



Guia do usuário para a versão 1.17.0

AWS SimSpace Weaver



AWS SimSpace Weaver: Guia do usuário para a versão 1.17.0

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

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestige a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

O que é o SimSpace Weaver?	1
Principais conceitos	1
Como o SimSpace Weaver funciona	2
Como usar o SimSpace Weaver	5
Esquema de simulação	6
Trabalhadores e unidades de recursos	6
Relógio da simulação	7
Partições	7
State Fabric	7
Entidades	8
Apps	8
Exemplo de casos de uso do	11
Configuração	13
Configurar a conta	13
Inscreva-se para um Conta da AWS	13
Criar um usuário com acesso administrativo	14
Adicionar permissões para usar o SimSpace Weaver	15
Configurar o ambiente local	17
AL2 no Docker	17
AL2 no WSL	19
Uso de software licenciado	23
Conceitos básicos	24
Tutorial de início rápido	24
Etapa 1: ativar o registro (opcional)	25
Etapa 2: Início rápido com o cliente do console (opção 1)	25
Etapa 2: Início rápido com o cliente Unreal Engine (opção 2)	26
Pare e exclua sua simulação	26
Solução de problemas	26
Tutorial detalhado	27
Etapa 1: ativar o registro (opcional)	28
Etapa 2: inicie sua simulação	28
Etapa 3: verifique os registros (opcional)	35
Etapa 4: veja sua simulação	37
Etapa 5: Pare e exclua sua simulação	38

Solução de problemas	39
Trabalhando com SimSpace Weaver	40
Configurar a simulação	40
Parâmetros de configuração da simulação	41
Versão do SDK	42
Propriedades da simulação	42
Operadores	43
Relógio	44
Estratégias de particionamento	47
Domínios	48
Duração máxima	58
Valor máximo	58
Valor padrão	59
Valor mínimo	59
Iniciar uma simulação usando o console	59
Status de uma simulação que atinge a duração máxima	59
Desenvolver aplicativos	59
Aplicativos espaciais	60
Aplicativos personalizados	61
Desenvolver aplicativos cliente	62
Obtenha o endereço IP e o número da porta	63
Lançamento do cliente de visualização Unreal Engine	66
Solução de problemas	67
Desenvolvimento local	68
Etapa 1: Inicie sua simulação local	68
Etapa 2: Visualize sua simulação local	69
Etapa 3: interrompa sua simulação local (opcional no Windows)	70
Solução de problemas de desenvolvimento local	71
Aplicativo do SDK do SimSpace Weaver	71
Os métodos da API retornam um Result	72
Interagir com o aplicativo do SDK no nível superior	73
Gerenciamento de simulações	73
Assinaturas	76
Entidades	77
Eventos da entidade	89
Result e tratamento de erros	96

Genéricos e tipos de domínio	98
Operações diversas do aplicativo do SDK	98
Estrutura de demonstração do SimSpace Weaver	101
Trabalhar com cotas	102
Obter os limites de um aplicativo	102
Obter a quantidade de recursos usados por um aplicativo	103
Redefinir métricas	104
Exceder um limite	104
Ficar sem memória	105
Práticas recomendadas	105
Depurar simulações	106
Use SimSpace Weaver Local e veja a saída do console	106
Veja seus registros no Amazon CloudWatch Logs	106
Use chamadas de API describe	106
Conectar um cliente	107
Depuração de simulações locais	107
Contêineres personalizados	108
Criar um contêiner personalizado	109
Modificar um projeto para usar um contêiner personalizado	110
Perguntas frequentes	113
Solução de problemas	114
Como trabalhar com Python	114
Criar um projeto do Python	115
Iniciar uma simulação em Python	117
Cliente de exemplo Python	117
Perguntas frequentes	118
Solução de problemas	118
Suporte para outros mecanismos	120
Unity	120
Unreal Engine	120
Uso de software licenciado	121
Gerenciar recursos com o AWS CloudFormation	121
Snapshots	123
Snapshots	124
Casos de uso para snapshots	125
SimSpace Weaver Console	126

AWS CLI	127
Perguntas frequentes	130
Sistema de mensagens	131
Casos de uso para mensagens	131
Usando as APIs de mensagens	132
Quando usar mensagens	139
Dicas para trabalhar com mensagens	143
Erros de mensagens e solução de problemas	144
Práticas recomendadas	147
Configurar alarmes de faturamento	147
Use SimSpace Weaver Local	147
Interrompa simulações que você não precisa	148
Exclua os recursos que você não precisa.	148
Faça backups	148
Segurança	150
Proteção de dados	151
Criptografia em repouso	152
Criptografia em trânsito	152
Privacidade do tráfego entre redes	153
Identity and Access Management	153
Público	154
Autenticando com identidades	154
Gerenciando acesso usando políticas	158
Como AWS SimSpace Weaver funciona com IAM	161
Exemplos de políticas baseadas em identidade	168
Permissões que SimSpace Weaver cria para você	172
Prevenção do problema do substituto confuso entre serviços	174
Solução de problemas	177
Registro e monitoramento de eventos de segurança no	180
Compliance Validation	181
Resiliência	182
Segurança da infraestrutura	182
Modelo de segurança de conectividade de rede	183
Análise de configuração e vulnerabilidade	184
Melhores práticas de segurança	184
Criptografe as comunicações entre os aplicativos e seus clientes	185

Faça backup periódico do estado da simulação	185
Faça a manutenção de seus aplicativos e SDKs	185
Logging e monitoramento	187
Login CloudWatch	187
Acessar os logs no SimSpace Weaver	187
SimSpace Weaver troncos	188
Monitoramento com CloudWatch	190
SimSpace Weaver métricas no nível da conta	191
CloudTrail troncos	191
SimSpace Weaver informações em CloudTrail	191
Entendendo as entradas do arquivo de SimSpace Weaver log	192
Endpoints e cotas de serviço	195
Service endpoints	195
Cotas de serviço	196
Cotas de mensagens	199
Taxas de clock	199
Cotas de serviço para o SimSpace Weaver Local	200
Solução de problemas	202
AssumeRoleAccessDenied	202
InvalidBucketName	204
ServiceQuotaExceededException	205
TooManyBuckets	205
Permissão negada durante o início da simulação	206
Problemas relacionados ao tempo de uso do Docker	206
O cliente do console não consegue se conectar	207
Não há <code>simspaceweaver</code> na AWS CLI	209
Referência de esquema	210
Exemplo de um esquema completo	210
Formato do esquema	212
SDKversão	213
Propriedades da simulação	213
Operadores	215
Relógio	216
Estratégias de particionamento	216
Domínios	218
Limitações de posicionamento	228

Referências de API	230
SimSpace Weaver versões	231
Versão mais recente	231
Como encontrar a sua versão atual	231
Baixe a versão mais recente.	231
Solução de problemas de downloads do SDK do aplicativo	232
Instalar a versão mais recente	233
Versões do serviço	233
1.17.0	248
Principais mudanças para 1.17.0	248
Atualizar um projeto para 1.17.0	249
Perguntas frequentes sobre a versão 1.17.0	250
1.15.1	251
Atualizar um projeto Python existente para o 1.15.1	251
Solução de problemas para a versão 1.15.1	252
Perguntas frequentes sobre a versão 1.15.1	252
Histórico do documento	254
Glossário	263
.....	cclxviii

O que é o AWS SimSpace Weaver?

O AWS SimSpace Weaver é um serviço que você pode usar para criar e executar simulações espaciais em grande escala no Nuvem AWS. Por exemplo, você pode criar simulações de multidões, grandes ambientes do mundo real e experiências imersivas e interativas.

Com o SimSpace Weaver, você pode distribuir cargas de trabalho de simulação em várias instâncias do Amazon Elastic Compute Cloud (Amazon EC2). O SimSpace Weaver implanta a infraestrutura subjacente da AWS para você e trata do gerenciamento de dados da simulação e da comunicação de rede entre as instâncias do Amazon EC2 que executam a simulação.

Conceitos principais do SimSpace Weaver

Simulações ou jogos são limitados pelo computador que os executa. À medida que aumenta o tamanho e a complexidade do mundo virtual, o desempenho do processamento começa a diminuir. Os cálculos demoram mais, os sistemas ficam sem memória e as taxas de quadros do cliente caem. Para simulações que não precisam de desempenho em tempo real, isso pode ser irritante. Ou pode ser uma situação crítica para os negócios, na qual o aumento dos atrasos no processamento resulta em aumento de custos. Se sua simulação ou jogo precisa de desempenho em tempo real, a degradação do desempenho é definitivamente um problema.

Uma solução comum para uma simulação que atinge o limite de desempenho é simplificar a simulação. Os jogos on-line com muitos usuários geralmente resolvem problemas de escala fazendo cópias de seu mundo virtual em servidores diferentes e distribuindo os usuários por eles.

O SimSpace Weaver resolve o problema de escalabilidade dividindo seu mundo virtual espacialmente e distribuindo as peças em um cluster de instâncias de computação que são executadas no Nuvem AWS. As instâncias de computação trabalham juntas para processar todo o mundo da simulação em paralelo. Seu mundo de simulação aparece como um único espaço integrado para tudo que está nele e para todos os clientes que se conectam a ele. Não é mais preciso simplificar uma simulação devido ao limite de desempenho do hardware. Em vez disso, adicione mais capacidade computacional na nuvem.

Tópicos

- [Como o SimSpace Weaver funciona](#)
- [Como usar o SimSpace Weaver](#)

- [Esquema de simulação](#)
- [Trabalhadores e unidades de recursos](#)
- [Relógio da simulação](#)
- [Partições](#)
- [State Fabric](#)
- [Entidades](#)
- [Apps](#)

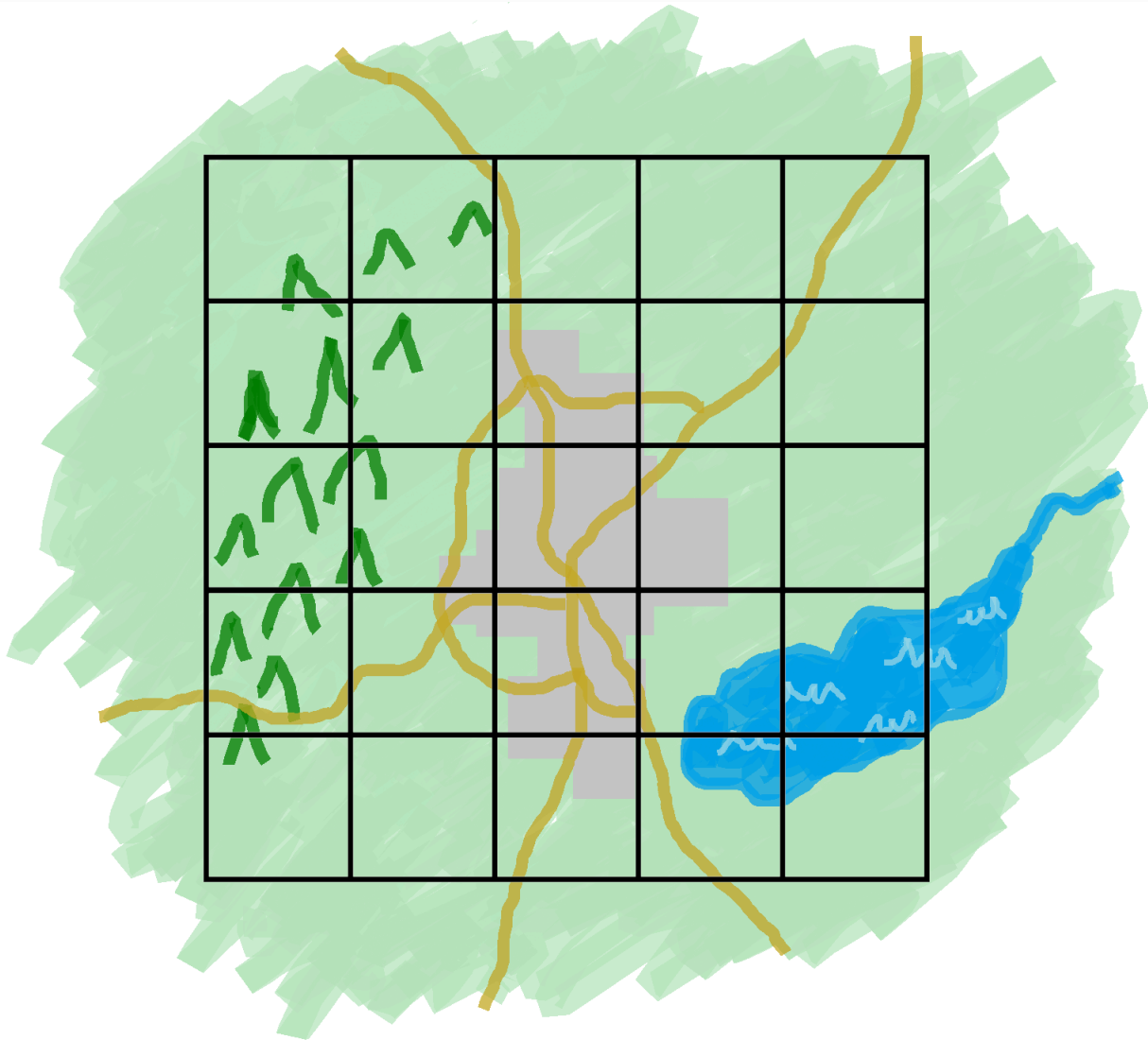
Como o SimSpace Weaver funciona

Sua simulação consiste em um mundo com objetos incluídos. Alguns dos objetos, como pessoas e veículos, se movem e fazem coisas. Outros objetos, como árvores e edifícios, são estáticos. No SimSpace Weaver, uma entidade é um objeto em seu mundo de simulação.

Defina os limites do mundo da simulação e divida-o em uma grade. Em vez de criar uma lógica de simulação que opera em toda a grade, crie uma lógica de simulação que opere em uma célula da grade. No SimSpace Weaver, um aplicativo espacial é um programa que você escreve que implementa a lógica de simulação para uma célula da sua grade. Isso inclui a lógica de todas as entidades nessa célula. A área de propriedade de um aplicativo espacial é a célula da grade que o aplicativo espacial controla.

Note

No SimSpace Weaver, o termo “aplicativo” pode se referir ao código de um aplicativo ou a uma instância em execução desse código.



Seu mundo de simulação dividido em uma grade

Divida seu mundo de simulação em uma grade. Cada aplicativo espacial implementa a lógica de simulação para uma única célula nessa grade.

O SimSpace Weaver executa uma instância do código do seu aplicativo espacial para cada célula da sua grade. Todas as instâncias espaciais do aplicativo são executadas em paralelo. Essencialmente, o SimSpace Weaver divide sua simulação geral em várias simulações menores. Cada uma das simulações menores lida com uma parte do mundo geral da simulação. O SimSpace Weaver pode distribuir e executar essas simulações menores em várias instâncias do Amazon Elastic Compute

Cloud (Amazon EC2) chamadas operadores no Nuvem AWS. Um único operador pode executar vários aplicativos espaciais.

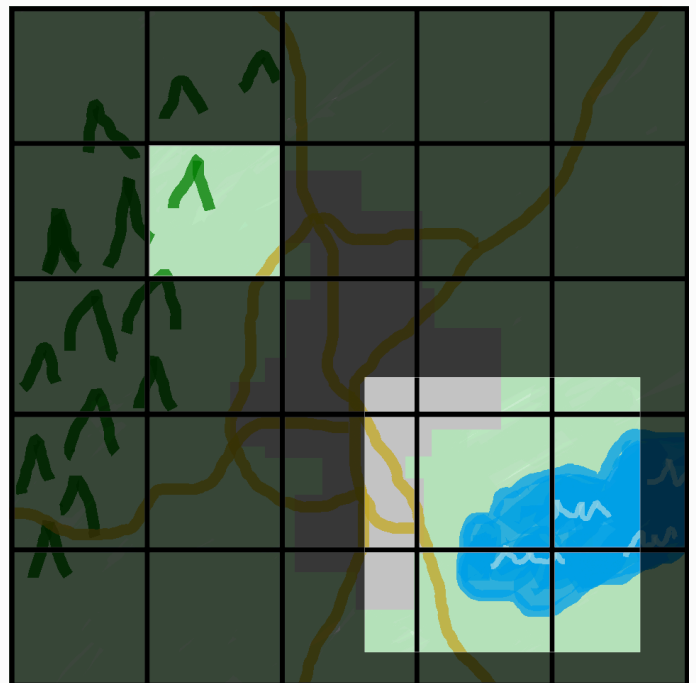
As entidades podem se mover pelo mundo da simulação. Se uma entidade entrar na área de propriedade de outro aplicativo espacial (outra célula na grade), o proprietário do aplicativo espacial da nova área assumirá o controle da entidade. Se sua simulação for executada em vários operadores, uma entidade poderá passar do controle de um aplicativo espacial em um operador para um aplicativo espacial em outro operador. Quando uma entidade se move para um operador diferente, o SimSpace Weaver lida com a comunicação de rede subjacente.

Assinaturas

A visão de mundo de um aplicativo espacial é sua própria área de propriedade. Para descobrir o que está acontecendo em outra parte do mundo da simulação, o aplicativo espacial cria uma assinatura. A área de assinatura é um subconjunto da área geral do mundo da simulação. Uma área de assinatura pode incluir partes de várias áreas de propriedade, incluindo a própria área de propriedade do aplicativo espacial. O SimSpace Weaver notifica o aplicativo espacial sobre todos os eventos da entidade (por exemplo, entrar, sair, criar, atualizar e excluir) que ocorrem na área de assinatura.



A visão do mundo de um aplicativo espacial



Uma visão espacial do aplicativo com uma área de assinatura adicional

A visão do mundo de um aplicativo espacial é sua área de propriedade, que é uma célula na grade do mundo.

Um aplicativo espacial usa uma assinatura para descobrir o que está acontecendo em outra parte do mundo da simulação. A área de assinatura pode conter várias células da grade e partes de células.

