



Guía del usuario para la versión 1.16.0

# AWS SimSpace Weaver



# AWS SimSpace Weaver: Guía del usuario para la versión 1.16.0

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 comerciales que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, relacionados 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
Registro para obtener una Cuenta de AWS .....	13
Crear un usuario administrativo .....	14
Agregar permisos para usar SimSpace Weaver .....	15
Configure su entorno local .....	16
AL2 en Docker .....	17
AL2 en WSL .....	18
Uso de software con licencia .....	22
Introducción .....	23
Tutorial de inicio rápido .....	23
Paso 1: Crear un proyecto .....	24
Paso 2: activar el registro .....	27
Paso 3: Ejecutar el script de inicio rápido .....	29
Paso 4: obtenga su dirección IP y número de puerto .....	32
Paso 5: Ver la simulación .....	39
Paso 12: Detenga y limpie la simulación .....	47
Tutorial introductorio .....	53
Paso 1: Crear un proyecto .....	54
Paso 2: activar el registro .....	57
Paso 3: cargar el esquema de simulación .....	59
Paso 4: Crear su proyecto .....	62

Paso 5: Subir aplicaciones .....	63
Paso 6: Iniciar la simulación .....	66
Paso 7: Obtenga los detalles de la simulación .....	70
Paso 8: Iniciar aplicaciones personalizadas .....	76
Paso 9: inicie el reloj .....	81
Paso 10: Verifique los registros .....	83
Paso 11: Ver la simulación .....	86
Paso 12: Detenga y limpie la simulación .....	94
Trabajar con SimSpace Weaver .....	101
Configuración de su simulación .....	101
Parámetros de configuración de la simulación .....	103
Versión de SDK .....	104
Propiedades de simulación .....	104
Trabajadores .....	105
Reloj .....	106
Estrategias de partición .....	109
Dominios .....	110
Duración máxima .....	121
Valor máximo .....	121
Valor predeterminado .....	121
Valor mínimo .....	121
Iniciar una simulación mediante scripts del SDK de la aplicación SimSpace Weaver .....	121
Iniciar una simulación mediante la consola .....	122
El estado de una simulación que alcanza su duración máxima .....	122
Desarrollo de aplicaciones .....	122
Aplicación espacial .....	123
Aplicaciones personalizadas .....	124
Desarrollo de aplicaciones de cliente .....	125
Desarrollo local .....	127
Compilación .....	127
Ejecute .....	128
Visualización .....	130
Detener .....	131
Debug .....	131
Diferencias en la 1.15.3 .....	132
SDK de la aplicación SimSpace Weaver .....	138

Los métodos de la API devuelven un Result .....	139
Interactuar con el SDK de aplicación de nivel superior .....	140
Gestión de la simulación .....	140
Suscripciones .....	143
Entidades .....	144
Eventos de entidad .....	156
Result y control de errores .....	163
Genéricos y tipos de dominio .....	165
Operaciones diversas del SDK de la aplicación .....	165
Marco de demostración de SimSpace Weaver .....	168
Trabajar con cuotas .....	169
Obtener los límites de una aplicación .....	169
Obtener la cantidad de recursos que utiliza una aplicación .....	170
Restablecimiento de métricas .....	171
Superar un límite .....	172
Agotamiento de la memoria .....	172
Prácticas recomendadas .....	172
Depuración de simulaciones .....	173
Utilice SimSpace Weaver Local y observe el resultado de la consola .....	173
Mira tus registros en Amazon CloudWatch Logs .....	173
Utilice llamadas a la API describe .....	174
Conectar un cliente .....	175
Depuración de simulaciones locales .....	175
Contenedores personalizados .....	176
Creación de un contenedor personalizado .....	177
Modificar un proyecto para usar un contenedor personalizado .....	178
Preguntas frecuentes .....	181
Solución de problemas .....	181
Trabajo con Python .....	182
Crear un proyecto de Python .....	183
Inicio de una simulación de Python .....	186
Cliente Python de muestra .....	186
Escribir sus propios scripts de compilación .....	187
Preguntas frecuentes .....	188
Solución de problemas .....	188
Compatibilidad con otros motores .....	189

Unity .....	190
Unreal Engine .....	191
Uso de software con licencia .....	192
Administrar los recursos con AWS CloudFormation .....	192
Instantáneas .....	194
Instantáneas .....	195
Casos de uso de instantáneas .....	196
SDK de la aplicación SimSpace Weaver .....	197
Consola de SimSpace Weaver .....	202
AWS CLI .....	204
Las API de SimSpace Weaver .....	206
Preguntas frecuentes .....	207
Mensajería .....	208
Casos de uso de mensajería .....	208
Uso de las API de mensajería .....	209
¿Cuándo usar la mensajería .....	216
Consejos para trabajar con la mensajería .....	220
Errores de mensajería y solución de problemas .....	221
Prácticas recomendadas .....	224
Configurar alarmas de facturación .....	224
Utilizar SimSpace Weaver Local .....	224
Puede detener las simulaciones que no necesite .....	225
Elimine los recursos que no necesite .....	225
Disponer de copias de seguridad .....	225
Seguridad .....	227
Protección de los datos .....	228
Cifrado en reposo .....	229
Cifrado en tránsito .....	229
Privacidad del tráfico entre redes .....	230
Identity and Access Management .....	230
Público .....	230
Autenticación con identidades .....	231
Administración de acceso mediante políticas .....	235
Cómo funciona AWS SimSpace Weaver con IAM .....	238
Ejemplos de políticas basadas en identidades .....	245
Permisos que SimSpace Weaver crea para usted .....	250

Prevención de la sustitución confusa entre servicios .....	252
Solución de problemas .....	254
Registro y supervisión de eventos de seguridad .....	258
Validación de la conformidad .....	258
Resiliencia .....	260
Seguridad de infraestructuras .....	260
Modelo de seguridad de la conectividad de red .....	261
Configuración y análisis de vulnerabilidades .....	261
Prácticas recomendadas de seguridad .....	262
Encripte las comunicaciones entre sus aplicaciones y sus clientes .....	262
Realice copias de seguridad periódicas del estado de la simulación .....	263
Mantenga sus aplicaciones y SDK .....	263
Registro y monitorización .....	264
Inicia sesión CloudWatch .....	264
Acceder a sus registros de SimSpace Weaver .....	264
Registros de SimSpace Weaver .....	265
Monitorización con CloudWatch .....	268
Métricas de SimSpace Weaver a nivel de cuenta .....	269
CloudTrail registros .....	269
SimSpace Weaver información en CloudTrail .....	269
Descripción de las entradas de los archivos de registro de SimSpace Weaver .....	270
Puntos de conexión y Service Quotas .....	273
Puntos de conexión de servicio .....	273
Service Quotas .....	274
Cuotas de mensajería .....	277
Velocidades del reloj .....	277
Service Quotas de SimSpace Weaver Local .....	277
Solución de problemas .....	279
AssumeRoleAccessDenied .....	279
InvalidBucketName .....	281
ServiceQuotaExceededException .....	282
TooManyBuckets .....	282
Permiso denegado durante el inicio de la simulación .....	283
Problemas relacionados con el tiempo de uso de Docker .....	284
El cliente de consola no se puede conectar .....	284
N hay <code>simspaceweaver</code> en el AWS CLI .....	286

Referencia del esquema .....	288
Ejemplos de un esquema completo .....	288
Formato de esquema .....	290
Versión de SDK .....	291
Propiedades de simulación .....	291
Trabajadores .....	293
Reloj .....	294
Estrategias de partición .....	294
Dominios .....	296
Restricciones para la ubicación .....	307
Referencias de API .....	308
Versiones de SimSpace Weaver .....	309
Última versión .....	309
Cómo encontrar la versión actual .....	309
Descargue la versión más reciente .....	309
Solución de problemas de descargas del SDK de aplicaciones .....	310
Instale la versión más reciente .....	311
Versiones del servicio .....	311
1.15.1 .....	328
Actualizar un proyecto de Python existente a 1.15.1 .....	328
Solución de problemas con la versión 1.15.1 .....	329
Preguntas frecuentes sobre la versión 1.15.1 .....	329
Historial de documentos .....	331
Glosario .....	340
.....	cccxliv

# ¿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.

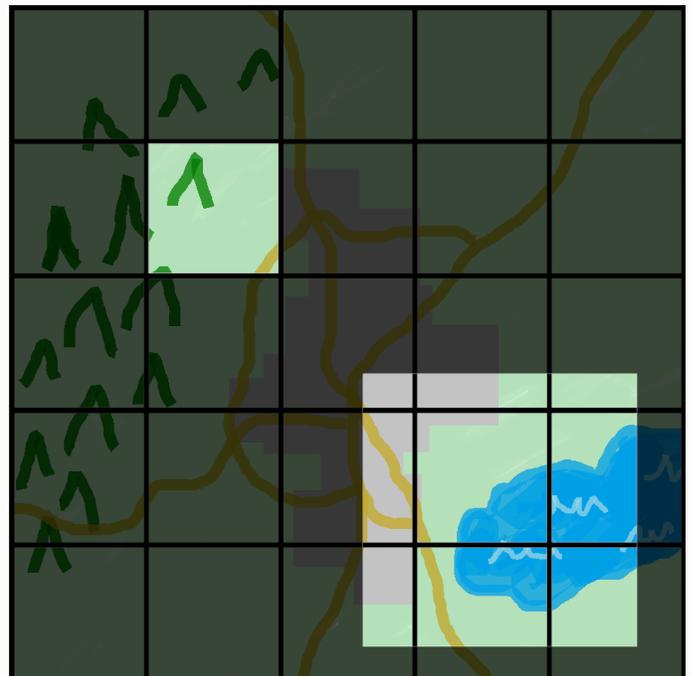
Las entidades pueden moverse por el mundo de la simulación. Si una entidad entra en el área de propiedad de otra aplicación espacial (otra celda de la cuadrícula), el propietario de la aplicación espacial de la nueva área asume el control de la entidad. Si la simulación se ejecuta en varios trabajadores, 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 [Referencia del esquema de simulación SimSpace Weaver](#).

## 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. [Configure su dispositivo Cuenta de AWS para usar SimSpace Weaver](#).
2. [Configuración del entorno de SimSpace Weaver](#).

## Configure su dispositivo Cuenta de AWS para usar SimSpace Weaver

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

### Registro para obtener una Cuenta de AWS

Si no dispone de una Cuenta de AWS, siga estos pasos para crear una.

Cómo registrarse en una Cuenta de AWS

1. Abra <https://portal.aws.amazon.com/billing/signup>.
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.

Al registrarse en una Cuenta de AWS, se crea un Usuario raíz de la cuenta de AWS. 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 administrativo](#) y utilice únicamente el usuario raíz para realizar [tareas que requieran acceso de usuario raíz](#).

AWS le enviará un correo electrónico de confirmación luego de completar el proceso de registro.

Puede ver la actividad de la cuenta y administrar la cuenta en cualquier momento entrando en <https://aws.amazon.com/> y seleccionando Mi cuenta.

## Crear un usuario administrativo

Después de registrarse para obtener una Cuenta de AWS, proteja su Usuario raíz de la cuenta de AWS, habilite AWS IAM Identity Center y cree un usuario administrativo para no utilizar el usuario raíz en las tareas cotidianas.

### Protección de su Usuario raíz de la cuenta de AWS

1. Inicie sesión en la [AWS Management Console](#) como propietario de cuenta, elija Usuario raíz e ingrese el email de su Cuenta de AWS. 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. Active la autenticación multifactor (MFA) para el usuario raíz.

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

### Creación de un usuario administrativo

1. Activar IAM Identity Center

Para ver las instrucciones, consulte [Enabling 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 administrativo.

Para ver un tutorial sobre el uso del Directorio de IAM Identity Center como origen de identidades, consulte [Configure user access with the default Directorio de IAM Identity Center](#) en la Guía del usuario de AWS IAM Identity Center.

### Cómo iniciar sesión como usuario administrativo

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

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

## Agregar permisos para usar SimSpace Weaver

Para dar acceso, añada 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 administrados en IAM a través de un proveedor de identidades:

Cree un rol para la federación de identidades. Siga las instrucciones de [Creación de un rol para un proveedor de identidades de terceros \(federación\)](#) en la Guía del usuario de IAM.

- Usuarios de IAM:

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

### Example Política de IAM para conceder permisos para usar 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

- [Configurar Amazon Linux 2 \(AL2\) en Docker](#)
- [Configurar Amazon Linux 2 \(AL2\) en Windows Subsystem for Linux \(WSL\)](#)

## Configurar Amazon Linux 2 (AL2) en Docker

En esta sección se proporcionan instrucciones para configurar su entorno AL2 local en Docker. Para obtener instrucciones para configurar AL2 en Windows Subsystem for Linux (WSL), consulte [Configurar Amazon Linux 2 \(AL2\) en Windows Subsystem for Linux \(WSL\)](#).

### Requisitos

- Microsoft Windows 10 o superior
- [Microsoft Visual Studio 2019](#) o superior, con la carga de trabajo de [Desktop development with C++](#) instalada
- [CMake3](#)
- [Git](#)
- [Docker Desktop](#)
- [AWS CLI](#)

### Para configurar AL2 en Docker

1. Si aún no ha configurado sus AWS credenciales para AWS CLI, siga estas instrucciones: [Configuración de la AWS CLI](#). Si solo está utilizando SimSpace Weaver, puede configurar AWS CLI para que use sus credenciales de SimSpace Weaver de forma predeterminada.
2. [Descargue el paquete distribuible del SDK de la aplicación de SimSpace Weaver](#). Contiene lo siguiente:
  - Archivos binarios y bibliotecas para el desarrollo de la aplicación SimSpace Weaver

- Scripts auxiliares que automatizan partes del flujo de trabajo de desarrollo
  - Ejemplos de aplicaciones que muestran conceptos de SimSpace Weaver
3. Descomprima el archivo en una *sdk-folder* de su elección.
  4. Vaya al *sdk-folder*.
  5. Introduzca el siguiente comando para crear una imagen de Docker.

```
docker-create-image.bat
```

 Note

Si aparece un error durante este paso, asegúrese de que Docker está funcionando.

## Configurar Amazon Linux 2 (AL2) en Windows Subsystem for Linux (WSL)

En esta sección se proporcionan instrucciones para configurar su entorno AL2 local en Windows Subsystem for Linux (WSL). Para obtener instrucciones para configurar AL2 en Docker, consulte [Configurar 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 AWS no asumen ninguna responsabilidad si decide 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 AL2 en WSL

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

```
wsl -d Amazon2
```

**⚠ Important**

Mientras esté en ejecución en WSL, utilice las instrucciones de Linux en lugar de las instrucciones de Windows de esta guía (si puede elegir). Si no hay instrucciones para Linux, utilice las siguientes sustituciones:

- Use `tools/linux` en lugar de `tools\windows`
- Utilice scripts de `.sh` en lugar de scripts de `.bat`

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
```

- Elimine los AWS CLI que yum ha instalado. Pruebe los dos comandos siguientes si no tiene claro si yum ha instalado un AWS CLI.

```
yum remove awscli
```

```
yum remove aws-cli
```

- Cree un directorio temporal y acceda a él.

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

- Descargue e instale el AWS CLI:

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

- Podrá eliminar el directorio temporal.

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

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

```
exec
```

- Configure sus credenciales de AWS para el AWS CLI en su entorno AL2. Para obtener más información, consulte [Configuración de AWS CLI](#). Si utiliza AWS IAM Identity Center, consulte [Configurar el AWS CLI para utilizar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface.

```
aws configure
```

- Instale Git.

```
yum install -y git
```

## 11. Instale wget.

```
yum install -y wget
```

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

```
mkdir sdk-folder
```

## 13. Acceda a su carpeta de SDK.

```
cd sdk-folder
```

## 14. Descargue el archivo distribuible del SDK de la aplicación SimSpace Weaver. Contiene lo siguiente:

- Archivos binarios y bibliotecas para el desarrollo de la aplicación SimSpace Weaver
- Scripts auxiliares que automatizan partes del flujo de trabajo de desarrollo
- Ejemplos de aplicaciones que muestran conceptos de 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 adicional.

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

### 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.

# Introducción a SimSpace Weaver

Esta sección brinda tutoriales de para ayudarle a comenzar a usar SimSpace Weaver. Estos tutoriales le presentan el flujo de trabajo general para crear simulaciones con SimSpace Weaver. En ambos tutoriales, aprenderá a 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. A continuación, consulte el tutorial detallado para obtener más información sobre cada paso del proceso.

Estos tutoriales utilizan una aplicación de muestra (PathfindingSample) incluida en el archivo.zip del SDK de la aplicación de SimSpace Weaver que descargó durante los [procedimientos de configuración](#). La aplicación de ejemplo muestra los conceptos que comparten todas las simulaciones de SimSpace Weaver, 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 utilizar una aplicación cliente independiente (incluida en el SDK de la aplicación de SimSpace Weaver) para ver la simulación.

## Temas

- [Tutorial de inicio rápido: cree y ejecute una simulación en minutos](#)
- [Tutorial detallado: conozca los detalles mientras crea la aplicación de muestra](#)

## Tutorial de inicio rápido: cree y ejecute una simulación en minutos

Este tutorial le guía a través del proceso para crear y ejecutar una simulación en cuestión de 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](#).

## Pasos

- [Paso 1: Crear un proyecto](#)
- [Paso 2: activar el registro \(opcional\)](#)
- [Paso 3: Ejecutar el script de inicio rápido](#)
- [Paso 4: obtenga su dirección IP y número de puerto](#)
- [Paso 5: Ver la simulación](#)
- [Paso 12: Detenga y limpie la simulación](#)

## Paso 1: Crear un proyecto

El SDK de la aplicación SimSpace Weaver que se puede distribuir contiene un script que crea un proyecto a partir del proyecto agrupado de PathfindingSample. Debe ejecutar el script desde su ubicación en el sistema de archivos. El script crea un *project-name* en una *ruta* interna utilizando los valores que usted proporciona en la línea de comandos.

### Docker

Para crear un proyecto

1. En una línea de comandos de Windows, cambie a la carpeta del proyecto.

```
cd sdk-folder
```

2. Ejecute el script `create-project.bat`.

```
.\create-project.bat --name project-name --path path
```

A continuación se muestra la lista completa de comandos y parámetros.

```
.\create-project.bat --name project-name --path path --app-sdk-version version-number --template template-name --overwriteproject
```

#### Important

No utilice un nombre de proyecto que supere los 20 caracteres. Si se supera este límite, se pueden producir errores.

**Note**

Si recibe un mensaje para compartir una unidad con Docker, seleccione Yes para compartirla.

**Important**

Si usa AWS IAM Identity Center o perfiles nombrados para AWS Command Line Interface (AWS CLI), debe usar la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. La última versión es la 1.16.0. Para obtener información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#). Los scripts del SDK de la aplicación de SimSpace Weaver utilizan AWS CLI. Si utiliza el centro de identidad de IAM, puede copiar su perfil del centro de identidad de IAM para el AWS CLI a su perfil de default o proporcionar el nombre de su perfil del centro de identidad de IAM a los scripts del SDK de la aplicación de SimSpace Weaver con el parámetro `--profile cli-profile-name`. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

## 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](#).

## Para crear un proyecto

1. En un aviso del intérprete de comandos de Linux, cambie a la carpeta del proyecto.

```
cd sdk-folder
```

## 2. Ejecute el script `create-project.sh`.

```
./create-project.sh --name project-name --path path
```

A continuación se muestra la lista completa de comandos y parámetros.

```
./create-project.sh --name project-name --path path --profile cli-profile-name  
--app-sdk-version version-number --template template-name --overwriteproject
```

### Important

No utilice un nombre de proyecto que supere los 20 caracteres. Si se supera este límite, se pueden producir errores.

### Important

Si usa AWS IAM Identity Center o perfiles nombrados para AWS Command Line Interface (AWS CLI), debe usar la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. La última versión es la 1.16.0. Para obtener información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#). Los scripts del SDK de la aplicación de SimSpace Weaver utilizan AWS CLI. Si utiliza el centro de identidad de IAM, puede copiar su perfil del centro de identidad de IAM para el AWS CLI a su perfil de default o proporcionar el nombre de su perfil del centro de identidad de IAM a los scripts del SDK de la aplicación de SimSpace Weaver con el parámetro `--profile cli-profile-name`. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

## Parámetros

### name

El nombre del proyecto.

## path

La ubicación del proyecto en el sistema de archivos. En ocasiones, la estructura del proyecto puede tener rutas de archivo largas que pueden superar el límite de longitud de las rutas del sistema operativo. Le recomendamos que utilice un nombre de ruta lo más corto posible.

## profile

El nombre del perfil de AWS CLI que el script debe usar para la autenticación. Para obtener más información, consulte [Configuración y ajustes de archivos de credenciales](#) en la Guía de usuario de AWS Command Line Interface. Este parámetro solo está disponible para la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. Para obtener más información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#).

## app-sdk-version

(opcional) La versión del SDK de la aplicación de SimSpace Weaver que usa su proyecto. Creará y vinculará sus aplicaciones con esta versión. Si el script no encuentra la versión en la ubicación del objeto de distribución, o si usted no proporciona el número de versión, el script descargará automáticamente la última versión.

## template

(opcional) Una plantilla de proyecto que el script utilizará para crear su proyecto. Si no proporciona una plantilla, el script utilizará `PathfindingSample`. Valores válidos:

- **PathfindingSample**: una aplicación de ejemplo que utiliza un único [trabajador](#).
- **MultiWorkerPathfindingSample**: una versión de la aplicación de ejemplo que utiliza varios [trabajadores](#).

## overwriteproject

(opcional) Utilice esta opción para sobrescribir una carpeta de proyecto existente con el mismo nombre y ruta del proyecto.

## Paso 2: activar el registro (opcional)

El registro está desactivado de forma predeterminada en el proyecto `PathfindingSample`. En este tutorial se presupone que el inicio de sesión está activado. Puede elegir activar el registro, pero es opcional.

**⚠ Important**

PathfindingSample genera grandes cantidades de datos de registro. Si decide activar el registro, recibirá los cargos de facturación por los datos de registro. Seguirá recibiendo los cargos de facturación por los datos de registro mientras existan. Le recomendamos encarecidamente que detenga esta simulación y realice los pasos de limpieza que aparecen al final del tutorial lo antes posible si activa el registro.

## Para activar el registro

1. Abra el siguiente archivo en un editor de texto.

## Docker

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

**i Note**

Su *carpeta de proyecto* está usando los valores de *path\project-name* que proporcionó al crear el proyecto.

## 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](#).

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

**i Note**

Su *carpeta de proyecto* está usando los valores de *path/project-name* que proporcionó al crear el proyecto.

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

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

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

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

4. 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>"
```

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

## Paso 3: Ejecutar el script de inicio rápido

La aplicación de ejemplo incluye un script de inicio rápido. El script creará, compilará, cargará e iniciará la simulación y sus aplicaciones.

### Docker

Para ejecutar el script de inicio rápido

1. Si aún no lo ha hecho, vaya a la carpeta de herramientas de su proyecto y plataforma. Su *project-folder* está usando los valores de *path\project-name* que proporcionó al crear el proyecto.

En la línea de comandos de Windows, escriba:

```
cd project-folder\tools\windows
```

2. Ejecute el script de inicio rápido de este proyecto.

```
.\quick-start-project-name-cli.bat
```

**⚠ Important**

A partir de la versión 1.12.3, el script `quick-start` inicia la simulación con una duración máxima de 1 hora. Puede utilizar el parámetro `--maximum-duration` para especificar una duración máxima diferente. En la versión 1.12.2 o inferior, no se puede proporcionar una duración máxima al guion y las simulaciones tienen una duración máxima de 14 días. Para obtener más información acerca de la duración máxima de una simulación, consulte [Duración máxima de una simulación](#).

**⚠ Important**

Si usa AWS IAM Identity Center o perfiles nombrados para AWS Command Line Interface (AWS CLI), debe usar la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. La última versión es la 1.16.0. Para obtener información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#). Los scripts del SDK de la aplicación de SimSpace Weaver utilizan AWS CLI. Si utiliza el centro de identidad de IAM, puede copiar su perfil del centro de identidad de IAM para el AWS CLI a su perfil de `default` o proporcionar el nombre de su perfil del centro de identidad de IAM a los scripts del SDK de la aplicación de SimSpace Weaver con el parámetro `--profile cli-profile-name`. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

**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](#).

Para ejecutar el script de inicio rápido

1. Si aún no lo ha hecho, vaya a la carpeta de herramientas de su proyecto y plataforma. Su *project-folder* utiliza los valores de *path/project-name* que proporcionó al crear el proyecto.

En un aviso de intérprete de comandos de Linux, introduzca:

```
cd project-folder/tools/linux
```

2. Ejecute el script de inicio rápido de este proyecto.

```
./quick-start-project-name-cli.sh
```

#### Important

A partir de la versión 1.12.3, el script `quick-start` inicia la simulación con una duración máxima de 1 hora. Puede utilizar el parámetro `--maximum-duration` para especificar una duración máxima diferente. En la versión 1.12.2 o inferior, no se puede proporcionar una duración máxima al guion y las simulaciones tienen una duración máxima de 14 días. Para obtener más información acerca de la duración máxima de una simulación, consulte [Duración máxima de una simulación](#).

#### Important

Si usa AWS IAM Identity Center o perfiles nombrados para AWS Command Line Interface (AWS CLI), debe usar la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. La última versión es la 1.16.0. Para obtener información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#). Los scripts del SDK de la aplicación de SimSpace Weaver utilizan AWS CLI. Si utiliza el centro de identidad de IAM, puede copiar su perfil del centro de identidad de IAM para el AWS CLI a su perfil de default o proporcionar el nombre de su perfil del centro de identidad de IAM a los scripts del SDK de la aplicación de SimSpace Weaver con el parámetro `--profile cli-profile-name`. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#)

en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

El script iniciará un bucle y se detendrá automáticamente una vez que todos los componentes estén STARTED. El resultado tiene un aspecto similar al siguiente:

```
[2022-10-04T22:15:28] [INFO] Describe Simulation Results:
[2022-10-04T22:15:28] [INFO] {
  "Status": "STARTED",
  "Name": "MyProjectSimulation_22-10-04_22_10_15",
  "RoleArn": "arn:aws:iam::111122223333:role/weaver-MyProject-app-role",
  "CreationTime": 1664921418.09,
```

## Paso 4: obtenga su dirección IP y número de puerto

Debe obtener la dirección IP y el número de puerto de su aplicación de visualización (personalizada) para poder conectarse a la simulación. En el siguiente procedimiento se supone que no sabe nada sobre la simulación (por ejemplo, el nombre de la simulación). Puede utilizar este procedimiento en cualquier momento para buscar la dirección IP y el número de puerto de una aplicación personalizada o una aplicación de servicio. A continuación, se muestra un ejemplo del resultado de un proyecto llamado MyProject.

