiere de la aplicación de la partición de origen a la aplicación de la partición de destino.

R

Unidad de recursos Consulte [???](#).

S

Esquema Documento YAML o JSON que describe la configuración de una simulación. SimSpace Weaver usa un esquema para crear un [recurso de simulación](#).

Aplicación de servicio Tipo de [aplicación](#) que se utiliza para leer el estado de la simulación e interactuar con él. Las aplicaciones de servicio pueden crear entidades

en la simulación, pero deben transferirlas al [dominio](#) espacial. SimSpace Weaver administra el [ciclo de vida](#) de una aplicación de servicio e inicia 1 (o más, según se especifique en el [esquema](#) de simulación) en cada [trabajador](#) de la simulación.

Dominio de servicio	Un dominio que contiene aplicaciones personalizadas .
Partición de servicio	La partición de una aplicación de servicio .
Simulación (recurso)	Una abstracción de un clúster de cómputo que ejecuta un espacio virtual simulado. Puede tener varias simulaciones. La simulación se configura mediante un esquema .
Aplicación espacial	Tipo de aplicación que encapsula la lógica principal de la simulación. Cada aplicación espacial posee 1 (y solo 1) partición .
Dominio espacial	Un dominio que contiene aplicaciones espaciales .
Partición espacial	La partición de una aplicación espacial .
State Fabric	Base de datos en memoria de SimSpace WeaverState FabricAlmacena el estado de las simulaciones, incluidas las entidades y los datos internos. SimSpace Weaver
Almacenar (datos del campo de la entidad)	Escribir los datos del campo de la entidad en State Fabric .
Suscripción	Una solicitud de larga duración para que una instancia de aplicación específica reciba datos de un área de suscripción . La aplicación que se suscribe utiliza una suscripción para detectar los cambios en las entidades que se encuentran dentro del área de suscripción.
Área de suscripción	Región bidimensional del espacio de simulación. Una suscripción se refiere a un área de suscripción. Un área de suscripción puede abarcar más de 1 partición y también incluir partes de particiones. Un área de suscripción es continua dentro de sus límites definidos.

T

Graduación	Un valor de tiempo discreto (ya sea el tiempo de un reloj de pared o el tiempo de simulación). Las aplicaciones pueden iterar más rápido
------------	--

que la duración de las marcas, pero se espera que escriban marcas específicas dentro de plazos específicos. Todas las operaciones de todas las aplicaciones relacionadas con una marca determinada deben completarse antes de que se inicie la siguiente.

Frecuencia de graduación

Consulte la frecuencia del reloj.

Hora (real)

El tiempo actual desde la perspectiva de la realidad. SimSpace Weaver utiliza una marca de tiempo POSIX de 64 bits, que es el número de nanosegundos transcurridos desde la época Unix (January 1, 1970, 00:00:00 UTC).

Hora (simulación)

La hora actual desde la perspectiva de la simulación. SimSpace Weaver utiliza un contador lógico de dígitos enteros de 64 bits, que puede no corresponder directamente a la hora real.

W

Trabajador

Una instancia de Amazon Elastic Compute Cloud (Amazon EC2) que ejecuta código de simulación.

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la versión original de inglés, prevalecerá la versión en inglés.