Por exemplo, um aplicativo que simula entidades interagindo fisicamente pode precisar conhecer entidades um pouco além dos limites espaciais de sua área de propriedade. Para isso, o aplicativo pode se inscrever em áreas que fazem fronteira com sua área de propriedade. Depois de criar a assinatura, o aplicativo recebe notificações sobre eventos de entidades nessas áreas e pode ler essas entidades. Outro exemplo é um veículo autônomo que precisa ver todas as entidades a 200 metros à sua frente, independente de qual aplicativo é dono da área. O aplicativo para o veículo pode criar uma assinatura com um filtro como uma caixa delimitadora alinhada ao eixo (AABB) que cobre a área visível.

É possível criar uma lógica de simulação que não seja responsável por gerenciar os aspectos espaciais da simulação. Um aplicativo personalizado é um programa executável executado em um único operador. Você controla o ciclo de vida (início e término) de um aplicativo personalizado. Os clientes de simulação podem se conectar a um aplicativo personalizado para visualizar ou interagir com a simulação. Também é possível criar um aplicativo de serviço que seja executado em cada operador. O SimSpace Weaver inicia uma instância do aplicativo de serviço em cada operador que executa a simulação.

Aplicativos personalizados e de serviço criam assinaturas para aprender sobre eventos de entidades e ler essas entidades. Esses aplicativos não têm áreas de propriedade porque não são espaciais. Usar uma assinatura é a única maneira de descobrir o que está acontecendo no mundo da simulação.

Como usar o SimSpace Weaver

Estas são as principais etapas a seguir ao usar o SimSpace Weaver:

1. Criar e gravar aplicativos C++ que integrem o aplicativo do SDK do SimSpace Weaver.
 - a. Seus aplicativos fazem chamadas de API para interagir com o estado da simulação.
2. Criar clientes que visualizem e interajam com sua simulação por meio de alguns aplicativos.
3. Configurar a simulação em um arquivo de texto.
4. Carregar os pacotes de aplicativos e da configuração da simulação para o serviço.

5. Iniciar a simulação.
6. Iniciar e interromper aplicativos personalizados conforme necessário.
7. Conectar clientes aos seus aplicativos personalizados ou de serviço para visualizar ou interagir com a simulação.
8. Verificar os logs de simulação no Amazon CloudWatch Logs.
9. Interromper a simulação.
10. Limpar a simulação.

Esquema de simulação

O esquema de simulação (ou esquema) é um arquivo de texto YAML formatado que contém informações de configuração para sua simulação. O SimSpace Weaver usa o esquema quando inicia uma simulação. O pacote distribuível do aplicativo do SDK do SimSpace Weaver inclui um esquema para um projeto de exemplo. Use esse cliente como um ponto de partida para seus próprios aplicativos. Para obter mais informações sobre o esquema da simulação, consulte [SimSpace Weaver referência do esquema de simulação](#).

Trabalhadores e unidades de recursos

Um operador é uma instância do Amazon EC2 que executa sua simulação. Você especifica um tipo de operador em seu esquema de simulação. O SimSpace Weaver mapeia o tipo de operador para um tipo específico de instância do Amazon EC2 que o serviço usa. O SimSpace Weaver inicia e interrompe seus operadores para você e gerencia a comunicação de rede entre os operadores. O SimSpace Weaver inicia um conjunto de operadores para cada simulação. Simulações diferentes usam operadores distintos.

A capacidade computacional disponível (processador e memória) em um operador é dividida em unidades lógicas chamadas unidades de recursos computacionais ou unidades de recursos. Uma unidade de recurso representa uma quantidade fixa de capacidade de processador e memória.

Note

Anteriormente, nos referíamos a uma unidade de recursos computacionais como um slot. Talvez você ainda veja esse termo em nossa documentação.

Relógio da simulação

Cada simulação tem seu próprio relógio. Inicia e interrompa o relógio usando chamadas de API ou o console do SimSpace Weaver. A simulação é atualizada somente quando o relógio está funcionando. Todas as operações na simulação ocorrem em segmentos de tempo chamados marcações. O relógio anuncia a hora de início de cada marcação para todos os operadores.

A taxa de clock ou taxa de atualização é o número de marcações por segundo (hertz ou Hz) que o relógio anuncia. A taxa de clock desejada para uma simulação faz parte do esquema da simulação. Todas as operações de uma marcação devem ser concluídas antes que a próxima marcação comece. Por esse motivo, a taxa de clock efetiva pode ser menor do que a taxa de clock desejada. A taxa de relógio efetiva não será maior do que a taxa de relógio desejada.

Partições

Uma partição é um segmento da memória compartilhada em um operador. Cada partição contém parte dos dados do estado da simulação.

Uma partição para um aplicativo espacial (também chamada partição espacial de aplicativo ou partição espacial) contém todas as entidades na área de propriedade de um aplicativo espacial. O SimSpace Weaver coloca entidades em partições espaciais de aplicativos com base na localização espacial de cada entidade. Isso significa que o SimSpace Weaver tenta posicionar entidades espacialmente próximas no mesmo operador. Isso minimiza a quantidade de conhecimento que um aplicativo exige de entidades que ele não possui para simular as entidades que ele possui.

State Fabric

O State Fabric é o sistema de memória compartilhada (a coleção de todas as partições) em todos os operadores. Ele contém todos os dados de estado da sua simulação.

O State Fabric usa um formato binário personalizado que descreve uma entidade como um conjunto de dados iniciais e um log de atualização para cada campo de dados dessa entidade. Com esse formato, você pode acessar o estado de uma entidade em um ponto anterior no tempo de simulação e mapeá-la de volta a um ponto no tempo real. O buffer tem um tamanho finito e não é possível voltar no tempo além do que está no buffer. O SimSpace Weaver usa um ponteiro para o deslocamento atual no registro de atualização de cada campo e atualiza um ponteiro como parte da atualização de campo. O SimSpace Weaver mapeia esses registros de atualização no espaço de processo de um aplicativo usando memória compartilhada.

Esse formato de objeto resulta em baixa sobrecarga e não tem custos de serialização. O SimSpace Weaver também usa esse formato de objeto para analisar e identificar campos de índice, como a posição da entidade.

Entidades

Uma entidade é o menor bloco de construção de dados na simulação. Exemplos de entidades incluem atores, como pessoas e veículos, e objetos estáticos, como edifícios e obstáculos. As entidades têm propriedades (como posição e orientação) que podem ser armazenadas como dados persistentes no SimSpace Weaver. As entidades existem dentro das partições.

Apps

Um SimSpace Weaver aplicativo é um software que você escreve e que contém uma lógica personalizada executada a cada verificação da simulação. O objetivo da maioria dos aplicativos é atualizar entidades à medida que a simulação é executada. Seus aplicativos chamam APIs no aplicativo do SDK do SimSpace Weaver para realizar ações (como leitura e atualização) em entidades na simulação.

Empacote os aplicativos e recursos necessários (como bibliotecas) como arquivos .zip e carregue-os no SimSpace Weaver. Um aplicativo é executado em um contêiner do Docker em um operador. O SimSpace Weaver aloca para cada aplicativo um número fixo de unidades de recursos no operador.

O SimSpace Weaver atribui a propriedade de uma (e somente uma) partição a cada aplicativo. Um aplicativo e sua partição estão localizados no mesmo operador. Cada partição tem apenas um proprietário do aplicativo. Um aplicativo pode criar, ler, atualizar e excluir entidades em sua partição. Um aplicativo é proprietário de todas as entidades em sua partição.

Há três tipos de aplicativos: espaciais, personalizados e de serviço. Eles diferem por casos de uso e ciclos de vida.

Note

No SimSpace Weaver, o termo “aplicativo” pode se referir ao código de um aplicativo ou de uma instância em execução desse código.

Aplicativos espaciais

Os aplicativos espaciais atualizam o estado das entidades que existem espacialmente em sua simulação. Por exemplo, você pode definir um aplicativo `Physics` responsável por mover e agrupar entidades para cada marcação com base em velocidade, forma e tamanho. Nesse caso, o SimSpace Weaver executa várias instâncias do aplicativo `Physics` em paralelo para lidar com o tamanho da carga de trabalho.

O SimSpace Weaver gerencia o ciclo de vida dos aplicativos espaciais. Especifique uma organização de partições espaciais do aplicativo no esquema de simulação. Ao iniciar a simulação, o SimSpace Weaver inicia um aplicativo espacial para cada partição do aplicativo espacial. Ao interromper a simulação, o SimSpace Weaver encerra os aplicativos espaciais.

Outros tipos de aplicativos podem criar entidades, mas somente aplicativos espaciais podem atualizá-las. Outros tipos de aplicativos devem transferir entidades que eles criam para um domínio espacial. O SimSpace Weaver usa a localização espacial de uma entidade para movê-la para a partição de um aplicativo espacial. Isso transfere a propriedade da entidade para o aplicativo espacial.

Aplicativos personalizados

Use aplicativos personalizados para interagir com a simulação. Um aplicativo personalizado lê os dados da entidade usando assinaturas. Um aplicativo personalizado pode criar entidades. No entanto, o aplicativo deve transferir uma entidade para um aplicativo espacial para incluir a entidade na simulação e atualizá-la. O SimSpace Weaver pode atribuir um endpoint de rede a um aplicativo personalizado. Os clientes de simulação podem se conectar ao endpoint da rede para interagir com a simulação. Defina os aplicativos personalizados no esquema de simulação, mas é responsável por iniciá-los e interrompê-los (usando chamadas de API do SimSpace Weaver). Depois de iniciar uma instância de aplicativo personalizada em um operador, o SimSpace Weaver não transfere a instância para outro operador.

Aplicativos de serviço

Use um aplicativo de serviço quando precisar de um processo somente para leitura em execução em cada operador. Por exemplo, use um aplicativo de serviço se tiver uma simulação grande e precisar de um cliente de visualização que a percorra e exiba somente as entidades visíveis para o usuário. Nesse caso, uma única instância de aplicativo personalizado não pode processar todas as entidades na simulação. É possível configurar um aplicativo de serviço para ser executado em cada operador.

Cada um desses aplicativos de serviço pode então filtrar as entidades no operador designado e enviar somente as entidades relevantes para os clientes conectados. O cliente de visualização pode então se conectar a diferentes aplicativos de serviço à medida que se move pelo espaço de simulação. Configure aplicativos de serviço no esquema de simulação. O SimSpace Weaver inicia e interrompe os aplicativos de serviço para você.

Resumo de APIs

A tabela a seguir resume as características dos diferentes tipos de aplicativos do SimSpace Weaver.

	Aplicativos espaciais	Aplicativos personalizados	Aplicativos de serviço
Ler entidades	Sim	Sim	Sim
Atualizar entidades	Sim	Não	Não
Criar entidades	Sim	Sim	Sim
Ciclo de vida	Gerenciado (o SimSpace Weaver o controla)	Não gerenciado (você o controla)	Gerenciado (o SimSpace Weaver o controla)
Método de início	O SimSpace Weaver inicia uma instância do aplicativo para cada partição espacial, conforme especificado no esquema.	Inicie cada instância do aplicativo.	O SimSpace Weaver inicia uma ou mais instâncias do aplicativo o em cada operador, conforme especificado no esquema.
Os clientes podem se conectar	Não	Sim	Sim

*Quando um aplicativo personalizado ou de serviço cria uma entidade, deve transferir a propriedade da entidade para um aplicativo espacial para que este atualize seu estado.

Domínios

Um domínio do SimSpace Weaver é uma coleção de instâncias de aplicativos que executam o mesmo código de aplicativo executável e têm as mesmas opções e comandos de execução. Nos referimos aos domínios pelos tipos de aplicativos que eles contêm: domínios espaciais, domínios personalizados e domínios de serviço. Configure seus aplicativos em domínios.

Assinaturas e replicação

Um aplicativo cria uma assinatura em uma região espacial para aprender eventos de entidades (por exemplo, entrar, sair, criar, atualizar e excluir) nessa região. Um aplicativo processa eventos de entidade de uma assinatura antes de ler dados de entidades em partições que não são de sua propriedade.

Uma partição pode existir no mesmo operador do aplicativo (chamada de partição local), mas outro aplicativo pode ser proprietário da partição. Uma partição também pode existir em um operador diferente (chamada de partição remota). Se a assinatura for para uma partição remota, o operador cria uma cópia local dela com um processo chamado replicação. Em seguida, o operador lê a cópia local (partição remota replicada). Se outro aplicativo no operador precisar ler dessa partição com a mesma marcação, o operador lerá a mesma cópia local.

Exemplos de uso do SimSpace Weaver

Você pode usar o SimSpace Weaver para modelos baseados em agentes e simulações de intervalos de tempo discretos com um componente espacial.

Criar simulações de grandes multidões

Você pode usar o SimSpace Weaver para simular multidões em ambientes do mundo real. O SimSpace Weaver permite que você escale suas simulações para milhões de objetos dinâmicos com seus próprios comportamentos.

Criar ambientes em escala urbana

Usar o SimSpace Weaver para criar um gêmeo digital de uma cidade inteira. Criar simulações para planejamento urbano, projetar rotas de tráfego e planejar a resposta a riscos ambientais. Você pode usar suas próprias fontes de dados geoespaciais como blocos de construção para seus ambientes.

Criar experiências imersivas e interativas

Crie experiências de simulação nas quais vários usuários possam participar e interagir. Usar ferramentas de desenvolvimento populares, como Unreal Engine e Unity, para criar mundos virtuais tridimensionais (3D). Personalizar a experiência 3D com conteúdo e comportamentos próprios.

Configuração do SimSpace Weaver

Antes de usar o SimSpace Weaver pela primeira vez, você deve configurar o ambiente da Conta da AWS e o ambiente local. Quando concluir essas tarefas, você estará pronto para os [Tutoriais de conceitos básicos](#).

Tarefas de configuração

1. [Configure seu Conta da AWS para usar SimSpace Weaver](#).
2. [Configurar o ambiente local para o SimSpace Weaver](#).

Configure seu Conta da AWS para usar SimSpace Weaver

Conclua as tarefas a seguir Conta da AWS para configurar seu uso SimSpace Weaver.

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a <https://portal.aws.amazon.com/billing/inscrição>.
2. Siga as instruções online.

Parte do procedimento de inscrição envolve receber uma chamada telefônica e inserir um código de verificação no teclado do telefone.

Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWS é criado. O usuário-raiz tem acesso a todos os Serviços da AWS e atributos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar [tarefas que exigem acesso de usuário-raiz](#).

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, você pode visualizar a atividade atual da sua conta e gerenciar sua conta acessando <https://aws.amazon.com/e> escolhendo Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login [AWS Management Console](#) como proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira sua senha.

Para obter ajuda ao fazer login usando o usuário raiz, consulte [Fazer login como usuário raiz](#) no Guia do usuário do Início de Sessão da AWS .

2. Ative a autenticação multifator (MFA) para seu usuário root.

Para obter instruções, consulte [Habilitar um MFA dispositivo virtual para seu usuário Conta da AWS root \(console\)](#) no Guia IAM do usuário.

Criar um usuário com acesso administrativo

1. Ative o IAM Identity Center.

Para obter instruções, consulte [Habilitar AWS IAM Identity Center](#) no Guia do usuário do AWS IAM Identity Center .

2. No IAM Identity Center, conceda acesso administrativo a um usuário.

Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte [Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM](#) no Guia AWS IAM Identity Center do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para entrar com seu usuário do IAM Identity Center, use o login URL que foi enviado ao seu endereço de e-mail quando você criou o usuário do IAM Identity Center.

Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte [Como fazer login no portal de AWS acesso](#) no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No IAM Identity Center, crie um conjunto de permissões que siga as melhores práticas de aplicação de permissões com privilégios mínimos.

Para obter instruções, consulte [Create a permission set](#) no Guia do usuário do AWS IAM Identity Center .

2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

Para obter instruções, consulte [Add groups](#) no Guia do usuário do AWS IAM Identity Center .

Adicionar permissões para usar o SimSpace Weaver

Para conceder acesso, adicione as permissões aos seus usuários, grupos ou perfis:

- Usuários e grupos em AWS IAM Identity Center:

Crie um conjunto de permissões. Siga as instruções em [Criação de um conjunto de permissões](#) no Guia do usuário do AWS IAM Identity Center .

- Usuários gerenciados IAM por meio de um provedor de identidade:

Crie um perfil para a federação de identidades. Siga as instruções em [Criar uma função para um provedor de identidade terceirizado \(federação\)](#) no Guia IAM do usuário.

- IAMusuários:

- Crie um perfil que seu usuário possa assumir. Siga as instruções em [Criar uma função para um IAM usuário](#) no Guia do IAM usuário.
- (Não recomendado) Vincule uma política diretamente a um usuário ou adicione um usuário a um grupo de usuários. Siga as instruções em [Adicionar permissões a um usuário \(console\)](#) no Guia do IAM usuário.

Example IAMpolítica para conceder permissões de uso SimSpace Weaver

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateAndRunSimulations",
      "Effect": "Allow",
```

```

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

```


Configurar o ambiente local para o SimSpace Weaver

Simulações do SimSpace Weaver executadas em ambientes containerizados do Amazon Linux 2 (AL2). Você deve ter um ambiente AL2 para compilar e vincular seus aplicativos ao aplicativo do SDK do SimSpace Weaver. O ambiente de desenvolvimento local padrão é um contêiner do AL2 no Docker. Se você optar por não usar o Docker, forneceremos instruções alternativas para executar um ambiente do AL2 no Windows Subsystem for Linux (WSL). Também é possível usar seu próprio método para criar um ambiente local do AL2. Para outras formas de executar o AL2 localmente, consulte a documentação do [Amazon EC2](#).

Important

O Docker no Microsoft Windows é o ambiente de desenvolvimento padrão. Para sua conveniência, sugerimos outras formas de configurar seu ambiente de desenvolvimento local, mas elas não são padrão e não têm suporte.

Tópicos

- [Configure o pacote SimSpace Weaver de distribuição para Amazon Linux 2 \(AL2\) em Docker](#)
- [Configure o pacote SimSpace Weaver de distribuição para Amazon Linux 2 \(AL2\) em Windows Subsystem for Linux \(WSL\)](#)

Configure o pacote SimSpace Weaver de distribuição para Amazon Linux 2 (AL2) em Docker

Esta seção fornece instruções para configurar seu zip de SimSpace Weaver distribuição local com um ambiente AL2 em Docker. Para obter instruções sobre como configurar com o AL2 Windows Subsystem for Linux (WSL), consulte [Configure o pacote SimSpace Weaver de distribuição para Amazon Linux 2 \(AL2\) em Windows Subsystem for Linux \(WSL\)](#).

Requisitos

- Microsoft Windows 10 ou superior, ou um sistema Linux compatível
- [Microsoft Visual Studio 2019](#) ou posterior, com workload instalada para [Desktop development with C++](#)
- [CMake3](#)

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

Para configurar o zip SimSpace Weaver de distribuição com AL2 em Docker

1. Se você ainda não configurou suas AWS credenciais para o AWS CLI, siga estas instruções: [Configurando a AWS CLI](#).
2. [Baixe o pacote distribuível do SDK do SimSpace Weaver aplicativo](#). Ele contém o seguinte:
 - Binários e bibliotecas para desenvolvimento de SimSpace Weaver aplicativos
 - Scripts de ajuda que automatizam partes do fluxo de trabalho de desenvolvimento
 - Exemplos de aplicativos que demonstram SimSpace Weaver conceitos
3. Descompacte o arquivo em uma *sdk-folder* de sua escolha.
4. Acesse o *sdk-folder*.
5. Digite o comando a seguir para instalar os pacotes Python necessários:

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

6. Insira o comando a seguir para configurar a SimSpace Weaver distribuição com uma imagem do Docker.