### Docker

Para obtener su dirección IP y número de puerto

1. Si aún no lo ha hecho, vaya a la carpeta de herramientas de su proyecto y plataforma. Su *project-folder* está usando los valores de *path\project-name* que proporcionó al crear el proyecto.

En la línea de comandos de Windows, escriba:

```
cd project-folder\tools\windows
```

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

```
.\weaver-project-name-cli.bat list-simulations
```

### Important

Si usa AWS IAM Identity Center o perfiles nombrados para AWS Command Line Interface (AWS CLI), debe usar la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. La última versión es la 1.16.0. Para obtener información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#). Los scripts del SDK de la aplicación de SimSpace Weaver utilizan AWS CLI. Si utiliza el centro de identidad de IAM, puede copiar su perfil del centro de identidad de IAM para el AWS CLI a su perfil de default o proporcionar el nombre de su perfil del centro de identidad de IAM a los scripts del SDK de la aplicación de SimSpace Weaver con el parámetro `--profile cli-profile-name`. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

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"
    }
  ]
}
```

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

```
.\weaver-project-name-cli.bat 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"
    }
  ],

```

4. Use la ListAppsAPI para obtener una lista de aplicaciones personalizadas en un dominio. 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.

```
.\weaver-project-name-cli.bat 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"
    }
]
}

```

5. Usa 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`.

```

.\weaver-project-name-cli.bat 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 SimSpace Weaver expone los clientes para que se conecten a su aplicación. SimSpace Weaver asigna el puerto `Declared` al puerto `Actual`.

## 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](#).

Para obtener su dirección IP y número de puerto

1. Si aún no lo ha hecho, vaya a la carpeta de herramientas de su proyecto y plataforma. Su *project-folder* utiliza los valores de *path/project-name* que proporcionó al crear el proyecto.

En un aviso de intérprete de comandos de Linux, introduzca:

```
cd project-folder/tools/linux
```

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

```
./weaver-project-name-cli.sh list-simulations
```

**Important**

Si usa AWS IAM Identity Center o perfiles nombrados para AWS Command Line Interface (AWS CLI), debe usar la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. La última versión es la 1.16.0. Para obtener información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#). Los scripts del SDK de la aplicación de SimSpace Weaver utilizan AWS CLI.

Si utiliza el centro de identidad de IAM, puede copiar su perfil del centro de identidad de IAM para el AWS CLI a su perfil de default o proporcionar el nombre de su perfil del centro de identidad de IAM a los scripts del SDK de la aplicación de SimSpace Weaver con el parámetro `--profile` *cli-profile-name*. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

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"
    }
  ]
}
```

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

```
./weaver-project-name-cli.sh 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. 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.

```

./weaver-project-name-cli.sh 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`.

```

./weaver-project-name-cli.sh 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 SimSpace Weaver expone los clientes para que se conecten a su aplicación. SimSpace Weaver asigna el puerto `Declared` al puerto `Actual`.

## Paso 5: Ver la simulación

El SDK de la aplicación de SimSpace Weaver ofrece diferentes opciones para ver la aplicación de muestra. Puede usar el cliente de consola de ejemplo si no tiene soporte local para el desarrollo

de Unreal Engine. En las instrucciones para el cliente de Unreal Engine se presupone que está 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.

## Windows console client

### Requisitos

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

Para conectarse a la aplicación de muestra con el cliente de consola de muestra

1. En una ventana de línea de comandos, vaya a la carpeta del cliente de consola (en la carpeta SDK de la aplicación).

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

2. Se utiliza CMake3 para crear una solución de Visual Studio en esta carpeta.

```
cmake .
```

### Note

Asegúrese de incluir el espacio y el punto al final.

**⚠ Important**

Mantenga abierta la ventana de la línea de comandos para seguir pasos.

3. En Visual Studio que creó en el paso anterior, abra `PathfindingSampleConsoleClient.sln`.
4. Seleccione la configuración de la compilación `RelWithDebInfo`.
5. Seleccione `Build > Build Solution`.
6. En su ventana de línea de comandos anterior, vaya a la carpeta de salida de la compilación en la carpeta del cliente de consola.

```
cd RelWithDebInfo
```

7. Ejecute el cliente con la dirección IP y el número de puerto de su `ViewApp`.

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

La ventana de la línea de comandos debería mostrar los números de los eventos de actualización, eliminación y creación de entidades, de forma similar al siguiente ejemplo de salida.

**📘 Note**

Las direcciones IP y los números de puerto del siguiente ejemplo de salida son marcadores de posición. Proporcione su dirección IP y su número de puerto de su `ViewApp` al cliente de la consola. Proporcione el número de puerto de `Actual` si desea conectarse a un `ViewApp` que se ejecute en el Nube de AWS. Proporcione la dirección IP y el número de puerto `127.0.0.1:7000` cuando se conecte a un sistema `ViewApp` que se ejecute en su sistema local. Para obtener más información, consulte [Desarrollo local](#).

```
##PathfindingSample#ViewApp Message Reader##
```

```
Added argument url:tcp://198.51.100.135:4321
Some subscription arguments are missing, restoring defaults.

*****
Sample usage without a MoveStrategy:
ConsoleClient --url tcp://198.51.100.135:4321 --subs-center-x 600 --subs-center-
y 500 --subs-radius 50
Sample usage with CircleMoveStrategy:
ConsoleClient --url tcp://198.51.100.135:4321 --subs-center-x 600 --subs-center-
y 500 --subs-radius 50 --subs-move-strategy circle --circle-center-x 500 --
circle-center-y 500 --circle-speed 0.001

*****
Starting NNG client. NNG version: 1.2.4
Creating socket ...done.
Connecting to View App ... done.
Initiating connection to tcp:// 198.51.100.135:4321 ... done.

Receiving messages ...
[2022-10-04 19:13:00.710] CreateEntity Count: 72
[2022-10-04 19:13:00.756] UpdateEntity Count: 42
[2022-10-04 19:13:00.794] DeleteEntity Count: 72
[2022-10-04 19:13:03.690] CreateEntity Count: 11
[2022-10-04 19:13:03.725] UpdateEntity Count: 2
[2022-10-04 19:13:03.757] UpdateEntity Count: 2
[2022-10-04 19:13:03.790] UpdateEntity Count: 2
```

 Note

Para obtener ayuda sobre la resolución de problemas, consulte [El cliente de consola PathFindingSample no se puede conectar.](#)

8. Pulse CTRL+C para salir del cliente de la consola.

## Linux console client

### Important

Proporcionamos estas instrucciones para su comodidad. Es posible que no funcionen en algunos entornos Linux. Estos procedimientos no son compatibles.

En este procedimiento, se presupone que está trabajando completamente en un entorno Linux. También puede visualizar la simulación con los clientes integrados en Windows.

### Requisitos

- CMake3
- Compilador C (ya incluido en Amazon Linux 2)
- Git

Para conectarse a la aplicación de muestra con el cliente de consola de muestra

1. En un Linux símbolo del intérprete de comandos, vaya a la carpeta del cliente de consola (en la carpeta SDK de la aplicación).

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

2. Cree una carpeta de compilación.

```
mkdir build
```

3. Vaya a la carpeta de compilación.

```
cd build
```

4. Use CMake3 para crear el cliente.

```
cmake3 ../ && cmake3 --build .
```

**Note**

Asegúrese de incluir el espacio y el punto al final.

5. Ejecute el cliente con la dirección IP y el número de puerto de su ViewApp.

```
./ConsoleClient --url tcp://ip-address:port-number
```

La ventana de la línea de comandos debería mostrar los números de los eventos de actualización, eliminación y creación de entidades, de forma similar al siguiente ejemplo de salida.

**Note**

Las direcciones IP y los números de puerto del siguiente ejemplo de salida son marcadores de posición. Proporcione su dirección IP y su número de puerto de su ViewApp al cliente de la consola. Proporcione el número de puerto de Actual si desea conectarse a un ViewApp que se ejecute en el Nube de AWS. Proporcione la dirección IP y el número de puerto 127.0.0.1:7000 cuando se conecte a un sistema ViewApp que se ejecute en su sistema local. Para obtener más información, consulte [Desarrollo local](#).

```
##PathfindingSample#ViewApp Message Reader##
```

```
Added argument url:tcp://198.51.100.135:4321
```

```
Some subscription arguments are missing, restoring defaults.
```

```
*****
```

```
Sample usage without a MoveStrategy:
```

```
ConsoleClient --url tcp://198.51.100.135:4321 --subs-center-x 600 --subs-center-y 500 --subs-radius 50
```

```
Sample usage with CircleMoveStrategy:
```

```
ConsoleClient --url tcp://198.51.100.135:4321 --subs-center-x 600 --subs-center-y 500 --subs-radius 50 --subs-move-strategy circle --circle-center-x 500 --circle-center-y 500 --circle-speed 0.001
```

```
*****
Starting NNG client. NNG version: 1.2.4
Creating socket ...done.
Connecting to View App ... done.
Initiating connection to tcp:// 198.51.100.135:4321 ... done.

Receiving messages ...
[2022-10-04 19:13:00.710] CreateEntity Count: 72
[2022-10-04 19:13:00.756] UpdateEntity Count: 42
[2022-10-04 19:13:00.794] DeleteEntity Count: 72
[2022-10-04 19:13:03.690] CreateEntity Count: 11
[2022-10-04 19:13:03.725] UpdateEntity Count: 2
[2022-10-04 19:13:03.757] UpdateEntity Count: 2
[2022-10-04 19:13:03.790] UpdateEntity Count: 2
```

#### Note

Para obtener ayuda sobre la resolución de problemas, consulte [El cliente de consola PathFindingSample no se puede conectar](#).

6. Pulse CTRL+C para salir del cliente de la consola.

## Unreal Engine on Windows

### Requisitos

- Entorno de desarrollo de Unreal Engine 5
- Microsoft .NET Framework 4.8 Developer Pack
- Cliente de consola Windows (consulte la pestaña del cliente de consola Windows en esta página)

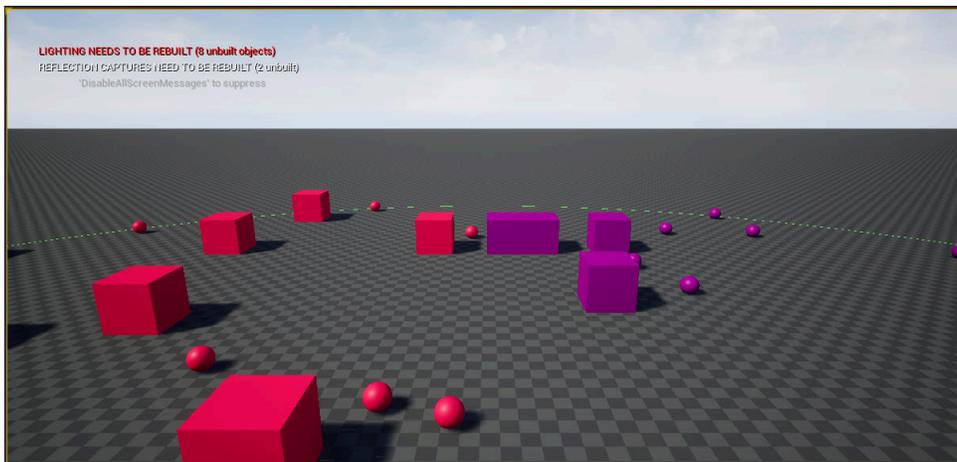
#### Important

Otras versiones de Unreal Engine y no .NET son compatibles y pueden causar problemas.

## Para conectarse a la aplicación de muestra con el cliente de muestra Unreal

1. El cliente Unreal Engine usa la biblioteca del cliente de consola NNG. Debe crear el cliente de consola para Windows el que aún no lo ha creado. Para obtener más información, consulte la pestaña del cliente de la consola Windows de esta página.
2. En una ventana del administrador de archivos, vaya a `sdk-folder\packaging-tools\clients\PathfindingSampleClients\UnrealClient`.
3. Abra `UnrealClient.uproject`.
4. Si el editor le pregunta si desea reconstruir los módulos de `UnrealClient`, seleccione `yes`.
5. En un editor de texto, abra `sdk-folder\packaging-tools\clients\PathfindingSampleClients\UnrealClient\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:port-number` (debería tener el aspecto de `tcp://198.51.100.135:1234`).
7. En el editor de Unreal, seleccione `play`.

El editor de Unreal debería mostrar una visualización de la simulación, similar a la siguiente captura de pantalla.



### Note

Dependiendo de la potencia del sistema de desarrollo local, el editor de Unreal puede tardar unos minutos en mostrar la simulación. Durante este tiempo, el sistema puede parecer que se está congelando.

Utilice las teclas W, A, S, D para mover el cliente Unreal. Mantenga pulsado el botón del ratón y arrastre el ratón para girar.

Puede pulsar la tecla [ (corchete izquierdo) para reducir el tamaño del área de suscripción. Puede pulsar la tecla ] (corchete derecho) para aumentar el tamaño del área de suscripción. El tamaño del área de suscripción determina la cantidad de entidades que aparecen en el cliente.

Puede pulsar la tecla C para crear una entidad en la simulación. El cliente envía un comando `CreateEntity` a la aplicación de visualización. A continuación, la aplicación de visualización creará la entidad y la transferirá al dominio espacial.

Puede examinar el código `ViewAppDriver::HandleEntityCreationRequests` en `project-folder\src\PathfindingSample\ViewApp\Driver\ViewAppDriver.cpp` para ver cómo la aplicación implementa este proceso.

## Paso 12: Detenga y limpie la simulación

Es importante limpiar las simulaciones cuando ya no las necesita. Los recursos de simulación de SimSpace Weaver se tienen en cuenta para sus Service Quotas (límites), incluso si la simulación está detenida. Seguirá cobrando los cargos de facturación por las simulaciones que se estén ejecutando. También puede recibir cargos de facturación por el almacenamiento de datos en servicios de apoyo, como Amazon CloudWatch Logs y Amazon Simple Storage Service. Para obtener más información sobre las cuotas de SimSpace Weaver servicio, consulte [SimSpace Puntos finales y cuotas de Weaver](#).

Siga los procedimientos de esta sección cuando desee limpiar la simulación.

### Important

No puede reiniciar una simulación después de detenerla.

### Important

No es posible recuperar una simulación eliminada.

## Limpie los recursos de simulación en SimSpace Weaver

Debe detener la simulación antes de poder eliminarla. Al eliminar una simulación, solo se eliminan los recursos en SimSpace Weaver. Debe realizar pasos distintos para eliminar los recursos que la simulación creó o utiliza en otros servicios (consulte la sección siguiente).

### Docker

#### Para limpiar la simulación

1. Si aún no lo ha hecho, vaya a la carpeta de herramientas de su proyecto y plataforma. Su *project-folder* está usando los valores de *path\project-name* que proporcionó al crear el proyecto.

En la línea de comandos de Windows, escriba:

```
cd project-folder\tools\windows
```

2. Busque los nombres de las simulaciones.

```
.\weaver-project-name-cli.bat list-simulations
```

#### Important

Si usa AWS IAM Identity Center o nombra perfiles para AWS Command Line Interface (AWS CLI), debe usar la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. La última versión es la 1.16.0. Para obtener información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#). Los scripts del SDK de la aplicación de SimSpace Weaver utilizan AWS CLI. Si utiliza el centro de identidad de IAM, puede copiar su perfil del centro de identidad de IAM para el AWS CLI a su perfil de default o proporcionar el nombre de su perfil del centro de identidad de IAM a los scripts del SDK de la aplicación de SimSpace Weaver con el parámetro `--profile cli-profile-name`. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

3. Detenga una simulación.

```
.\weaver-project-name-cli.bat stop-simulation --simulation simulation-name
```

4. Elimine una simulación detenida.

```
.\weaver-project-name-cli.bat delete-simulation --simulation simulation-name
```

## 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](#).

### Para limpiar la simulación

1. Si aún no lo ha hecho, vaya a la carpeta de herramientas de su proyecto y plataforma. Su *project-folder* utiliza los valores de *path/project-name* que proporcionó al crear el proyecto.

En un aviso de intérprete de comandos de Linux, introduzca:

```
cd project-folder/tools/linux
```

2. Busque los nombres de las simulaciones.

```
./weaver-project-name-cli.sh list-simulations
```

### Important

Si usa AWS IAM Identity Center o nombra perfiles para AWS Command Line Interface (AWS CLI), debe usar la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. La última versión es la 1.16.0. Para obtener información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#). Los scripts del SDK de la aplicación de SimSpace Weaver utilizan AWS CLI. Si utiliza el centro de identidad de IAM, puede copiar su perfil del centro de identidad

de IAM para el AWS CLI a su perfil de default o proporcionar el nombre de su perfil del centro de identidad de IAM a los scripts del SDK de la aplicación de SimSpace Weaver con el parámetro `--profile cli-profile-name`. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

3. Detenga una simulación.

```
./weaver-project-name-cli.sh stop-simulation --simulation simulation-name
```

4. Elimine una simulación detenida.

```
./weaver-project-name-cli.sh delete-simulation --simulation simulation-name
```

## AWS Management Console

Para limpiar la simulación

1. Abra la consola de SimSpace Weaver en la consola de [SimSpace Weaver](#).
2. En el panel de navegación, seleccione Simulations.
3. En la lista de Simulations, seleccione la opción situada junto al nombre de la simulación que desea eliminar.
4. Si la Status de simulación que ha seleccionado es STARTED:
  - a. Seleccione Actions en el menú desplegable.
  - b. Elija Stop.
  - c. Para confirmar, introduzca el nombre de la simulación.
  - d. Elija Stop.
  - e. Espere hasta que el Status de su simulación sea STOPPED.
5. Seleccione Actions en el menú desplegable.
6. Elija Delete.
7. Seleccione Delete para confirmar.

## Limpie los recursos de simulación de los servicios de apoyo

Para respaldar la simulación, SimSpace Weaver crea recursos en otros servicios. SimSpace Weaver no elimina estos recursos al eliminar la simulación. Puede eliminar estos recursos adicionales si no los necesita.

### Important

Es posible que reciba cargos de facturación por cualquiera de estos recursos que no elimine.

Para eliminar los recursos de soporte de su proyecto

1. Si ha terminado con su proyecto, elimine su pila de AWS CloudFormation. Para obtener más información sobre trabajar con AWS CloudFormation, consulte [Eliminación de una pila en la consola de AWS CloudFormation](#) en la Guía del usuario de AWS CloudFormation.

- `weaver-project-name-stack`

### Important

Las simulaciones que se iniciaron a partir del mismo proyecto comparten recursos, como el rol de la aplicación. Cuando se elimina la pila de AWS CloudFormation, también se elimina la función de aplicación. No elimine su pila de AWS CloudFormation si tiene otras simulaciones que comparten los mismos recursos.

### Note

Es probable que su pila de AWS CloudFormation muestre un informe de DELETE\_FAILED porque no puede eliminar los buckets de Amazon S3 que no estén vacíos. Eliminará buckets de Amazon S3 en el siguiente paso.

2. Si ha terminado con su proyecto, elimine sus buckets de Amazon S3. Para obtener más información sobre trabajar con buckets de Amazon S3, consulte [Eliminación de un bucket](#) en la Guía del usuario de Amazon Simple Storage Service.

- `weaver-lowercase-project-name-account-number-region`

Por ejemplo, el proyecto denominado MyProject en la cuenta 111122223333 en la región us-west-2 tiene el siguiente bucket.

- `weaver-myproject-111122223333-us-west-2`

 Note

Debe eliminar el contenido de un bucket de Amazon S3 antes de poder eliminar el bucket.

 Note

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

- *weaver- lowercase-project-name- número de cuenta -app-zips- región*
- *weaver- - número de cuenta -esquemas- región lowercase-project-name*

3. Si ha activado el registro en la simulación, elimine el grupo de registros. CloudWatch Para obtener más información sobre cómo trabajar con CloudWatch registros, consulte [Trabajar con grupos de registros y flujos](#) de CloudWatch registros en la Guía del usuario de Amazon Logs.

El nombre del grupo de registro de la simulación se especifica en su esquema (archivo de configuración): *project-folder\tools\project-name.yaml*

El nombre del grupo de registro es el valor de `log_destination_resource_name`. El siguiente fragmento de esquema muestra que el grupo de registro de la aplicación de ejemplo es `MySimulationLogs`.

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

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

#### Warning

Si inicia varias simulaciones que especifican el mismo grupo de registro, los datos de registro de todas esas simulaciones irán al mismo grupo de registro. Si elimina el grupo de registro, eliminará los datos de registro de todas las simulaciones que utilizan ese grupo de registro. Si elimina un grupo de registro para una simulación en ejecución, la simulación fallará.

#### Important

Si el esquema de la simulación especifica un `log_destination_service: "logs"` y `log_destination_resource_name`, pero no encuentra el grupo de CloudWatch registros en Logs, asegúrese de comprobar el mismo en el Región de AWS que se ejecutó la simulación.

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

Este tutorial le guiará a través del mismo procedimiento general que se presentó en el tutorial de inicio rápido, pero con más detalle. El tutorial de inicio rápido simplifica muchos de los pasos y oculta los detalles de la automatización. Este tutorial expone y explica esos detalles.

Le recomendamos que complete el [tutorial de inicio rápido](#) antes de continuar con este tutorial.

### Requisitos

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

### Pasos

- [Paso 1: Crear un proyecto](#)
- [Paso 2: activar el registro \(opcional\)](#)

- [Paso 3: cargar el esquema de simulación](#)
- [Paso 4: Crear su proyecto](#)
- [Paso 5: Subir aplicaciones](#)
- [Paso 6: Iniciar la simulación](#)
- [Paso 7: Obtenga los detalles de la simulación](#)
- [Paso 8: Iniciar aplicaciones personalizadas](#)
- [Paso 9: inicie el reloj](#)
- [Paso 10: Verifique los registros](#)
- [Paso 11: Ver la simulación](#)
- [Paso 12: Detenga y limpie la simulación](#)

## Paso 1: Crear un proyecto

El SDK de la aplicación SimSpace Weaver que se puede distribuir contiene un script que crea un proyecto a partir del proyecto agrupado de PathfindingSample. Debe ejecutar el script desde su ubicación en el sistema de archivos. El script crea un *project-name* en una *ruta* interna utilizando los valores que usted proporciona en la línea de comandos.

### Docker

Para crear un proyecto

1. En una línea de comandos de Windows, cambie a la carpeta del proyecto.

```
cd sdk-folder
```

2. Ejecute el script `create-project.bat`.

```
.\create-project.bat --name project-name --path path
```

A continuación se muestra la lista completa de comandos y parámetros.

```
.\create-project.bat --name project-name --path path --app-sdk-version version-number --template template-name --overwriteproject
```

 Important

No utilice un nombre de proyecto que supere los 20 caracteres. Si se supera este límite, se pueden producir errores.

 Note

Si recibe un mensaje para compartir una unidad con Docker, seleccione Yes para compartirla.

 Important

Si usa AWS IAM Identity Center o perfiles nombrados para AWS Command Line Interface (AWS CLI), debe usar la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. La última versión es la 1.16.0. Para obtener información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#). Los scripts del SDK de la aplicación de SimSpace Weaver utilizan AWS CLI. Si utiliza el centro de identidad de IAM, puede copiar su perfil del centro de identidad de IAM para el AWS CLI a su perfil de default o proporcionar el nombre de su perfil del centro de identidad de IAM a los scripts del SDK de la aplicación de SimSpace Weaver con el parámetro `--profile cli-profile-name`. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

## 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](#).

## Para crear un proyecto

1. En un aviso del intérprete de comandos de Linux, cambie a la carpeta del proyecto.

```
cd sdk-folder
```

2. Ejecute el script `create-project.sh`.

```
./create-project.sh --name project-name --path path
```

A continuación se muestra la lista completa de comandos y parámetros.

```
./create-project.sh --name project-name --path path --profile cli-profile-name  
--app-sdk-version version-number --template template-name --overwriteproject
```

### Important

No utilice un nombre de proyecto que supere los 20 caracteres. Si se supera este límite, se pueden producir errores.

### Important

Si usa AWS IAM Identity Center o perfiles nombrados para AWS Command Line Interface (AWS CLI), debe usar la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. La última versión es la 1.16.0. Para obtener información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#). Los scripts del SDK de la aplicación de SimSpace Weaver utilizan AWS CLI. Si utiliza el centro de identidad de IAM, puede copiar su perfil del centro de identidad de IAM para el AWS CLI a su perfil de default o proporcionar el nombre de su perfil del centro de identidad de IAM a los scripts del SDK de la aplicación de SimSpace Weaver con el parámetro `--profile cli-profile-name`. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

## Parámetros

### name

El nombre del proyecto.

### path

La ubicación del proyecto en el sistema de archivos. En ocasiones, la estructura del proyecto puede tener rutas de archivo largas que pueden superar el límite de longitud de las rutas del sistema operativo. Le recomendamos que utilice un nombre de ruta lo más corto posible.

### profile

El nombre del perfil de AWS CLI que el script debe usar para la autenticación. Para obtener más información, consulte [Configuración y ajustes de archivos de credenciales](#) en la Guía de usuario de AWS Command Line Interface. Este parámetro solo está disponible para la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. Para obtener más información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#).

### app-sdk-version

(opcional) La versión del SDK de la aplicación de SimSpace Weaver que usa su proyecto. Creará y vinculará sus aplicaciones con esta versión. Si el script no encuentra la versión en la ubicación del objeto de distribución, o si usted no proporciona el número de versión, el script descargará automáticamente la última versión.

### template

(opcional) Una plantilla de proyecto que el script utilizará para crear su proyecto. Si no proporciona una plantilla, el script utilizará `PathfindingSample`. Valores válidos:

- **PathfindingSample**: una aplicación de ejemplo que utiliza un único [trabajador](#).
- **MultiWorkerPathfindingSample**: una versión de la aplicación de ejemplo que utiliza varios [trabajadores](#).

### overwriteproject

(opcional) Utilice esta opción para sobrescribir una carpeta de proyecto existente con el mismo nombre y ruta del proyecto.

## Paso 2: activar el registro (opcional)

El registro está desactivado de forma predeterminada en el proyecto `PathfindingSample`. En este tutorial se presupone que el inicio de sesión está activado. Puede elegir activar el registro, pero es opcional.

### Important

`PathfindingSample` genera grandes cantidades de datos de registro. Si decide activar el registro, recibirá los cargos de facturación por los datos de registro. Seguirá recibiendo los cargos de facturación por los datos de registro mientras existan. Le recomendamos encarecidamente que detenga esta simulación y realice los pasos de limpieza que aparecen al final del tutorial lo antes posible si activa el registro.

Para activar el registro

1. Abra el siguiente archivo en un editor de texto.

Docker

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

### Note

Su *carpeta de proyecto* está usando los valores de `path\project-name` que proporcionó al crear el proyecto.

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](#).