```
python setup.py
```

Esse comando faz o seguinte:

- Cria uma imagem docker AL2 com todos os requisitos para projetos de SimSpace Weaver construção instalados.
- Cria os CloudFormation recursos necessários para iniciar uma simulação.
 - O modelo de CloudFormation pilha de amostra pode ser encontrado em *sdk-folder/PackagingTools/sample-stack-template.yaml*
- Configura os projetos de amostra fornecidos com os caminhos corretos para seu sistema local.

Solução de problemas

- O Docker parece preso
 - Se a saída do console parecer travada após a chamada dos comandos do Docker, tente reiniciar o mecanismo do Docker. Se isso não funcionar, reinicie o computador.

Configure o pacote SimSpace Weaver de distribuição para Amazon Linux 2 (AL2) em Windows Subsystem for Linux (WSL)

Esta seção fornece instruções para configurar seu zip SimSpace Weaver de distribuição com um ambiente AL2 em Windows Subsystem for Linux (WSL). Para obter instruções de configuração do AL2 no Docker, consulte [Configure o pacote SimSpace Weaver de distribuição para Amazon Linux 2 \(AL2\) em Docker](#).

Important

Esta seção descreve uma solução que usa uma versão do AL2 que não é de propriedade, desenvolvida ou compatível com a Amazon. Essa solução é fornecida somente para a sua conveniência, caso você opte por não usar o Docker. A Amazon não AWS assume nenhuma responsabilidade se você optar por usar essa solução.

Requisitos

- [Hyper-V no Windows 10](#)
- [Windows Subsystem for Linux \(WSL\)](#)
- Distribuição do AL2 de código aberto de terceiros para WSL ([download version 2.0.20200722.0-update.2](#)) (Consulte [Instruções](#))

Important

Nossas instruções WSL usam a versão [2.0.20200722.0-update.2](#) da distribuição AL2 para WSL. Pode haver erros ao usar qualquer outra versão.

Para configurar o zip de SimSpace Weaver distribuição com AL2 em WSL

1. Em um prompt de comando do Windows, inicie seu ambiente AL2 no WSL.

```
wsl -d Amazon2
```

Important

Enquanto estiver executando WSL, inclua a `--al2` opção ao executar um dos scripts auxiliares do `quick-start.py` Python localizados em `sdky-folder/Samples/sample-name/tools/cloud/quick-start.py`

2. Em um prompt de shell do Linux, atualize seu gerenciador de pacotes yum.

```
yum update -y
```

Important

Se essa etapa atingir o tempo limite, talvez seja necessário alternar para o WSL1 e repetir os procedimentos. Saia da sessão do AL2 do WSL e digite o seguinte no prompt de comando do Windows:

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

3. Instale a ferramenta de descompactação.

```
yum install -y unzip
```

4. Remova qualquer um AWS CLI que yum esteja instalado. Experimente os dois comandos a seguir se não tiver certeza se yum instalou um AWS CLI.

```
yum remove awscli
```

```
yum remove aws-cli
```

5. Crie um diretório temporário e vá até ele.

```
mkdir ~/temp
```

```
cd ~/temp
```

6. Baixe e instale o AWS CLI:

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

7. É possível remover o diretório temporário.

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

8. Reinicie a sessão do shell para atualizar o caminho no ambiente.

```
exec
```

9. Configure suas AWS credenciais para o AWS CLI em seu ambiente AL2. Para obter mais informações, consulte [Configuração da AWS CLI](#). Se você usa AWS IAM Identity Center, consulte [Configurando o AWS CLI para uso AWS IAM Identity Center](#) no Guia do AWS Command Line Interface usuário.

```
aws configure
```

10. Instale o Git.

```
yum install -y git
```

11. Instalar o wget.

```
yum install -y wget
```

12. Crie uma pasta para o SDK do SimSpace Weaver aplicativo.

```
mkdir sdk-folder
```

13. Acesse sua pasta do SDK.

```
cd sdk-folder
```

14. Baixe o pacote distribuível do SDK do SimSpace Weaver aplicativo. Ele contém o seguinte:

- Binários e bibliotecas para desenvolvimento de SimSpace Weaver aplicativos
- Scripts de ajuda que automatizam partes do fluxo de trabalho de desenvolvimento
- Exemplos de aplicativos que demonstram SimSpace Weaver conceitos

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

15. Descompacte o arquivo.

```
unzip *.zip
```

16. Execute o script de configuração do WSL.

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

17. Digite o comando a seguir para instalar os pacotes Python necessários:

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

18. Execute o script de configuração do zip de SimSpace Weaver distribuição:

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

Esse comando faz o seguinte:

- Cria os CloudFormation recursos necessários para iniciar uma simulação.
 - O modelo de CloudFormation pilha de amostra pode ser encontrado em *sdk-folder/* `PackagingTools/sample-stack-template.yaml`
- Configura os projetos de amostra fornecidos com os caminhos corretos para seu sistema local.

Note

Você só precisa fazer isso uma vez para seu ambiente AL2 no WSL.

Uso de software licenciado com o AWS SimSpace Weaver

O AWS SimSpace Weaver permite criar simulações com o conteúdo e o mecanismo de simulação de sua escolha. Em relação ao uso do SimSpace Weaver, você é responsável por obter, manter e aderir aos termos de licença de qualquer software ou conteúdo usado em suas simulações. Verifique se o contrato de licenciamento permite implantar o software e o conteúdo em um ambiente de host virtual.

Começando com SimSpace Weaver

Esta seção fornece tutoriais para ajudar você a começar. SimSpace Weaver Esses tutoriais apresentam o fluxo de trabalho geral para criar simulações com. SimSpace Weaver Esses tutoriais demonstram como criar, implantar e executar simulações no. SimSpace Weaver Recomendamos que você comece com o tutorial de início rápido para executar uma simulação em minutos. Depois disso, consulte os outros tutoriais para saber mais.

[Esses tutoriais usam um aplicativo de amostra \(PathfindingSample\) incluído no arquivo.zip do SDK do SimSpace Weaver aplicativo que você baixou durante os procedimentos de configuração.](#)

O aplicativo de amostra demonstra os conceitos que todas as SimSpace Weaver simulações compartilham, incluindo particionamento espacial, transferência de entidades entre partições, aplicativos e assinaturas.

Nos tutoriais, você criará uma simulação com quatro partições espaciais. Uma instância separada do aplicativo espacial PathfindingSample gerencia cada partição individual. Os aplicativos espaciais criam entidades em suas próprias partições. As entidades se movem para uma posição específica no mundo da simulação, evitando obstáculos à medida que se movem. Você pode usar um aplicativo cliente separado (incluído no SDK do SimSpace Weaver aplicativo) para visualizar a simulação.

Tópicos

- [Tutorial de início rápido para SimSpace Weaver](#)
- [Tutorial detalhado: conheça os detalhes ao criar o aplicativo de exemplo](#)

Tutorial de início rápido para SimSpace Weaver

Este tutorial orienta você pelo processo de criação e execução de uma simulação no SimSpace Weaver em apenas alguns minutos. Recomendamos que você comece com este tutorial e depois passe pelo [tutorial detalhado](#).

Requisitos

Antes de começar, você deve concluir as etapas em [Configuração do SimSpace Weaver](#).

Note

Os scripts usados aqui são fornecidos para sua conveniência e NÃO são obrigatórios. Veja o [tutorial detalhado](#) para saber como essas etapas podem ser executadas manualmente.

Etapa 1: ativar o registro (opcional)

Como ativar o registro

1. Navegue até:

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

2. Abra o arquivo do esquema em um editor de texto:

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

3. Encontre a seção `simulation_properties:` no início do arquivo:

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

4. Insira as duas linhas a seguir após a linha `simulation_properties::`

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

5. Confirme se a seção `simulation_properties:` é a mesma que a seguinte:

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

6. Salve o arquivo e saia do seu editor de texto.

Etapa 2: Início rápido com o cliente do console (opção 1)

Navegue até:

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

Execute um dos seguintes comandos:

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

Por padrão, isso iniciará uma simulação com uma única partição em um único trabalhador. Outras configurações podem ser iniciadas passando a `--schema {file name}.yaml` partir da `/Samples/PathfindingSample/tools/` pasta.

Note

Consulte [Tutorial detalhado: conheça os detalhes ao criar o aplicativo de exemplo](#) para obter uma explicação detalhada sobre o que esse script faz.

Etapa 2: Início rápido com o cliente Unreal Engine (opção 2)

Consulte [Lançamento do cliente de visualização Unreal Engine](#).

Pare e exclua sua simulação

Navegue até:

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

Encontre os nomes das suas simulações:

```
aws simspaceweaver list-simulations
```

Pare e exclua a simulação

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

Solução de problemas

- `FileNotFoundException`: `cmake`

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

- **Resolução:** O script não consegue encontrar o comando `cmake`. Certifique-se de ter a versão mínima recomendada do CMake instalada e de que ela possa ser chamada com o `cmake` comando no PATH. Use o comando `cmake --version` para verificar.
- **ImportError:** Falha no carregamento da DLL ao importar `libweaver_app_sdk_python_v1`: O módulo especificado não foi encontrado.
 - **Resolução:** Esse erro ocorre quando o Python 3.9 não está sendo usado para iniciar o SDK Weaver Python. Certifique-se de que a versão python associada ao comando “python” seja Python 3.9. Você pode verificar executando o `python --version` comando.
- O script de início rápido parece travado após a inicialização do Docker Build.
 - **Resolução:** Às vezes, o Docker precisa de alguns minutos para se aquecer. Se esse problema persistir por mais de 5 minutos, reinicie o Docker ou seu sistema.
- `target_compile_features` nenhum recurso conhecido para o compilador CXX “GNU”:
 - **Resolução:** limpe o cache do Docker, exclua a imagem do Docker `weaverappbuilder`, exclua os artefatos de construção do projeto e execute novamente. `setup.py` Isso deve redefinir seu ambiente Docker e resolver o erro.

Tutorial detalhado: conheça os detalhes ao criar o aplicativo de exemplo

O [tutorial de início rápido](#) abordou como criar, iniciar, parar e excluir uma simulação de amostra usando `quick-start.py` `stop-and-delete.py` e. Este tutorial abordará detalhadamente como esses scripts funcionam e os parâmetros adicionais que esses scripts podem usar para maximizar a flexibilidade das simulações personalizadas do Weaver.

Requisitos

Antes de começar, você deve concluir as etapas em [Configuração do SimSpace Weaver](#).

Etapa 1: ativar o registro (opcional)

Como ativar o registro

1. Navegue até:

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

2. Abra o arquivo do esquema em um editor de texto:

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

3. Encontre a seção `simulation_properties`: no início do arquivo:

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

4. Insira as duas linhas a seguir após a linha `simulation_properties::`:

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

5. Confirme se a seção `simulation_properties`: é a mesma que a seguinte:

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

6. Salve o arquivo e saia do seu editor de texto.

Etapa 2: inicie sua simulação

Conforme mostrado no [tutorial de início rápido](#), as etapas mais básicas para iniciar uma simulação de amostra são:

1. Navegue até:

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

2. Execute um dos seguintes comandos:

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

Esse script automatiza comandos comuns do terminal, todos os quais podem ser executados manualmente usando o AWS CLI. Essas etapas são:

1. Faça o upload do esquema Weaver para o S3.
 - SimSpace Weaver usa um esquema para configurar sua simulação. O esquema é um arquivo de texto simples formatado em YAML. Para obter mais informações, consulte [Configurar a simulação](#).
2. Crie e faça upload de um contêiner personalizado (opcional).
 - Se seu esquema definir um contêiner personalizado, o script de início rápido criará a imagem do docker e a enviará para o Amazon ECR. Para ter mais informações, consulte [Contêineres personalizados](#). Consulte o PythonBubblesSample esquema para ver um exemplo desse recurso.
3. Crie o projeto.
 - `quick-start.py` chama a `build_project` função definida em `build.py`. Essa etapa variará de acordo com o projeto. Para o PathfindingSample, o CMake é usado. Os comandos CMake e Docker podem ser encontrados em `build.py`.
4. Faça o upload dos artefatos de construção para o S3.
 - Você pode verificar seus buckets do S3 para garantir que todos os uploads tenham sido bem-sucedidos. Para obter informações sobre como gerenciar arquivos com o Amazon S3, consulte [Criar, configurar e trabalhar com buckets do Amazon S3](#) no Guia do usuário do Amazon Simple Storage Service.
 - Os zips do aplicativo de exemplo e o bucket do S3 usam o seguinte formato de nome:
 - `weaver-sample-bucket-account-number-region`
 - Aplicativo espacial: `ProjectNameSpatial.zip`
 - Exibir aplicativo (personalizado): `ProjectNameView.zip`
5. Inicie a simulação.

- Este é um resumo da `aws simspaceweaver start-simulation` AWS CLI ligação. Para obter mais informações, consulte a [Referência de AWS CLI Comandos](#) para SimSpace Weaver.
- O script será repetido até que o status da simulação seja `STARTED` ou `FAILED`. Poderá levar alguns minutos para que a simulação seja iniciada.

6. Veja os detalhes da simulação.

- A API `DescribeSimulation` fornece detalhes sobre a simulação, incluindo o estado. Uma simulação pode estar em um dos seguintes estados:

Estados do ciclo de vida da simulação

1. **STARTING**: estado inicial após chamar `StartSimulation`
2. **STARTED**: todos os aplicativos espaciais são lançados e estão saudáveis
3. **STOPPING**: estado inicial após chamar `StopSimulation`
4. **STOPPED**: todos os recursos computacionais estão parados
5. **DELETING**: estado inicial após chamar `DeleteSimulation`
6. **DELETED**: todos os recursos atribuídos à simulação são excluídos
7. **FAILED**: a simulação teve um erro/falha crítica e foi interrompida
8. **SNAPSHOT_IN_PROGRESS**: um [snapshot](#) está em andamento

Como obter os detalhes da simulação

1. Chame a API `ListSimulations`.

```
aws simspaceweaver list-simulations
```

O script deve exibir detalhes sobre cada uma das simulações, semelhantes aos seguintes:

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

```
"TargetStatus": "STARTED"
}
```

2. Chame `DescribeSimulation` para obter os detalhes da simulação. Substitua *simulation-name* pelo Name da simulação a partir da saída da etapa anterior.

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

O script deve exibir mais detalhes sobre a simulação que você especificou, semelhantes aos seguintes:

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

7. Inicie aplicativos personalizados.

- SimSpace Weaver não gerencia o ciclo de vida de aplicativos personalizados. Você deve iniciar os aplicativos personalizados. É uma prática recomendada iniciar os aplicativos personalizados antes de iniciar o relógio de simulação, mas você também pode fazê-lo depois.

Você pode chamar a `StartApp` API para iniciar seus aplicativos personalizados.

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

A chamada de API `StartApp` criará e iniciará uma nova instância do aplicativo personalizado usando o nome que você fornecer. Se você fornecer o nome de um aplicativo que já existe, receberá um erro. Se quiser reiniciar um aplicativo específico (instância), primeiro interrompa-o e depois exclua-o.

Note

O status da sua simulação deve ser STARTED antes de iniciar aplicativos personalizados.

O aplicativo de exemplo fornece o aplicativo ViewApp personalizado para visualizar sua simulação. Este aplicativo fornece um endereço IP estático e um número de porta para conectar os clientes de simulação (você fará isso em uma etapa posterior deste tutorial). Você pode pensar em um domain como uma classe de aplicativos com o mesmo código executável e as mesmas opções de inicialização. O app name identifica a instância do aplicativo. Para obter mais informações sobre SimSpace Weaver conceitos, consulte [Conceitos principais do SimSpace Weaver](#).

É possível usar a API DescribeApp para verificar o status de um aplicativo personalizado depois de iniciá-lo.

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

Como iniciar o aplicativo de visualização neste tutorial

1. Ligue StartApp para ViewApp.

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

2. Chame DescribeApp para verificar o status do seu aplicativo personalizado.

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

Depois que o status do seu aplicativo personalizado (instância) for STARTED, a saída de DescribeApp incluirá o endereço IP e o número da porta desse aplicativo personalizado (instância). No exemplo da saída a seguir, o endereço IP é o valor de Address e o número da porta é o valor de Actual no bloco 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


O valor de `Declared` é o número da porta à qual o código do seu aplicativo deve ser vinculado. O valor de `Actual` é o número da porta que é SimSpace Weaver exposta aos clientes para se conectarem ao seu aplicativo. SimSpace Weaver mapeia a `Declared` porta até a `Actual` porta.

Note

Você pode usar o procedimento descrito em [Obtenha o endereço IP e o número da porta de um aplicativo personalizado](#) para obter o endereço IP e o número da porta de qualquer aplicativo personalizado iniciado.

8. Inicie o relógio.


- Quando você cria uma simulação pela primeira vez, ela tem um relógio, mas ele não está funcionando. Quando o relógio não está funcionando, a simulação não atualiza seu estado. Depois de iniciar o relógio, ele começará a enviar marcações para os aplicativos. A cada clique, seus aplicativos espaciais examinam as entidades que possuem e enviam os resultados para SimSpace Weaver

 Note

Pode levar de 30 a 60 segundos para iniciar o relógio.

Chame a API StartClock.

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

 Note

A API StartClock usa o *simulation-name*, que você pode encontrar usando a API ListSimulations:

```
aws simspaceweaver list-simulations
```

parâmetros de início rápido

- -h, --help
 - Liste esses parâmetros.
- --limpar
 - Exclua o conteúdo do diretório de compilação antes de compilar.
- --al2
 - É construído diretamente na máquina nativa em vez do Docker. Use isso somente se estiver executando em um ambiente Amazon Linux 2, como WSL.
- --somente upload
 - Faça o upload apenas do esquema e dos zips do aplicativo para o Amazon S3, não inicie a simulação.

- `--sem construção`
 - Pule a reconstrução do projeto.
- `--sem contêiner`
 - Ignore a reconstrução do contêiner de simulação listado no esquema.
- `--cliente do console`
 - Crie e conecte automaticamente o cliente do console listado em `config.py`.
- `--esquema esquema`
 - Qual esquema essa invocação usará. O padrão é o valor de 'SCHEMA' em `config.py`.
- `--name NAME`
 - Qual nome a simulação terá. O padrão é o valor de 'project_name'-date-time em `config.py`.

Etapa 3: verifique os registros (opcional)

SimSpace Weaver grava mensagens de gerenciamento de simulação e a saída do console de seus aplicativos no Amazon CloudWatch Logs. Para obter mais informações sobre como trabalhar com registros, consulte [Trabalho com grupos de registros e fluxos de registros](#) no Guia do usuário do Amazon CloudWatch Logs.

Cada simulação que você cria tem seu próprio grupo de CloudWatch registros em Registros. O nome do grupo de logs é especificado no esquema de simulação. No trecho de esquema a seguir, o valor de `log_destination_service` é `logs`. Isso significa que o valor de `log_destination_resource_name` é o nome de um grupo de logs. Nesse caso, o grupo de logs é `MySimulationLogs`.

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

Você também pode usar a API `DescribeSimulation` para encontrar o nome do grupo de logs para simulação depois de iniciá-lo.

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

O exemplo a seguir mostra a parte da saída de DescribeSimulation que descreve a configuração de registro. O nome do grupo de logs é mostrado no final do LogGroupArn.

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

Cada grupo de logs de simulação contém vários fluxos de logs:

- Fluxo de registros de gerenciamento — mensagens de gerenciamento de simulação produzidas pelo SimSpace Weaver serviço.

```
/sim/management
```

- Fluxo de registro de erros — mensagens de erro produzidas pelo SimSpace Weaver serviço. Esse fluxo de log só existe se houver erros. SimSpace Weaver armazena erros gravados por seus aplicativos em seus próprios fluxos de registro de aplicativos (consulte os fluxos de registro a seguir).

```
/sim/errors
```

- Fluxos de logs de aplicativos espaciais (um para cada aplicativo espacial em cada operador): saída do console produzida por aplicativos espaciais. Cada aplicativo espacial grava em seu próprio fluxo de logs. O *spatial-app-id* são todos os caracteres após o delimitador no final do *worker-id*.

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

- Fluxos de logs de aplicativos personalizados (um para cada instância de aplicativo personalizado): saída de console produzida por aplicativos personalizados. Cada instância do aplicativo personalizado grava em seu próprio fluxo de logs.

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

- Fluxos de logs do aplicativo de serviço (um para cada instância do aplicativo de serviço): saída do console produzida pelos aplicativos de serviço. Cada aplicativo de serviço grava em seu próprio fluxo de logs. O *service-app-id* são todos os caracteres após o delimitador no final do *service-app-name*.

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

Note

O aplicativo de exemplo não tem aplicativos de serviço.

Etapa 4: veja sua simulação

O SDK do SimSpace Weaver aplicativo fornece opções diferentes para visualizar o aplicativo de amostra. Você pode usar o cliente de console de amostra se não tiver suporte local para o desenvolvimento do Unreal Engine. As instruções para o cliente Unreal Engine pressupõem que você esteja usando o Windows.

O cliente do console exibe uma lista dos eventos da entidade à medida que eles ocorrem. O cliente obtém as informações do evento da entidade do ViewApp. Se o cliente do console exibir a lista de eventos, ele confirmará a conectividade da rede com a atividade do ViewApp e na simulação.

A simulação PathfindingSample cria entidades estacionárias e móveis em um plano bidimensional. As entidades móveis se movem em torno das entidades estacionárias. O cliente do Unreal Engine fornece uma visualização dos eventos da entidade.

Cliente de console

O cliente do console pode ser criado e conectado automaticamente ao iniciar uma amostra, `quick-start.py` se você incluir a `--consoleclient` opção. Para criar e conectar o cliente do console depois de `quick-start.py` já ter sido chamado, faça o seguinte:

Navegue até:

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

Execute o script para criar e conectar o cliente:

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

O script fará o seguinte:

1. Crie o cliente do console com o CMake.
2. Inicie o executável construído com o endereço IP e o número da porta fornecidos.

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

Cliente Unreal Engine

Consulte [Lançamento do cliente de visualização Unreal Engine](#).

Etapa 5: Pare e exclua sua simulação

Navegue até:

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

Encontre os nomes das suas simulações:

```
aws simspaceweaver list-simulations
```

Pare e exclua a simulação:

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

O script `stop-and-delete.py` fará o seguinte:

1. Chame o AWS CLI comando para interromper uma simulação.
 - `aws simspaceweaver stop-simulation`
 - Para obter mais informações, consulte [Referência de AWS CLI comandos](#) para SimSpace Weaver.

2. Chame o AWS CLI comando para excluir uma simulação.

- `aws simpaceweaver delete-simulation`
- Para obter mais informações, consulte [Referência de AWS CLI comandos](#) para SimSpace Weaver.

stop-and-delete parâmetros

- `-h, --help`
 - Liste esses parâmetros.
- `--simulação SIMULAÇÃO`
 - O nome da simulação para stop-and-delete
- `--parar`
 - Apenas pare a simulação. Não o exclui.
- `--excluir`
 - Exclua somente uma simulação. Só funcionará se a simulação for uma STOPPED ou outra FAILED.

Solução de problemas

Veja [Solução de problemas](#) no tutorial de início rápido.

Trabalhando com SimSpace Weaver

Este capítulo fornece informações e orientações para ajudá-lo a criar seus próprios aplicativos no SimSpace Weaver.

Tópicos

- [Configurar a simulação](#)
- [Duração máxima de uma simulação](#)
- [Desenvolver aplicativos](#)
- [Desenvolver aplicativos cliente](#)
- [Obtenha o endereço IP e o número da porta de um aplicativo personalizado](#)
- [Lançamento do cliente de visualização Unreal Engine](#)
- [Desenvolvimento local em SimSpace Weaver](#)
- [Aplicativo do SDK do AWS SimSpace Weaver](#)
- [Estrutura de demonstração do AWS SimSpace Weaver](#)
- [Trabalhar com cotas de serviço](#)
- [Depurar simulações](#)
- [Contêineres personalizados](#)
- [Como trabalhar com Python](#)
- [Suporte para outros mecanismos](#)
- [Uso de software licenciado com AWS SimSpace Weaver](#)
- [Gerenciar recursos com o AWS CloudFormation](#)
- [Snapshots](#)
- [Sistema de mensagens](#)

Configurar a simulação

Um esquema de simulação (ou esquema) é um arquivo de texto YAML formatado que especifica a configuração de uma simulação. É possível usar o mesmo esquema para iniciar várias simulações. O arquivo do esquema está localizado na pasta do projeto para a simulação. Você pode usar qualquer editor de texto para editar o arquivo. SimSpace Weaver só lê seu esquema quando

inicia a simulação. Todas as edições feitas em um arquivo de esquema afetam somente as novas simulações, iniciadas após as edições.

Para configurar sua simulação, edite seu arquivo de esquema de simulação (use o separador de caminho apropriado para seu sistema operacional):

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

Carregue o esquema de simulação ao criar uma nova simulação. O script de ajuda de início rápido do projeto fará o carregamento do esquema como parte do processo para criar a simulação:

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

Para obter mais informações sobre como executar o script de início rápido, consulte o [Tutorial detalhado](#) [Conceitos básicos](#) capítulo deste guia.

Parâmetros de configuração da simulação

O esquema de simulação contém informações de inicialização, incluindo:

- Propriedades da simulação: versão do SDK e configuração computacional (tipo e número de [operadores](#))
- Relógios: taxa de atualização e tolerâncias
- Estratégias de particionamento espacial: topologia espacial (como a grade), limites e grupos de posicionamento (agrupamento de partição espacial em operadores)
- Domínios e seus aplicativos: bucket do aplicativo, caminho e comandos de inicialização

SimSpace Weaver usa sua configuração de esquema para configurar e organizar partições espaciais, iniciar aplicativos e avançar na simulação na taxa de ticks especificada.

Note

O script create-project no SDK do SimSpace Weaver aplicativo gerará automaticamente um esquema de simulação para você, com base no aplicativo de amostra.

Os tópicos a seguir descrevem os parâmetros no esquema de simulação. Para obter uma descrição completa do esquema de simulação, consulte [SimSpace Weaver referência do esquema de simulação](#).

Tópicos

- [Versão do SDK](#)
- [Propriedades da simulação](#)
- [Operadores](#)
- [Relógio](#)
- [Estratégias de particionamento](#)
- [Domínios](#)

Versão do SDK

O `sdk_version` campo especifica a versão para a SimSpace Weaver qual o esquema está formatado. Valores válidos: 1.17, 1.16, 1.15, 1.14, 1.13, 1.12

Important

O valor de `sdk_version` inclui somente o número da versão principal e o número da primeira versão secundária. Por exemplo, o valor 1.12 especifica todas as versões 1.12.x, como 1.12.0, 1.12.1, e 1.12.2.

Propriedades da simulação

A seção `simulation_properties` do esquema especifica a configuração de registro e um tipo de dados para o campo de índice (geralmente a localização espacial) das entidades.

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

O valor de `log_destination_service` determina a interpretação do valor de `log_destination_resource_name`. Atualmente, o único valor compatível é `logs`. Isso significa

que o valor de `log_destination_resource_name` é o nome de um grupo de logs no Amazon CloudWatch Logs.

Note

O registro é opcional. Se você não configurar as propriedades de destino do log, sua simulação não produzirá registros.

A propriedade `default_entity_index_key_type` é obrigatória. O único valor válido é `Vector3<f32>`.

Operadores

A seção `workers` especifica o tipo e o número de operadores desejados para a simulação. O SimSpace Weaver usa seus próprios tipos de operadores que são mapeados para os tipos de instância do Amazon EC2.

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

Ativar simulações com vários operadores

Você pode criar uma simulação que use mais de um operador. Por padrão, as simulações usam um operador. Você deve modificar o esquema de simulação antes de iniciar a simulação.

Note

Você não pode alterar uma simulação que já foi iniciada. Se você quiser habilitar vários operadores para uma simulação em execução, primeiro interrompa e exclua a simulação.

Para usar mais de um operador, defina o número `desired` de instâncias de computação para um valor maior que 1. Há um número máximo de aplicativos para cada operador. Para obter mais informações, consulte [SimSpace Terminais e cotas do Weaver](#). O SimSpace Weaver usará mais

de um operador apenas quando o número de aplicativos para um operador exceder esse limite. O SimSpace Weaver pode posicionar um aplicativo em qualquer um dos operadores disponíveis. O posicionamento do aplicativo em um funcionário específico não é garantido.

O trecho de esquema a seguir demonstra uma configuração para uma simulação que solicita dois operadores. O SimSpace Weaver tentará alocar o segundo operador se o número de aplicativos exceder o máximo de aplicativos para um operador.

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

Relógio

A seção `clock` especifica as propriedades do relógio da simulação. Atualmente, você só pode configurar a taxa de atualização (o número de marcações por segundo que o relógio envia aos aplicativos). A taxa de atualização é a taxa máxima. A taxa efetiva de marcações pode ser menor porque todas as operações (como atualizações de entidades) de uma marcação devem terminar antes que a próxima marcação possa começar. A taxa de atualização também é chamada de taxa de relógio.

Os valores válidos para a `tick_rate` dependem da `sdk_version` especificada em seu esquema.

Valores válidos para a taxa de atualização

- Versões anteriores a "1.14":
 - 10
 - 15
 - 30
- Versão "1.14" ou posterior.
 - "10"
 - "15"
 - "30"
 - "unlimited"

Para obter mais informações, consulte [Taxa de atualização ilimitada](#).

Important

- Para esquemas com uma `sdk_version` anterior a "1.14" o valor da `tick_rate` é um número inteiro, como 30.
- Para esquemas com uma `sdk_version` de "1.14" ou posterior, o valor da `tick_rate` é uma string, como "30". O valor deve incluir aspas duplas.

Se você converter um esquema com versão "1.12" ou "1.13" em versão "1.14" ou posterior, deverá colocar o valor da `tick_rate` entre aspas duplas.

Taxa de atualização ilimitada

Você pode definir a `tick_rate` para "unlimited" para permitir que a simulação seja executada tão rapidamente quanto seu código puder ser executado. Com uma taxa de atualização ilimitada, o SimSpace Weaver envia a próxima marcação imediatamente após todos os aplicativos concluírem as confirmações da marcação atual.

Important

A taxa de atualização ilimitada não é compatível com as versões do SimSpace Weaver anteriores a 1.14.0. O valor mínimo da `sdk_version` no esquema é "1.14".

Taxa de atualização ilimitada no SimSpace Weaver Local

O SimSpace Weaver Local implementa "unlimited" como se o esquema especificasse uma taxa de atualização de 10 kHz (10.000). O efeito é o mesmo de uma taxa de atualização ilimitada no Nuvem AWS. Ainda é preciso especificar `tick_rate`: "unlimited" no esquema. Para ter mais informações sobre o SimSpace Weaver Local, consulte [Desenvolvimento local em SimSpace Weaver](#).

Perguntas frequentes sobre o relógio

P1. Posso alterar uma simulação STARTED para usar uma taxa de atualização diferente?

Não é possível alterar a taxa de atualização de uma simulação que já existe no Nuvem AWS em nenhum estágio de seu ciclo de vida. Também não é possível alterar a taxa de atualização de uma

simulação em execução no SimSpace Weaver Local. Você pode definir a `tick_rate` no esquema e iniciar uma nova simulação a partir desse esquema.

P2. Posso executar minha simulação com uma taxa de atualização ilimitada em uma versão anterior a 1.14?

Não, a taxa de atualização ilimitada não é suportada nas versões anteriores a 1.14.0.

Solução de problemas do relógio

Se sua simulação falhar no início, você pode verificar o valor de `StartError` na saída da API `DescribeSimulation`. Um valor inválido de `tick_rate` no esquema produzirá os seguintes erros.

Note

A saída de erro mostrada aqui é exibida em várias linhas para melhorar a legibilidade. A saída real do erro consiste em uma única linha.

- A `sdk_version` é anterior a "1.14" e o valor da `tick_rate` é um número inteiro inválido. Valores válidos: 10, 15, 30

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

- A `sdk_version` é anterior a "1.14" e o valor da `tick_rate` é uma string. 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"}"]"
```

- A `sdk_version` é "1.14" ou posterior e o valor da `tick_rate` é uma string inválida. Valores válidos: "10", "15", "30", "unlimited"

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

- A `sdk_version` é "1.14" ou posterior e o valor da `tick_rate` é um número inteiro. 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"}]"
```

Estratégias de particionamento

A seção `partitioning_strategies` especifica as propriedades de configuração para as partições de aplicativos espaciais. Você fornece seu próprio nome para uma estratégia de particionamento (um conjunto de propriedades nesta seção) e o usa na configuração do aplicativo espacial.

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

A propriedade `topology` especifica o tipo de sistema de coordenadas que sua simulação usará. O valor `Grid` especifica uma grade bidimensional (2D).

Para uma topologia de `Grid`, o espaço de simulação é modelado como uma caixa delimitadora (AABB) alinhada ao eixo. Você especifica os limites das coordenadas para cada eixo da AABB na propriedade `aabb_bounds`. Todas as entidades que existem espacialmente em sua simulação devem ter uma posição dentro da AABB.

Grupos de posicionamento de grade

Um grupo de posicionamento é uma coleção de partições espaciais de aplicativos que você deseja que o SimSpace Weaver coloque no mesmo operador. Você especifica o número e a organização

dos grupos de posicionamento (em uma grade) na propriedade `grid_placement_groups`. O SimSpace Weaver tentará distribuir uniformemente as partições entre os grupos de posicionamento. As áreas de propriedade de aplicativos espaciais com partições no mesmo grupo de posicionamento serão espacialmente adjacentes.

Recomendamos que $x * y$ seja igual ao número desejado de operadores. Se não for igual, o SimSpace Weaver tentará equilibrar seus grupos de posicionamento entre os operadores disponíveis.

Se uma configuração de grupo de posicionamento não for especificada, o SimSpace Weaver calculará uma para você.

Domínios

Dê um nome para um conjunto de propriedades de configuração de um domínio. A configuração de lançamento de aplicativos em um domínio determina o tipo de domínio:

- **launch_apps_via_start_app_call**: domínio personalizado
- **launch_apps_by_partitioning_strategy**: domínio espacial
- **launch_apps_per_worker**: domínio de serviço (não incluído no aplicativo de exemplo)

Important

SimSpace Weaver suporta até 5 domínios para cada simulação. Isso inclui todos os domínios espaciais, personalizados e de serviço.

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



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

Note

SimSpace Weaver Os projetos da SDK versão 1.12.x do aplicativo usam buckets separados para os arquivos.zip do aplicativo e o esquema:

- `tecelão-lowercase-project-name-account-number-app-zips-region`
- `tecelão-lowercase-project-name-account-number-esquemas-region`

Tópicos

- [Configuração do aplicativo](#)
- [Configurar domínios espaciais](#)
- [Endpoints de rede](#)
- [Configurar domínios de serviço](#)

Configuração do aplicativo

Especifique a configuração de um aplicativo (`app_config`) como parte da configuração de seu domínio. Todos os tipos de domínios usam essas mesmas propriedades de configuração do aplicativo.

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