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

**Note**

Su *carpeta de proyecto* está usando los valores de *path/project-name* que proporcionó al crear el proyecto.

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

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

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

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

4. 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>"
```

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

## Paso 3: cargar el esquema de simulación

SimSpace Weaver utiliza un esquema para configurar la simulación. El esquema es un archivo YAML de texto plano con formato. Para obtener más información, consulte [Configuración de su simulación](#).

### Docker

La aplicación de ejemplo incluye un esquema preconfigurado. Puede encontrar el archivo de esquema de la aplicación de ejemplo en la carpeta de herramientas de su proyecto:

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

## Para cargar el esquema

1. Si aún no lo ha hecho, vaya a la carpeta de herramientas de su proyecto y plataforma. Su *project-folder* está usando los valores de *path\project-name* que proporcionó al crear el proyecto.

En la línea de comandos de Windows, escriba:

```
cd project-folder\tools\windows
```

2. Utilice el script auxiliar para cargar el esquema.

```
.\upload-schema-project-name.bat
```

### Important

Si usa AWS IAM Identity Center o perfiles nombrados para AWS Command Line Interface (AWS CLI), debe usar la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. La última versión es la 1.16.0. Para obtener información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#). Los scripts del SDK de la aplicación de SimSpace Weaver utilizan AWS CLI. Si utiliza el centro de identidad de IAM, puede copiar su perfil del centro de identidad de IAM para el AWS CLI a su perfil de default o proporcionar el nombre de su perfil del centro de identidad de IAM a los scripts del SDK de la aplicación de SimSpace Weaver con el parámetro `--profile cli-profile-name`. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

## 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](#).

La aplicación de ejemplo incluye un esquema preconfigurado. Puede encontrar el archivo de esquema de la aplicación de ejemplo en la carpeta de herramientas de su proyecto:

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

Para cargar el esquema

1. Si aún no lo ha hecho, vaya a la carpeta de herramientas de su proyecto y plataforma. Su *project-folder* utiliza los valores de *path/project-name* que proporcionó al crear el proyecto.

En un aviso de intérprete de comandos de Linux, introduzca:

```
cd project-folder/tools/linux
```

2. Utilice el script auxiliar para cargar el esquema.

```
./upload-schema-project-name.sh
```

#### Important

Si usa AWS IAM Identity Center o perfiles nombrados para AWS Command Line Interface (AWS CLI), debe usar la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. La última versión es la 1.16.0. Para obtener información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#). Los scripts del SDK de la aplicación de SimSpace Weaver utilizan AWS CLI. Si utiliza el centro de identidad de IAM, puede copiar su perfil del centro de identidad de IAM para el AWS CLI a su perfil de default o proporcionar el nombre de su perfil del centro de identidad de IAM a los scripts del SDK de la aplicación de SimSpace Weaver con el parámetro `--profile cli-profile-name`. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

## Paso 4: Crear su proyecto

Ahora está en disposición de crear las aplicaciones espaciales y personalizadas para el proyecto de muestra. El proyecto de ejemplo incluye un script auxiliar que crea estas aplicaciones por usted.

### Docker

El script lanzará un contenedor de Docker con la imagen de Docker que creó al [configurar su entorno local](#). El script ejecuta la compilación en un entorno Amazon Linux en el contenedor de Docker. Escribe los artefactos de compilación y sus dependencias en su carpeta de Windows `project-folder\build`.

Para compilar su proyecto

1. Si aún no lo ha hecho, vaya a la carpeta de herramientas de su proyecto y plataforma. Su `project-folder` está usando los valores de `path\project-name` que proporcionó al crear el proyecto.

En la línea de comandos de Windows, escriba:

```
cd project-folder\tools\windows
```

2. Utilice el script auxiliar para compilar el proyecto.

```
.\build-project-name.bat
```

### 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](#).

El script escribe los artefactos de compilación y sus dependencias en su carpeta `project-folder/build`.

## Para compilar su proyecto

1. Si aún no lo ha hecho, vaya a la carpeta de herramientas de su proyecto y plataforma. Su *project-folder* utiliza los valores de *path/project-name* que proporcionó al crear el proyecto.

En un aviso de intérprete de comandos de Linux, introduzca:

```
cd project-folder/tools/linux
```

2. Utilice el script auxiliar para compilar el proyecto.

```
./build-project-name.sh
```

## Paso 5: Subir aplicaciones

El script de compilación ha empaquetado sus aplicaciones como archivos zip. Debe cargar estos archivos zip en buckets específicos de Amazon Simple Storage Service para poder ejecutar la simulación de SimSpace Weaver en la nube. El SDK de la aplicación de SimSpace Weaver proporciona un script auxiliar para gestionar la carga.

### Docker

#### Para cargar sus aplicaciones

1. Si aún no lo ha hecho, vaya a la carpeta de herramientas de su proyecto y plataforma. Su *project-folder* está usando los valores de *path\project-name* que proporcionó al crear el proyecto.

En la línea de comandos de Windows, escriba:

```
cd project-folder\tools\windows
```

2. Use el script auxiliar para cargar sus aplicaciones.

```
.\upload-app-project-name.bat
```

**⚠ Important**

Si usa AWS IAM Identity Center o perfiles nombrados para AWS Command Line Interface (AWS CLI), debe usar la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. La última versión es la 1.16.0. Para obtener información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#). Los scripts del SDK de la aplicación de SimSpace Weaver utilizan AWS CLI. Si utiliza el centro de identidad de IAM, puede copiar su perfil del centro de identidad de IAM para el AWS CLI a su perfil de default o proporcionar el nombre de su perfil del centro de identidad de IAM a los scripts del SDK de la aplicación de SimSpace Weaver con el parámetro `--profile cli-profile-name`. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

## 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](#).

Para cargar sus aplicaciones

1. Si aún no lo ha hecho, vaya a la carpeta de herramientas de su proyecto y plataforma. Su *project-folder* utiliza los valores de *path/project-name* que proporcionó al crear el proyecto.

En un aviso de intérprete de comandos de Linux, introduzca:

```
cd project-folder/tools/linux
```

2. Use el script auxiliar para cargar sus aplicaciones.

```
./upload-app-project-name.sh
```

### Important

Si usa AWS IAM Identity Center o perfiles nombrados para AWS Command Line Interface (AWS CLI), debe usar la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. La última versión es la 1.16.0. Para obtener información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#). Los scripts del SDK de la aplicación de SimSpace Weaver utilizan AWS CLI. Si utiliza el centro de identidad de IAM, puede copiar su perfil del centro de identidad de IAM para el AWS CLI a su perfil de default o proporcionar el nombre de su perfil del centro de identidad de IAM a los scripts del SDK de la aplicación de SimSpace Weaver con el parámetro `--profile cli-profile-name`. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

## Compruebe sus recursos de Amazon S3

Puede comprobar sus depósitos de Amazon S3 para asegurarse 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.

En el caso de la aplicación de muestra, el esquema (que cargó en un paso anterior) y los recursos de la aplicación utilizan los siguientes formatos de nombre:

- Bucket de esquemas: `simspaceweaver-project-name-lowercase-account-number-schemas-region`
  - Archivo de esquema: `project-name-schema.yaml`
- Bucket de aplicaciones: `simspaceweaver-project-name-lowercase-account-number-app-zips-region`
  - Aplicación espacial: `project-nameSpatial.zip`
  - Ver aplicación (personalizada): `project-nameView.zip`

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 esquema y los recursos de la aplicación tendrían los siguientes nombres:

- Bucket de esquemas: `simspaceweaver-myproject-111122223333-schemas-us-west-2`
  - Archivo de esquema: `MyProject-schema.yaml`
- Bucket de aplicaciones: `simspaceweaver-myproject-111122223333-apps-zips-us-west-2`
  - Aplicación espacial: `MyProjectSpatial.zip`
  - Ver aplicación (personalizada): `MyProjectView.zip`

## Paso 6: Iniciar la simulación

El SDK de la aplicación SimSpace Weaver proporciona un script auxiliar para iniciar la simulación. El script es un envoltorio alrededor de una llamada a la `StartSimulationAPI`. Proporciona los siguientes parámetros para la llamada a la API:

- El nombre del esquema de simulación (que cargó en un paso anterior)
- El nombre de la simulación
- El punto de conexión del servicio SimSpace Weaver

### Docker

Para iniciar la simulación

1. Si aún no lo ha hecho, vaya a la carpeta de herramientas de su proyecto y plataforma. Su *project-folder* está usando los valores de *path\project-name* que proporcionó al crear el proyecto.

En la línea de comandos de Windows, escriba:

```
cd project-folder\tools\windows
```

## 2. Utilice el script auxiliar para cargar e iniciar la simulación.

```
.\start-simulation-project-name.bat
```

### Important

A partir de la versión 1.12.3, el script `start-simulation` inicia la simulación con una duración máxima de 1 hora. Puede utilizar el parámetro `--maximum-duration` para especificar una duración máxima diferente. En la versión 1.12.2 o inferior, no se puede proporcionar una duración máxima al guion y las simulaciones tienen una duración máxima de 14 días. Para obtener más información acerca de la duración máxima de una simulación, consulte [Duración máxima de una simulación](#).

### Important

Si usa AWS IAM Identity Center o perfiles nombrados para AWS Command Line Interface (AWS CLI), debe usar la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. La última versión es la 1.16.0. Para obtener información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#). Los scripts del SDK de la aplicación de SimSpace Weaver utilizan AWS CLI. Si utiliza el centro de identidad de IAM, puede copiar su perfil del centro de identidad de IAM para el AWS CLI a su perfil de default o proporcionar el nombre de su perfil del centro de identidad de IAM a los scripts del SDK de la aplicación de SimSpace Weaver con el parámetro `--profile cli-profile-name`. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

### Note

`run-project-name.bat` es un script auxiliar alternativo que también iniciará el reloj de simulación. En este tutorial, en un paso posterior, iniciará el reloj por separado.

A partir de la versión 1.12.3, el script `run` inicia la simulación con una duración máxima de 1 hora. Puede utilizar el parámetro `--maximum-duration` para especificar una duración máxima diferente. En la versión 1.12.2 o inferior, no se puede proporcionar una duración máxima al guion y las simulaciones tienen una duración máxima de 14 días. Para obtener más información acerca de la duración máxima de una simulación, consulte [Duración máxima de una simulación](#).

## 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](#).

### Para iniciar la simulación

1. Si aún no lo ha hecho, vaya a la carpeta de herramientas de su proyecto y plataforma. Su *project-folder* utiliza los valores de *path/project-name* que proporcionó al crear el proyecto.

En un aviso de intérprete de comandos de Linux, introduzca:

```
cd project-folder/tools/linux
```

2. Utilice el script auxiliar para cargar e iniciar la simulación.

```
./start-simulation-project-name.sh
```

### Important

A partir de la versión 1.12.3, el script `start-simulation` inicia la simulación con una duración máxima de 1 hora. Puede utilizar el parámetro `--maximum-duration` para especificar una duración máxima diferente. En la versión 1.12.2 o inferior, no se puede proporcionar una duración máxima al guion y las simulaciones tienen una

duración máxima de 14 días. Para obtener más información acerca de la duración máxima de una simulación, consulte [Duración máxima de una simulación](#).

#### Important

Si usa AWS IAM Identity Center o perfiles nombrados para AWS Command Line Interface (AWS CLI), debe usar la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. La última versión es la 1.16.0. Para obtener información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#). Los scripts del SDK de la aplicación de SimSpace Weaver utilizan AWS CLI. Si utiliza el centro de identidad de IAM, puede copiar su perfil del centro de identidad de IAM para el AWS CLI a su perfil de default o proporcionar el nombre de su perfil del centro de identidad de IAM a los scripts del SDK de la aplicación de SimSpace Weaver con el parámetro `--profile cli-profile-name`. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

#### Note

`run-project-name.sh` es un script auxiliar alternativo que también iniciará el reloj de simulación. En este tutorial, en un paso posterior, iniciará el reloj por separado. A partir de la versión 1.12.3, el script `run` inicia la simulación con una duración máxima de 1 hora. Puede utilizar el parámetro `--maximum-duration` para especificar una duración máxima diferente. En la versión 1.12.2 o inferior, no se puede proporcionar una duración máxima al guion y las simulaciones tienen una duración máxima de 14 días. Para obtener más información acerca de la duración máxima de una simulación, consulte [Duración máxima de una simulación](#).

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. Si la simulación se inicia correctamente, se mostrará un resultado similar al que se incluye a continuación:

```
[2022-10-04T22:15:28] [INFO] Describe Simulation Results:
[2022-10-04T22:15:28] [INFO] {
  "Status": "STARTED",
  "Name": "MyProjectSimulation_22-10-04_22_10_15",
  "RoleArn": "arn:aws:iam::111122223333:role/weaver-MyProject-app-role",
  "CreationTime": 1664921418.09,
  "SchemaS3Location": {
    "ObjectKey": "MyProject-schema.yaml",
    "BucketName": "weaver-myproject-111122223333-us-west-2"
  },
}
```

### Note

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

- *weaver- lowercase-project-name- número de cuenta -app-zips- región*
- *weaver- - número de cuenta -esquemas- región lowercase-project-name*

## Paso 7: Obtenga los detalles de la simulación

El SDK de la aplicación SimSpace Weaver proporciona un script de ayuda que se basa en AWS CLI. El script simplifica las llamadas al AWS CLI al proporcionar el punto de conexión del servicio SimSpace Weaver. Utilice este script auxiliar para llamar a las API de SimSpace Weaver. 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

## Docker

Para obtener los detalles de la simulación

1. Si aún no lo ha hecho, vaya a la carpeta de herramientas de su proyecto y plataforma. Su *project-folder* está usando los valores de *path\project-name* que proporcionó al crear el proyecto.

En la línea de comandos de Windows, escriba:

```
cd project-folder\tools\windows
```

2. Utilice el script de ayuda CLI para llamar a la API de ListSimulations.

```
.\weaver-project-name-cli.bat list-simulations
```

### Important

Si usa AWS IAM Identity Center o perfiles nombrados para AWS Command Line Interface (AWS CLI), debe usar la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. La última versión es la 1.16.0. Para obtener información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#). Los scripts del SDK de la aplicación de SimSpace Weaver utilizan AWS CLI. Si utiliza el centro de identidad de IAM, puede copiar su perfil del centro de identidad de IAM para el AWS CLI a su perfil de default o proporcionar el nombre de su perfil del centro de identidad de IAM a los scripts del SDK de la aplicación de SimSpace Weaver con el parámetro `--profile cli-profile-name`. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

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"
}
```

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

```
.\weaver-project-name-cli.bat describe-simulation --simulation simulation-name
```

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

```
{
  "Name": "MyProjectSimulation_22-10-04_22_10_15",
  "ExecutionId": "1a2b3c4d-0ab1-1234-567a-12ab34cd5e6f",
  "Arn": "arn:aws:simspaceweaver:us-west-2:111122223333:simulation/MyProjectSimulation_22-10-04_22_10_15",
  "RoleArn": "arn:aws:iam::111122223333:role/weaver-MyProject-app-role",
  "CreationTime": 1664921418.09,
  "Status": "STARTED",
  "TargetStatus": "STARTED",
  "SchemaS3Location": {
    "ObjectKey": "MyProject-schema.yaml",
    "BucketName": "weaver-myproject-111122223333-us-west-2"
  },
  "SchemaError": "[]",
  "LoggingConfiguration": {
    "Destinations": [
      {
        "CloudWatchLogsLogGroup": {
          "LogGroupArn": "arn:aws:logs:us-west-2:111122223333:log-group:MySimulationLogs"
        }
      }
    ]
  }
}
```

```
    }
  }
]
},
"LiveSimulationState": {
  "Domains": [
    {
      "Type": "",
      "Name": "MySpatialSimulation",
      "Lifecycle": "Unknown"
    },
    {
      "Type": "",
      "Name": "MyViewDomain",
      "Lifecycle": "ByRequest"
    }
  ],
  "Clocks": [
    {
      "Status": "STARTED",
      "TargetStatus": "STARTED"
    }
  ]
},
"MaximumDuration": "1H",
"StartError": "[]"
}
```

## 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](#).

Para obtener los detalles de la simulación

1. Si aún no lo ha hecho, vaya a la carpeta de herramientas de su proyecto y plataforma. Su *project-folder* utiliza los valores de *path/project-name* que proporcionó al crear el proyecto.

En un aviso de intérprete de comandos de Linux, introduzca:

```
cd project-folder/tools/linux
```

2. Utilice el script de ayuda CLI para llamar a la API de ListSimulations.

```
./weaver-project-name-cli.sh list-simulations
```

#### Important

Si usa AWS IAM Identity Center o perfiles nombrados para AWS Command Line Interface (AWS CLI), debe usar la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. La última versión es la 1.16.0. Para obtener información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#). Los scripts del SDK de la aplicación de SimSpace Weaver utilizan AWS CLI. Si utiliza el centro de identidad de IAM, puede copiar su perfil del centro de identidad de IAM para el AWS CLI a su perfil de default o proporcionar el nombre de su perfil del centro de identidad de IAM a los scripts del SDK de la aplicación de SimSpace Weaver con el parámetro `--profile cli-profile-name`. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

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"
}

```

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

```

./weaver-project-name-cli.sh describe-simulation --simulation simulation-name

```

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

```

{
  "Name": "MyProjectSimulation_22-10-04_22_10_15",
  "ExecutionId": "1a2b3c4d-0ab1-1234-567a-12ab34cd5e6f",
  "Arn": "arn:aws:simspaceweaver:us-west-2:111122223333:simulation/
MyProjectSimulation_22-10-04_22_10_15",
  "RoleArn": "arn:aws:iam::111122223333:role/weaver-MyProject-app-role",
  "CreationTime": 1664921418.09,
  "Status": "STARTED",
  "TargetStatus": "STARTED",
  "SchemaS3Location": {
    "ObjectKey": "MyProject-schema.yaml",
    "BucketName": "weaver-myproject-111122223333-us-west-2"
  },
  "SchemaError": "[]",
  "LoggingConfiguration": {
    "Destinations": [
      {
        "CloudWatchLogsLogGroup": {
          "LogGroupArn": "arn:aws:logs:us-west-2:111122223333:log-
group:MySimulationLogs"
        }
      }
    ]
  },
  "LiveSimulationState": {
    "Domains": [

```

```
{
  {
    "Type": "",
    "Name": "MySpatialSimulation",
    "Lifecycle": "Unknown"
  },
  {
    "Type": "",
    "Name": "MyViewDomain",
    "Lifecycle": "ByRequest"
  }
],
"Clocks": [
  {
    "Status": "STARTED",
    "TargetStatus": "STARTED"
  }
]
},
"MaximumDuration": "1H",
"StartError": "[]"
}
```

## Paso 8: Iniciar 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.

Puede usar el script de ayuda de CLI para llamar a la API de StartApp e iniciar sus aplicaciones personalizadas.

### Important

Si usa AWS IAM Identity Center o perfiles nombrados para AWS Command Line Interface (AWS CLI), debe usar la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. La última versión es la 1.16.0. Para obtener información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#). Los scripts del SDK de la aplicación de SimSpace Weaver utilizan AWS CLI. Si utiliza el centro de identidad de

IAM, puede copiar su perfil del centro de identidad de IAM para el AWS CLI a su perfil de default o proporcionar el nombre de su perfil del centro de identidad de IAM a los scripts del SDK de la aplicación de SimSpace Weaver con el parámetro `--profile cli-profile-name`. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

## Docker

```
.\weaver-project-name-cli.bat start-app --simulation simulation-name --name app-name  
--domain domain-name
```

## 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](#).

```
./weaver-project-name-cli.sh 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. Para comprobar el estado de la simulación, consulte [Paso 7: Obtenga los detalles de la simulación](#).

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 los conceptos de SimSpace Weaver, consulte [Conceptos clave de SimSpace Weaver](#).

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

## Docker

```
.\weaver-project-name-cli.bat describe-app --simulation simulation-name --app app-name --domain domain-name
```

## 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](#).

```
./weaver-project-name-cli.sh describe-app --simulation simulation-name --app app-name --domain domain-name
```

## Docker

Para iniciar la aplicación de visualización en este tutorial

1. Si aún no lo ha hecho, vaya a la carpeta de herramientas de su proyecto y plataforma. Su `project-folder` está usando los valores de `path\project-name` que proporcionó al crear el proyecto.

En la línea de comandos de Windows, escriba:

```
cd project-folder\tools\windows
```

- Utilice el script de ayuda CLI para llamar a StartApp para ViewApp.

```
.\weaver-project-name-cli.bat start-app --simulation simulation-name --name  
ViewApp --domain MyViewDomain
```

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

```
.\weaver-project-name-cli.bat describe-app --simulation simulation-name --app  
ViewApp --domain MyViewDomain
```

## 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](#).

Para iniciar la aplicación de visualización en este tutorial

- Si aún no lo ha hecho, vaya a la carpeta de herramientas de su proyecto y plataforma. Su *project-folder* utiliza los valores de *path/project-name* que proporcionó al crear el proyecto.

En un aviso de intérprete de comandos de Linux, introduzca:

```
cd project-folder/tools/linux
```

- Utilice el script de ayuda CLI para llamar a StartApp para ViewApp.

```
./weaver-project-name-cli.sh start-app --simulation simulation-name --name  
ViewApp --domain MyViewDomain
```

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

```
./weaver-project-name-cli.sh 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 SimSpace Weaver expone los clientes para que se conecten a su aplicación. SimSpace Weaver asigna el puerto `Declared` al puerto `Actual`.

**Note**

Puede utilizar el [procedimiento del tutorial de inicio rápido](#) para obtener la dirección IP y el número de puerto de cualquier aplicación personalizada iniciada, independientemente de este flujo de trabajo.

## Paso 9: inicie 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 graduación, sus aplicaciones espaciales recorren las entidades que son de su propiedad y con las que archivan los resultados en SimSpace Weaver

**Note**

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

**Important**

Si usa AWS IAM Identity Center o perfiles nombrados para AWS Command Line Interface (AWS CLI), debe usar la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. La última versión es la 1.16.0. Para obtener información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#). Los scripts del SDK de la aplicación de SimSpace Weaver utilizan AWS CLI. Si utiliza el centro de identidad de IAM, puede copiar su perfil del centro de identidad de IAM para el AWS CLI a su perfil de default o proporcionar el nombre de su perfil del centro de identidad de IAM a los scripts del SDK de la aplicación de SimSpace Weaver con el parámetro `--profile cli-profile-name`. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

## Docker

Para poner en marcha el reloj

1. Si aún no lo ha hecho, vaya a la carpeta de herramientas de su proyecto y plataforma. Su *project-folder* está usando los valores de *path\project-name* que proporcionó al crear el proyecto.

En la línea de comandos de Windows, escriba:

```
cd project-folder\tools\windows
```

2. Utilice el script de ayuda CLI para llamar a la API de StartClock.

```
.\weaver-project-name-cli.bat start-clock --simulation simulation-name
```

### Note

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

```
.\weaver-project-name-cli.bat list-simulations
```

## 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](#).

Para poner en marcha el reloj

1. Si aún no lo ha hecho, vaya a la carpeta de herramientas de su proyecto y plataforma. Su *project-folder* utiliza los valores de *path/project-name* que proporcionó al crear el proyecto.

En un aviso de intérprete de comandos de Linux, introduzca:

```
cd project-folder/tools/linux
```

2. Utilice el script de ayuda CLI para llamar a la API de StartClock.

```
./weaver-project-name-cli.sh start-clock --simulation simulation-name
```

#### Note

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

```
./weaver-project-name-cli.sh list-simulations
```

## Paso 10: Verifique los registros

SimSpace Weaver describe 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.

### Important

Si usa AWS IAM Identity Center o perfiles nombrados para AWS Command Line Interface (AWS CLI), debe usar la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. La última versión es la 1.16.0. Para obtener información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#). Los scripts del SDK de la aplicación de SimSpace Weaver utilizan AWS CLI. Si utiliza el centro de identidad de IAM, puede copiar su perfil del centro de identidad de IAM para el AWS CLI a su perfil de default o proporcionar el nombre de su perfil del centro de identidad de IAM a los scripts del SDK de la aplicación de SimSpace Weaver con el parámetro `--profile cli-profile-name`. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

## Docker

```
project-folder\tools\windows\weaver-project-name-cli.bat describe-simulation --simulation simulation-name
```

## 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](#).

```
project-folder/tools/linux/weaver-project-name-cli.sh 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 servicio de SimSpace Weaver.

```
/sim/management
```

- Flujo de registro de errores: mensajes de error producidos por el servicio de SimSpace Weaver. Este flujo de registro solo existe si hay errores. SimSpace Weaver almacena los errores escritos por sus 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 11: Ver la simulación

El SDK de la aplicación de SimSpace Weaver ofrece diferentes opciones para ver la aplicación de muestra. Puede usar el cliente de consola de ejemplo si no tiene soporte local para el desarrollo de Unreal Engine. En las instrucciones para el cliente de Unreal Engine se presupone que está 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.

### Windows console client

#### Requisitos

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

- [CMake3](#)
- [Git](#)

Para conectarse a la aplicación de muestra con el cliente de consola de muestra

1. En una ventana de línea de comandos, vaya a la carpeta del cliente de consola (en la carpeta SDK de la aplicación).

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

2. Se utiliza CMake3 para crear una solución de Visual Studio en esta carpeta.

```
cmake .
```

 Note

Asegúrese de incluir el espacio y el punto al final.

 Important

Mantenga abierta la ventana de la línea de comandos para seguir pasos.

3. En Visual Studio que creó en el paso anterior, abra `PathfindingSampleConsoleClient.sln`.
4. Seleccione la configuración de la compilación `RelWithDebInfo`.
5. Seleccione `Build > Build Solution`.
6. En su ventana de línea de comandos anterior, vaya a la carpeta de salida de la compilación en la carpeta del cliente de consola.

```
cd RelWithDebInfo
```

7. Ejecute el cliente con la dirección IP y el número de puerto de su `ViewApp`.

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

La ventana de la línea de comandos debería mostrar los números de los eventos de actualización, eliminación y creación de entidades, de forma similar al siguiente ejemplo de salida.