```
compute: 1
```

Note

SimSpace Weaver Os projetos da SDK versão 1.12.x do aplicativo usam buckets separados para os arquivos.zip do aplicativo e o esquema:

- `tecelão-lowercase-project-name-account-number-app-zips-region`
- `tecelão-lowercase-project-name-account-number-esquemas-region`

A `package` propriedade especifica o S3 URI de um arquivo zip em um bucket do S3. O arquivo .zip contém o executável do aplicativo (também chamado de binário) e quaisquer outros recursos necessários (como bibliotecas). Cada instância do executável do aplicativo é executada em um Docker contêiner em um trabalhador.

A propriedade `launch_command` especifica o nome do executável e todas as opções de linha de comando para executar o aplicativo. O valor de `launch_command` é uma matriz. Cada token de toda a cadeia de comando de inicialização é um elemento na matriz.

Exemplo

- Para o comando de lançamento: `MyTestApp --option1 value1`
- Especifique: `launch_command: ["MyTestApp", "-option1", "value1"]`

A `required_resource_units` propriedade especifica o número de unidades de recursos computacionais que SimSpace Weaver devem ser alocadas para esse aplicativo. Uma unidade de recursos computacionais é uma quantidade fixa de capacidade de processamento (vCPU) e memória (RAM) em um trabalhador. É possível aumentar esse valor para incrementar a quantidade de poder de computação disponível para o aplicativo quando ele é executado em um operador. Há um número limitado de unidades de recursos computacionais em cada operador. Para obter mais informações, consulte [SimSpace Terminais e cotas do Weaver](#).

Configurar domínios espaciais

Para domínios espaciais, você deve especificar uma `partitioning_strategy`. O valor dessa propriedade é o nome que você deu a uma estratégia de particionamento definida em outra parte do esquema.

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

Note

SimSpace Weaver Os projetos da SDK versão 1.12.x do aplicativo usam buckets separados para os arquivos.zip do aplicativo e o esquema:

- `tecelão-lowercase-project-name-account-number-app-zips-region`
- `tecelão-lowercase-project-name-account-number-esquemas-region`

Uma estratégia de particionamento com uma Grid topologia (a única topologia suportada nesta versão) SimSpace Weaver direciona a organização das partições espaciais de aplicativos desse domínio em uma grade. A propriedade `grid_partition` especifica o número de linhas e colunas da grade de partição.

SimSpace Weaver iniciará 1 instância do aplicativo espacial para cada célula na grade de partição. Por exemplo, se um domínio espacial tiver `grid_partition` valores `x: 2` e `y: 2` houver $2 * 2 = 4$ partições no domínio espacial. SimSpace Weaver iniciará 4 instâncias do aplicativo configurado no domínio espacial e atribuirá 1 partição a cada instância do aplicativo.

Tópicos

- [Requisitos de recursos para domínios espaciais](#)
- [Vários domínios espaciais](#)
- [Perguntas frequentes sobre domínios espaciais](#)
- [Solucionar problemas de domínios espaciais](#)

Requisitos de recursos para domínios espaciais

É possível atribuir até 17 unidades de recursos de computação a cada operador. Especifique o número de unidades de recursos computacionais que cada aplicativo espacial usa na seção `app_config` do seu domínio espacial.

Example Trecho de esquema mostrando as unidades de recursos computacionais para um aplicativo 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 o número de unidades de recursos computacionais que um domínio exige, multiplique o número de células em sua grade (em sua `grid_partition`, $x*y$) pelo número de unidades de recursos computacionais atribuídas aos aplicativos espaciais.

No exemplo anterior, o domínio `MySpatialDomain` especifica:

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

A grade para `MySpatialDomain` tem $2 \times 2 = 4$ células. O domínio espacial requer $4 \times 1 = 4$ unidades de recursos computacionais.

O número total de unidades de recursos computacionais para todos os domínios especificados em seu esquema deve ser menor ou igual ao número `desired` de operadores multiplicado pelo número máximo de unidades de recursos computacionais para cada operador (17).

Vários domínios espaciais

É possível configurar sua simulação para usar mais de um domínio espacial. Por exemplo, você pode usar um domínio espacial para controlar os principais atores em uma simulação (como pessoas e carros) e um domínio espacial diferente para controlar o ambiente.

Você também pode usar vários domínios espaciais para atribuir recursos distintos a diferentes partes da simulação. Por exemplo, se a simulação tem um tipo de entidade que tem dez vezes mais instâncias de entidade do que outro tipo, é possível criar domínios diferentes para lidar com cada tipo de entidade e alocar mais recursos para o domínio com mais entidades.

Important

SimSpace Weaver versões anteriores à 1.14.0 não oferecem suporte a vários domínios espaciais.

Important

AWS SimSpace Weaver Local atualmente não oferece suporte a vários domínios espaciais. Para obter mais informações sobre SimSpace Weaver Local, consulte [Desenvolvimento local em SimSpace Weaver](#).

Important

SimSpace Weaver suporta até 5 domínios para cada simulação. Isso inclui todos os domínios espaciais, personalizados e de serviço.

Configurar vários domínios espaciais

Para configurar mais de um domínio espacial, adicione as outras definições de domínio espacial como seções nomeadas distintas em seu esquema. Cada domínio deve especificar a chave `launch_apps_by_partitioning_strategy`. Veja o esquema de exemplo a seguir.

```
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
```

Posicionar domínios espaciais juntos

Em alguns cenários, você pode desejar posicionar partições para um domínio espacial em operadores ao lado de partições de outro domínio. Isso pode melhorar as características de desempenho se essas partições criarem assinaturas entre domínios.

Adicione a chave de nível superior `placement_constraints` ao seu esquema para especificar quais domínios SimSpace Weaver devem ser colocados juntos. A chave `on_workers` necessária deve se referir a uma configuração de `workers` nomeada no esquema.

Example Trecho de esquema mostrando domínios espaciais posicionados juntos

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

Important

- Se você usa grupos de posicionamento:
 - Certifique-se de que $X \times Y$ seja um múltiplo do número de operadores.
 - Certifique-se de que os valores do grupo de posicionamento sejam divisores comuns para as dimensões da grade dos domínios que você posicionar juntos.
- Se você não usa grupos de posicionamento:
 - Certifique-se de que um eixo de suas grades de domínio espacial tenha um divisor comum igual ao número de operadores.

Para mais informações sobre grupos de posicionamento, consulte [Estratégias de particionamento](#).

Perguntas frequentes sobre domínios espaciais

P1. Como adicionar outro domínio espacial a uma simulação existente?

- Para uma simulação em execução: não é possível alterar a configuração de uma simulação em execução. Altere a configuração do domínio no esquema, faça o carregamento do esquema e dos `.zips` do aplicativo e inicie uma nova simulação.
- Para uma nova simulação: adicione a configuração do domínio ao esquema, carregue o esquema e os `.zips` do aplicativo e inicie a nova simulação.

Solucionar problemas de domínios espaciais

O erro a seguir ocorre ao iniciar a simulação com uma configuração de domínio inválida.

```
"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
  [grid_partition] values, or adjusting the
  dimensions of your [grid_placement_groups]."}]"
```

Possíveis causas

- O esquema aloca mais unidades de recursos computacionais para aplicativos do que as disponíveis para os operadores.
- SimSpace Weaver não consegue determinar um acordo para colocar domínios juntos em trabalhadores. Isso acontece quando vários domínios espaciais são especificados, mas não há um divisor comum ou um múltiplo entre as grades de domínio, como entre uma grade 2 x 4 e uma grade 3 x 5).

Endpoints de rede

Aplicativos personalizados e de serviço podem ter endpoints de rede aos quais clientes externos podem se conectar. Especifique uma lista de números de porta como valor para `ingress_ports` no `endpoint_config`. Esses números de porta são ambos TCP UDP e. O aplicativo personalizado ou de serviço deve ser vinculado aos números de porta que você especifica em `ingress_ports`. SimSpace Weaver aloca dinamicamente os números das portas em tempo de execução e mapeia essas portas para as portas dinâmicas. Você pode ligar para o aplicativo de descrição API depois que seus aplicativos começarem a encontrar os números de porta dinâmicos (reais). Para obter mais informações, consulte [Obtenha o endereço IP e o número da porta de um aplicativo personalizado](#) no tutorial de início rápido.

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



```

compute: 1
endpoint_config:
ingress_ports:
  - 7000

```

Note

SimSpace Weaver Os projetos da SDK versão 1.12.x do aplicativo usam buckets separados para os arquivos.zip do aplicativo e o esquema:

- `tecelão-lowercase-project-name-account-number-app-zips-region`
- `tecelão-lowercase-project-name-account-number-esquemas-region`

Note

`endpoint_config` é uma propriedade opcional para aplicativos personalizados e de serviço. Se você não especificar um `endpoint_config`, o aplicativo não terá um endpoint de rede.

Configurar domínios de serviço

A presença de `launch_apps_per_worker`: em uma configuração de domínio indica que é um domínio de serviço que tem aplicativos de serviço. SimSpace Weaver inicia e interrompe aplicativos de serviço para você. Quando SimSpace Weaver inicia e interrompe um aplicativo, considera-se que o aplicativo tem um ciclo de vida gerenciado. SimSpace Weaver atualmente suporta a inicialização de 1 ou 2 aplicativos de serviço em cada trabalhador.

Example Exemplo de um domínio configurado para iniciar um aplicativo de serviço em cada operador

```

domains:
  MyServiceDomain:
    launch_apps_per_worker:
      count: 1
    app_config:
      package: "s3://weaver-myproject-111122223333-app-zips-us-west-2/
PlayerConnectionServiceApp.zip"

```

```

launch_command: ["PlayerConnectionServiceApp"]
required_resource_units:
  compute: 1
endpoint_config:
  ingress_ports:
    - 9000
    - 9001

```

Example Exemplo de um domínio configurado para iniciar dois aplicativos de serviço em cada operador

```

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

```

Duração máxima de uma simulação

Cada simulação AWS SimSpace Weaver tem uma configuração de duração máxima que especifica o tempo máximo em que a simulação pode ser executada. Você fornece a duração máxima como parâmetro ao iniciar uma simulação. A [interface de programação de aplicações \(API\) StartSimulation](#) tem um parâmetro de `MaximumDuration` opcional. O valor do parâmetro é um número de minutos (m ou M), horas (h ou H) ou dias (d ou D). Por exemplo, 1h ou 1H significa uma hora. O SimSpace Weaver interrompe a simulação ao atingir esse limite.

Valor máximo

O maior valor válido para `MaximumDuration` é 14D, ou seu equivalente em horas (336H) ou minutos (20160M).

Valor padrão

O parâmetro `MaximumDuration` é opcional. Se você não fornecer um valor, SimSpace Weaver use um valor de 14D.

Valor mínimo

O menor valor válido para `MaximumDuration` é numericamente equivalente a 0. Por exemplo, os valores 0M, 0H e 0D são numericamente equivalentes a 0.

Se você fornecer o valor mínimo para a duração máxima, sua simulação transitará imediatamente para o STOPPING estado assim que atingir o STARTED estado.

Iniciar uma simulação usando o console

É possível fornecer um valor para a Duração máxima ao iniciar uma simulação no [console do SimSpace Weaver](#). Insira o valor no campo Duração máxima do formulário de Configurações da simulação ao escolher Iniciar simulação.

Important

Se você não fornecer um valor para a Duração máxima, o SimSpace Weaver usará o [valor padrão](#) (14D).

Status de uma simulação que atinge a duração máxima

Quando interrompe SimSpace Weaver automaticamente uma simulação que atinge sua duração máxima, o status da simulação é STOPPING (se estiver em andamento) ou STOPPED. No [console do SimSpace Weaver](#), o status de destino da simulação ainda é STARTED, porque esse foi o último estado solicitado por um usuário.

Desenvolver aplicativos

SimSpace Weaver o desenvolvimento requer um Amazon Linux 2 (AL2) ambiente para criar aplicativos porque suas simulações são executadas Amazon Linux no AWS Cloud. Se você estiver usando Windows, poderá usar scripts no SDK do SimSpace Weaver aplicativo para criar e iniciar um

Docker contêiner executado AL2 com as dependências necessárias para criar SimSpace Weaver aplicativos. Você também pode iniciar um ambiente em AL2 usando o Windows Subsystem for Linux (WSL) ou usar um sistema nativo AL2. Para ter mais informações, consulte [Configurar o ambiente local para o SimSpace Weaver](#).

Note

Independentemente de como você configurar seu ambiente de desenvolvimento local, seus aplicativos são executados em contêineres de Docker quando você os carrega para execução na Nuvem AWS. Seus aplicativos não têm acesso direto ao sistema operacional do host.

Fluxo geral de um SimSpace Weaver aplicativo

1. Crie uma aplicação.
2. Loop:
 - a. Começar a atualização criando uma `Transaction`.
 - Sair do loop se a simulação estiver sendo encerrada.
 - b. Processar eventos da entidade de assinatura e propriedade.
 - c. Atualizar a simulação.
 - d. Confirmar a `Transaction` para finalizar a atualização.
3. Destruir o aplicativo.

Aplicativos espaciais

Cada aplicativo espacial tem uma área de propriedade que é uma região espacial do mundo da simulação. As entidades localizadas na área de propriedade de um aplicativo espacial são armazenadas na partição atribuída ao aplicativo. O aplicativo espacial único tem propriedade total (permissões de leitura e gravação) sobre todas as entidades dentro da partição atribuída. Nenhum outro aplicativo pode gravar nessas entidades. O aplicativo espacial avança o estado de suas entidades. Cada aplicativo espacial possui apenas uma partição. O SimSpace Weaver usa a localização espacial de uma entidade para indexá-la e atribuí-la a uma partição espacial do aplicativo.

O SDK do SimSpace Weaver aplicativo fornece um aplicativo de amostra. Você pode encontrar o código-fonte do aplicativo espacial do aplicativo de amostra na seguinte pasta (use o separador de caminho correto para seu sistema operacional):

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

Aplicativos personalizados

Você cria e usa aplicativos personalizados para interagir com a simulação.

Aplicativos personalizados podem

- Criar entidades
- Assinar outras partições
- Confirme as alterações

Fluxo geral de um aplicativo personalizado

1. Crie uma aplicação.
2. Inscreva-se em uma região específica na simulação:
 - a. Crie uma `Transaction` para começar a primeira atualização.
 - b. Crie uma assinatura para a região específica.
 - c. Confirme a `Transaction` para finalizar a primeira atualização.
3. Loop:
 - a. Criar uma `Transaction` para começar a atualização.
 - Sair do loop se a simulação estiver sendo encerrada.
 - b. Processar as alterações no processo.
 - c. Confirmar a `Transaction` para finalizar a atualização.
4. Destruir o aplicativo.

Depois que um aplicativo personalizado cria uma entidade, ele deve transferir a entidade para um domínio espacial para que a entidade exista espacialmente na simulação. SimSpace Weaver usa a localização espacial da entidade para colocar a entidade na partição espacial apropriada do

aplicativo. O aplicativo personalizado que criou a entidade não pode atualizar nem excluir a entidade depois de transferi-la para um domínio espacial.

O SDK do SimSpace Weaver aplicativo fornece um aplicativo de amostra. É possível usar os aplicativos personalizados incluídos no aplicativo de exemplo como modelos para seus próprios aplicativos personalizados. Você pode encontrar o código-fonte do aplicativo de visualização (um aplicativo personalizado) do aplicativo de amostra na seguinte pasta (use o separador de caminho correto para seu sistema operacional):

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

Desenvolver aplicativos cliente

Alguns motivos para conectar um cliente a uma simulação incluem:

- Injetar informações de trânsito em tempo real em uma simulação em escala urbana.
- Criar simulações human-in-the-loop nas quais um operador humano controla alguns aspectos da simulação.
- Permitir que os usuários interajam com a simulação, como em uma simulação de treinamento.

Os aplicativos personalizados nesses exemplos funcionam como a interface entre o estado da simulação e o mundo externo. Os clientes se conectam aos aplicativos personalizados para interagir com a simulação.

SimSpace Weaver não lida com os aplicativos clientes e sua comunicação com seus aplicativos personalizados. Você é responsável pelo design, criação, operação e segurança dos aplicativos cliente e pela comunicação deles com os aplicativos personalizados. O SimSpace Weaver expõe apenas um endereço IP e um número de porta para cada um dos aplicativos personalizados para que os clientes possam se conectar a eles.

O SDK do SimSpace Weaver aplicativo fornece clientes para seu aplicativo de amostra. É possível usar esses clientes como modelos para seus próprios aplicativos cliente. É possível encontrar o código-fonte dos exemplos de aplicativos cliente na seguinte pasta:

Docker

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

WSL

⚠ Important

Fornecemos essas instruções para a sua conveniência. Elas devem ser usadas com o Windows Subsystem for Linux (WSL) e não são compatíveis. Para ter mais informações, consulte [Configurar o ambiente local para o SimSpace Weaver](#).

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

Para obter mais informações sobre como criar e usar os clientes de aplicativos de amostra, consulte os tutoriais em [Começando com SimSpace Weaver](#)

Obtenha o endereço IP e o número da porta de um aplicativo personalizado

Para visualizar sua simulação, você cria um aplicativo personalizado e se conecta a ele com um cliente. Para obter mais informações, consulte os tutoriais em [Começando com SimSpace Weaver](#). Você pode usar o procedimento a seguir para obter o endereço IP e o número da porta do seu aplicativo personalizado. Use o separador de caminho apropriado para seu sistema operacional (por exemplo, \ no Windows e / no Linux).

Como obter o endereço IP e o número da porta

1. Use a `ListSimulationsAPI` para obter o nome da sua simulação.