 Note

Las direcciones IP y los números de puerto del siguiente ejemplo de salida son marcadores de posición. Proporcione su dirección IP y su número de puerto de su ViewApp al cliente de la consola. Proporcione el número de puerto de Actual si desea conectarse a un ViewApp que se ejecute en el Nube de AWS. Proporcione la dirección IP y el número de puerto 127.0.0.1:7000 cuando se conecte a un sistema ViewApp que se ejecute en su sistema local. Para obtener más información, consulte [Desarrollo local](#).

```
##PathfindingSample#ViewApp Message Reader##

Added argument url:tcp://198.51.100.135:4321
Some subscription arguments are missing, restoring defaults.

*****
Sample usage without a MoveStrategy:
ConsoleClient --url tcp://198.51.100.135:4321 --subs-center-x 600 --subs-center-y 500 --subs-radius 50
Sample usage with CircleMoveStrategy:
ConsoleClient --url tcp://198.51.100.135:4321 --subs-center-x 600 --subs-center-y 500 --subs-radius 50 --subs-move-strategy circle --circle-center-x 500 --circle-center-y 500 --circle-speed 0.001

*****
Starting NNG client. NNG version: 1.2.4
Creating socket ...done.
Connecting to View App ... done.
Initiating connection to tcp:// 198.51.100.135:4321 ... done.

Receiving messages ...
[2022-10-04 19:13:00.710] CreateEntity Count: 72
[2022-10-04 19:13:00.756] UpdateEntity Count: 42
```

```
[2022-10-04 19:13:00.794] DeleteEntity Count: 72
[2022-10-04 19:13:03.690] CreateEntity Count: 11
[2022-10-04 19:13:03.725] UpdateEntity Count: 2
[2022-10-04 19:13:03.757] UpdateEntity Count: 2
[2022-10-04 19:13:03.790] UpdateEntity Count: 2
```

 Note

Para obtener ayuda sobre la resolución de problemas, consulte [El cliente de consola PathFindingSample no se puede conectar](#).

8. Pulse CTRL+C para salir del cliente de la consola.

## Linux console client

 Important

Proporcionamos estas instrucciones para su comodidad. Es posible que no funcionen en algunos entornos Linux. Estos procedimientos no son compatibles.

En este procedimiento, se presupone que está trabajando completamente en un entorno Linux. También puede visualizar la simulación con los clientes integrados en Windows.

### Requisitos

- CMake3
- Compilador C (ya incluido en Amazon Linux 2)
- Git

Para conectarse a la aplicación de muestra con el cliente de consola de muestra

1. En un Linux símbolo del intérprete de comandos, vaya a la carpeta del cliente de consola (en la carpeta SDK de la aplicación).

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

2. Cree una carpeta de compilación.

```
mkdir build
```

3. Vaya a la carpeta de compilación.

```
cd build
```

4. Use CMake3 para crear el cliente.

```
cmake3 ../ && cmake3 --build .
```

 Note

Asegúrese de incluir el espacio y el punto al final.

5. Ejecute el cliente con la dirección IP y el número de puerto de su ViewApp.

```
./ConsoleClient --url tcp://ip-address:port-number
```

La ventana de la línea de comandos debería mostrar los números de los eventos de actualización, eliminación y creación de entidades, de forma similar al siguiente ejemplo de salida.

 Note

Las direcciones IP y los números de puerto del siguiente ejemplo de salida son marcadores de posición. Proporcione su dirección IP y su número de puerto de su ViewApp al cliente de la consola. Proporcione el número de puerto de Actual si desea conectarse a un ViewApp que se ejecute en el Nube de AWS. Proporcione la dirección IP y el número de puerto 127.0.0.1:7000 cuando se conecte a un sistema ViewApp que se ejecute en su sistema local. Para obtener más información, consulte [Desarrollo local](#).

```
##PathfindingSample#ViewApp Message Reader##
```

```
Added argument url:tcp://198.51.100.135:4321
Some subscription arguments are missing, restoring defaults.

*****
Sample usage without a MoveStrategy:
ConsoleClient --url tcp://198.51.100.135:4321 --subs-center-x 600 --subs-center-
y 500 --subs-radius 50
Sample usage with CircleMoveStrategy:
ConsoleClient --url tcp://198.51.100.135:4321 --subs-center-x 600 --subs-center-
y 500 --subs-radius 50 --subs-move-strategy circle --circle-center-x 500 --
circle-center-y 500 --circle-speed 0.001

*****
Starting NNG client. NNG version: 1.2.4
Creating socket ...done.
Connecting to View App ... done.
Initiating connection to tcp:// 198.51.100.135:4321 ... done.

Receiving messages ...
[2022-10-04 19:13:00.710] CreateEntity Count: 72
[2022-10-04 19:13:00.756] UpdateEntity Count: 42
[2022-10-04 19:13:00.794] DeleteEntity Count: 72
[2022-10-04 19:13:03.690] CreateEntity Count: 11
[2022-10-04 19:13:03.725] UpdateEntity Count: 2
[2022-10-04 19:13:03.757] UpdateEntity Count: 2
[2022-10-04 19:13:03.790] UpdateEntity Count: 2
```

#### Note

Para obtener ayuda sobre la resolución de problemas, consulte [El cliente de consola PathFindingSample no se puede conectar.](#)

6. Pulse CTRL+C para salir del cliente de la consola.

## Unreal Engine on Windows

### Requisitos

- Entorno de desarrollo de Unreal Engine 5
- Microsoft .NET Framework 4.8 Developer Pack

- Cliente de consola Windows (consulte la pestaña del cliente de consola Windows en esta página)

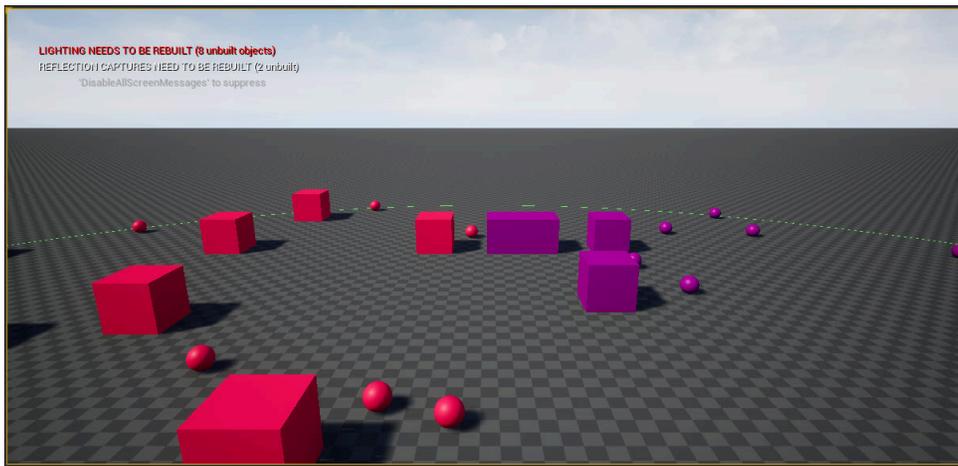
**⚠ Important**

Otras versiones de Unreal Engine y no .NET son compatibles y pueden causar problemas.

Para conectarse a la aplicación de muestra con el cliente de muestra Unreal

1. El cliente Unreal Engine usa la biblioteca del cliente de consola NNG. Debe crear el cliente de consola para Windows el que aún no lo ha creado. Para obtener más información, consulte la pestaña del cliente de la consola Windows de esta página.
2. En una ventana del administrador de archivos, vaya a `sdk-folder\packaging-tools\clients\PathfindingSampleClients\UnrealClient`.
3. Abra `UnrealClient.uproject`.
4. Si el editor le pregunta si desea reconstruir los módulos de `UnrealClient`, seleccione yes.
5. En un editor de texto, abra `sdk-folder\packaging-tools\clients\PathfindingSampleClients\UnrealClient\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:port-number` (debería tener el aspecto de `tcp://198.51.100.135:1234`).
7. En el editor de Unreal, seleccione play.

El editor de Unreal debería mostrar una visualización de la simulación, similar a la siguiente captura de pantalla.

**Note**

Dependiendo de la potencia del sistema de desarrollo local, el editor de Unreal puede tardar unos minutos en mostrar la simulación. Durante este tiempo, el sistema puede parecer que se está congelando.

Utilice las teclas W, A, S, D para mover el cliente Unreal. Mantenga pulsado el botón del ratón y arrastre el ratón para girar.

Puede pulsar la tecla [ (corchete izquierdo) para reducir el tamaño del área de suscripción. Puede pulsar la tecla ] (corchete derecho) para aumentar el tamaño del área de suscripción. El tamaño del área de suscripción determina la cantidad de entidades que aparecen en el cliente.

Puede pulsar la tecla C para crear una entidad en la simulación. El cliente envía un comando `CreateEntity` a la aplicación de visualización. A continuación, la aplicación de visualización creará la entidad y la transferirá al dominio espacial.

Puede examinar el código `ViewAppDriver::HandleEntityCreationRequests` en `project-folder\src\PathfindingSample\ViewApp\Driver\ViewAppDriver.cpp` para ver cómo la aplicación implementa este proceso.

**Note**

Si no conoce la dirección IP y el número de puerto de su aplicación de visualización, puede usar el [procedimiento del tutorial de inicio rápido](#) para obtener esa información.

## Paso 12: Detenga y limpie la simulación

Es importante limpiar las simulaciones cuando ya no las necesita. Los recursos de simulación de SimSpace Weaver se tienen en cuenta para sus Service Quotas (límites), incluso si la simulación está detenida. Seguirá cobrando los cargos de facturación por las simulaciones que se estén ejecutando. También puede recibir cargos de facturación por el almacenamiento de datos en servicios de apoyo, como Amazon CloudWatch Logs y Amazon Simple Storage Service. Para obtener más información sobre las cuotas de SimSpace Weaver servicio, consulte [SimSpace Puntos finales y cuotas de Weaver](#).

Siga los procedimientos de esta sección cuando desee limpiar la simulación.

### Important

No puede reiniciar una simulación después de detenerla.

### Important

No es posible recuperar una simulación eliminada.

## Limpie los recursos de simulación en SimSpace Weaver

Debe detener la simulación antes de poder eliminarla. Al eliminar una simulación, solo se eliminan los recursos en SimSpace Weaver. Debe realizar pasos distintos para eliminar los recursos que la simulación creó o utiliza en otros servicios (consulte la sección siguiente).

### Docker

Para limpiar la simulación

1. Si aún no lo ha hecho, vaya a la carpeta de herramientas de su proyecto y plataforma. Su *project-folder* está usando los valores de *path\project-name* que proporcionó al crear el proyecto.

En la línea de comandos de Windows, escriba:

```
cd project-folder\tools\windows
```

## 2. Busque los nombres de las simulaciones.

```
.\weaver-project-name-cli.bat list-simulations
```

### Important

Si usa AWS IAM Identity Center o nombra perfiles para AWS Command Line Interface (AWS CLI), debe usar la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. La última versión es la 1.16.0. Para obtener información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#). Los scripts del SDK de la aplicación de SimSpace Weaver utilizan AWS CLI. Si utiliza el centro de identidad de IAM, puede copiar su perfil del centro de identidad de IAM para el AWS CLI a su perfil de default o proporcionar el nombre de su perfil del centro de identidad de IAM a los scripts del SDK de la aplicación de SimSpace Weaver con el parámetro `--profile cli-profile-name`. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

## 3. Detenga una simulación.

```
.\weaver-project-name-cli.bat stop-simulation --simulation simulation-name
```

## 4. Elimine una simulación detenida.

```
.\weaver-project-name-cli.bat delete-simulation --simulation simulation-name
```

## 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](#).

## Para limpiar la simulación

1. Si aún no lo ha hecho, vaya a la carpeta de herramientas de su proyecto y plataforma. Su *project-folder* utiliza los valores de *path/project-name* que proporcionó al crear el proyecto.

En un aviso de intérprete de comandos de Linux, introduzca:

```
cd project-folder/tools/linux
```

2. Busque los nombres de las simulaciones.

```
./weaver-project-name-cli.sh list-simulations
```

### Important

Si usa AWS IAM Identity Center o nombra perfiles para AWS Command Line Interface (AWS CLI), debe usar la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. La última versión es la 1.16.0. Para obtener información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#). Los scripts del SDK de la aplicación de SimSpace Weaver utilizan AWS CLI. Si utiliza el centro de identidad de IAM, puede copiar su perfil del centro de identidad de IAM para el AWS CLI a su perfil de default o proporcionar el nombre de su perfil del centro de identidad de IAM a los scripts del SDK de la aplicación de SimSpace Weaver con el parámetro `--profile cli-profile-name`. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

3. Detenga una simulación.

```
./weaver-project-name-cli.sh stop-simulation --simulation simulation-name
```

4. Elimine una simulación detenida.

```
./weaver-project-name-cli.sh delete-simulation --simulation simulation-name
```

## AWS Management Console

Para limpiar la simulación

1. Abra la consola de SimSpace Weaver en la consola de [SimSpace Weaver](#).
2. En el panel de navegación, seleccione Simulations.
3. En la lista de Simulations, seleccione la opción situada junto al nombre de la simulación que desea eliminar.
4. Si la Status de simulación que ha seleccionado es STARTED:
  - a. Seleccione Actions en el menú desplegable.
  - b. Elija Stop.
  - c. Para confirmar, introduzca el nombre de la simulación.
  - d. Elija Stop.
  - e. Espere hasta que el Status de su simulación sea STOPPED.
5. Seleccione Actions en el menú desplegable.
6. Elija Delete.
7. Seleccione Delete para confirmar.

Limpie los recursos de simulación de los servicios de apoyo

Para respaldar la simulación, SimSpace Weaver crea recursos en otros servicios. SimSpace Weaver no elimina estos recursos al eliminar la simulación. Puede eliminar estos recursos adicionales si no los necesita.

### Important

Es posible que reciba cargos de facturación por cualquiera de estos recursos que no elimine.

Para eliminar los recursos de soporte de su proyecto

1. Si ha terminado con su proyecto, elimine su pila de AWS CloudFormation. Para obtener más información sobre trabajar con AWS CloudFormation, consulte [Eliminación de una pila en la consola de AWS CloudFormation](#) en la Guía del usuario de AWS CloudFormation.
  - `weaver-project-name-stack`

**⚠ Important**

Las simulaciones que se iniciaron a partir del mismo proyecto comparten recursos, como el rol de la aplicación. Cuando se elimina la pila de AWS CloudFormation, también se elimina la función de aplicación. No elimine su pila de AWS CloudFormation si tiene otras simulaciones que compartan los mismos recursos.

**ℹ Note**

Es probable que su pila de AWS CloudFormation muestre un informe de DELETE\_FAILED porque no puede eliminar los buckets de Amazon S3 que no estén vacíos. Eliminará buckets de Amazon S3 en el siguiente paso.

2. Si ha terminado con su proyecto, elimine sus buckets de Amazon S3. Para obtener más información sobre trabajar con buckets de Amazon S3, consulte [Eliminación de un bucket](#) en la Guía del usuario de Amazon Simple Storage Service.

- `weaver-lowercase-project-name-account-number-region`

Por ejemplo, el proyecto denominado MyProject en la cuenta 111122223333 en la región us-west-2 tiene el siguiente bucket.

- `weaver-myproject-111122223333-us-west-2`

**ℹ Note**

Debe eliminar el contenido de un bucket de Amazon S3 antes de poder eliminar el bucket.

**ℹ Note**

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

- *weaver- lowercase-project-name- número de cuenta -app-zips- región*
- *weaver- - número de cuenta -esquemas- región lowercase-project-name*

3. Si ha activado el registro en la simulación, elimine el grupo de registros. CloudWatch Para obtener más información sobre cómo trabajar con CloudWatch registros, consulte [Trabajar con grupos de registros y flujos](#) de CloudWatch registros en la Guía del usuario de Amazon Logs.

El nombre del grupo de registro de la simulación se especifica en su esquema (archivo de configuración): *project-folder\tools\project-name.yaml*

El nombre del grupo de registro es el valor de `log_destination_resource_name`. El siguiente fragmento de esquema muestra que el grupo de registro de la aplicación de ejemplo es `MySimulationLogs`.

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

#### Warning

Si inicia varias simulaciones que especifican el mismo grupo de registro, los datos de registro de todas esas simulaciones irán al mismo grupo de registro. Si elimina el grupo de registro, eliminará los datos de registro de todas las simulaciones que utilizan ese grupo de registro. Si elimina un grupo de registro para una simulación en ejecución, la simulación fallará.

#### Important

Si el esquema de la simulación especifica un `log_destination_service: "logs"` y `log_destination_resource_name`, pero no encuentra el grupo de CloudWatch

registros en Logs, asegúrese de comprobar el mismo en el Región de AWS que se ejecutó la simulación.

# Trabajar 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](#)
- [Desarrollo local](#)
- [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 modificar 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.

## Docker

Para configurar la simulación, edite el archivo de esquema de simulación:

```
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-project-name-cli.bat
```

También puede usar el script auxiliar de carga de esquemas para su proyecto si no está usando el script de inicio rápido para crear la simulación:

```
project-folder\tools\windows\upload-schema-project-name.bat
```

## 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](#).

La simulación se configura editando su archivo de esquema de simulación:

```
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/linux/quick-start-project-name-cli.sh
```

También puede usar el script auxiliar de carga de esquemas para su proyecto si no está usando el script de inicio rápido para crear la simulación:

```
project-folder/tools/linux/upload-schema-project-name.sh
```

### Important

Si usa AWS IAM Identity Center o perfiles nombrados para AWS Command Line Interface (AWS CLI), debe usar la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. La última versión es la 1.16.0. Para obtener información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#). Los scripts del SDK de la aplicación de SimSpace Weaver utilizan AWS CLI. Si utiliza el centro de identidad de IAM, puede copiar su perfil del centro de identidad de IAM para el AWS CLI a su perfil de default o proporcionar el nombre de su perfil del centro de identidad de IAM a los scripts del SDK de la aplicación de SimSpace Weaver con el parámetro `--profile cli-profile-name`. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

## 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 con la frecuencia de bits especificada.

### Note

El script de creación de proyectos del SDK de la aplicación SimSpace Weaver 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 [Referencia del esquema de simulación SimSpace Weaver](#).

## Temas

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

## Versión de SDK

El campo `sdk_version` especifica la versión de SimSpace Weaver para la que está formateado el esquema. Valores válidos: 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](#).

## 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

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

- *weaver-lowercase-project-name-account-number-app-zips-region*
- *weaver-lowercase-project-name-account-number-schemas-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

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

- `weaver-lowercase-project-name-account-number-app-zips-region`
- `weaver-lowercase-project-name-account-number-schemas-region`

La propiedad `package` especifica el URI de S3 de un archivo zip de 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 contenedor Docker en un dispositivo de trabajo.

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 propiedad `required_resource_units` especifica el número de unidades de recursos informáticos que SimSpace Weaver debe asignar a esta aplicación. Una unidad de recursos informáticos es una cantidad fija de capacidad de procesamiento (vCPU) y memoria (RAM) de 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

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

- `weaver-lowercase-project-name-account-number-app-zips-region`
- `weaver-lowercase-project-name-account-number-schemas-region`

Una estrategia de particionamiento con una topología Grid (la única topología compatible en esta versión) dirige SimSpace Weaver para organizar las particiones de 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 por 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

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

#### Important

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

#### 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.

Agregue 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
  [\u0027\u0027grid_partition\u0027\u0027] values, or adjusting the
  dimensions of your [\u0027\u0027grid_placement_groups\u0027\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 puede determinar un acuerdo para agrupar los dominios de 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 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 a la API describe-app una vez que sus aplicaciones hayan empezado a encontrar los números de puerto dinámicos (reales). Para obtener más información, consulte [Paso 4: obtenga su dirección IP y número de puerto](#) 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

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

- `weaver-lowercase-project-name-account-number-app-zips-region`
- `weaver-lowercase-project-name-account-number-schemas-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 SimSpace Weaver se inicia y detiene una aplicación,

se considera que la aplicación tiene un ciclo de vida administrado. 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://example-bucket/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://example-bucket/PlayerConnectionServiceApp.zip"
      launch_command: ["PlayerConnectionServiceApp"]
      required_resource_units:
        compute: 1
      endpoint_config:
        ingress_ports:
          - 9000
          - 9001
```

## Duración máxima de una simulación

Cada simulación en 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 pasará inmediatamente al estado STOPPING en cuanto alcance el estado STARTED.

## Iniciar una simulación mediante scripts del SDK de la aplicación SimSpace Weaver

Puede proporcionar un valor para el parámetro `maximum-duration` cuando utilice uno de los siguientes scripts para iniciar una simulación:

- `quick-start-project-name-cli.bat --maximum-duration value`
- `start-simulation-project-name.bat --maximum-duration value`
- `run-project-name.bat --maximum-duration value`

Cada script pasa el valor de `maximum-duration` a la API de `StartSimulation`.

**⚠ Important**

Si no proporciona un valor para `maximum-duration`, SimSpace Weaver utiliza el [valor por defecto](#) (14D).

## 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 SimSpace Weaver detiene 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

El desarrollo de SimSpace Weaver requiere un entorno Amazon Linux 2 (AL2) para crear aplicaciones porque sus simulaciones se ejecutan en Amazon Linux en AWS Cloud. Si está usando Windows, puede usar los scripts del SDK de la aplicación SimSpace Weaver para crear y lanzar un contenedor Docker que ejecute AL2 con las dependencias que necesita para crear aplicaciones SimSpace Weaver. 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 aplicación SimSpace Weaver proporciona una aplicación de ejemplo. Puede encontrar el código fuente de la aplicación espacial de la aplicación de ejemplo en la siguiente carpeta:

### Docker

```
project-folder\src\PathfindingSample\SpatialApp
```

## 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](#).

```
project-folder/src/PathfindingSample/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 aplicación SimSpace Weaver proporciona una aplicación de ejemplo. 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:

### Docker

```
project-folder\src\PathfindingSample\ViewApp
```

### 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](#).

```
project-folder/src/PathfindingSample/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 las 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 aplicación SimSpace Weaver proporciona clientes para su aplicación de ejemplo. 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 el tutorial de inicio rápido de [Paso 5: Ver la simulación](#) en esta guía.

## Desarrollo local

Puede implementar sus aplicaciones de SimSpace Weaver localmente para realizar pruebas y depuraciones rápidas. SimSpace Weaver Local solo admite Microsoft Windows para construir.

### Important

Para obtener información acerca de cómo trabajar con Unity, consulte Unreal Engine y [Compatibilidad con otros motores](#).

### Important

Si está trabajando en la versión 1.15.3 con C++, Python, Unity o Unreal Engine, consulte [Diferencias de desarrollo local en la versión 1.15.3](#).

### Requisitos

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

### Temas

- [Crear su simulación para SimSpace Weaver Local](#)
- [Ejecute la simulación con SimSpace Weaver Local](#)
- [Visualizar su simulación local](#)
- [Detener su simulación local](#)
- [Depuración de simulaciones locales](#)
- [Diferencias de desarrollo local en la versión 1.15.3](#)

## Crear su simulación para SimSpace Weaver Local

Para aprender a usar SimSpace Weaver Local, puede utilizar la misma aplicación Pathfinding Sample que utilizó en la nube durante los tutoriales de [Introducción a SimSpace Weaver](#), pero esta vez en su hardware local.

## Para crear la aplicación de muestra en SimSpace Weaver Local

1. En la línea de comandos, vaya a *project-folder*\tools\local.
2. Ejecute `generate_visual_studio_project.bat`.
3. Abra *project-folder*\buildlocal\PathfindingSampleLocal.sln en Visual Studio.
4. Establezca la configuración de compilación en RelWithDebInfo.
5. Seleccione Build > Build Solution.

Visual Studio colocará tus artefactos de construcción en:

- *project-folder*\buildlocal\out\RelWithDebInfo.

Dentro de esa carpeta, debería ver los archivos ejecutables siguientes:

- PathfindingSampleLocalSpatial.exe
- PathfindingSampleLocalView.exe

## Ejecute la simulación con SimSpace Weaver Local

Puede utilizar SimSpace Weaver Local para ejecutar cualquier combinación de hasta 24 aplicaciones espaciales o personalizadas en su ordenador local. El reloj de simulación se inicia después de que se inicien todas las aplicaciones espaciales definidas en el esquema.

Para ejecutar sus aplicaciones con SimSpace Weaver Local

1. En una ventana de selección de archivos, vaya a *project-folder*\buildlocal\out\RelWithDebInfo.
2. Las aplicaciones de SimSpace Weaver Local requieren un archivo de esquema nombrado `schema.yaml` en el directorio de trabajo de las aplicaciones. Si no se lee la información requerida del esquema, se cancelan las aplicaciones.

El esquema de SimSpace Weaver Local no tiene por qué ser idéntico al de *project-folder*\tools\*project-name*-schema.yaml, pero puede usarlo como punto de partida.

Seleccione una de las siguientes opciones:

- Copie ese esquema a *project-folder*\buildlocal\out\RelWithDebInfo\schema.yaml.

- Defina la variable de entorno `WEAVERLOCAL_SCHEMA_PATH` con el nombre de un archivo de esquema con una ruta o un nombre de archivo diferentes.

### Example Ejemplo

```
set WEAVERLOCAL_SCHEMA_PATH=c:\projects\MyProject\tools\MyProject-schema.yaml
```

#### Note

Si configura la variable de entorno desde la línea de comandos, solo podrá acceder a esa variable de entorno (con ese valor) desde esa sesión de línea de comandos (ventana de consola).

3. El esquema de la aplicación de ejemplo define una cuadrícula de 2 x 2, que crea 4 particiones. Ejecutará un script que lanzará 4 instancias de la aplicación espacial para que coincidan con el número de aplicaciones espaciales especificado en el esquema. El script también abrirá una aplicación de 1 vista. Cuando se inicien todas las aplicaciones espaciales y se les asigne una partición, la simulación empezará a funcionar automáticamente.

### Para lanzar sus aplicaciones

- a. En la línea de comandos, acceda a la carpeta de herramientas local del proyecto.

```
cd project-folder\tools\local
```

- b. Ejecute el script para iniciar las aplicaciones.

```
launch_simulation_locally.bat
```

#### Note

Si utiliza `WEAVERLOCAL_SCHEMA_PATH` como nombre de un archivo de esquema, debe iniciar las aplicaciones espaciales en la línea de comandos de la misma sesión (ventana) en la que configuró la variable de entorno.

**⚠ Important**

Si aparece una ventana emergente de seguridad de Windows, seleccione Allow Access para poder conectarse a la aplicación de visualización para visualizar la simulación.

**ℹ Note**

También puede iniciar sus aplicaciones espaciales y de visualización manualmente. Para ello, debe iniciar manualmente 4 instancias de la aplicación espacial y una aplicación de visualización.

- Aplicación espacial: `start PathfindingSampleLocalSpatial.exe`
- Ver aplicación: `start PathfindingSampleLocalView.exe`

## Visualizar su simulación local

Para ver la 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 [Paso 5: Ver la simulación](#) del tutorial de inicio rápido.

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

## Detener su simulación local

La simulación local seguirá ejecutándose si las aplicaciones espaciales locales están activas. Si cierra una de las ventanas de la aplicación espacial, detendrá toda la simulación. Cierre todas las demás ventanas para limpiar el resto de la simulación.

Puede cerrar manualmente cada ventana de la aplicación o puede usar el siguiente script para cerrarlas todas automáticamente:

- `project-folder\tools\local\terminate_local_simulation.bat`

### Note

Aunque cerrar una ventana espacial de la aplicación detenga la simulación, asegúrese de cerrar las demás ventanas de la aplicación. No podrá iniciar otra simulación local correctamente si aún quedan ventanas abiertas de una simulación anterior.

## 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.

## Diferencias de desarrollo local en la versión 1.15.3

En esta sección se describen los cambios en el desarrollo con SimSpace Weaver Local a partir de la versión 1.15.3. Estos cambios afectan a los flujos de trabajo para proyectos SimSpace Weaver Local en C++, Python, Unity y Unreal Engine.

### Temas

- [Cambios en los archivos](#)
- [Actualizar un proyecto de C++ existente a SimSpace Weaver Local 1.15.3](#)
- [Ejecute un nuevo proyecto de Python en SimSpace Weaver Local 1.15.3](#)
- [Actualizar un proyecto de Unity existente a SimSpace Weaver Local 1.15.3](#)
- [Actualizar un proyecto de Unreal Engine existente a SimSpace Weaver Local 1.15.3](#)
- [Preguntas frecuentes sobre SimSpace Weaver Local 1.15.3](#)

### Cambios en los archivos

- Los archivos de biblioteca `SimSpaceWeaverAppSdkLocal` ahora tienen un nombre `weaver_app_sdk_cxx_v1_full_local`.
  - Puede encontrar estos archivos en `sdk-folder\SimSpaceWeaverAppSdk-1.15.3\lib\weaverlocal\windows`.
- Sus proyectos ya no deben incluir enlaces a `SimSpaceWeaverAppSDK-1.15.1\include\aws\weaverruntime\local_ffi`.
- Los scripts de Python para SimSpace Weaver Local ya no precisan `cmake`.
- Hay scripts de Python nuevos o renombrados para SimSpace Weaver Local:
  - `build-local`: crea la aplicación Python para su lanzamiento. Coloca todo el código Python y los archivos de biblioteca SimSpace Weaver Local en el directorio `buildlocal`.

- `local-config`: define las variables de entorno para los scripts locales.
- `start-python-locally`: configura la `PythonPath` de la aplicación e inicia Python.
- Los siguientes scripts son equivalentes a los scripts en la nube con nombres similares:
  - `quick-start-local`
  - `start-simulation-local`
  - `stop-simulation-local`

## Actualizar un proyecto de C++ existente a SimSpace Weaver Local 1.15.3

### Requisitos

- [Versión 1.15.3 del SDK de la aplicación SimSpace Weaver](#)
- Un proyecto de C++ existente para la versión de SimSpace Weaver Local 1.15.2 o anterior

### Para actualizar su proyecto

1. Asegúrese de que su proyecto incluya el encabezado necesario:

```
#include <aws/weaverruntime/ffi/weaver_app_sdk_cxx_ffi_v1/src/lib.rs.h>
```

#### Note

Por lo general, se incluye en `aws\weaverruntime\detail.h`

2. Asegúrese de que su proyecto no incluya el siguiente encabezado obsoleto:

```
#include <aws/weaverruntime/local_ffi/Bridge.h>
```

#### Note

Por lo general, se incluye en `aws\weaverruntime\detail.h`

3. En su archivo CMake o script de compilación, reemplace los nombres de las bibliotecas estáticas `SimSpaceWeaverAppSdkLocal` por el nuevo nombre de `weaver_app_sdk_cxx_v1_full_local`.

4. Si no ha ejecutado `sdk-folder\docker-create-image.bat` desde la versión 1.15.3, ejecútela ahora. Solamente es necesario hacerlo una vez.
5. Siga los [procedimientos habituales](#) para crear y ejecutar el proyecto.

## Solución de problemas

Se producen errores en el enlazador (símbolo externo no resuelto)

El compilador genera errores de enlazador como los siguientes (se añaden saltos de línea para facilitar la lectura).

```
Error    LNK2019    unresolved external symbol
           "class outcome_v2_92ee5284::basic_result<class
           Aws::WeaverRuntime::Application,
           enum Aws::WeaverRuntime::ffi::weaver_app_sdk_cxx_ffi_v1::ErrorCode
```

Para solucionar el problema

- Asegúrese de que su proyecto no incluya `<aws/weaverruntime/local_ffi/Bridge.h>`.

Aparece un error de incompatibilidad de tipos

Puede generar un error como el siguiente (se añaden saltos de línea para facilitar la lectura).

```
the object has type qualifiers that are not compatible with the member function
object type is: const rust::cxxbridge1::String
```

Para solucionar el problema

1. Actualice sus archivos de cabecera de SimSpace Weaver con las versiones 1.15.3.
2. Asegúrese de que cualquier uso de `rust::cxxbridge1::String` sea similar al siguiente:

```
rust::cxxbridge1::String domain_name = domain.name.value;
if (domain.type_ == Api::DomainType::Spatial && Name.Compare(domain_name.c_str())
    == 0)
```

## Ejecute un nuevo proyecto de Python en SimSpace Weaver Local 1.15.3

### Requisitos

- [Versión 1.15.3 del SDK de la aplicación SimSpace Weaver](#)
- Un nuevo proyecto de PythonBubblesSample de SimSpace Weaver Local versión 1.15.3

### Para ejecutar su proyecto

1. En una ventana de símbolo del sistema, vaya a `project-folder\tools\local\windows`.
2. Ejecute `quick-start-local.bat`

Deberían iniciarse localmente 4 aplicaciones espaciales, 1 aplicación de visualización y 1 cliente. El cliente debe mostrar burbujas.

### Solución de problemas

Aparece un error que dice que **GLIBCXX** no se encuentra

El siguiente error indica un fallo al importar el SDK de la aplicación SimSpace Weaver Python. Una causa probable es que las bibliotecas de C++ estén desactualizadas.

```
ImportError: /lib64/libstdc++.so.6: version `GLIBCXX_3.4.29' not found
```

### Para solucionar el problema

1. Actualice el código fuente de las bibliotecas `libstdc++` (por ejemplo, `gcc` o `msvc`) a una versión que incluya la versión especificada de `GLIBCXX` (en el ejemplo proporcionado, la versión es la 3.4.29).
2. Asegúrese de que la variable de entorno del sistema `LD_LIBRARY_PATH` esté configurada en la ruta correcta a `lib64`.

Se produce un error en el script de inicio rápido

Se produce un error en el script de inicio rápido y aparece un mensaje como el siguiente.

```
python: can't open file 'C:\usr\project\buildlocal\bIn\bubbles_tkinter_client.py':
```

```
[Errno 2] No such file or directory
```

Para solucionar el problema

1. Vaya a *project-folder*\tools\local\windows y edite local-config.bat.
2. Asegúrese de que las siguientes variables de entorno estén configuradas en las rutas correctas para su sistema local:
  - TOOLS\_DIR está configurado en su *project-folder*\tools
  - TOOLS\_DIR\_WINDOWS está configurado en su *project-folder*\tools\local\windows
  - PROJECT\_ROOT está configurado en su *project-folder*
  - BUILD\_DIR está configurado en su *project-folder*\buildlocal
  - APP\_SDK\_DIR está configurado en su *sdk-folder*

## Actualizar un proyecto de Unity existente a SimSpace Weaver Local 1.15.3

Requisitos

- [Versión 1.15.3 del SDK de la aplicación SimSpace Weaver](#)
- Un proyecto de Unity existente para la versión de SimSpace Weaver Local 1.15.2 o anterior

Para actualizar su proyecto

1. En Unity, elimine el paquete de AWS SimSpace Weaver existente.
  - a. Abra su proyecto de Unity existente.
  - b. En la ventana del editor, seleccione Ventana > Administrador de paquetes.
  - c. En Paquetes: Unity Technologies, seleccione AWS SimSpace Weaver y seleccione Eliminar.
2. En su versión *sdk-folder* 1.15.3, ejecute download-unity-package.bat
3. Siga las instrucciones de Unity\_SDK\_for\_AWS\_SimSpace\_Weaver.pdf para añadir el paquete recién descargado SimSpaceWeaverUnityPackage.zip como paquete de AWS SimSpace Weaver en el editor de Unity.

## Actualizar un proyecto de Unreal Engine existente a SimSpace Weaver Local 1.15.3

### Requisitos

- [Versión 1.15.3 del SDK de la aplicación SimSpace Weaver](#)
- Un proyecto de Unreal Engine existente para la versión de SimSpace Weaver Local 1.15.2 o anterior

### Para actualizar su proyecto

1. Cierre todas las ventanas del proyecto y del editor de código de Unreal.
2. En una ventana de línea de comandos, vaya a la *sdk-folder* versión 1.15.3.
3. Ejecute `update-unreal-project.bat --path project-folder --name project-name`.

#### Note

Esto reemplaza su complemento existente por el nuevo. Se borra cualquier modificación.

4. Siga las instrucciones de `AWS_SimSpace_Weaver_Unreal_Guide.pdf` para crear su proyecto para SimSpace Weaver Local.

### Solución de problemas

#### Error al eliminar archivos de complementos existentes

Podría obtener errores similares a los siguientes.

```
cannot remove '/usr/src/project/{PROJECT_NAME}/src/PathfindingSampleUnrealSpatial/PathfindingUnrealProject/Plugins/SimSpaceWeaverAppSdkPlugin/Binaries/Win64/UnrealEditor-WeaverAppSdk.dll': Operation not permitted
cannot remove '/usr/src/project/{PROJECT_NAME}/src/PathfindingSampleUnrealSpatial/PathfindingUnrealProject/Plugins/SimSpaceWeaverAppSdkPlugin/Binaries/Win64/UnrealEditor-WeaverAppSdkLocal.dll': Operation not permitted
cannot remove '/usr/src/project/{PROJECT_NAME}/src/PathfindingSampleUnrealSpatial/PathfindingUnrealProject/Plugins/SimSpaceWeaverAppSdkPlugin/Binaries/Win64/UnrealEditor-WeaverCppMetrics.dll': Operation not permitted
```

## Para solucionar el problema

1. Asegúrese de que la ventana del editor de proyectos de Unreal y los editores de código estén cerrados.
2. Ejecute `update-unreal-project.bat`.

## Preguntas frecuentes sobre SimSpace Weaver Local 1.15.3

- P1: ¿Cómo cambio las variables de entorno SimSpace Weaver Local para mi proyecto de Python?
  - Edite `project-folder\tools\windows\local-config` para su proyecto de Python.

## 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 [Referencias de API de SimSpace Weaver](#).

## 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

Crema 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

- [Reparar 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 entityIdsByAction;

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

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

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

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

```

```

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

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

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

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

```

```

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

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

  m_entityIdsByPreviousOwnershipAction = entityIdByOwnershipAction;

  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 distribuable 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](#).

## 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.

**⚠ Important**

Si usa AWS IAM Identity Center o perfiles nombrados para AWS Command Line Interface (AWS CLI), debe usar la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. La última versión es la 1.16.0. Para obtener información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#). Los scripts del SDK de la aplicación de SimSpace Weaver utilizan AWS CLI. Si utiliza el centro de identidad de IAM, puede copiar su perfil del centro de identidad de IAM para el AWS CLI a su perfil de default o proporcionar el nombre de su perfil del centro de identidad de IAM a los scripts del SDK de la aplicación de SimSpace Weaver con el parámetro `--profile cli-profile-name`. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

- ListSimulations: obtener una lista de todas sus simulaciones en Nube de AWS.

### Example Ejemplo

```
tools\windows\weaver-MyProject-cli.bat list-simulations
```

- DescribeSimulation: obtener detalles sobre una simulación.

## Example Ejemplo

```
tools\windows\weaver-MyProject-cli.bat describe-simulation --simulation MySimulation
```

- DescribeApp: obtener detalles sobre una aplicación.

## Example Ejemplo

```
tools\windows\weaver-MyProject-cli.bat describe-app --simulation MySimulation --  
domain MyCustomDomain --app MyCustomApp
```

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

## 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 aplicación SimSpace Weaver 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 acerca de cómo crear y ejecutar los clientes de ejemplo, consulte [Paso 5: Ver la simulación](#).

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

Las aplicaciones de AWS SimSpace Weaver se ejecutan en entornos contenerizados Amazon Linux 2 (AL2). En Nube de AWS, SimSpace Weaver ejecuta simulaciones en contenedores Docker creados a partir de una imagen de `amazonlinux:2` proporcionada por 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.

Un Dockerfile para ejecutar aplicaciones AWS SimSpace Weaver comienza con la imagen Amazon Linux 2 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 administrada a la identidad de IAM e intentarlo de nuevo. 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 sabe cómo usar AWS SimSpace Weaver y quiere que sus flujos de trabajo de almacenamiento y desarrollo de aplicaciones en Nube de AWS sean más eficientes.

### Requisitos previos

- Está modificando un proyecto de SimSpace Weaver existente creado por el script `create-project.bat`.
- 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 a la función 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 de la función de la aplicación de simulación de su proyecto:
  - i. En un editor de texto, abra la plantilla AWS CloudFormation para el proyecto:

```
project-folder\cloudformation\weaver-project-name-stack.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.

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 contenedor de SimSpace Weaver estándar.

Example Fragmento de esquema que incluye una configuración de contenedor personalizada

```

sdk_version: "1.16.0"
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 administrada a la identidad de IAM e intentarlo de nuevo. 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. Para obtener más información, consulte el [tutorial detallado](#).
- Pruebe la simulación sin un contenedor personalizado.
- Pruebe la simulación localmente. Para obtener más información, consulte [Desarrollo local](#).

## Trabajo con Python

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

### Important

SimSpace Weaver es compatible con Python versión 3.9.

### Important

La compatibilidad de SimSpace Weaver con Python requiere SimSpace Weaver, versión 1.15.0 o posterior.

## Temas

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

## Crear un proyecto de Python

Utiliza el script `create-project.bat` para crear un proyecto de Python, de la misma manera que lo haría para crear un proyecto que no sea de Python. Puede usar la plantilla `PythonBubblesSample` como punto de partida para tu proyecto de Python. Consulte [Crear un proyecto de Python](#) a continuación.

### Contenedor de Python personalizado

Para ejecutar su simulación de SimSpace Weaver 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 `create-custom-container.bat` (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 `create-custom-container.bat` 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

Debe ejecutar `create-custom-container.bat` después de cualquier cambio en `Dockerfile` o `requirements.txt`.

**⚠ 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 utiliza un contenedor personalizado que tenga Python (como la imagen del contenedor creada por el script `create-custom-container.bat`), debe incluir Python y las dependencias necesarias en cada archivo zip de aplicación que cargue a SimSpace Weaver.

## Crear un proyecto de Python

El siguiente procedimiento es para Microsoft Windows. Si utiliza el subsistema de Windows para Linux (WSL), utilice en su lugar las versiones `.sh` de los scripts `.bat`. Debe completar la configuración de Amazon Elastic Container Registry (Amazon ECR) para utilizar este procedimiento. Para obtener más información, consulte [Configuración de Amazon ECR](#) en la Guía del usuario de Amazon ECR.

Para crear un proyecto de Python

1. En una ventana de símbolo del sistema, vaya a su carpeta SDK SimSpace Weaver.

```
cd sdk-folder
```

2. Ejecute `create-project.bat` con la plantilla `PythonBubblesSample`.

```
.\create-project.bat --name project-name --path project-folder-parent-path --  
template PythonBubblesSample
```

3. Acceda a la carpeta `tools` del proyecto. La *carpeta de tu proyecto* es *project-folder-parent-path\project-name*.

```
cd project-folder\tools
```

4. Creación de un contenedor personalizado

```
.\create-custom-container.bat
```

## Inicio de una simulación de Python

Puede iniciar su simulación basada en Python de la misma manera que una simulación de SimSpace Weaver normal, SimSpace Weaver Local tanto en SimSpace Weaver como en Nube de AWS. Para obtener más información, consulte lo siguiente:

### SimSpace Weaver Local

- [Desarrollo local](#)

### Nube de AWS

- [Paso 3: Ejecutar el script de inicio rápido](#) en el tutorial de inicio rápido
- [Tutorial detallado: conozca los detalles mientras crea la aplicación de muestra](#)

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 símbolo del sistema, vaya a la carpeta `src\PythonBubblesSample\bin` de su proyecto.

```
cd project-folder\src\PythonBubblesSample\bin
```

2. Ejecute el cliente Python.

```
python bubbles_tkinter_client.py --host ip-address --port port-number --  
simsizes max-entitites
```

## 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 [consola de SimSpace Weaver](#) o seguir el procedimiento [Paso 4: obtenga su dirección IP y número de puerto](#) 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 puerto Actual. Puede encontrar el número de puerto de la simulación en la [consola de SimSpace Weaver](#) o utilizar el procedimiento [Paso 4: obtenga su dirección IP y número de puerto](#) 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.

## Escribir sus propios scripts de compilación

Puede escribir sus propios scripts de compilación para su simulación de Python. Para que la compilación se realice correctamente, se deben seguir los siguientes pasos:

1. Copie el contenido de `src/PythonBubblesSample/` en el directorio `Build/out`.
2. Copie el contenido de `#{WEAVER_SDK_DIRECTORY}/lib/weaver/weaver_python_app_sdk_v1` en el directorio `Build/out/lib/weaver_app_sdk_v1`.
3. Copie `#{WEAVER_SDK_DIRECTORY}/lib/weaver/libweaver_app_sdk_python_v1_39.so` en el directorio `Build/out/lib/weaver_app_sdk_v1`.
4. Cambie el nombre de `Build/out/lib/weaver_app_sdk_v1/libweaver_app_sdk_python_v1_39.so` a `libweaver_app_sdk_python_v1.so`.
5. Comprima en zip el contenido del directorio `Build/out/`.
6. Repita el proceso de compresión para cada aplicación comprimida especificada en el esquema de la simulación. Para el `PythonBubblesSample`, el esquema espera a `project-nameSpatial.zip` y `project-nameView.zip`.

Tras estos pasos, los archivos zip estarán listos para cargarse en el bucket de Amazon S3 del proyecto.

## Preguntas frecuentes sobre el uso de Python

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

SimSpace Weaver 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 error ModuleNotFound](#)

### Error durante la creación de un contenedor personalizado

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

```
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 admita como compatible. Para obtener más información, consulte [SimSpace Puntos finales y cuotas de Weaver](#).

Después de ejecutar el comando `aws ecr`, vuelva a ejecutar `create-custom-container.bat`.

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](#).

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

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 su `requirements.txt`. Recuerde `create-custom-container.bat` 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

Puede usar su propio motor personalizado de C++ 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 2021.3.7f1 como mínimo)
- [Unreal Engine](#) (versión 5.0 como mínimo)

## Unity

Debe tener el entorno de desarrollo Unity ya instalado antes de crear simulaciones SimSpace Weaver con Unity. Descargue el archivo separado Unity SDK para AWS SimSpace Weaver (Unity SDK) y siga las instrucciones de ese paquete.

**⚠ Important**

Debe usar la versión más reciente del SDK de la aplicación SimSpace Weaver. La última versión es la 1.16.0. Para obtener más información, consulte [Versiones de AWS SimSpace Weaver](#).

## Descarga y uso de la Unity SDK

1. En la línea de comandos de Windows, acceda a la carpeta *sdk-folder*.
2. Ejecute el script de descarga. Sustituya la *región* por la Región de AWS que empezará la simulación (por ejemplo, us-west-2).

```
.\download-unity-package.bat --region region
```

El script se descarga y se descomprime SimSpaceWeaverUnityPackage.zip en la carpeta actual.

3. Lea SimSpaceWeaverUnityPackage\Release\Documentation\Unity\_SDK\_for\_AWS\_SimSpace\_Weaver.pdf.

**⚠ Important**

Si recibes un error en Unity sobre la falta de un espacio de nombres para `JsonProperty` o `JsonAttribute`, sigue estos pasos para añadir el paquete: `Newtonsoft.Json`

1. En el editor de Unidad, seleccione Ventana > Package Manager en la barra de menús.
2. En la ventana Package Manager, pulse el botón + (más) en la parte superior de la ventana.
3. Seleccione Añadir paquete desde la URL de git.
4. Introduzca lo siguiente:

```
com.unity.nuget.newtonsoft-json
```

5. Elija Añadir.

**⚠ Important**

Unity SDK no admite perfiles con nombre para AWS Command Line Interface (AWS CLI). Si usa nuestros perfiles AWS IAM Identity Center o AWS CLI, debe copiar o cambiar el nombre del perfil designado al perfil `default` antes de usar Unity SDK. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

## Unreal Engine

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

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

```
sdk-folder\AWS_SimSpace_Weaver_Unreal_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 el SDK de la aplicación SimSpace Weaver para trabajar con instantáneas](#)
- [Utilice la consola SimSpace Weaver para trabajar con instantáneas](#)

- [Utilice la AWS CLI para trabajar con instantáneas](#)
- [Utilice las API SimSpace Weaver 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 [Prácticas recomendadas a la hora de 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 el SDK de la aplicación SimSpace Weaver para trabajar con instantáneas

Puede usar los scripts que se proporcionan en el SDK de la aplicación SimSpace Weaver (versión 1.13 como mínimo) para crear y usar instantáneas.

El SDK de la aplicación SimSpace Weaver organiza las simulaciones por proyecto. Puede iniciar varias simulaciones desde un solo proyecto. Cada una de esas simulaciones utiliza el mismo esquema y los mismos archivos .zip de la aplicación. Los scripts del SDK de la aplicación SimSpace Weaver colocan los activos de una simulación en un bucket de Amazon S3 específico en función del nombre, el número Cuenta de AWS y Región de AWS. Los scripts funcionan con archivos de instantáneas ubicados en una carpeta snapshot en la raíz de ese bucket. El URI de Amazon S3 a la carpeta snapshot tiene el siguiente formato:

```
s3://weaver-project-name-lowercase-account-number-artifacts-region/snapshot
```

### Ejemplo

- Nombre del proyecto: MyProject
- Número de Cuenta de AWS: 111122223333
- Región de AWS: us-west-2
- URI de Amazon S3 de la carpeta de instantáneas: s3://weaver-myproject-111122223333-artifacts-us-west-2/snapshot

Si desea utilizar un bucket de Amazon S3 diferente, consulte las siguientes formas alternativas de trabajar con instantáneas.

### Otras formas de trabajar con instantáneas

- [Consola de SimSpace Weaver](#)
- [AWS CLI](#)
- [API de SimSpace Weaver](#)

## Temas

- [Utilice el SDK de la aplicación SimSpace Weaver para crear una instantánea](#)
- [Utilice el SDK de la aplicación SimSpace Weaver para iniciar una simulación a partir de una instantánea](#)
- [Utilice el SDK de la aplicación SimSpace Weaver para iniciar rápidamente una simulación a partir de una instantánea](#)
- [Use el SDK de la aplicación SimSpace Weaver para enumerar las instantáneas de un proyecto](#)

## Utilice el SDK de la aplicación SimSpace Weaver para crear una instantánea

Para crear una instantánea, la simulación debe estar en el estado STARTED. La creación de la instantánea comienza cuando finaliza la marca actual. SimSpace Weaver deja de enviar graduaciones a las aplicaciones, pero el estado del reloj sigue indicando STARTED. El estado de la simulación cambia a SNAPSHOT\_IN\_PROGRESS. Una vez finalizada la instantánea, el estado de la simulación vuelve a cambiar a STARTED y las aplicaciones vuelven a recibir graduaciones.

Para crear una instantánea

1. En la línea de comandos de Windows, vaya a la carpeta de herramientas del proyecto.

```
cd project-folder\tools\windows
```

2. Si no sabe el nombre de la simulación, llame a la API `list-simulations` para ver una lista de sus recursos de simulación. Asegúrese de que el estado de la simulación sea STARTED.

```
.\weaver-project-name-cli.bat list-simulations
```

3. Ejecute el script `create-snapshot` de su proyecto.

```
.\create-snapshot-project-name.bat --simulation simulation-name
```

### Ejemplo

```
.\create-snapshot-MyProject.bat --simulation MyProjectSimulation_23-04-29_12_00_00
```

SimSpace Weaver crea el archivo de instantáneas en el bucket de artefactos de su proyecto.

## Ejemplo

- Nombre del proyecto: MyProject
- Número de Cuenta de AWS: 111122223333
- Región de AWS: us-west-2
- URI de Amazon S3 de la carpeta de instantáneas: s3://weaver-myproject-111122223333-artifacts-us-west-2/snapshot
- Nombre de la simulación: MyProjectSimulation\_23-04-29\_12\_00\_00
- Hora de la instantánea: 29 de abril de 2023 a las 15:30:27 UTC
- Nombre del archivo de la instantánea:  
MyProjectSimulation\_23-04-29\_12\_00\_00-230429-1530-27.zip
- URI de Amazon S3 del archivo de instantáneas: s3://weaver-myproject-111122223333-artifacts-us-west-2/snapshot/  
MyProjectSimulation\_23-04-29\_12\_00\_00-230429-1530-27.zip

## Utilice el SDK de la aplicación SimSpace Weaver para iniciar una simulación a partir de una instantánea

Cuando se utiliza un script del SDK de una aplicación para iniciar una simulación a partir de una instantánea, los scripts crean un nuevo nombre de simulación del mismo modo que lo hacen al iniciar una simulación sin una instantánea.

El archivo de instantáneas debe existir en la ubicación de la instantánea en Amazon S3 con el siguiente URI de Amazon S3:

```
s3://weaver-project-name-lowercase-account-number-artifacts-region/snapshot
```

Los archivos.zip de la aplicación deben estar en la misma ubicación en la que se encontraban cuando se creó la instantánea.

SimSpace Weaver crea un nuevo recurso de simulación, inicializa State Fabric con los datos de la entidad almacenados en la instantánea, inicia nuevas instancias de las mismas aplicaciones espaciales y de servicio que se estaban ejecutando cuando se creó la instantánea y pone el reloj en la casilla correspondiente. Debe iniciar las aplicaciones personalizadas por separado siguiendo el proceso normal.

El script `start-from-snapshot` es la versión instantánea del script `start-simulation`. Al igual que el guion `start-simulation`, el guion `start-from-snapshot` no pone en marcha el reloj por usted. Debe arrancar el reloj por separado.

Iniciar una simulación a partir de una instantánea

1. En la línea de comandos de Windows, vaya a la carpeta de herramientas del proyecto.

```
cd project-folder\tools\windows
```

2. Ejecute el script `start-from-snapshot`.