```
aws simspaceweaver list-simulations
```

Resultado do exemplo:

```
{
  "Simulations": [
    {
      "Status": "STARTED",
```

```

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

```

2. Use a `DescribeSimulationAPI` para obter uma lista de domínios em sua simulação.

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

Procure a seção `Domains` na seção `LiveSimulationState` da saída.

Resultado do exemplo:

```

"LiveSimulationState": {
  "Domains": [
    {
      "Type": "",
      "Name": "MySpatialSimulation",
      "Lifecycle": "Unknown"
    },
    {
      "Type": "",
      "Name": "MyViewDomain",
      "Lifecycle": "ByRequest"
    }
  ],
}

```

3. Use a `ListAppsAPI` para obter uma lista de aplicativos personalizados em um domínio. Por exemplo, o nome de domínio do aplicativo de visualização (personalizado) no projeto de amostra é `MyViewDomain`. Procure o nome do aplicativo na saída.

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

Resultado do exemplo:


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

- Use a `DescribeAppAPI` para obter o endereço IP e o número da porta. Para o projeto de exemplo, o nome do domínio é `MyViewDomain` e o nome do aplicativo é `ViewApp`.

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

O endereço IP e o número da porta estão no bloco do `EndpointInfo` na saída. O endereço IP é o valor de `Address` e o número da porta é o valor de `Actual`.

Resultado do exemplo:

```
{
  "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

O valor de `Declared` é o número da porta à qual o código do seu aplicativo deve ser vinculado. O valor de `Actual` é o número da porta que é SimSpace Weaver exposta aos clientes para se conectarem ao seu aplicativo. SimSpace Weaver mapeia a `Declared` porta até a `Actual` porta.

Lançamento do cliente de visualização Unreal Engine

Navegue até:

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

1. Execute um dos seguintes comandos:
 - Docker: `python quick-start.py`
 - WSL: `python quick-start.py --a12`
2. Obtenha o endereço IP e o número da porta “real”. Eles estarão na saída do console da execução do `quick-start.py` ou obtenha-os seguindo os procedimentos em [Obtenha o endereço IP e o número da porta de um aplicativo personalizado](#).

3. Navegue até:

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

4. Execute os seguintes comandos para criar a biblioteca NNG:

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

5. Em um editor de texto, abra `view_app_url.txt`.
6. Atualize a URL com o endereço IP e o número da porta do seu aplicativo de visualização: `tcp://ip-address:actual-port-number` (deve ser semelhante a `tcp://198.51.100.135:1234`).
7. No editor Unreal, escolha jogar.

Solução de problemas

- A etapa de instalação do NNG CMake falha com “Talvez precise de privilégios administrativos “:

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

- **Resolução:** Se `nng.lib` ou `nng.so` existir no diretório `UnrealClient/lib`, esse erro pode ser ignorado com segurança. Caso contrário, tente executar os comandos `cmake build` em um terminal com privilégios de administrador.
- “CMake para encontrar um arquivo de configuração de pacote fornecido pelo nng”:

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

- **Resolução:** o CMake está tendo problemas para encontrar o `Findnng.cmake` arquivo. Ao criar com o CMake, adicione o argumento `-DTHIRD_PARTY_LIB_PATH sdk-folder/ThirdParty`. Certifique-se de que o `Findnng.cmake` arquivo ainda esteja no `ThirdParty` diretório antes de executar novamente a compilação do CMake.

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

Desenvolvimento local em SimSpace Weaver

Você pode implantar seus SimSpace Weaver aplicativos localmente para testes e depuração rápidos.

Requisitos

- Siga as etapas em [Configuração do SimSpace Weaver](#).

Tópicos

- [Etapa 1: Inicie sua simulação local](#)
- [Etapa 2: Visualize sua simulação local](#)
- [Etapa 3: interrompa sua simulação local \(opcional no Windows\)](#)
- [Solução de problemas de desenvolvimento local em SimSpace Weaver](#)

Etapa 1: Inicie sua simulação local

1. Navegue até

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

2. Execute o comando a seguir para criar e iniciar sua simulação localmente.

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

Esse script fará o seguinte:

1. Crie o projeto.
 - `quick-start.py` chama a `build_project` função definida em `build.py`. Essa etapa variará de acordo com o projeto. Para o `PathfindingSample`, o `CMake` é usado. Os comandos `CMake` e `Docker` podem ser encontrados em `build.py`.
2. Inicie sua simulação local
 - O script iniciará um processo local para cada partição espacial definida no esquema.
 - O script iniciará um processo para cada aplicativo personalizado definido no esquema.
 - Os aplicativos espaciais serão lançados primeiro, seguidos pelos aplicativos personalizados — cada um na ordem em que aparecem no esquema.

⚠ Important

Ao iniciar em um ambiente que não suporta GUI, como uma sessão SSH do console, use a `--noappwindow` opção para redirecionar toda a saída para o terminal atual.

⚠ Important

Para usuários de Linux, o script pressupõe que seu sistema tenha o `xterm` comando. Se sua distribuição Linux não tiver o `xterm` comando, use a `--noappwindow` opção para redirecionar toda a saída para o terminal atual.

- `-h, --help`
 - Liste esses parâmetros.
- `--limpar`
 - Exclua o conteúdo do diretório de compilação antes de compilar.
- `--sem construção`
 - Pule a reconstrução do projeto.
- `--nenhuma janela de aplicativo`
 - Não abra uma nova janela para cada aplicativo. Em vez disso, redirecione o `stdout` para o terminal atual.
- `--arquivo de log`
 - Grave a saída do console em um arquivo de registros.
- `--cliente do console`
 - Conecte automaticamente o cliente do console listado na configuração.
- `--esquema esquema`
 - Qual esquema essa invocação usará. O padrão é 'SCHEMA' em `config.py`.

Etapa 2: Visualize sua simulação local

Para visualizar sua simulação local, você pode usar qualquer um dos clientes incluídos no `SimSpaceWeaverAppSdkDistributable`. Para obter mais informações sobre como criar e usar os clientes de amostra, consulte os tutoriais em [Começando com SimSpace Weaver](#)

Você deve atualizar o endereço IP e o número da porta no cliente para se conectar ao aplicativo de visualização para a simulação local. Use sempre os seguintes valores com o SimSpace Weaver Local:

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

Dependendo do cliente selecionado, você pode atualizar o endereço IP e o número da porta da seguinte forma:

- Unreal: altera a URL na linha 1 do `view_app_url.txt`
- Console: inicia o cliente com o endereço IP e a URL do número da porta como parâmetro

Etapa 3: interrompa sua simulação local (opcional no Windows)

Note

Essa etapa é obrigatória no Linux, mas opcional no Windows.

1. Navegue até:

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

2. Execute o comando a seguir para interromper sua simulação local e excluir todos os recursos de memória compartilhada.

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

Esse script fará o seguinte:

- Pare os processos locais.
- Exclua o objeto de memória compartilhada (necessário somente no Linux).

parâmetros do `stop-and-delete.py`

- `-h, --help`
- Liste esses parâmetros.

- `--parar`
 - Apenas tente interromper os processos.
- `--excluir`
 - Apenas tente excluir os recursos de memória compartilhada.
- `--processo`
 - O nome do processo a interromper. Use isso se o nome do seu processo não corresponder ao nome do pacote no esquema.
- `--esquema esquema`
 - Qual esquema essa invocação usará. O padrão é o valor de 'SCHEMA' em `config.py`.

Solução de problemas de desenvolvimento local em SimSpace Weaver

- Linux: comando `xterm` não encontrado/não pode ser aberto
 - Os scripts locais presumem que o comando `xterm` existe quando executado no Linux. Se você não tiver o comando `xterm` ou estiver executando em um ambiente que não oferece suporte à GUI, use a `--noappwindow` opção ao executar o script de início rápido.
- Nenhuma janela de aplicativo está abrindo!
 - Isso acontece quando a simulação local falha imediatamente. Para ver a saída do console após a falha, use as `--logfile` opções `--noappwindow` ou ao executar o script de início rápido.
- A simulação não funciona depois que o aplicativo de visualização é iniciado ou o cliente de visualização se conecta!
 - A execução com a `--noappwindow` opção normalmente resolve esses tipos de problemas. Caso contrário, reiniciar algumas vezes também terá sucesso (embora em uma taxa muito menor).

Aplicativo do SDK do AWS SimSpace Weaver

O aplicativo do SDK do SimSpace Weaver fornece APIs que você pode usar para controlar as entidades na simulação e responder aos eventos do SimSpace Weaver. Isso inclui o seguinte namespace:

- API: principais definições da API e seu uso

Link com a seguinte biblioteca:

- `libweaver_app_sdk_cxx_v1_full.so`

Important

A biblioteca está disponível para vinculação dinâmica quando você executa seus aplicativos no Nuvem AWS. Não é necessário carregá-la com seus aplicativos.

Note

As APIs do aplicativo do SDK do SimSpace Weaver controlam os dados em sua simulação. Essas APIs são diferentes das APIs de serviço do SimSpace Weaver, que controlam seus recursos de serviço no SimSpace Weaver (como simulações, aplicativos e relógios) na AWS. Para ter mais informações, consulte [SimSpace Weaver Referências de API](#).

Tópicos

- [Os métodos da API retornam um Result](#)
- [Interagir com o aplicativo do SDK no nível superior](#)
- [Gerenciamento de simulações](#)
- [Assinaturas](#)
- [Entidades](#)
- [Eventos da entidade](#)
- [Result e tratamento de erros](#)
- [Genéricos e tipos de domínio](#)
- [Operações diversas do aplicativo do SDK](#)

Os métodos da API retornam um Result

A maioria das funções da API do SimSpace Weaver tem um tipo de retorno

`Aws::WeaverRuntime::Result<T>`. Se a função for executada com sucesso, o `Result` conterá `T`. Caso contrário, o `Result` conterá um `Aws::WeaverRuntime::ErrorCode` que representa um código de erro do Rust App SDK.

Example Exemplo

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

Esse método:

- Retorna `Transaction` se `BeginUpdate()` for executado com sucesso.
- Retorna `Aws::WeaverRuntime::ErrorCode` caso o `BeginUpdate()` falhe.

Interagir com o aplicativo do SDK no nível superior

Ciclo de vida

- O aplicativo do SDK do SimSpace Weaver gerencia o ciclo de vida do aplicativo. Não é preciso ler ou gravar o estado do ciclo de vida de um aplicativo.

Partições

- Use `Result <PartitionSet> AssignedPartitions(Transaction& txn);` para obter partições próprias.
- Use `Result <PartitionSet> AllPartitions(Transaction& txn);` para obter todas as partições na simulação.

Gerenciamento de simulações

Esta seção descreve soluções para tarefas comuns de gerenciamento de simulações.

Tópicos

- [Iniciar uma simulação](#)
- [Atualizar uma simulação](#)
- [Encerrar uma simulação](#)

Iniciar uma simulação

Como criar um aplicativo

Example Exemplo

```

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()));

```

Atualizar uma simulação

Use as seguintes funções de `BeginUpdate` para atualizar o aplicativo:

- `Result<Transaction> BeginUpdate(Application& app)`
- `Result<bool> BeginUpdateWillBlock(Application& app)`: informa se o `BeginUpdate()` bloqueará ou não.

Execute `Result<void> Commit(Transaction& txn)` para confirmar as alterações.

Example Exemplo

```

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();
}

```

Encerrar uma simulação

Use `Result<void> DestroyApplication(Application&& app)` para encerrar o aplicativo e a simulação.

Outros aplicativos descobrem que a simulação é encerrada quando recebem `ErrorCode::ShuttingDown` de chamadas para `BeginUpdateWillBlock()` ou `BeginUpdate()`. Quando um aplicativo recebe `ErrorCode::ShuttingDown`, ele pode chamar `Result<void> DestroyApplication(Application&& app)` para se encerrar.

Example Exemplo

```

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

Apenas chame `Result<void> DestroyApplication(Application&& app)` após `Api::Commit()`. Destruir um aplicativo durante uma atualização pode causar um comportamento indefinido.

⚠ Important

Você deve chamar `DestroyApplication()` antes do encerramento do programa para garantir que o aplicativo seja relatado como encerrado com êxito. Deixar de chamar `DestroyApplication()` quando o programa for encerrado fará com que o status seja considerado como FATAL.

Assinaturas

Crie uma assinatura com uma área de assinatura e um ID de domínio. O ID do domínio representa o domínio que possui essa área de assinatura. Uma `BoundingBox2F32` descreve a área de assinatura. Use a seguinte função para criar uma assinatura:

```

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

```

Example Exemplo

```

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();
}

```

Você pode usar o `Api::SubscriptionHandle` retornado por `CreateSubscriptionBoundingBox2F32()` para modificar a assinatura. Transfira-lo como argumento para as seguintes funções:

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

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

Entidades

Chame as APIs `Load` e `Store` que usam a `Api::Entity` do `Result<Api::Entity>` retornadas de `CreateEntity()` ou de um evento de alteração de propriedade quando uma entidade entra na área de assinatura do aplicativo. Para obter mais informações, consulte [Eventos da entidade](#). Recomendamos que você rastreie seus objetos `Api::Entity` para usá-los com essas APIs.

Tópicos

- [Criar entidades](#)
- [Transferir uma entidade para um domínio espacial](#)
- [Gravar e ler dados do campo da entidade](#)
- [Armazenar a posição de uma entidade](#)
- [Carregar a posição de uma entidade](#)

Criar entidades

Use `CreateEntity()` para criar uma entidade. Defina o significado de `Api::TypeId` que você transferirá para essa função.

```
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

Os valores de 0-511 para `Api::BuiltinTypeId` são reservados. Sua entidade `TypeId` (`k_entityTypeId` neste exemplo) deve ter um valor de 512 ou superior.

Transferir uma entidade para um domínio espacial

Depois que um aplicativo personalizado ou de serviço cria uma entidade, deve transferir a entidade para um domínio espacial para que a entidade exista espacialmente na simulação. As entidades em um domínio espacial podem ser lidas por outros aplicativos e atualizadas por um aplicativo espacial. Use a API `ModifyEntityDomain()` para transferir uma entidade para um domínio espacial.

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

Se `DomainId` não corresponder à `Partition` atribuída pelo aplicativo de chamada, então `DomainId` deve ser para um `Domain` de `DomainType::Spatial`. A transferência de propriedade para o novo `Domain` ocorre durante `Commit(Transaction&&)`.

Parâmetros

`txn`

A `Transaction` atual.

`entity`

A `Entity` de destino para a mudança de `Domain`.

`domainId`

O `DomainId` do `Domain` de destino para a `Entity`.

Essa API retornará `Success` se o domínio da entidade tiver sido alterado com sucesso.

Gravar e ler dados do campo da entidade

Todos os campos de dados da entidade são do tipo blob. É possível gravar até 1.024 bytes de dados em uma entidade. Recomendamos manter os blobs tão pequenos quanto possível, pois tamanhos maiores reduzirão o desempenho. Ao gravar em um blob, você transmite ao SimSpace Weaver um ponteiro para os dados e seu comprimento. Quando você lê um blob, o SimSpace Weaver fornece um ponteiro e um comprimento para a leitura. Todas as leituras devem ser concluídas antes da chamada `Commit()` no aplicativo. Os ponteiros retornados de uma chamada de leitura são invalidados quando o aplicativo chama `Commit()`.

Important

- A leitura de um ponteiro de blob em cache após `Commit()` não é suportada e pode fazer com que a simulação falhe.
- A gravação em um ponteiro blob retornado de uma chamada de leitura não é suportada e pode fazer com que a simulação falhe.

Tópicos

- [Armazenar dados de campo de uma entidade](#)
- [Carregar os dados de campo de uma entidade](#)
- [Carregar dados de campo de entidades removidas](#)

Armazenar dados de campo de uma entidade

Os exemplos a seguir demonstram como armazenar (gravar no State Fabric) os dados de campo de uma entidade de propriedade do aplicativo. Esses exemplos usam a função a seguir:

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

O parâmetro `Api::TypeId keyTypeId` representa o tipo de dados transmitidos.

O parâmetro `Api::TypeId keyTypeId` deve receber o correspondente `Api::TypeId` de `Api::BuiltinTypeId`. Se não houver uma conversão apropriada, você poderá usar `Api::BuiltinTypeId::Dynamic`.

Para tipos de dados complexos, use `Api::BuiltinTypeId::Dynamic`.

Note

O valor de `FieldIndex index` deve ser maior que 0. O valor 0 é reservado para a chave de índice (consulte `StoreEntityIndexKey()`).

Example Exemplo de uso de tipos de dados 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 Exemplo de uso de um struct para armazenar os dados

```

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));
}

```

Carregar os dados de campo de uma entidade

Os exemplos a seguir demonstram como carregar (ler no State Fabric) os dados de campo de uma entidade. Esses exemplos usam a seguinte função:

```

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

```

O parâmetro `Api::TypeId keyTypeId` deve receber o correspondente `Api::TypeId` de `Api::BuiltinTypeId`. Se não houver uma conversão apropriada, use `Api::BuiltinTypeId::Dynamic`.

Note

O valor do índice `FieldIndex` deve ser maior que 0. O valor 0 é reservado para a chave de índice (consulte `StoreEntityIndexKey()`).

Exemplo de uso de tipos de dados 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 Exemplo de uso de um struct para armazenar os dados

```

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);
}

```

Carregar dados de campo de entidades removidas

Você não pode carregar (ler no State Fabric) dados de campos de entidades que foram removidas das áreas de propriedade e assinatura do aplicativo. O exemplo a seguir resulta em um erro porque chama `Api::LoadIndexKey()` em uma entidade como resultado de uma `Api::ChangeListAction::Remove`. O segundo exemplo mostra uma forma correta de armazenar e carregar dados da entidade diretamente no aplicativo.

Exemplo Exemplo de código incorreto

```
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 Exemplo de uma forma correta de armazenar e carregar dados da entidade no aplicativo

```

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;
    }
}
}

/* ... */
}

```

Armazenar a posição de uma entidade

Você pode armazenar (gravar no State Fabric) a posição de uma entidade usando uma estrutura de dados inteira. Esses exemplos usam a seguinte função:

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

Note

Forneça `Api::BuiltinTypeId::Vector3F32` para `Api::StoreEntityIndexKey()`, conforme mostrado nos exemplos a seguir.

Example Exemplo de uso de uma matriz para representar a posição

```
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 Exemplo com struct para representar a posição

```
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));
}

```

Carregar a posição de uma entidade

Você pode carregar (ler no State Fabric) a posição de uma entidade usando uma estrutura de dados inteira. Esses exemplos usam a seguinte função:

Note

Forneça `Api::BuiltinTypeId::Vector3F32` para `Api::LoadEntityIndexKey()`, conforme mostrado nos exemplos a seguir.