```
.\start-from-snapshot-project-name.bat --snapshot-s3-file snapshot-file-name
```

Ejemplo

```
.\start-from-snapshot-MyProject.bat --snapshot-s3-file  
MyProjectSimulation_23-04-29_12_00_00-230429-1530-27.zip
```

Utilice el SDK de la aplicación SimSpace Weaver para iniciar rápidamente una simulación a partir de una instantánea

Inicio rápido de una simulación a partir de una instantánea Esto es similar a un inicio rápido sin una instantánea.

El archivo de instantáneas debe existir en la ubicación de la instantánea en Amazon S3 con el siguiente URI de Amazon S3:

```
s3://weaver-project-name-lowercase-account-number-artifacts-region/snapshot
```

Los archivos.zip de la aplicación deben estar en la misma ubicación en la que se encontraban cuando se creó la instantánea.

SimSpace Weaver crea un nuevo recurso de simulación, inicializa State Fabric con los datos de la entidad almacenados en la instantánea, inicia nuevas instancias de las mismas aplicaciones espaciales y de servicio que se estaban ejecutando cuando se creó la instantánea y pone el reloj en la casilla correspondiente. Debe iniciar las aplicaciones personalizadas por separado siguiendo el proceso normal.

El script `quick-start-from-snapshot` es la versión instantánea del script `quick-start`. Al igual que el guion `quick-start`, el guion `quick-start-from-snapshot` pone en marcha el reloj por usted. También inicia la aplicación de visualización del proyecto de ejemplo de búsqueda de rutas.

Iniciar una simulación a partir de una instantánea

1. En la línea de comandos de Windows, vaya a la carpeta de herramientas del proyecto.

```
cd project-folder\tools\windows
```

2. Ejecute el script `quick-start-from-snapshot`.

```
.\quick-start-from-snapshot-project-name-cli.bat --snapshot-s3-file snapshot-file-name
```

### Ejemplo

```
.\quick-start-from-snapshot-MyProject-cli.bat --snapshot-s3-file  
MyProjectSimulation_23-04-29_12_00_00-230429-1530-27.zip
```

## Use el SDK de la aplicación SimSpace Weaver para enumerar las instantáneas de un proyecto

Puede usar el script `list-snapshots` para enumerar las instantáneas de un proyecto. Este script muestra los archivos de la carpeta `snapshot` del proyecto. Los proyectos son exclusivos del SDK de la aplicación SimSpace Weaver, por lo que solo puede hacerlo con los scripts del SDK de la aplicación y solo para los proyectos. El script asume que todos los archivos de la carpeta `snapshot` de Amazon S3 son archivos de instantáneas. Si mueve o elimina archivos de la carpeta, esos archivos no aparecerán en la lista.

Para crear una lista de las instantáneas de un proyecto

1. En la línea de comandos de Windows, vaya a la carpeta de herramientas del proyecto.

```
cd project-folder\tools\windows
```

2. Ejecute el script `list-snapshots`.

```
.\list-snapshots-project-name.bat
```

## Ejemplo

```
.\list-snapshots-MyProject.bat
```

## Utilice la consola SimSpace Weaver para trabajar con instantáneas

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

Otras formas de trabajar con instantáneas

- [Scripts del SDK de la aplicación SimSpace Weaver](#)
- [AWS CLI](#)
- [API de SimSpace Weaver](#)

## 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 AWS Management Console y conéctese a la consola de [SimSpace Weaver](#).
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 bucket o un bucket y una carpeta en los que desee que SimSpace Weaver cree 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.

**ℹ Note**

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

**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 AWS Management Console y conéctese a la consola de [SimSpace Weaver](#).
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 la AWS CLI para trabajar con instantáneas

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

Otras formas de trabajar con instantáneas

- [Scripts del SDK de la aplicación SimSpace Weaver](#)
- [Consola de SimSpace Weaver](#)
- [API de SimSpace Weaver](#)

Temas

- [Utilice el AWS CLI para crear una instantánea](#)
- [Utilice el 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 bucket. SimSpace Weaver crea la instantánea dentro de una carpeta `snapshot` 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 API de `CreateSnapshot`, consulte [CreateSnapshot](#) en la Referencia de la API de AWS SimSpace Weaver.

Utilice el 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

### nombre

El nombre de la nueva simulación. El nombre de la simulación debe ser exclusivo de 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 API de `StartSimulation`, consulte [StartSimulation](#) en la Referencia de la API de AWS SimSpace Weaver.

## Utilice las API SimSpace Weaver para trabajar con instantáneas

Puede llamar a las API SimSpace Weaver directamente para trabajar con las instantáneas. Para obtener más información sobre la API, consulte la [referencia de la API de AWS SimSpace Weaver](#).

Otras formas de trabajar con instantáneas

- [Scripts del SDK de la aplicación SimSpace Weaver](#)
- [Consola de SimSpace Weaver](#)
- [AWS CLI](#)

## Crear una instantánea

Puede llamar a la API `CreateSnapshot` para crear una instantánea para una simulación. El estado de la simulación debe ser `STARTED`. SimSpace Weaver crea el archivo de instantáneas en una carpeta `snapshot` del bucket de Amazon S3 y el prefijo de objeto que especifique. Para obtener más información, consulte [CreateSnapshot](#) en la Referencia de la API de AWS SimSpace Weaver.

## Iniciar una simulación a partir de una instantánea

Puede proporcionar una instantánea cuando llame a la API `StartSimulation` para iniciar una nueva simulación. Debe proporcionar una cadena JSON como argumento del parámetro `SnapshotS3Location`. La cadena especifica el nombre del bucket de Amazon S3 y la clave de objeto del archivo de instantánea. Si proporciona un `SnapshotS3Location`, no podrá proporcionar un `SchemaS3Location`. Para obtener más información, consulte [StartSimulation](#) en la Referencia de la API de AWS SimSpace Weaver.

## 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. Cree 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.

# Prácticas recomendadas a la hora de trabajar con SimSpace Weaver

Recomendamos las siguientes prácticas cuando trabaje con la SimSpace Weaver.

## Temas

- [Configurar alarmas de facturación](#)
- [Utilizar SimSpace Weaver Local](#)
- [Puede detener las simulaciones que no necesite](#)
- [Elimine los recursos que no necesite](#)
- [Disponer de copias de seguridad](#)

## Configurar alarmas de facturación

Es fácil aprovisionar los recursos en 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:

- [Crear una alarma de facturación para monitorizar los cargos estimados de AWS](#)
- [¿Qué es AWS Cost Management?](#)

## Utilizar SimSpace Weaver Local

Le recomendamos que utilice SimSpace Weaver Local para desarrollar y probar sus simulaciones antes de cargarlas al servicio de SimSpace Weaver en el Nube de AWS. Las ventajas 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

- Puede modificar, reconstruir y reiniciar 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 en SimSpace Weaver también crea recursos en otros servicios de AWS. 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 servicios de AWS. 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. Para obtener más información, consulte [Paso 12: Detenga y limpie la simulación](#) en el tutorial de inicio rápido.

## Disponer de copias de seguridad

Es una buena idea disponer de copias de seguridad y planes de respaldo para todo. No debe dar por sentado que no tiene que hacer copias de seguridad por el hecho de que sus datos estén en

AWS. Debe crear su propio sistema si necesita hacer una copia de seguridad del estado de la simulación. Considere usar varios Regiones de AWS y 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 compatible con SimSpace Weaver, consulte [SimSpace Puntos finales y cuotas de Weaver](#).

# Seguridad en AWS SimSpace Weaver

En AWS, la seguridad en la nube es la máxima prioridad. Como cliente de AWS, se beneficiará de una arquitectura de red y de centros de datos diseñados para satisfacer los requisitos de seguridad de las organizaciones más exigentes.

La seguridad es una responsabilidad compartida entre AWS 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 servicios de AWS 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 [Programas de conformidad de AWS](#) . Para obtener información sobre los programas de conformidad que se aplican a AWS SimSpace Weaver, consulte [Servicios de AWS en el ámbito del programa de conformidad](#).
- Seguridad en la nube: su responsabilidad viene determinada por el servicio de AWS que utilice. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y la normativa aplicables.

Esta documentación le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida a la hora de utilizar SimSpace Weaver. Los siguientes temas le mostrarán cómo configurar SimSpace Weaver para satisfacer sus objetivos de seguridad y de conformidad. También puede aprender a utilizar otros servicios de AWS que lo ayuden a monitorizar y proteger los recursos de SimSpace Weaver.

## Temas

- [Protección de los datos en AWS SimSpace Weaver](#)
- [Administración de identidades y accesos para AWS SimSpace Weaver](#)
- [Registro y supervisión de eventos de seguridad en AWS SimSpace Weaver](#)
- [Validación de la conformidad para AWS SimSpace Weaver](#)
- [Resiliencia en AWS SimSpace Weaver](#)
- [Seguridad de infraestructuras en AWS SimSpace Weaver](#)
- [Configuración y análisis de vulnerabilidades en AWS SimSpace Weaver](#)
- [Prácticas recomendadas de seguridad para SimSpace Weaver](#)

# Protección de los datos en AWS SimSpace Weaver

El [modelo de responsabilidad compartida](#) de AWS se aplica a la protección de datos de AWS SimSpace Weaver. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global que ejecuta toda la 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 las [Preguntas frecuentes sobre la privacidad de datos](#). Para obtener información sobre la protección de datos en Europa, consulte la publicación de blog [AWS Shared Responsibility Model and GDPR](#) en el Blog de seguridad de AWS.

Con fines de protección de datos, recomendamos proteger las credenciales de la cuenta de Cuenta de AWS y configurar cuentas de usuario 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 autenticación multifactor (MFA) en cada cuenta.
- Utilice SSL/TLS para comunicarse con los recursos de AWS. Se recomienda el uso de TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad del usuario con AWS CloudTrail.
- Utilice las soluciones de cifrado de AWS, junto con todos los controles de seguridad predeterminados dentro de los 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 módulos criptográficos validados FIPS 140-2 al acceder a AWS a través de una interfaz de la línea de comandos o una API, utilice un punto de conexión de FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte [Estándar de procesamiento de la información federal \(FIPS\) 140-2](#).

Se recomienda encarecidamente no ingresar 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 las situaciones en las que debe trabajar con la SimSpace Weaver u otros Servicios de AWS a través de la consola, la API, la AWS CLI o los SDK de AWS. 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, recomendamos encarecidamente que no incluya la información de las credenciales en la URL para validar la solicitud para 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 se utiliza 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

El resto de los datos que SimSpace Weaver utiliza internamente no se conservan después de detener la simulación.

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

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

## Cifrado en tránsito

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

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

### Important

Las comunicaciones entre sus aplicaciones y sus clientes no implican a 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 los servicios de AWS compatibles con sus soluciones de cifrado, consulte [el blog de seguridad de AWS](#).

## Privacidad del tráfico entre redes

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

## Administración de identidades y accesos para AWS SimSpace Weaver

AWS Identity and Access Management (IAM) es un servicio de AWS que ayuda a los administradores a controlar de forma segura el acceso a los recursos de AWS. Los administradores de IAM controlan quién está autenticado (ha iniciado sesión) y autorizado (tiene permisos) para utilizar recursos de SimSpace Weaver. IAM es un servicio de Servicio de AWS que se puede utilizar sin cargo adicional.

### Temas

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

## Público

La forma en que utilice AWS Identity and Access Management (IAM) difiere en función del trabajo que realice en SimSpace Weaver.

Usuario de servicio: si utiliza el servicio de SimSpace Weaver para realizar su trabajo, su administrador le proporciona las credenciales y los permisos que necesita. A medida que utilice más características de SimSpace Weaver 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 de identidades de AWS SimSpace Weaver y accesos](#).

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

Administrador de IAM: si es un administrador de IAM, es posible que quiera conocer más detalles sobre cómo escribir políticas para administrar el acceso a SimSpace Weaver. Para consultar ejemplos de políticas basadas en identidad de SimSpace Weaver que puede utilizar en IAM, consulte [Ejemplos de políticas basadas en identidades de AWS SimSpace Weaver](#).

## Autenticación con identidades

La autenticación es la manera de iniciar sesión en AWS mediante credenciales de identidad. Debe estar autenticado (haber iniciado sesión en AWS) como Usuario raíz de la cuenta de AWS, como un usuario de IAM o asumiendo un rol de IAM.

Puede iniciar sesión en AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad de AWS IAM Identity Center. Los usuarios (del 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, su administrador habrá configurado previamente la federación de identidades mediante roles de IAM. Cuando accede a AWS mediante la federación, está asumiendo un rol de forma indirecta.

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

Si accede a AWS mediante programación, AWS proporciona un kit de desarrollo de software (SDK) y una interfaz de la línea de comandos (CLI) para firmar criptográficamente las solicitudes mediante el uso de las credenciales. Si no utiliza las herramientas de AWS, debe firmar usted mismo las solicitudes. Para obtener más información sobre la firma de solicitudes, consulte [Firma de solicitudes API de AWS](#) en la Guía del usuario de IAM.

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

## Usuario raíz de Cuenta de AWS

Cuando se crea una Cuenta de AWS, se comienza con una identidad de inicio de sesión que tiene acceso completo a todos los recursos y Servicios de AWS de la cuenta. Esta identidad recibe el nombre de usuario raíz de la Cuenta de AWS y se accede a ella iniciando sesión con el email y la contraseña que utilizó 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 las tareas que requieren que inicie sesión como usuario raíz, consulte [Tareas que requieren credenciales de usuario raíz](#) en la Guía del usuario de IAM.

## Identidad federada

Como práctica recomendada, solicite que los usuarios humanos, incluidos los que requieren acceso de administrador, utilicen la federación con un proveedor de identidades para acceder a Servicios de AWS utilizando credenciales temporales.

Una identidad federada es un usuario del directorio de usuarios de su empresa, un proveedor de identidad web, el AWS Directory Service, el directorio del Identity Center, o cualquier usuario que acceda a Servicios de AWS utilizando credenciales proporcionadas a través de una fuente de identidad. Cuando identidades federadas acceden a las Cuentas de AWS, asumen roles y los roles proporcionan credenciales temporales.

Para una administración de acceso centralizada, le recomendamos que utilice AWS IAM Identity Center. Puede crear usuarios y grupos en el IAM Identity Center o puede conectarse y sincronizar con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todas sus

aplicaciones y Cuentas de AWS. Para más información, consulte [¿Qué es IAM Identity Center?](#) en la Guía del usuario de AWS IAM Identity Center.

## Usuarios y grupos de IAM

Un [usuario de IAM](#) es una identidad en su Cuenta de AWS que dispone de permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos emplear credenciales temporales, en lugar de crear usuarios de IAM que tengan credenciales de larga duración como contraseñas y claves de acceso. No obstante, si tiene casos de uso específicos que requieran credenciales de larga duración con usuarios de IAM, recomendamos rotar las claves de acceso. Para más información, consulte [Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración](#) en la Guía del usuario de IAM.

Un [grupo de IAM](#) es una identidad que especifica un conjunto de usuarios de IAM. 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 de grandes conjuntos de usuarios. Por ejemplo, podría tener un grupo cuyo nombre fuese IAMAdmins y conceder permisos a dicho grupo para administrar los recursos de IAM.

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 permanentes a largo plazo y los roles proporcionan credenciales temporales. Para más información, consulte [Cuándo crear un usuario de IAM \(en lugar de un rol\)](#) en la Guía del usuario de IAM.

## Roles de IAM

Un [rol de IAM](#) es una identidad de tu Cuenta de AWS que dispone de permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una determinada persona. Puede asumir temporalmente un rol de IAM en la AWS Management Console [cambiando de roles](#). Puede asumir un rol llamando a una operación de AWS CLI o de la API de AWS, o utilizando una URL personalizada. Para más información sobre los métodos para el uso de roles, consulte [Uso de roles de IAM](#) en la Guía del usuario de IAM.

Los roles de IAM 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 acerca de roles para federación,

consulte [Creación de un rol para un proveedor de identidades de terceros](#) en la Guía del usuario de IAM. Si utiliza el IAM Identity Center, debe configurar un conjunto de permisos. El Centro de identidades de IAM correlaciona el conjunto de permisos con un rol en IAM para controlar a qué pueden acceder sus identidades después de autenticarse. Para obtener información acerca de los conjuntos de permisos, consulte [Conjuntos de permisos](#) en la Guía del usuario de AWS IAM Identity Center.

- **Permisos de usuario de IAM temporales:** un usuario de IAM puede asumir un rol de IAM para recibir temporalmente permisos distintos que le permitan realizar una tarea concreta.
- **Acceso entre cuentas:** puede utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta acceda a los recursos de la cuenta. Los roles son la forma principal de conceder acceso entre cuentas. No obstante, con algunos Servicios de AWS se puede asociar una política directamente a un recurso (en lugar de utilizar un rol como representante). Para obtener información acerca de la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte [Cómo los roles de IAM difieren de las políticas basadas en recursos](#) en la Guía del usuario de IAM.
- **Acceso entre servicios:** algunos Servicios de AWS utilizan características de otros Servicios de AWS. Por ejemplo, cuando realiza una llamada en un servicio, es común 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 a servicios.
- **Reenviar sesiones de acceso (FAS):** cuando utiliza un rol o un usuario de IAM para llevar a cabo acciones en AWS, se le considera una entidad principal. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos de la entidad principal para llamar a un Servicio de AWS, combinados con el Servicio de AWS solicitante para realizar solicitudes a servicios posteriores. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS o recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte [Reenviar sesiones de acceso](#).
- **Rol de servicio:** un rol de servicio es un [rol de IAM](#) que adopta un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para más información, consulte [Creación de un rol para delegar permisos a un Servicio de AWS](#) en la Guía del usuario de IAM.
- **Rol vinculado al servicio:** un rol vinculado al 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 a servicios aparecen en la Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

- Aplicaciones que se ejecutan en Amazon EC2: puede utilizar un rol de IAM que le permita administrar credenciales temporales para las aplicaciones que se ejecutan en una instancia de EC2 y realizan solicitudes a la AWS CLI o a la API de AWS. Es preferible hacerlo de este modo a almacenar claves de acceso en la instancia EC2. Para asignar un rol de AWS a una instancia de EC2 y ponerla a disposición de todas las aplicaciones, cree un perfil de instancia asociado a la instancia. Un perfil de instancia contiene el rol y permite a los programas que se ejecutan en la instancia EC2 obtener credenciales temporales. Para más información, consulte [Uso de un rol de IAM para conceder permisos a aplicaciones que se ejecutan en instancias Amazon EC2](#) en la Guía del usuario de IAM.

Para obtener información sobre el uso de los roles de IAM, consulte [Cuándo crear un rol de IAM \(en lugar de un usuario\)](#) en la Guía del usuario de IAM.

## Administración de acceso mediante políticas

Para controlar el acceso en AWS, se crean políticas y se adjuntan a identidades o recursos de AWS. Una política es un objeto de AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando una entidad principal (sesión de rol, usuario o usuario raíz) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan en AWS como documentos JSON. Para obtener más información sobre la estructura y el contenido de los documentos de política JSON, consulte [Información general de políticas JSON](#) en la Guía del usuario de IAM.

Los administradores pueden utilizar las políticas JSON de AWS 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 permiso a los usuarios para realizar acciones en los recursos que necesiten, un administrador puede crear políticas de IAM. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Las políticas de IAM definen 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 dicha política puede obtener información del rol de la AWS Management Console, la AWS CLI o la API de AWS.

## Políticas basadas en identidades

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, 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 más información sobre cómo crear una política basada en identidad, consulte [Creación de políticas de IAM](#) en la Guía del usuario de 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 asociar a varios usuarios, grupos y roles de su Cuenta de AWS. Las políticas administradas incluyen las políticas administradas de AWS y las políticas administradas por el cliente. Para más información sobre cómo elegir una política administrada o una política insertada, consulte [Elegir entre políticas administradas y políticas insertadas](#) en la Guía del usuario de IAM.

## Políticas basadas en recursos

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM 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. Las entidades 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 se puede utilizar políticas de IAM administradas de AWS en una política basada en recursos.

## Listas de control de acceso (ACL)

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3, AWS WAF y Amazon VPC son ejemplos de servicios que admiten las ACL. Para obtener más información sobre las ACL, consulte [Información general de Lista de control de acceso \(ACL\)](#) en la Guía para desarrolladores de Amazon Simple Storage Service.

## Otros tipos de políticas

AWS admite otros tipos de políticas adicionales menos frecuentes. 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 característica avanzada que le permite establecer los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM (usuario o rol de IAM). 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 especifique 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 [Límites de permisos para las entidades de IAM](#) en la Guía del usuario de IAM.
- **Políticas de control de servicio (SCP):** las SCP son políticas de JSON que especifican los permisos máximos de una empresa o una unidad organizativa en AWS Organizations. AWS Organizations es un servicio que le permite agrupar y administrar de manera centralizada varias Cuentas de AWS que posea su empresa. Si habilita todas las características en una empresa, entonces podrá aplicar políticas de control de servicio (SCP) a una o todas sus cuentas. Una SCP limita los permisos para las entidades de las cuentas de miembros, incluido cada Usuario raíz de la cuenta de AWS. Para más información sobre Organizations y las SCP, consulte [Funcionamiento de las SCP](#) en la Guía del usuario de AWS Organizations.
- **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 más información, consulte [Políticas de sesión](#) en la Guía del usuario de IAM.

## Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para obtener información acerca de cómo AWS decide si permitir o no una solicitud cuando hay varios tipos de políticas implicados, consulte [Lógica de evaluación de políticas](#) en la Guía del usuario de IAM.

## Cómo funciona AWS SimSpace Weaver con IAM

Antes de utilizar IAM para administrar el acceso a SimSpace Weaver, conozca qué características de IAM se pueden utilizar con SimSpace Weaver.

### Características de IAM que puede utilizar con AWS SimSpace Weaver

Característica de IAM	Compatibilidad con SimSpace Weaver
<a href="#">Políticas basadas en identidad</a>	Sí
<a href="#">Políticas basadas en recursos</a>	No
<a href="#">Acciones de políticas</a>	Sí
<a href="#">Recursos de políticas</a>	Sí
<a href="#">Claves de condición de política (específicas del servicio)</a>	Sí
<a href="#">ACL</a>	No
<a href="#">ABAC (etiquetas en políticas)</a>	Sí
<a href="#">Credenciales temporales</a>	Sí
<a href="#">Permisos de entidades principales</a>	Sí
<a href="#">Roles de servicio</a>	Sí
<a href="#">Roles vinculados al servicio</a>	No

Para obtener una perspectiva general sobre cómo funcionan SimSpace Weaver y otros servicios de AWS con las características de IAM, consulte [Servicios de AWS que funcionan con IAM](#) en la Guía del usuario de IAM.

## Políticas de SimSpace Weaver basadas en identidades

Compatibilidad con las políticas basadas en identidades      Sí

Las políticas basadas en identidades son documentos de políticas de permisos JSON que puede adjuntar a una identidad, como un usuario, un grupo de usuarios o un rol de IAM. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte [Creación de políticas de IAM](#) en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, 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 más información sobre los elementos que puede utilizar en una política de JSON, consulte [Referencia de los elementos de las políticas de JSON de IAM](#) en la Guía del usuario de IAM.

### Ejemplos de políticas basadas en identidades de SimSpace Weaver

Para ver ejemplos de políticas basadas en identidad de SimSpace Weaver, consulte [Ejemplos de políticas basadas en identidades de AWS SimSpace Weaver](#).

## Políticas basadas en recursos de SimSpace Weaver

Compatibilidad con las políticas basadas en recursos      No

Las políticas basadas en recursos son documentos de políticas JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM 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. Las entidades principales pueden incluir cuentas, usuarios, roles, usuarios federados o servicios de Servicios de AWS.

Para habilitar el acceso entre cuentas, puede especificar toda una cuenta o entidades de IAM de otra cuenta como la entidad principal de una política en función de 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 la entidad principal y el recurso se encuentran en Cuentas de AWS diferentes, un administrador de IAM de la cuenta de confianza también debe conceder a la entidad principal (usuario o rol) permiso 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 más información, consulte [Cómo los roles de IAM difieren de las políticas basadas en recursos](#) en la Guía del usuario de IAM.

## Acciones de política para SimSpace Weaver

Admite acciones de políticas

Sí

Los administradores pueden utilizar las políticas JSON de AWS para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Las acciones de la política generalmente tienen el mismo nombre que la operación de API AWS asociada. Hay algunas excepciones, como acciones de solo permiso que no tienen una operación de API coincidente. 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 las acciones de SimSpace Weaver, consulte [Acciones definidas por AWS SimSpace Weaver](#) en la Referencia de autorizaciones de servicio.

Las acciones de políticas de 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 basadas en identidad de SimSpace Weaver, consulte [Ejemplos de políticas basadas en identidades de AWS SimSpace Weaver](#).

## Recursos de políticas para SimSpace Weaver

Admite recursos de políticas	Sí
------------------------------	----

Los administradores pueden utilizar las políticas JSON de AWS 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` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso utilizando el [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 recursos de SimSpace Weaver y sus ARN, consulte [Recursos definidos por AWS SimSpace Weaver](#) en la Referencia de autorizaciones de servicio. Para obtener información sobre las acciones con las que puede especificar el ARN de cada recurso, consulte [Acciones definidas por AWS SimSpace Weaver](#).

Para ver ejemplos de políticas basadas en identidad de SimSpace Weaver, consulte [Ejemplos de políticas basadas en identidades de AWS SimSpace Weaver](#).

## Claves de condición de políticas para SimSpace Weaver

Admite claves de condición de políticas específicas del servicio	Sí
--	----

Los administradores pueden utilizar las políticas JSON de AWS 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 lógica AND. Si especifica varios valores para una única clave de condición, AWS evalúa la condición con una operación OR 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 un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM. Para más información, consulte [Elementos de la política de IAM: variables y etiquetas](#) en la Guía del usuario de IAM.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición globales de AWS, consulte [Claves de contexto de condición globales de AWS](#) en la Guía del usuario de IAM.

Para ver una lista de las claves de condición de SimSpace Weaver, consulte [Claves de condición para AWS SimSpace Weaver](#) en la Referencia de autorizaciones de servicio. Para obtener más información sobre las acciones y los recursos con los que puede utilizar una clave de condición, consulte [Acciones definidas por AWS SimSpace Weaver](#).

Para ver ejemplos de políticas basadas en identidad de SimSpace Weaver, consulte [Ejemplos de políticas basadas en identidades de AWS SimSpace Weaver](#).

## Listas de control de acceso (ACL) en SimSpace Weaver

Admite las ACL	No
----------------	----

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

## Control de acceso basado en atributos (ABAC) con SimSpace Weaver

Admite ABAC (etiquetas en las políticas)	Sí
--	----

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos en función de atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a entidades de IAM (usuarios o roles) y a muchos recursos de AWS. El etiquetado de entidades y recursos es el primer paso de ABAC. A continuación, designa las políticas de ABAC para permitir operaciones cuando la etiqueta de la entidad principal coincida con la etiqueta del recurso al que se intenta acceder.

ABAC es útil en entornos que crecen con rapidez y ayuda en situaciones en las que la administración de las políticas resulta 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 sobre ABAC, consulte [¿Qué es ABAC?](#) en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulte [Uso del control de acceso basado en atributos \(ABAC\)](#) en la Guía del usuario de IAM.

## Uso de credenciales temporales con SimSpace Weaver

Admite el uso de credenciales temporales	Sí
--	----

Algunos Servicios de AWS no funcionan cuando inicia sesión con credenciales temporales. Para obtener información adicional, incluida la información sobre qué servicios de Servicios de

AWS funcionan con credenciales temporales, consulte [Servicios de Servicios de AWS que funcionan con IAM](#) en la Guía del usuario de IAM.

Utilice credenciales temporales si inicia sesión en la AWS Management Console con cualquier método excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accede a AWS utilizando el enlace de inicio de sesión único (SSO) de la empresa, ese proceso crea automáticamente credenciales temporales. También crea automáticamente credenciales temporales cuando inicia sesión en la consola como usuario y luego cambia de rol. Para más información sobre el cambio de roles, consulte [Cambio a un rol \(consola\)](#) en la Guía del usuario de IAM.

Puede crear credenciales temporales de forma manual mediante la AWS CLI o la API de AWS. A continuación, puede usar esas credenciales temporales para acceder a AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de usar claves de acceso a largo plazo. Para más información, consulte [Credenciales de seguridad temporales en IAM](#).

## Permisos de entidades principales entre servicios de SimSpace Weaver

Admite Forward access sessions (FAS)	Sí
--------------------------------------	----

Cuando utiliza un usuario o un rol de IAM para llevar a cabo acciones en AWS, se lo considera una entidad principal. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos de la entidad principal para llamar a un Servicio de AWS, combinados con el Servicio de AWS solicitante para realizar solicitudes a servicios posteriores. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS o recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información detallada sobre las políticas a la hora de realizar solicitudes de FAS, consulte [Forward access sessions](#).

## Roles de servicio para SimSpace Weaver

Compatible con roles de servicio	Sí
----------------------------------	----

Un rol de servicio es un [rol de IAM](#) que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para más información, consulte [Creación de un rol para delegar permisos a un Servicio de AWS](#) en la Guía del usuario de IAM.

**⚠ Warning**

Cambiar los permisos de un rol de servicio podría interrumpir la funcionalidad de SimSpace Weaver. Edite los roles de servicio sólo cuando SimSpace Weaver proporcione orientación para hacerlo.

Los scripts del SDK de la aplicación SimSpace Weaver utilizan una plantilla de AWS CloudFormation para crear recursos en otros servicios de AWS que sirvan de apoyo a la simulación. Uno de estos recursos es la función de la aplicación para su 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](#).

## Roles vinculados a servicios de SimSpace Weaver

Compatible con roles vinculados al servicio	No
---	----

Un rol vinculado al 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 a servicios aparecen en la Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

Para más información sobre cómo crear o administrar roles vinculados a servicios, consulte [Servicios de AWS que funcionan con IAM](#). Busque un servicio en la tabla que incluya Yes en la columna Rol vinculado a servicios. Seleccione el vínculo Sí para ver la documentación acerca del rol vinculado a servicios para ese servicio.

## Ejemplos de políticas basadas en identidades de 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 mediante la AWS Management Console, la AWS Command Line Interface (AWS CLI) o la API de AWS. Para conceder permiso a los usuarios para realizar acciones en los recursos que necesiten, un administrador de IAM puede crear políticas de IAM. A continuación, el administrador puede agregar las políticas de IAM a los roles y los usuarios pueden asumirlos.

Para obtener información acerca de cómo crear una política basada en identidades de IAM mediante el uso de estos documentos de políticas JSON de ejemplo, consulte [Creación de políticas de IAM](#) en la Guía del usuario de IAM.

Para obtener más información sobre las acciones y los tipos de recursos definidos por SimSpace Weaver, incluido el formato de los ARN para cada tipo de recurso, consulte [Acciones, recursos y claves de condición para AWS SimSpace Weaver](#) en la Referencia de autorizaciones de servicio.

## Temas

- [Prácticas recomendadas sobre las políticas](#)
- [Mediante la consola de SimSpace Weaver](#)
- [Cómo permitir a los usuarios consultar sus propios permisos](#)
- [Permitir a los usuarios crear y ejecutar simulaciones](#)

## Prácticas recomendadas sobre las políticas

Las políticas basadas en identidades determinan si alguien puede crear, acceder o eliminar los recursos de SimSpace Weaver de la cuenta. Estas acciones pueden generar costes adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas administradas de AWS y continúe con los permisos de privilegio mínimo: a fin de comenzar a conceder permisos a los usuarios y las cargas de trabajo, utilice las políticas administradas de AWS, que conceden permisos para muchos casos de uso comunes. Están disponibles en su Cuenta de AWS. Se recomienda definir políticas administradas por el cliente de AWS específicas para los casos de uso a fin de reducir aún más los permisos. Con el fin de obtener más información, consulte las [políticas administradas por AWS](#) o las [políticas administradas por AWS para funciones de trabajo](#) en la Guía del usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, 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. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte [Políticas y permisos en IAM](#) en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben

enviarse utilizando SSL. También puede usar condiciones para conceder acceso a acciones de servicios si se emplean a través de un Servicio de AWS determinado, como por ejemplo AWS CloudFormation. Para más información, consulte [Elementos de la política JSON de IAM: condición](#) en la Guía del usuario de IAM.

- Utilice el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte la [política de validación del Analizador de acceso de IAM](#) en la Guía del usuario de IAM.
- Solicite la autenticación multifactor (MFA): si se encuentra en una situación en la que necesita usuarios raíz o de IAM en su Cuenta de AWS, active la MFA para mayor seguridad. Para solicitar la MFA cuando se invocan las operaciones de la API, agregue las condiciones de la MFA a sus políticas. Para más información, consulte [Configuración de acceso a una API protegida por MFA](#) en la Guía del usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte las [Prácticas recomendadas de seguridad en IAM](#) en la Guía del usuario de IAM.

## Mediante la consola de SimSpace Weaver

Para acceder a la consola de AWS SimSpace Weaver, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y consultar los detalles sobre los recursos de SimSpace Weaver en su 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 conceda permisos mínimos para la consola a los usuarios que solo realizan llamadas a la AWS CLI o a la API de AWS. En su lugar, permite acceso únicamente a las acciones que coincidan con la operación de API que intentan realizar.

Para asegurarse de que los usuarios y los roles puedan seguir utilizando la consola de SimSpace Weaver, asocie también a las entidades la política administrada por SimSpace Weaver *ConsoleAccess* o *ReadOnly* AWS. Para más información, consulte [Adición de permisos a un usuario](#) en la Guía del usuario de IAM:

## Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas administradas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para realizar esta acción en la consola o mediante programación con la AWS CLI o la API de AWS.

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

## Permitir a los usuarios crear y ejecutar simulaciones

Este ejemplo de política de IAM 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 proyecto de SimSpace Weaver, el servicio creará un rol (de IAM) de AWS Identity and Access Management con el nombre `weaver-project-name-app-role` y una política de confianza de IAM. La política de confianza de 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 siguiente ejemplo.