Example Exemplo de uso de uma matriz para representar a posição

```

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 Exemplo de uso da struct para representar a posição

```

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 da entidade

Você pode usar as seguintes funções no aplicativo do SDK do SimSpace Weaver para obter todos os eventos de propriedade e de assinatura:

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

Você pode usar a estrutura de demonstração do SimSpace Weaver para o processamento de eventos de entidade orientado por retorno de chamada. Para mais informações, consulte o seguinte arquivo de cabeçalho:

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

Também é possível criar seu próprio processamento de eventos de entidade.

Tópicos

- [Iterar eventos para entidades próprias](#)
- [Iterar eventos para entidades assinadas](#)
- [Iterar eventos de mudança de propriedade para entidades](#)

Iterar eventos para entidades próprias

Use `OwnershipChanges()` para obter uma lista de eventos para entidades próprias (entidades na área de propriedade do aplicativo). A função tem a assinatura a seguir:

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

Faça a iteração das entidades com um loop, conforme demonstrado no exemplo a seguir.

Example Exemplo

```
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: a entidade está na área e os dados de posição e campo não foram modificados.
- Remove: a entidade foi removida da área.
- Add: a entidade foi adicionada à área.
- Update: a entidade está na área e foi modificada.
- Reject: o aplicativo falhou ao remover a entidade da área.

Note

No caso de um evento `Reject`, o aplicativo tentará a transferência novamente na próxima marcação.

Iterar eventos para entidades assinadas

Use `AllSubscriptionEvents()` para obter uma lista de eventos para entidades assinadas (entidades na área de assinatura do aplicativo). A função possui a assinatura a seguir:

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

Faça a iteração das entidades com um loop, conforme demonstrado no exemplo a seguir.

Example Exemplo

```
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: a entidade está na área e os dados de posição e campo não foram modificados.
- Remove: a entidade foi removida da área.
- Add: a entidade foi adicionada à área.
- Update: a entidade está na área e foi modificada.
- Reject: o aplicativo falhou ao remover a entidade da área.

Note

No caso de um evento `Reject`, o aplicativo tentará a transferência novamente na próxima marcação.

Iterar eventos de mudança de propriedade para entidades

Para obter eventos em que uma entidade se move entre uma área de propriedade e uma área de assinatura, compare as mudanças entre os eventos atuais e anteriores de propriedade e assinatura da entidade.

Lide com esses eventos lendo:

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

Compare as alterações com os dados armazenados anteriormente.

O exemplo a seguir mostra como lidar com eventos de mudança de propriedade da entidade. Este exemplo pressupõe que, para entidades em transição entre entidades assinadas e de propriedade (em qualquer direção), o evento de remoção/adição de propriedade ocorra primeiro, seguido pelo evento de remoção/adição de assinatura na próxima marca.

Example Exemplo

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

    static EntityIdsByAction m_entityIdsByPreviousOwnershipAction;

    EntityIdSetByAction entityIdSetByAction;

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

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

```

const Api::ChangeListAction action = event.action;

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

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

EntityIdsByAction entityIdByAction;

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

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

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

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

```

```

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

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

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

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

```

```

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

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

m_entityIdsByPreviousOwnershipAction = entityIdsByOwnershipAction;

return Success();
}

```

Result e tratamento de erros

A classe `Aws::WeaverRuntime::Result<T>` usa uma biblioteca `Outcome` de terceiros. Use o padrão a seguir para verificar `Result` e capturar os erros retornados pelas chamadas de 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 declaração de controle de Result

Dentro de uma função com um tipo de retorno `Aws::WeaverRuntime::Result<T>`, é possível usar a macro `WEAVERRUNTIME_TRY` em vez do padrão de código anterior. A macro executará a função passada para ela. Se a função passada falhar, a macro fará com que a função envolvente retorne um erro. Se a função passada for bem-sucedida, a execução avançará para a próxima linha. O exemplo a seguir mostra a nova gravação da função `DoBeginUpdate()` anterior. Essa versão usa a macro `WEAVERRUNTIME_TRY` em vez da estrutura de controle `if-else`. O tipo de retorno da função é `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();
}

```

Se `BeginUpdate()` falhar, a macro `DoBeginUpdate()` retorna mais cedo com uma falha. Você pode usar a macro `WEAVERRUNTIME_EXPECT_ERROR` para obter o `Aws::WeaverRuntime::ErrorCode` de `BeginUpdate()`. O exemplo a seguir mostra como a função `Update()` chama `DoBeginUpdate()` e obtém o código de erro em caso de falha.

```
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);
    }
}
```

Você pode tornar o código de erro `BeginUpdate()` disponível para uma função que chamará `Update()` alterando o tipo de retorno de `Update()` para `Aws::WeaverRuntime::Result<void>`. Repita o processo para continuar enviando o código de erro mais abaixo na pilha de chamadas.

Genéricos e tipos de domínio

O aplicativo do SDK do SimSpace Weaver fornece os tipos de dados de precisão única `Api::Vector2F32` e `Api::BoundingBox2F32`, e a precisão dupla `Api::Vector2F64` e `Api::BoundingBox2F64`. Esses tipos de dados são estruturas de dados passivas sem métodos convenientes. Observe que a API usa apenas `Api::Vector2F32` e `Api::BoundingBox2F32`. Use esses tipos de dados para criar e modificar assinaturas.

A estrutura de demonstração do SimSpace Weaver fornece uma versão mínima da biblioteca matemática `AzCore`, que contém `Vector3` e `Aabb`. Para obter mais informações, consulte os arquivos de cabeçalho em:

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

Operações diversas do aplicativo do SDK

Tópicos

- [AllSubscriptionEvents e OwnershipChanges contêm eventos da última chamada](#)
- [Liberar bloqueios de leitura após o processamento de SubscriptionChangeList](#)
- [Criar uma instância de aplicativo independente para testes](#)

AllSubscriptionEvents e OwnershipChanges contêm eventos da última chamada

Os valores de retorno das chamadas para `Api::AllSubscriptionEvents()` e `Api::OwnershipChanges()` contêm eventos da última chamada, não da última marcação. No exemplo a seguir, `secondSubscriptionEvents` e `secondOwnershipChangeList` estão vazios porque suas funções são chamadas imediatamente após as primeiras chamadas.

Se você esperar dez marcações e depois chamar `Api::AllSubscriptionEvents()` e `Api::OwnershipChanges()`, os resultados conterão eventos e alterações das últimas dez marcações (e não da última).

Example Exemplo

```
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

A função `AllSubscriptionEvents()` é implementada, mas a função `SubscriptionEvents()` não está implementada.

Liberar bloqueios de leitura após o processamento de SubscriptionChangeList

Ao iniciar uma atualização, há segmentos de memória compartilhada para os dados confirmados em outras partições da marca anterior. Esses segmentos de memória compartilhada podem ser bloqueados pelos leitores. Um aplicativo não pode ser confirmado totalmente até que todos os leitores tenham liberado os bloqueios. Como otimização, um aplicativo deve chamar `Api::ReleaseReadLeases()` para liberar os bloqueios após o processamento dos itens de `Api::SubscriptionChangeList`. Isso reduz a contenção no momento da confirmação. A `Api::Commit()` libera as concessões de leitura por padrão, mas é uma prática recomendada liberá-las manualmente após o processamento das atualizações da assinatura.

Example Exemplo

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

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

    WEAVERRUNTIME_EXPECT(Api::ReleaseReadLeases(transaction));

    ...
}
```

Criar uma instância de aplicativo independente para testes

Use `Api::CreateStandaloneApplication()` para criar um aplicativo independente e testar a lógica dele antes de executar o código em uma simulação real.

Example Exemplo

```
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);
}
```

```
...  
}
```

Estrutura de demonstração do AWS SimSpace Weaver

A estrutura de demonstração do AWS SimSpace Weaver é uma biblioteca de utilitários que você pode usar para desenvolver aplicativos do SimSpace Weaver.

A estrutura de demonstração fornece

- Amostras de código e padrões de programação para você usar e examinar
- Abstrações e funções utilitárias que simplificam o desenvolvimento de aplicativos simples
- Uma maneira mais simples de testar recursos experimentais do SimSpace Weaver do aplicativo do SDK

Projetamos o SimSpace Weaver do aplicativo do SDK com acesso de baixo nível às APIs do SimSpace Weaver para entregar maior desempenho. Por outro lado, projetamos a estrutura de demonstração para fornecer abstrações de alto nível e acesso a APIs que facilitam o uso do SimSpace Weaver. O custo da facilidade de uso é um nível de desempenho mais baixo em comparação com o uso direto do SimSpace Weaver do aplicativo do SDK. Simulações que podem tolerar menor desempenho (como aquelas sem requisitos de desempenho em tempo real) podem ser boas candidatas para usar a estrutura de demonstração. Recomendamos que você use a funcionalidade nativa no SimSpace Weaver do aplicativo do SDK para aplicativos complexos porque a estrutura de demonstração não é um kit de ferramentas completo.

A estrutura de demonstração inclui

- Exemplos de código de trabalho que oferecem suporte e demonstram:
 - Gerenciamento do fluxo de aplicativos
 - Processamento de eventos de entidades orientado por retorno de chamada
- Um conjunto de bibliotecas utilitárias de terceiros:
 - spdlog (uma biblioteca de logs)
 - Uma versão mínima do AZCore (uma biblioteca matemática) que contém somente:
 - Vector3
 - Aabb

- `cxxopts` (uma biblioteca de analisadores de opções de linha de comando)
- Funções utilitárias específicas para o SimSpace Weaver

A estrutura de demonstração consiste em uma biblioteca, arquivos de origem e CMakeLists. Os arquivos estão incluídos no pacote distribuível do SimSpace Weaver do aplicativo do SDK.

Trabalhar com cotas de serviço

Esta seção descreve como trabalhar com as cotas de serviço do SimSpace Weaver. As cotas também são chamadas de limites. Para uma lista de cotas de serviço, consulte [SimSpace Terminais e cotas do Weaver](#). As APIs nesta seção são do conjunto de APIs de aplicativo. As APIs de aplicativo são diferentes das APIs de serviço. As APIs de aplicativo fazem parte do SDK do aplicativo do SimSpace Weaver. Você pode encontrar a documentação das APIs de aplicativo na pasta do aplicativo do SDK em seu sistema local:

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

Tópicos

- [Obter os limites de um aplicativo](#)
- [Obter a quantidade de recursos usados por um aplicativo](#)
- [Redefinir métricas](#)
- [Exceder um limite](#)
- [Ficar sem memória](#)
- [Práticas recomendadas](#)

Obter os limites de um aplicativo

Você pode usar a API de aplicativo `RuntimeLimits` para consultar os limites de um aplicativo.

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

Parâmetros

`Application& app`

Uma referência ao aplicativo.

LimitTypeTipo

Uma enumeração com os seguintes tipos de limite:

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

O exemplo a seguir consulta o limite de contagem de entidades.

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

Obter a quantidade de recursos usados por um aplicativo

Você pode chamar a API de aplicativo RuntimeMetrics para obter a quantidade de recursos usados por um aplicativo:

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

Parâmetros

Application& app

Uma referência ao aplicativo.

A API retorna uma referência a uma struct que contém as métricas. Um contador de métricas contém um valor total contínuo crescente. Um medidor de métricas contém um valor que pode aumentar ou diminuir. O tempo de execução do aplicativo atualiza um contador sempre que um evento aumenta em valor. O tempo de execução só atualiza os medidores quando você chama a API. O SimSpace Weaver garante que a referência seja válida durante toda a vida útil do aplicativo. Chamadas repetidas para a API não alterarão a referência.

```
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  
}
```

Redefinir métricas

A API de aplicativo `ResetRuntimeMetrics` redefine os valores no `AppRuntimeMetrics` struct.

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

O exemplo a seguir demonstra como fazer chamadas `ResetRuntimeMetrics` pelo app.

```
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));  
}
```

Exceder um limite

Uma chamada de API do aplicativo que excede um limite retornará um `ErrorCode::CapacityExceeded`, exceto para transferências de entidades. O SimSpace Weaver trata as transferências de entidades de forma assíncrona como parte das operações da API de

aplicativo Confirmar e BeginUpdate. Portanto, não há uma operação específica que retorne um erro se uma transferência falhar devido ao limite de transferência da entidade. Para detectar falhas de transferência, compare os valores atuais de `rejected_incoming_transfer_counter` e `rejected_outgoing_transfer_counter` (no `AppRuntimeMetricsstruct`) com os valores anteriores. As entidades rejeitadas não estarão na partição, mas o aplicativo ainda poderá simulá-las.

Ficar sem memória

O SimSpace Weaver usa um processo de coleta de lixo para limpar e liberar memória. É possível gravar dados mais rápido do que o coletor de lixo pode liberar memória. Se isso acontecer, as operações de gravação poderão exceder o limite de memória reservada do aplicativo. O SimSpace Weaver retornará um erro interno com uma mensagem que contém `OutOfMemory` e detalhes adicionais. Para obter mais informações, consulte [Distribua gravações ao longo do tempo](#).

Práticas recomendadas

As práticas recomendadas a seguir são diretrizes gerais para criar aplicativos e evitar ultrapassar limites. Elas podem não se aplicar ao design específico do seu aplicativo.

Monitorar com frequência e diminuir a velocidade

Você deve monitorar suas métricas com frequência e desacelerar as operações que estão perto de atingir um limite.

Evite exceder os limites de assinatura e de transferência

Se possível, projete sua simulação para reduzir o número de assinaturas remotas e transferências de entidades. Você pode usar grupos de posicionamento para colocar várias partições no mesmo operador e reduzir a necessidade de transferências remotas de entidades entre operadores.

Distribua gravações ao longo do tempo

O número e o tamanho das atualizações em uma marcação podem ter um impacto significativo no tempo e na memória necessários para confirmar uma transação. Grandes requisitos de memória podem fazer com que o tempo de execução do aplicativo fique sem memória. Você pode distribuir as gravações ao longo do tempo para reduzir o tamanho total médio das atualizações por marcação. Isso pode ajudar a melhorar o desempenho e evitar ultrapassar os limites. Recomendamos que você não escreva mais do que uma média de 12 MB em cada marcação ou 1,5 KB para cada entidade.

Depurar simulações

Use os seguintes métodos para obter informações sobre as simulações.

Tópicos

- [Use SimSpace Weaver Local e veja a saída do console](#)
- [Veja seus registros no Amazon CloudWatch Logs](#)
- [Use chamadas de API describe](#)
- [Conectar um cliente](#)

Use SimSpace Weaver Local e veja a saída do console

Recomendamos que você desenvolva suas simulações localmente primeiro e depois as execute na Nuvem AWS. É possível ver a saída do console diretamente ao executar com o SimSpace Weaver Local. Para ter mais informações, consulte [Desenvolvimento local em SimSpace Weaver](#).

Veja seus registros no Amazon CloudWatch Logs

Quando você executa sua simulação no console, Nuvem AWS a saída dos seus aplicativos é enviada para fluxos de log no Amazon CloudWatch Logs. Sua simulação também grava outros dados de log. Você deve habilitar o registro em seu esquema de simulação se quiser que ela grave dados de log. Para ter mais informações, consulte [SimSpace Weaver loga no Amazon CloudWatch Logs](#).

Warning

A simulação pode produzir grandes quantidades de dados de log. Os dados de registro podem crescer muito rapidamente. Você deve observar seus registros de perto e interromper as simulações quando não precisar mais que sejam executadas. Os registros podem gerar grandes custos.

Use chamadas de API describe

É possível usar as seguintes APIs de serviço para obter informações sobre suas simulações na Nuvem AWS.

- ListSimulations— obtenha uma lista de todas as suas simulações no Nuvem AWS.

Example Exemplo