```

{
  "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 entre servicios puede dar lugar al problema de la sustitución confusa. 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 global [aws:SourceArn](#) y [aws:SourceAccount](#) en las políticas de recursos para limitar los permisos que AWS SimSpace Weaver concede a otro servicio para el recurso. Si el valor de `aws:SourceArn` no contiene el ID de cuenta, como un nombre de recurso de Amazon (ARN) de bucket de Amazon S3, 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 `aws:SourceArn` debe usar el ARN de la extensión.

La forma más eficaz de protegerse contra el problema de la sustitución confusa es utilizar la clave de contexto de condición global de `aws:SourceArn` con el ARN completo del recurso. Si no conoce el ARN completo de la extensión o si está especificando varias extensiones, utilice la clave de condición de contexto global de `aws:SourceArn` con comodines (\*) para las partes desconocidas del ARN. Por ejemplo, `arn:aws:simspaceweaver:*:111122223333:*`.

El siguiente ejemplo muestra cómo se pueden utilizar las claves contextuales de condición global `aws:SourceArn` y `aws:SourceAccount` en SimSpace Weaver para evitar el problema del suplente confundido. Esta política solo permitirá a SimSpace Weaver asumir la función cuando la solicitud provenga de la cuenta de origen especificada y se suministre con el ARN especificado. *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 su `aws:SourceArn` incluye explícitamente un número de cuenta, puede omitir la prueba de elementos de `Condition` para `aws:SourceAccount` (consulte la [Guía del usuario de IAM](#) 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 de identidades de AWS SimSpace Weaver y accesos

Utilice la siguiente información para diagnosticar y solucionar los problemas comunes que puedan surgir cuando trabaje con SimSpace Weaver e IAM.

### Temas

- [No tengo autorización para realizar una acción en SimSpace Weaver](#)
- [No estoy autorizado a realizar tareas como: PassRole](#)
- [Quiero ver mis claves de acceso](#)
- [Soy administrador y deseo permitir que otros obtengan acceso a SimSpace Weaver](#)
- [Quiero permitir a personas externas a mi Cuenta de AWS el acceso a mis recursos de SimSpace Weaver](#)

## No tengo autorización para realizar una acción en SimSpace Weaver

Si la AWS Management Console le indica que no está autorizado para llevar a cabo una acción, debe ponerse en contacto con su administrador para recibir ayuda. Su administrador es la persona que le facilitó su nombre de usuario y contraseña.

En el siguiente ejemplo, el error se produce cuando el usuario de IAM `mateojackson` intenta utilizar la consola para consultar los detalles acerca de un recurso ficticio `my-example-widget`, pero no tiene los permisos ficticios `simspaceweaver:GetWidget`.

```
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 tareas como: 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 le permiten transferir un rol existente a dicho servicio en lugar de crear un nuevo rol de servicio o uno vinculado al servicio. Para ello, debe tener permisos para transferir el rol al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en 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 su administrador de AWS. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

## Quiero ver mis claves de acceso

Después de crear sus claves de acceso de usuario de IAM, puede 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](#). Si lo hace, podría conceder a otra persona acceso permanente a su 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 pierde la clave de acceso secreta, debe agregar nuevas claves de acceso a su usuario de IAM. 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 consultar las instrucciones, consulte [Administración de claves de acceso](#) en la Guía del usuario de IAM.

## Soy administrador y deseo permitir que otros obtengan acceso a SimSpace Weaver

Para permitir que otros obtengan acceso a SimSpace Weaver, debe crear una entidad de IAM (usuario o rol) para la persona o la aplicación que necesita acceso. Esta persona utilizará las credenciales de la entidad para acceder a AWS. A continuación, debe asociar una política a la entidad que le conceda los permisos correctos en SimSpace Weaver.

Para comenzar de inmediato, consulte [Creación del primer grupo y usuario delegado de IAM](#) en la Guía del usuario de IAM.

## Quiero permitir a personas externas a mi Cuenta de AWS el acceso a mis recursos de SimSpace Weaver

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 admitan las políticas basadas en recursos o las listas de control de acceso (ACL), puede utilizar dichas políticas para conceder a las personas acceso a sus recursos.

Para más información, consulte lo siguiente:

- Para obtener información acerca de si SimSpace Weaver admite estas características, consulte [Cómo funciona AWS SimSpace Weaver con IAM](#).
- Para obtener información acerca de cómo proporcionar acceso a los recursos de las Cuentas de AWS de su propiedad, consulte [Proporcionar acceso a un usuario de IAM a otra Cuenta de AWS de la que es propietario](#) en la Guía del usuario de IAM.
- Para obtener información acerca de cómo proporcionar acceso a los recursos a Cuentas de AWS de terceros, consulte [Proporcionar acceso a Cuentas de AWS que son propiedad de terceros](#) en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulte [Proporcionar acceso a usuarios autenticados externamente \(federación de identidades\)](#) en la Guía del usuario de IAM.
- Para obtener información sobre la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte [Cómo los roles de IAM difieren de las políticas basadas en recursos](#) en la Guía del usuario de IAM.

# Registro y supervisión de eventos de seguridad en AWS SimSpace Weaver

La supervisión es un aspecto importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de SimSpace Weaver y sus soluciones de AWS. Debe recopilar datos de monitorización de todas las partes de su solución de AWS para que le resulte más sencillo depurar un error que se produce en distintas partes del código, en caso de que ocurra.

AWS y SimSpace Weaver proporcionan varias herramientas para supervisar sus 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 tema de Amazon SNS 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 a la API de AWS SimSpace Weaver mediante AWS CloudTrail](#).

# Validación de la conformidad para AWS SimSpace Weaver

SimSpace Weaver no está en el ámbito de los programas de conformidad de AWS.

Los auditores externos evalúan la seguridad y la conformidad de otros servicios de AWS en distintos programas de conformidad de AWS. Estos incluyen SOC, PCI, FedRAMP, HIPAA y otros.

Para saber si un servicio de Servicio de AWS está incluido en el ámbito de programas de conformidad específicos, consulte [Servicios de Servicios de AWS en el ámbito del programa de conformidad](#) y escoja el programa de conformidad que le interese. Para obtener información general, consulte [AWSProgramas de conformidad de](#).

Puede descargar los informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte [Descarga de informes en AWS Artifact](#).

Su responsabilidad de conformidad al utilizar Servicios de AWS se determina en función de la sensibilidad de los datos, los objetivos de conformidad de su empresa y la legislación y los reglamentos correspondientes. AWS proporciona los siguientes recursos para ayudar con la conformidad:

- [Guías de inicio rápido de seguridad y conformidad](#): estas guías de implementación tratan consideraciones sobre arquitectura y ofrecen pasos para implementar los entornos de referencia centrados en la seguridad y la conformidad en AWS.
- [Arquitectura para la seguridad y el cumplimiento de la HIPAA en Amazon Web Services](#): en este documento técnico, se describe cómo las empresas pueden utilizar AWS para crear aplicaciones aptas para HIPAA.

 Note

No todos los Servicios de AWS son aptos para HIPAA. Para obtener más información, consulte la [Referencia de servicios aptos para HIPAA](#).

- [Recursos de conformidad de AWS](#): este conjunto de manuales y guías podría aplicarse a su sector y ubicación.
- [Guías de cumplimiento para clientes de AWS](#): comprenda el modelo de responsabilidad compartida desde el punto de vista del cumplimiento. Las guías resumen las mejores prácticas para garantizar la seguridad de los Servicios de AWS y orientan los controles de seguridad en varios marcos (incluidos el Instituto Nacional de Estándares y Tecnología (NIST, por sus siglas en inglés), el Consejo de Estándares de Seguridad de la Industria de Tarjetas de Pago (PCI, por sus siglas en inglés) y la Organización Internacional de Normalización (ISO, por sus siglas en inglés)).

- [Evaluación de recursos con reglas](#) en la Guía para desarrolladores de AWS Config: el servicio AWS Config evalúa en qué medida las configuraciones de sus recursos cumplen las prácticas internas, las directrices del sector y las normativas.
- [AWS Security Hub](#): este Servicio de AWS proporciona una visión completa de su estado de seguridad en AWS. Security Hub utiliza controles de seguridad para evaluar sus recursos de AWS y comprobar su conformidad 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](#).
- [AWS Audit Manager](#): este servicio de Servicio de AWS le ayuda a auditar continuamente el uso de AWS con el fin de simplificar la forma en que administra el riesgo y la conformidad con las normativas y los estándares del sector.

## Resiliencia en AWS SimSpace Weaver

La infraestructura global de AWS se divide en Regiones de AWS y zonas de disponibilidad. Las Regiones de AWS proporcionan varias zonas de disponibilidad físicamente independientes y aisladas que se encuentran conectadas mediante redes con un alto nivel de rendimiento y redundancia, además de baja latencia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que conmutan automáticamente entre zonas sin interrupción. 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 Regiones de AWS y las zonas de disponibilidad, consulte [Infraestructura global de AWS](#).

## Seguridad de infraestructuras en AWS SimSpace Weaver

Como se trata de un servicio administrado, AWS SimSpace Weaver está protegido por la seguridad de red global de AWS. Para obtener información sobre los servicios de seguridad de AWS y cómo AWS protege la infraestructura, consulte [Seguridad en la nube de AWS](#). Para diseñar su entorno de AWS conforme a las prácticas recomendadas de seguridad de la infraestructura, consulte [Protección de la infraestructura](#) en Pilar de seguridad del Marco de AWS Well-Architected.

Puede utilizar llamadas a la API publicadas en AWS para obtener acceso a SimSpace Weaver a través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.

- Conjuntos de cifrado con confidencialidad directa total (PFS) como DHE (Ephemeral Diffie-Hellman) o 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 estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad principal de 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 una VPC de Amazon ubicada en una región de AWS que seleccione. Una VPC de Amazon es una red virtual en la nube de AWS que aísla la infraestructura por carga de trabajo o entidad organizativa. Las comunicaciones entre las instancias informáticas de la VPC de Amazon permanecen dentro de la red de AWS y no circulan por Internet. Algunas comunicaciones de los servicios internos se transmiten por Internet y están cifradas. Las simulaciones de todos los clientes que se ejecutan en la misma región de AWS comparten la misma VPC de Amazon. Las simulaciones para distintos clientes utilizan instancias informáticas independientes dentro de la misma VPC de Amazon.

Las comunicaciones entre sus clientes de simulación y las simulaciones que se están ejecutando en SimSpace Weaver circulan por Internet. SimSpace Weaver no gestiona estas conexiones. Es su responsabilidad proteger las conexiones de sus clientes.

Sus conexiones al servicio de SimSpace Weaver se transmiten por Internet y están cifradas. Esto incluye las conexiones mediante AWS Management Console, AWS Command Line Interface (AWS CLI), los kits de desarrollo de software (SDK) de AWS y el SDK de la aplicación SimSpace Weaver.

## Configuración y análisis de vulnerabilidades en AWS SimSpace Weaver

La configuración y los controles de TI son una responsabilidad compartida entre AWS y usted. Para obtener más información, consulte el [modelo de responsabilidad compartida de AWS](#). 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 de desastres de la infraestructura de AWS. 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 sus simulaciones para utilizar las versiones más recientes del SDK, incluidos el SDK de AWS y el SDK de la aplicación de SimSpace Weaver.

#### 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 prácticas recomendadas de seguridad específicas de SimSpace Weaver. Para obtener más información sobre las prácticas recomendadas de seguridad en AWS, consulte las [Prácticas recomendadas sobre 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 SDK](#)

## 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 llamada a la API, una opción de consola o 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 SDK

Mantenga sus aplicaciones, las instalaciones locales de los kits de desarrollo de software (SDK) de AWS y el SDK de la aplicación SimSpace Weaver. Puede descargar e instalar versiones nuevas de los SDK de AWS. Pruebe las nuevas versiones del SDK de la aplicación de SimSpace Weaver con compilaciones de aplicaciones que no sean de producción para asegurarse de que las aplicaciones sigan funcionando según lo previsto. 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.

# Registro y monitorización en SimSpace Weaver

La supervisión es una parte importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de SimSpace Weaver y de sus otras soluciones de AWS. AWS ofrece las siguientes herramientas de supervisión para vigilar SimSpace Weaver, informar cuando algo no va bien y tomar medidas automáticamente cuando proceda:

- 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 las llamadas a la API y otros eventos relacionados que realiza la Cuenta de AWS o que se realizan en nombre de esta. Además, entrega los archivos de registro a un bucket de Amazon S3 especificado. También pueden identificar qué usuarios y cuentas llamaron a AWS, la dirección IP de origen de las llamadas y el momento en que estas se realizaron. Para más información, consulte la [Guía del usuario de AWS CloudTrail](#).

## Temas

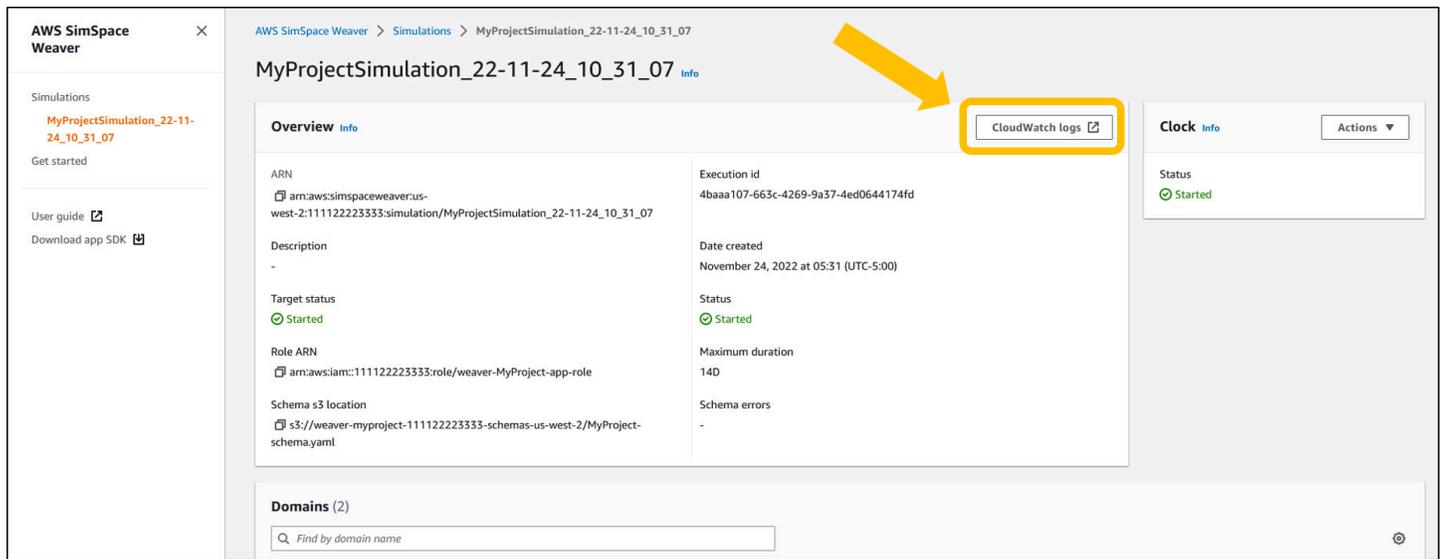
- [SimSpace Weaver inicia sesión en Amazon CloudWatch Logs](#)
- [Monitorización SimSpace Weaver con Amazon CloudWatch](#)
- [Registrar llamadas a la API de AWS SimSpace Weaver mediante AWS CloudTrail](#)

## SimSpace Weaver inicia sesión en Amazon CloudWatch Logs

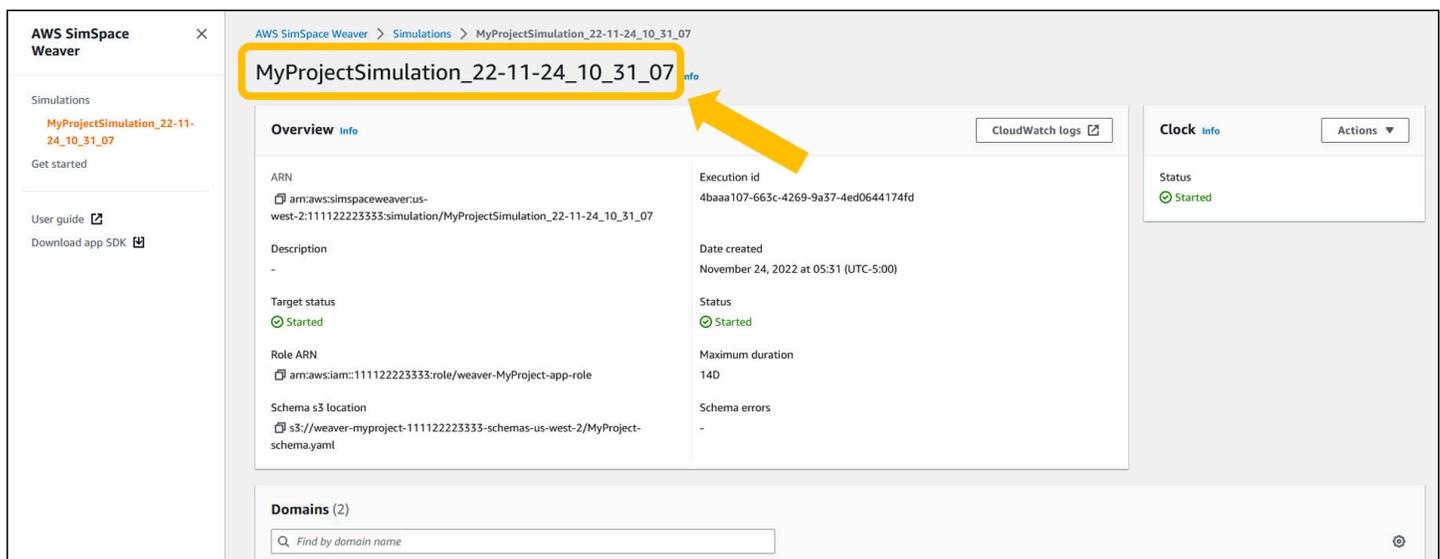
### Acceder a sus registros de SimSpace Weaver

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.



## Registros de SimSpace Weaver

SimSpace Weaver describe 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.

#### Important

Si usa AWS IAM Identity Center o perfiles nombrados para AWS Command Line Interface (AWS CLI), debe usar la versión 1.12.1 o superior del SDK de la aplicación de SimSpace Weaver. La última versión es la 1.16.0. Para obtener información acerca de las versiones de SimSpace Weaver, consulte [Versiones de SimSpace Weaver](#). Los scripts del SDK de la aplicación de SimSpace Weaver utilizan AWS CLI. Si utiliza el centro de identidad de IAM, puede copiar su perfil del centro de identidad de IAM para el AWS CLI a su perfil de `default` o proporcionar el nombre de su perfil del centro de identidad de IAM a los scripts del SDK de la aplicación de SimSpace Weaver con el parámetro `--profile cli-profile-name`. Para obtener más información, consulte [Configuración de AWS CLI para usar AWS IAM Identity Center](#) en la Guía del usuario de AWS Command Line Interface y [Ajustes de configuración y archivos de credenciales](#) en la Guía del usuario de AWS Command Line Interface.

## Docker

```
project-folder\tools\windows\weaver-project-name-cli.bat describe-simulation --
simulation simulation-name
```

## 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](#).

```
project-folder/tools/linux/weaver-project-name-cli.sh 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 servicio de SimSpace Weaver.

```
/sim/management
```

- Flujo de registro de errores: mensajes de error producidos por el servicio de SimSpace Weaver. Este flujo de registro solo existe si hay errores. SimSpace Weaver almacena los errores escritos

por sus 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
```

## 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 servicio de SimSpace Weaver informa de las siguientes métricas en el espacio de nombres de AWS/simspaceweaver.

## Métricas de SimSpace Weaver a nivel de cuenta

El espacio de nombres de SimSpace Weaver incluye las siguientes métricas relacionadas con la actividad de AWS a nivel de cuenta.

Métrica	Descripción
SimulationCount	El número de simulaciones de la cuenta actual.  Unidades: recuento  Dimensiones: ninguna  Estadísticas: promedio, mínimo, máximo

## Registrar llamadas a la API de AWS SimSpace Weaver mediante 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 las llamadas a la API SimSpace Weaver como eventos. Las llamadas capturadas incluyen las llamadas desde la consola de SimSpace Weaver y las llamadas desde el código a las operaciones de la API de SimSpace Weaver. 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 ver los eventos más recientes en la CloudTrail consola en Event history. Con la información recopilada por usted 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 mantener un registro continuo de eventos en la Cuenta de AWS, incluidos los eventos de SimSpace Weaver, cree un registro de seguimiento. 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. El registro de seguimiento registra los eventos de todas las regiones de la partición de AWS y envía los archivos de registro al bucket de Amazon S3 especificado. 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 las notificaciones de Amazon SNS 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 [referencia de la AWS SimSpace Weaver API](#). Por ejemplo, las llamadas a `DescribeSimulation` y `DeleteSimulation` las acciones generan entradas en los archivos de CloudTrail registro. `ListSimulations`

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 de AWS Identity and Access Management (IAM) o credenciales de usuario raíz.
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro servicio de AWS.

Para obtener más información, consulte el [elemento `userIdentity` de CloudTrail](#).

## Descripción de las entradas de los archivos de registro de SimSpace Weaver

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 llamadas a la API pública, por lo que no aparecen en ningún orden específico.

En el siguiente ejemplo, se 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	<a href="#">Sí</a>	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	<a href="#">Sí</a>	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](#)
- [El cliente de consola PathFindingSample 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. Para obtener más información, consulte [SimSpace Puntos finales y cuotas de Weaver](#). Para obtener información sobre cómo limpiar las simulaciones innecesarias, consulte [Paso 12: Detenga y limpie la simulación](#) en el tutorial de inicio rápido.

## 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 buckets que puede tener en su cuenta de AWS (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 cuenta de AWS diferente.

#### Note

La API de `DeleteSimulation` en 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. Para obtener instrucciones sobre cómo limpiar la simulación, consulte [Paso 12: Detenga y limpie la simulación](#) en el tutorial de inicio rápido.

## 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
```

## El cliente de consola PathFindingSample no se puede conectar

Es posible que reciba el siguiente error del cliente de la consola al conectarse a la PathfindingSample simulación descrita en el [tutorial de inicio rápido](#) y en el [tutorial detallado](#). 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 API ListSimulations para obtener el estado de la simulación. Para obtener más información, consulte [Paso 4: obtenga su dirección IP y número de puerto](#). También puede utilizar la [consola de SimSpace Weaver](#) para verificar el estado de las simulaciones.
- ¿Están funcionando sus aplicaciones? Puede usar la API DescribeApp para obtener el estado de sus aplicaciones. Para obtener más información, consulte [Paso 4: obtenga su dirección IP y](#)

[número de puerto](#). También puede utilizar la [consola de SimSpace Weaver](#) para verificar el estado de las simulaciones.

- ¿Están funcionando sus aplicaciones? Puede usar la API DescribeApp para obtener el estado de sus aplicaciones. Para obtener más información, consulte [Paso 4: obtenga su dirección IP y número de puerto](#). 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. Puede encontrar la IP Address y el número de puerto Actual en el bloque de salida de la API de EndpointInfo DescribeApp. 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](#).
- ¿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](#).
- ¿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.

# Referencia del esquema de simulación SimSpace Weaver

SimSpace Weaver usa un archivo YAML para configurar las propiedades de una simulación. Este archivo se denomina esquema de simulación (o simplemente esquema). El ejemplo de simulación incluido en el SDK de la aplicación de SimSpace Weaver incluye un esquema que puede copiar y editar para su propia simulación.

## Temas

- [Ejemplos de un esquema completo](#)
- [Formato de esquema](#)

## Ejemplos de un esquema completo

El siguiente ejemplo muestra el archivo YAML de texto con formato que describe una SimSpace Weaver simulación. Estos ejemplos 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: "1.16"
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

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

## Versión de SDK

La sección `sdk_version` (obligatoria) identifica la versión del SDK de la aplicación SimSpace Weaver que admite este esquema. Valores válidos: 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: "1.16"
```

## 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 SimSpace Weaver escribirá 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 un `logging_destination_resource_name` que no es un ARN.

Obligatorio: debe especificar esta propiedad si se especifica `log_destination_resource_name` y no es un ARN. No puede especificar esta propiedad si `log_destination_resource_name` no se especifica 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`)

## Trabajadores

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 prefiera. 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 una estrategia de partición.

Para especificar las propiedades de una estrategia de particionamiento, *partitioning-strategy-name* sustitúyalas por el nombre que prefiera. 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 de (AABB) para la 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, *spatial-domain-name* sustitúyala por el nombre que prefiera. 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: el URI de Amazon S3 del paquete en un bucket de Amazon S3. Por ejemplo, `s3://example-bucket/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) en 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 propiedad `image` (opcional) especifica la ubicación de una imagen de contenedor que SimSpace Weaver utiliza para ejecutar aplicaciones en este dominio (no es compatible con las versiones 1.13 y 1.12). Proporcione la URI a un repositorio en 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 prefiera. 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 lanzar sus aplicaciones personalizadas mediante la API StartApp.

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: el URI de Amazon S3 del paquete en un bucket de Amazon S3. Por ejemplo, `s3://example-bucket/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) en 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 TCP y UDP. Use la API de `DescribeApp` 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 propiedad `image` (opcional) especifica la ubicación de una imagen de contenedor que SimSpace Weaver utiliza para ejecutar aplicaciones en este dominio (no es compatible con las versiones 1.13 y 1.12). Proporcione la URI a un repositorio en 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, *service-domain-name* sustitúyalas por el nombre que prefiera. 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: el URI de Amazon S3 del paquete en un bucket de Amazon S3. Por ejemplo, `s3://example-bucket/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) en 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 TCP y UDP. Use la API de `DescribeApp` 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 propiedad `image` (opcional) especifica la ubicación de una imagen de contenedor que SimSpace Weaver utiliza para ejecutar aplicaciones en este dominio (no es compatible con las versiones 1.13 y 1.12). Proporcione la URI a un repositorio en 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 debe colocar 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

# Referencias de API de SimSpace Weaver

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 software (SDK) principal de AWS y puede utilizar la interfaz de la línea de comandos (CLI) de AWS para llamarlos. También puede utilizar el práctico script de la carpeta de herramientas de su proyecto y plataforma (por ejemplo, `project-folder\tools\windows\weaver-project-name-cli.bat`) para llamar a estas API. 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, consulte la documentación del SDK de la aplicación SimSpace Weaver en la carpeta del SDK de la aplicación descomprimida: `sdk-folder\SimSpaceWeaverAppSdk-1.16.0\documentation`

## Note

*sdk-folder es la carpeta* en la que descomprimiste el paquete. SimSpaceWeaverAppSdkDistributable Si `sdk-folder\SimSpaceWeaverAppSdk-1.16.0` no existe, asegúrate de tener la versión 1.16.0 del SDK de la aplicación. SimSpace Weaver SimSpaceWeaverAppSdkDistributable No contiene todo el SDK de la SimSpace Weaver aplicación. La primera vez que utilices los scripts del SDK de la SimSpace Weaver aplicación para crear un proyecto, estos crean la carpeta SimSpaceWeaverAppSdk -1.16.0 dentro de la carpeta `sdk` y descargan en ella el resto del SDK de la SimSpace Weaver aplicación.

# Versiones de AWS SimSpace Weaver

Mejoramos AWS SimSpace Weaver continuamente. Es necesario que descargue el último SDK de la aplicación SimSpace Weaver cuando publiquemos una nueva versión si quiere aprovechar las nuevas funciones y actualizaciones de características. 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 cubre 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: 1.16.0

## Cómo encontrar la versión actual

Si creó una simulación con el SDK de la aplicación SimSpace Weaver, el script `create-project` descargó una versión de las bibliotecas del SDK en un subdirectorio de su `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 1.15.3 están disponibles. `SimSpaceWeaverAppSdk-1.15.3`

También puede encontrar la versión del paquete distribuible del SDK de la aplicación SimSpace Weaver en el archivo de texto `app_sdk_distributable_version.txt` de su `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 puede descargar el paquete distribuible completo del SDK de la aplicación SimSpace Weaver desde la consola de [SimSpace Weaver](#) en AWS Management Console. En el panel de navegación, seleccione Descargar el SDK de la aplicación en el panel de navegación.

#### Warning

No utilice AWS CLI para descargar nada que parezca ser el paquete distribuible del SDK de la aplicación SimSpace Weaver. 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.

- Se muestra en tu navegador 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. Ejecute `docker-create-image.bat` (o `docker-create-image.sh` para WSL) desde la carpeta descomprimida de la última versión del SDK de la aplicación SimSpace Weaver.
4. Use la carpeta descomprimida de la última versión del SDK de la aplicación SimSpace Weaver en lugar de la versión anterior.

## Versiones del servicio

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
1.16.0	<p>Nueva característica:</p> <ul style="list-style-type: none"> <li>• 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</li> </ul>	12 de febrero de 2024	Esta guía	<ul style="list-style-type: none"> <li>• <a href="#">Paquete completo</a></li> <li>• <a href="#">Solo bibliotecas</a></li> </ul> <p>Consulte <a href="#">Solución de problemas</a></p>

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
	disponible para C++, Python y las integraciones de Unity y Unreal Engine 5. Para obtener más información, consulte <a href="#">Mensajería</a> .			

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"><li>• 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. Para obtener más información, consulte <a href="#">Diferencias de desarrollo local en la versión 1.15.3</a>.</li></ul>	4 de diciembre de 2023	Consulte el catálogo de <a href="#">AWS SimSpace Weaverguías</a> .	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"><li>• Hemos actualizado Dockerfile para usar la versión específica requerida de cmake. Las compilaciones de contenedores de Docker podrían fallar sin este cambio.</li></ul>	2 de noviembre de 2023	Consulte el <a href="#">catálogo AWS SimSpace Weaver de guías</a> .	No está disponible para su descarga

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
1.15.1	<p>Actualización de una característica:</p> <ul style="list-style-type: none"><li>• SDK de Python: esta versión corrige un problema que provocaba que las simulaciones basadas en Python fallaran en Nube de AWS. Para obtener más información, consulte las <a href="#">notas de la versión</a>.</li></ul>	22 de septiembre de 2023	Consulte el <a href="#">catálogo AWS SimSpace Weaver de guías</a> .	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 1390" style="list-style-type: none"><li data-bbox="272 401 548 1390">• SDK de Python: Ahora puede desarrollar sus simulaciones en Python. El paquete distribuido del SDK de la aplicación SimSpace Weaver incluye una plantilla para un proyecto de Python de muestra y su cliente de visualización de Python. Para obtener más información, consulte <a href="#">Trabajo con Python</a>.</li></ul>	31 de agosto de 2023	Consulte el <a href="#">catálogo AWS SimSpace Weaver de guías</a> .	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"><li>• Contenedores personalizados: 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. Para obtener más información, consulte <a href="#">Contenedores personalizados</a>.</li><li>• Varios dominios espaciales: cree más de un dominio espacial en una simulación. Separe la lógica de simulación</li></ul>	26 de julio de 2023	Consulte el <a href="#">catálogo AWS SimSpace Weaver de guías</a> .	No está disponible para su descarga

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
	<p>n en lugar de combinarla toda en una sola aplicación espacial. Asigne diferentes recursos a los dominios espaciales en función de sus requisitos. Para obtener más información, consulte <a href="#">Configuración de dominios espaciales</a>.</p> <ul style="list-style-type: none"><li>• 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</li></ul>			

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
	<p>operaciones de confirmación de la graduación actual. Para obtener más información, consulte <a href="#">Reloj</a>.</p> <p>SDK de la aplicación: SimSpace Weaver</p> <ul style="list-style-type: none"><li>• El valor de <code>tick_rate</code> ahora es una cadena. El valor debe incluir comillas dobles ("). La frecuencia de graduación de las versiones anteriores sigue siendo un número entero. Para obtener más información, consulte <a href="#">Reloj</a>.</li></ul>			

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
1.13.1	<p data-bbox="272 275 553 401">SDK de la aplicación: SimSpace Weaver</p> <ul data-bbox="272 449 553 911" style="list-style-type: none"><li data-bbox="272 449 553 911">• Actualización de una característica: la creación de proyectos ahora funciona correctamente con la plantilla PathfindingSampleUnreal .</li></ul>	7 de junio de 2023	Consulte el <a href="#">catálogo AWS SimSpace Weaver de guías</a> .	No está disponible para su descarga

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
1.13.0	<p>API de servicio: SimSpace Weaver</p> <ul style="list-style-type: none"> <li>• Nueva acción de <a href="#">CreateSnapshot</a></li> <li>• Cambios en la acción de <a href="#">StartSimulation</a> : <ul style="list-style-type: none"> <li>• Se ha añadido un parámetro de SnapshotS3Location para comenzar desde una instantánea.</li> <li>• El parámetro SchemaS3Location ahora es opcional.</li> </ul> </li> <li>• Cambios en el resultado de <a href="#">DescribeSimulation</a> :</li> </ul>	29 de abril de 2023	<a href="#">Consulte el catálogo de guíasAWS SimSpace Weaver.</a>	No está disponible para su descarga

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
	<ul style="list-style-type: none"> <li>• <code>SchemaError</code> está obsoleto.</li> <li>• Se ha añadido el campo <code>StartError</code>.</li> <li>• Se ha añadido el campo <code>SnapshotS3Location</code>.</li> <li>• Se ha añadido un estado de simulación <code>SNAPSHOT_IN_PROGRESS</code>.</li> <li>• Nuevo tipo de datos de <a href="#">S3Destination</a>.</li> </ul> <p>Consola de SimSpace Weaver:</p> <ul style="list-style-type: none"> <li>• Nueva funcionalidad para crear instantáneas.</li> </ul>			

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
	<ul style="list-style-type: none"> <li>• Nueva funcionalidad para iniciar una simulación a partir de una instantánea.</li> </ul> <p>SDK de la aplicación: SimSpace Weaver</p> <ul style="list-style-type: none"> <li>• Nuevos scripts para admitir instantáneas               <ul style="list-style-type: none"> <li>• create-snapshot- <i>project-name</i> .bat</li> <li>• start-from-snapshot- <i>project-name</i> .bat</li> <li>• quick-start-from-snapshot- <i>project-name</i> -cli.bat</li> <li>• list-snapshots- <i>project-name</i> .bat</li> </ul> </li> <li>• <i>Los proyectos ahora utilizan un único bucket</i></li> </ul>			

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
	<p><i>de Amazon S3 por proyecto: weaver-lowercase-project-name - número de cuenta - región</i></p> <p>Para obtener más información acerca de las instantáneas, consulte <a href="#">Instantáneas</a>.</p>			

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
1.12.3	<p>SDK de la aplicación: SimSpace Weaver</p> <ul style="list-style-type: none"> <li>• Los siguientes scripts ahora admiten el parámetro <code>--maximum-duration</code> : <ul style="list-style-type: none"> <li>• <code>quick-start-<i>project-name</i> - cli.bat</code></li> <li>• <code>quick-start-<i>project-name</i> - cli.sh</code></li> <li>• <code>start-simulation-<i>project-name</i> .bat</code></li> <li>• <code>start-simulation-<i>project-name</i> .sh</code></li> <li>• <code>run-<i>project-name</i> .bat</code></li> <li>• <code>run-<i>project-name</i> .sh</code></li> </ul> </li> </ul> <p>Para obtener más información, consulte <a href="#">Duración</a></p>	27 de marzo de 2023	<p>Consulte el catálogo de <a href="#">AWS SimSpace Weaverguías</a>.</p>	No está disponible para su descarga

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
	<a href="#"><u>máxima de una simulación.</u></a>			
1.12.2	SDK de la aplicación: SimSpace Weaver <ul style="list-style-type: none"><li>• Corrección de error: <code>docker-create-image.bat</code> ahora funciona correctamente.</li></ul>	1 de marzo de 2023	Consulte el catálogo de <a href="#"><u>AWS SimSpace Weaverguías.</u></a>	No está disponible para su descarga

Versión	Notas	Fecha de publicación	Documentación	Descargar el SDK de la aplicación
1.12.1	<p>SDK de la aplicación: SimSpace Weaver</p> <ul style="list-style-type: none"><li>• Los scripts ahora aceptan un perfil AWS CLI para usarlo en la autenticación en AWS.</li><li>• Los scripts ahora admiten AWS IAM Identity Center para la autenticación en AWS.</li></ul> <p>SimSpace Weaver Local:</p> <ul style="list-style-type: none"><li>• 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.</li></ul>	28 de febrero de 2023	Consulte el <a href="#">catálogo AWS SimSpace Weaver de guías</a> .	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 <a href="#">catálogo AWS SimSpace Weaver de guías</a> .	No está disponible para su descarga

## 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 de AWS SimSpace Weaver

En la siguiente tabla, se describen los cambios importantes que se han realizado en la documentación de SimSpace Weaver.

Date	Cambio	Actualizaciones de la documentación	Versiones de API actualizadas
12 de febrero de 2024	Contenido actualizado	Se actualizó el <a href="#">Versiones de AWS SimSpace Weaver</a> capítulo para la versión 1.16.0.	N/A
12 de febrero de 2024	Nuevo contenido	Se agregó la <a href="#">Mensajería</a> 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 <a href="#">Referencia del esquema de simulación SimSpace Weaver</a> capítulo para la versión 1.16.0.	N/A
12 de febrero de 2024	Contenido actualizado	Se agregaron cuotas de servicio de mensajería al <a href="#">SimSpace Puntos finales y cuotas de Weaver</a> capítulo.	N/A

Date	Cambio	Actualizaciones de la documentación	Versiones de API actualizadas
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 <a href="#">catálogo de AWS SimSpace Weaver guías</a> (disponible en la <a href="#">página de inicio de la documentación principal</a> ) para acceder a las guías de las versiones anteriores.	N/A
4 de diciembre de 2023	Contenido actualizado	Se ha actualizado el capítulo <a href="#">Versiones de AWS SimSpace Weaver</a> para la versión 1.15.3.	N/A
4 de diciembre de 2023	Contenido actualizado	Se actualizó el <a href="#">Versiones de AWS SimSpace Weaver</a> capítulo para incluir las instrucciones de instalación de la última versión.	N/A
4 de diciembre de 2023	Contenido actualizado	Se han actualizado los <a href="#">Service Quotas de SimSpace Weaver Local</a> .	N/A

Date	Cambio	Actualizaciones de la documentación	Versiones de API actualizadas
4 de diciembre de 2023	Contenido nuevo y actualizado	Se reestructuró la sección <a href="#">Desarrollo local</a> 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 <a href="#">Configuración del entorno de SimSpace Weaver</a> .	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 <a href="#">Versiones del servicio</a> .	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 usar AWS CLI para descargar el paquete distribuible del SDK de la aplicación. Para obtener más información, consulte <a href="#">Descargue la versión más reciente</a> .	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 <a href="#">AWS SimSpace Weaver versión 1.15.1</a> .	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 situaciones en las que AWS CLI no reconoce SimSpace Weaver. Para obtener más información, consulte <a href="#">La AWS CLI no reconoce simspaceweaver</a> .	N/A
10 de septiembre de 2023	Contenido actualizado	Se han actualizado las instrucciones de instalación de AWS CLI en WSL. Para obtener más información, consulte <a href="#">Configurar Amazon Linux 2 (AL2) en Windows Subsystem for Linux (WSL)</a> .	N/A
7 de septiembre de 2023	Actualización de API	BucketName y ahora ObjectKey son necesarios para el tipo de datos <a href="#">S3Location</a> . BucketName y ahora son necesarios para el tipo de datos <a href="#">S3Destination</a> .	SDK de AWS: 07/09/2023
31 de agosto de 2023	Nuevo contenido	Se ha añadido una nueva sección para la versión 1.15.0: <a href="#">Trabajo con Python</a> .	N/A

Date	Cambio	Actualizaciones de la documentación	Versiones de API actualizadas
15 de agosto de 2023	Contenido actualizado	Se han actualizado las instrucciones de descarga de <a href="#">Versiones de AWS SimSpace Weaver</a> para que solo aparezcan los buckets de S3 oficiales de Amazon de SimSpace Weaver. Otras ubicaciones de descarga no están controladas por AWS y pueden contener códigos maliciosos.	N/A
26 de julio de 2023	Contenido actualizado	Actualizado <a href="#">Reloj</a> .	N/A
26 de julio de 2023	Contenido actualizado	Actualizado <a href="#">Configuración de dominios espaciales</a> .	N/A
26 de julio de 2023	Nuevo contenido	Se ha añadido una nueva sección: <a href="#">Contenedores personalizados</a> .	N/A
26 de julio de 2023	Contenido actualizado	Se ha actualizado <a href="#">Versiones de AWS SimSpace Weaver</a> a la versión 1.14.0.	N/A
6 de julio de 2023	Nuevo contenido	Se ha añadido una nueva sección: <a href="#">El cliente de consola PathFindi ngSample no se puede conectar</a> .	N/A

Date	Cambio	Actualizaciones de la documentación	Versiones de API actualizadas
7 de junio de 2023	Contenido actualizado	Se ha actualizado <a href="#">Versiones de AWS SimSpace Weaver</a> a la versión 1.13.1.	N/A
15 de mayo de 2023	Nuevo contenido	Se ha añadido una nueva sección: <a href="#">Uso de instantáneas con AWS CloudFormation</a> .	N/A
29 de abril de 2023	Nuevo contenido	Se ha añadido contenido para la versión 1.13.0. Para obtener más información, consulte <a href="#">Versiones de AWS SimSpace Weaver</a> .	SDK de AWS: 28/04/2023
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

Date	Cambio	Actualizaciones de la documentación	Versiones de API actualizadas
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 ha añadido un capítulo que describe las versiones de SimSpace Weaver.	N/A
28 de febrero de 2023	Cambio del contenido	Se ha modificado el contenido sobre la autenticación para incluir el uso de AWS IAM Identity Center y los perfiles con nombre para AWS Command Line Interface (AWS CLI).	N/A
17 de febrero de 2023	Nuevo contenido	Se ha añadido una sección sobre cómo administrar sus recursos con AWS CloudFormation.	N/A
23 de enero de 2023	Nuevo contenido	Se han añadido instrucciones para depurar simulaciones locales.	N/A

Date	Cambio	Actualizaciones de la documentación	Versiones de API actualizadas
29 de noviembre de 2022	Lanzamiento del servicio	Se han publicado la Guía del usuario y la Referencia de la API para SimSpace Weaver.	SDK de AWS: 29/11/2022

# Glosario

---

Este glosario define los términos que son específicos sobre AWS SimSpace Weaver.

Para ver la terminología más reciente de AWS, consulte el [Glosario de AWS](#) en la Referencia general de 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 <a href="#">entidades</a> .
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 <a href="#">entidades</a> y realizar un seguimiento del tiempo de simulación. Para obtener más información, consulte <a href="#">SDK de la aplicación SimSpace Weaver</a> .

## C

---

Cliente	Procesos (o sus definiciones) que existen fuera de SimSpace Weaver e interactúan con la simulación a través de una <a href="#">aplicación personalizada</a> o una <a href="#">aplicación de servicio</a> . Puede usar un cliente para ver o cambiar el estado de la simulación.
Reloj	Una abstracción de los procesos de programación interna de SimSpace Weaver. El reloj publica las <a href="#">graduaciones</a> en las <a href="#">aplicaciones</a> para mantener la sincronización horaria. Cada simulación tiene su propio reloj.
Frecuencia del reloj	El número de <a href="#">graduaciones</a> por segundo que el <a href="#">reloj</a> publica en las <a href="#">aplicaciones</a> . Para obtener más información acerca de las frecuencias del reloj, consulte <a href="#">SimSpace Puntos finales y cuotas de Weaver</a> .
Frecuencia de la graduación del reloj	Consulte la <a href="#">frecuencia del reloj</a> .

Unidad de recursos de cómputo	Unidad de recursos de cómputo (procesador y memoria) de un <a href="#">trabajador</a> . A una sola instancia de una <a href="#">aplicación</a> 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 <a href="#">aplicación</a> 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 <a href="#">dominio espacial</a> . Usted controla el ciclo de vida de una aplicación personalizada mediante las API de la aplicación. Para obtener más información sobre las API de SimSpace Weaver, consulte <a href="#">Referencias de API de SimSpace Weaver</a> .
Dominio personalizado	Un <a href="#">dominio</a> que contiene <a href="#">aplicaciones personalizadas</a> .
Partición personalizada	La <a href="#">partición</a> de una <a href="#">aplicación personalizada</a> .

## D

---

Fecha límite	<a href="#">El tiempo real</a> en el que debe completarse una operación (como el procesamiento de una <a href="#">graduación</a> ).
Dominio	Grupo de instancias de <a href="#">aplicaciones</a> 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 servicio de SimSpace Weaver.
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 <a href="#">aplicaciones personalizadas</a> y <a href="#">aplicaciones de servicio</a> .
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 (SimSpace Weaverse inicia y se detiene la aplicación) o no se administran (usted 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 <a href="#">aplicación</a> 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 <a href="#">dominio</a> espacial. SimSpace Weaver administra el <a href="#">ciclo de vida</a> de una aplicación de servicio e inicia 1 (o más, según se especifique en el <a href="#">esquema</a> de simulación) en cada <a href="#">trabajador</a> de la simulación.
Dominio de servicio	Un <a href="#">dominio</a> que contiene <a href="#">aplicaciones personalizadas</a> .
Partición de servicio	La <a href="#">partición</a> de una <a href="#">aplicación de servicio</a> .
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 <a href="#">esquema</a> .
Aplicación espacial	Tipo de <a href="#">aplicación</a> que encapsula la lógica principal de la simulación. Cada aplicación espacial posee 1 (y solo 1) <a href="#">partición</a> .
Dominio espacial	Un <a href="#">dominio</a> que contiene <a href="#">aplicaciones espaciales</a> .
Partición espacial	La <a href="#">partición</a> de una <a href="#">aplicación espacial</a> .
State Fabric	Base de datos en memoria de SimSpace Weaver State Fabric almacena el estado de las simulaciones, incluidas las entidades y los datos internos de SimSpace Weaver.
Almacenar (datos del campo de la entidad)	Escribir los datos del campo de la entidad en <a href="#">State Fabric</a> .
Suscripción	Una solicitud de larga duración para que una instancia de <a href="#">aplicación</a> específica reciba datos de un <a href="#">área de suscripción</a> . La aplicación que se suscribe utiliza una suscripción para detectar los cambios en las <a href="#">entidades</a> que se encuentran dentro del área de suscripción.
Área de suscripción	Región bidimensional del espacio de simulación. Una <a href="#">suscripción</a> se refiere a un área de suscripción. Un área de suscripción puede abarcar más de 1 <a href="#">partición</a> 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 <a href="#">aplicaciones</a> 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 Unix del tiempo 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 version original de inglés, prevalecerá la version en inglés.