```
aws simspaceweaver list-simulations
```

- DescribeSimulation: obtenha detalhes sobre uma simulação.

Example Exemplo

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

- DescribeApp: obtenha detalhes sobre um aplicativo.

Example Exemplo

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

Para obter mais informações sobre as SimSpace Weaver APIs, consulte [SimSpace Weaver Referências de API](#).

Conectar um cliente

É possível conectar um cliente a um aplicativo personalizado ou de serviço em execução definido com um `endpoint_config` no esquema de simulação. O SDK do SimSpace Weaver aplicativo inclui clientes de amostra que você pode usar para visualizar o aplicativo de amostra. É possível examinar o código-fonte desses clientes de exemplo e o aplicativo de exemplo para ver como criar seus próprios clientes. Para obter mais informações sobre como criar e executar os clientes de amostra, consulte os tutoriais em [Começando com SimSpace Weaver](#)

É possível encontrar o código-fonte dos clientes de exemplo na seguinte pasta:

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

Depuração de simulações locais

É possível depurar aplicativos do SimSpace Weaver Local com o Microsoft Visual Studio. Para obter mais informações sobre como depurar com o Visual Studio, consulte [Microsoft Visual Studio documentation](#).

Como depurar uma simulação local

1. Verifique se o `schema.yaml` está no diretório de trabalho.
2. No Visual Studio, abra o menu de contexto de cada aplicativo que deseja depurar (como `PathfindingSampleLocalSpatial` ou `PathfindingSampleLocalView`) e defina o diretório de trabalho na seção de depuração.
3. Abra o menu de contexto do aplicativo que deseja depurar e selecione Definir como projeto de inicialização.
4. Escolha F5 para começar a depurar o aplicativo.

Os requisitos para depurar uma simulação são os mesmos que os requisitos para executar normalmente uma simulação. Você deve iniciar o número de aplicativos espaciais especificados no esquema. Por exemplo, se seu esquema especificar uma grade 2x2 e você iniciar um aplicativo espacial no modo de depuração, a simulação não será executada até que você inicie mais três aplicativos espaciais (no modo de depuração ou não).

Para depurar um aplicativo personalizado, você deve primeiro iniciar os aplicativos espaciais e depois o aplicativo personalizado no depurador.

Observe que a simulação é executada em sincronia. Assim que um aplicativo atingir um ponto de interrupção, todos os outros aplicativos serão pausados. Depois de continuar a partir desse ponto de interrupção, os outros aplicativos também continuarão.

Contêineres personalizados

AWS SimSpace Weaver aplicativos executados em ambientes em contêineres Amazon Linux 2 (AL2). No Nuvem AWS, SimSpace Weaver executa suas simulações em contêineres Docker criados a partir de uma `amazonlinux:2` imagem servida pelo Amazon Elastic Container Registry (Amazon ECR). É possível criar uma imagem do Docker personalizada, armazená-la no Amazon ECR e usar essa imagem para sua simulação em vez da imagem do Docker que fornecemos.

É possível usar um contêiner personalizado para gerenciar suas dependências de software e incluir componentes de software adicionais que não estão na imagem do Docker padrão. Por exemplo, você pode adicionar as bibliotecas de software disponíveis publicamente que seu aplicativo usa ao contêiner e colocar somente seu código personalizado no arquivo `.zip` do aplicativo.

⚠ Important

Só oferecemos compatibilidade a imagens AL2 do Docker hospedadas em repositórios do Amazon ECR, seja na Galeria Pública do Amazon ECR ou em seu registro privado do Amazon ECR. Não oferecemos compatibilidade a imagens do Docker hospedadas fora do Amazon ECR. Para obter mais informações sobre o Amazon ECR, consulte a [documentação do Amazon Elastic Container Registry](#).

Tópicos

- [Criar um contêiner personalizado](#)
- [Modificar um projeto para usar um contêiner personalizado](#)
- [Perguntas frequentes sobre contêineres personalizados](#)
- [Solução de problemas com contêineres](#)

Criar um contêiner personalizado

Essas instruções pressupõem que você saiba como usar o Docker e o Amazon Elastic Container Registry (Amazon ECR). Para obter mais informações sobre o Amazon ECR, consulte o [Guia do usuário da Amazon ECR](#).

Pré-requisitos

- A identidade (uso ou função) do IAM que você usa para realizar essas ações tem as permissões corretas para usar o Amazon ECR
- O Docker está instalado em seu sistema local

Como criar um contêiner personalizado

1. Crie o Dockerfile.

A) Dockerfile executar AWS SimSpace Weaver aplicativos começa com a Amazon Linux 2 imagem no Amazon ECR.

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

2. Compile seu `Dockerfile`.
3. Carregue a imagem no registro do contêiner do Amazon ECR.
 - [Use o AWS Management Console](#)
 - [Use a AWS Command Line Interface](#)

Note

Se você receber um erro `AccessDeniedException` ao tentar carregar sua imagem de contêiner para o Amazon ECR, sua identidade do IAM (usuário ou função) pode não ter as permissões necessárias para usar o Amazon ECR. Você pode anexar a política `AmazonEC2ContainerRegistryPowerUser` AWS gerenciada à sua identidade do IAM e tentar novamente. Para obter informações sobre como anexar políticas, consulte [Adicionar e remover permissões de identidade do IAM](#) no Guia do usuário do AWS Identity and Access Management .

Modificar um projeto para usar um contêiner personalizado

Essas instruções pressupõem que você já saiba como usar AWS SimSpace Weaver e queira tornar seus fluxos de trabalho de desenvolvimento e armazenamento de aplicativos Nuvem AWS mais eficientes.

Pré-requisitos

- Você tem um contêiner personalizado no Amazon Elastic Container Registry (Amazon ECR). Para obter mais informações sobre como criar um contêiner personalizado, consulte [Criar um contêiner personalizado](#).

Como modificar seu projeto para usar um contêiner personalizado

1. Adicione permissões à função do aplicativo de simulação do seu projeto para usar o Amazon ECR.
 - a. Se você ainda não tem uma política do IAM com as seguintes permissões, crie a política. Sugerimos o nome `simspaceweaver-ecr` para a política. Para obter informações sobre como criar políticas, consulte [Criar políticas do IAM](#) no Guia do usuário do AWS Identity and Access Management .

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

- b. Encontre o nome da função do aplicativo de simulação do seu projeto:
 - i. Em um editor de texto, abra o AWS CloudFormation modelo:

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

- ii. Encontre a propriedade RoleName abaixo de WeaverAppRole. O valor é o nome da função do aplicativo de simulação do seu projeto.

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:
                - 'sinspaceweaver.amazonaws.com'
```

- c. Anexe a política `simspaceweaver-ecr` à função do aplicativo de simulação do projeto. Para obter informações sobre como anexar políticas, consulte [Adicionar e remover permissões de identidade do IAM](#) no Guia do usuário do AWS Identity and Access Management.
- d. Navegue até o comando a seguir `sdk-folder` e execute-o para atualizar a SimSpace Weaver pilha de amostras:

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

2. Especifique suas imagens de contêiner no esquema de simulação do projeto.

- É possível adicionar a propriedade `default_image` opcional abaixo de `simulation_properties` para especificar uma imagem de contêiner personalizada padrão para todos os domínios.
- Adicione a propriedade `image` em `app_config` para um domínio no qual você deseja usar uma imagem de contêiner personalizada. Especifique o URI do repositório do Amazon ECR como o valor. É possível especificar uma imagem diferente para cada domínio.
- Se uma `image` não for especificada para um domínio e uma `default_image` for especificada, os aplicativos desse domínio usarão a imagem padrão.
- Se `image` não for especificado para um domínio e `default_image` não for especificado, os aplicativos nesse domínio serão executados em um SimSpace Weaver contêiner padrão.

Example Trecho de esquema que inclui configurações personalizadas de contêiner

```
sdk_version: "1.17.0"
simulation_properties:
  log_destination_service: "logs"
  log_destination_resource_name: "MySimulationLogs"
  default_entity_index_key_type: "Vector3<f32>"
  default_image: "111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest" # image to use if no image specified for a domain
domains:
  MyCustomDomain:
    launch_apps_via_start_app_call: {}
    app_config:
      package: "s3://weaver-myproject-111122223333-us-west-2/MyViewApp.zip"
      launch_command: ["MyViewApp"]
      required_resource_units:
```



```
    compute: 1
    endpoint_config:
      ingress_ports:
        - 7000
    image: "111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-
repository:latest" # custom container image to use for this domain
  MySpatialDomain:
    launch_apps_by_partitioning_strategy:
      partitioning_strategy: "MyGridPartitioning"
      grid_partition:
        x: 2
        y: 2
    app_config:
      package: "s3://weaver-myproject-111122223333-us-west-2/MySpatialApp.zip"
      launch_command: ["MySpatialApp"]
      required_resource_units:
        compute: 1
    image: "111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-
repository:latest" # custom container image to use for this domain
```

3. Crie e carregue o seu projeto normalmente.

Perguntas frequentes sobre contêineres personalizados

P1. O que eu faço se eu quiser alterar o conteúdo do meu contêiner?

- Para uma simulação em execução: você não pode alterar o contêiner para uma simulação em execução. Você deve criar um novo contêiner e iniciar uma nova simulação que use esse contêiner.
- Para uma nova simulação: crie um novo contêiner, faça o upload para o Amazon Elastic Container Registry (Amazon ECR) e inicie uma nova simulação que use esse contêiner.

P2. Como posso alterar a imagem do contêiner para minha simulação?

- Para uma simulação em execução: você não pode alterar o contêiner para uma simulação em execução. Você deve iniciar uma nova simulação que use o novo contêiner.
- Para uma nova simulação: especifique a nova imagem do contêiner no esquema de simulação do seu projeto. Para ter mais informações, consulte [Modificar um projeto para usar um contêiner personalizado](#).

Solução de problemas com contêineres

Tópicos

- [AccessDeniedException ao enviar sua imagem para o Amazon Elastic Container Registry \(Amazon ECR\)](#)
- [Uma simulação que usa um contêiner personalizado falha ao iniciar](#)

AccessDeniedException ao enviar sua imagem para o Amazon Elastic Container Registry (Amazon ECR)

Se você receber um erro `AccessDeniedException` ao tentar carregar sua imagem de contêiner para o Amazon ECR, sua identidade do IAM (usuário ou função) pode não ter as permissões necessárias para usar o Amazon ECR. Você pode anexar a política `AmazonEC2ContainerRegistryPowerUser` AWS gerenciada à sua identidade do IAM e tentar novamente. Para obter informações sobre como anexar políticas, consulte [Adicionar e remover permissões de identidade do IAM](#) no Guia do usuário do AWS Identity and Access Management .

Uma simulação que usa um contêiner personalizado falha ao iniciar

Dicas de solução de problemas

- Se o registro estiver ativado para sua simulação, verifique seus logs de erro.
- Teste sua simulação sem um contêiner personalizado.
- Teste sua simulação localmente. Para ter mais informações, consulte [Desenvolvimento local em SimSpace Weaver](#).

Como trabalhar com Python

Você pode usar o Python para seus SimSpace Weaver aplicativos e clientes. O kit de desenvolvimento de software Python (Python SDK) está incluído como parte do pacote distribuível padrão SimSpace Weaver do SDK do aplicativo. O desenvolvimento com Python funciona de forma semelhante ao desenvolvimento em outras linguagens compatíveis.

Important

SimSpace Weaver suporta apenas Python versão 3.9.

⚠ Important

SimSpace Weaver o suporte para Python requer a SimSpace Weaver versão 1.15.0 ou posterior.

Tópicos

- [Criar um projeto do Python](#)
- [Iniciar uma simulação em Python](#)
- [Cliente de exemplo Python](#)
- [Perguntas frequentes sobre o uso do Python](#)
- [Solução de problemas relacionados ao Python](#)

Criar um projeto do Python

Contêiner personalizado do Python

Para executar sua SimSpace Weaver simulação baseada em Python no Nuvem AWS, você pode criar um contêiner personalizado que inclua as dependências necessárias. Para ter mais informações, consulte [Contêineres personalizados](#).

Um contêiner personalizado do Python deve incluir o seguinte:

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

Se você usar o modelo PythonBubblesSample para criar seu projeto, poderá executar o script `quick-start.py` (localizado na pasta `tools` do projeto) para criar uma imagem do Docker com

as dependências necessárias. O script carrega a imagem para o Amazon Elastic Container Registry (Amazon ECR).

O script `quick-start.py` usa o seguinte 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
```

É possível adicionar suas próprias dependências ao Dockerfile:

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

O arquivo `requirements.txt` contém uma lista dos pacotes Python necessários para a simulação da amostra `PythonBubblesSample`:

```
Flask==2.1.1
```

É possível adicionar suas próprias dependências do pacote Python ao `requirements.txt`:

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

O Dockerfile e o `requirements.txt` estão na pasta `tools` do projeto.

Important

Tecnicamente, você não precisa usar um contêiner personalizado com sua simulação do Python, mas é altamente recomendável que você use um contêiner personalizado. O

contêiner padrão do Amazon Linux 2 (AL2) que fornecemos não tem Python. Portanto, se você não usar um contêiner personalizado que tenha Python, deverá incluir o Python e as dependências necessárias em cada arquivo zip do aplicativo para o qual você carrega. SimSpace Weaver

Iniciar uma simulação em Python

Você pode iniciar sua simulação baseada em Python da mesma forma que uma SimSpace Weaver simulação normal, tanto no SimSpace Weaver Local quanto no SimSpace Weaver . Nuvem AWS Para obter mais informações, consulte os tutoriais em [Começando com SimSpace Weaver](#)

O PythonBubblesSample inclui seu próprio cliente de exemplo em Python. Para ter mais informações, consulte [Cliente de exemplo Python](#).

Cliente de exemplo Python

Se você usar o modelo PythonBubblesSample para criar um projeto, ele conterá um cliente de exemplo em Python. É possível usar o cliente de exemplo para visualizar a simulação PythonBubblesSample. Você também pode usar o cliente de exemplo como ponto de partida para criar seu próprio cliente Python.

O procedimento a seguir pressupõe que você tenha criado um projeto PythonBubblesSample e iniciado sua simulação.

Como iniciar o cliente Python

1. Em uma janela do prompt de comando, acesse a pasta do projeto de PyBubbleClient amostra.

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

2. Execute o cliente Python.

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

Parâmetros

host

O endereço IP da simulação. Para uma simulação iniciada no Nuvem AWS, você pode encontrar o endereço IP da simulação no [SimSpace Weaver console](#) ou usar o procedimento [Obtenha o endereço IP e o número da porta de um aplicativo personalizado](#) do tutorial de início rápido. Para uma simulação local, use `127.0.0.1` como endereço IP.

port

O número da porta da simulação. Para uma simulação iniciada no Nuvem AWS, esse é o número da Actual porta. É possível encontrar o número da porta da simulação no [console do SimSpace Weaver](#) ou usar o procedimento em [Obtenha o endereço IP e o número da porta de um aplicativo personalizado](#) do tutorial de início rápido. Para uma simulação local, use `7000` como número da porta.

simsize

O número máximo de entidades a serem exibidas no cliente.

Perguntas frequentes sobre o uso do Python

P1. Quais versões do Python são compatíveis?

SimSpace Weaver suporta apenas Python versão 3.9.

Solução de problemas relacionados ao Python

Tópicos

- [Falha durante a criação do contêiner personalizado](#)
- [A simulação do Python falha ao iniciar](#)
- [Um cliente de simulação ou visualização do Python gera um erro ModuleNotFound](#)

Falha durante a criação do contêiner personalizado

Se você receber um erro de `no basic auth credentials` após a execução de `quick-start.py`, pode haver um problema com suas credenciais temporárias para o Amazon ECR. Execute o seguinte comando com seu Região da AWS ID e número AWS da conta:

```
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

Certifique-se de que a Região da AWS especificado seja o mesmo que você usa para sua simulação. Use um desses Regiões da AWS SimSpace Weaver suportes. Para ter mais informações, consulte [SimSpace Terminais e cotas do Weaver](#).

Depois de executar o comando `aws ecr`, execute o `quick-start.py` novamente.

Outros recursos de solução de problemas a serem verificados

- [Solução de problemas com contêineres](#)
- [Solução de problemas do Amazon ECR](#) no Guia do usuário do Amazon ECR
- [Confirmar com o Amazon ECR](#) no Guia do usuário do Amazon ECR

A simulação do Python falha ao iniciar

É possível ver o erro `Unable to start app` no log de gerenciamento da sua simulação. Isso pode acontecer se a criação do contêiner personalizado falhar. Para ter mais informações, consulte [Falha durante a criação do contêiner personalizado](#). Para obter mais informações sobre logs, consulte [SimSpace Weaver loga no Amazon CloudWatch Logs](#).

Se você tiver certeza de que não há nada de errado com seu contêiner, verifique o código-fonte Python do aplicativo. É possível usar o SimSpace Weaver Local para testar o aplicativo. Para ter mais informações, consulte [Desenvolvimento local em SimSpace Weaver](#).

Um cliente de simulação ou visualização do Python gera um erro `ModuleNotFound`

O Python gera um erro `ModuleNotFound` quando não consegue localizar um pacote Python necessário.

Se sua simulação estiver no Nuvem AWS, certifique-se de que seu contêiner personalizado tenha todas as dependências necessárias listadas em `seurequirements.txt`. Lembre-se de executar o `quick-start.py` novamente se você editar o `requirements.txt`.

Se você receber o erro para o cliente PythonBubblesSample, use `pip` para instalar o pacote indicado:

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

Suporte para outros mecanismos

Você pode usar seu próprio C++ mecanismo personalizado com SimSpace Weaver. No momento, estamos desenvolvendo compatibilidade para os seguintes mecanismos. Há documentação separada para cada um desses mecanismos.

Important

As integrações com os mecanismos listados aqui são experimentais. Eles estão disponíveis para visualização.

Mecanismos

- [Unity](#)(versão mínima 2022.3.19.F1)
- [Unreal Engine](#) (versão mínima 5.0)

Unity

Você deve ter o ambiente de Unity desenvolvimento já instalado antes de criar SimSpace Weaver simulações com o Unity. Para obter mais informações, consulte as instruções separadas:

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

Unreal Engine

Você deve criar um servidor dedicado para o Unreal Engine a partir do código-fonte. SimSpaceWeaverAppSdkDistributable Inclui uma versão do PathfindingSample forUnreal Engine. Para obter mais informações, consulte as instruções separadas:

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

Uso de software licenciado com AWS SimSpace Weaver

O AWS SimSpace Weaver permite que você crie simulações com o mecanismo e o conteúdo de simulação de sua escolha. Em relação ao uso de SimSpace Weaver, você é responsável por obter, manter e aderir aos termos de licença de qualquer software ou conteúdo usado em suas simulações. Verifique se seu contrato de licenciamento permite implantar seu software e conteúdo em um ambiente de host virtual.

Gerenciar recursos com o AWS CloudFormation

Você pode usar o AWS CloudFormation para gerenciar seus recursos no AWS SimSpace Weaver. O AWS CloudFormation é um serviço da AWS distinto, que ajuda a especificar, provisionar e gerenciar a infraestrutura AWS como código. Com o AWS CloudFormation, você cria um arquivo JSON ou YAML chamado [modelo](#). O modelo especifica os detalhes da infraestrutura. O AWS CloudFormation usa seu modelo para provisionar a infraestrutura como uma única unidade, chamada [pilha](#). Ao excluir a pilha, você pode fazer o AWS CloudFormation excluir tudo na pilha ao mesmo tempo. Você pode gerenciar o modelo usando processos de gerenciamento de código-fonte padrão (por exemplo, rastreando-o em um sistema de controle de versão como o [Git](#)). Para obter mais informações sobre o AWS CloudFormation, consulte o [Guia do usuário do AWS CloudFormation](#).

Seu recurso de simulação

Na AWS, um recurso é uma entidade com a qual você pode trabalhar. Exemplos incluem uma instância do Amazon EC2, um bucket do Amazon S3 ou um perfil do IAM. Sua simulação do SimSpace Weaver é um recurso. Nas configurações, você geralmente especifica um recurso AWS no formulário de `AWS::service::resource`. Para o SimSpace Weaver, especifique o recurso de simulação como `AWS::SimSpaceWeaver::Simulation`. Para obter mais informações sobre seu recurso de simulação no AWS CloudFormation, consulte a seção [SimSpace Weaver](#) no Guia do usuário do AWS CloudFormation.

Como posso usar o AWS CloudFormation com o SimSpace Weaver?

Você pode criar um modelo do AWS CloudFormation que especifique os recursos da AWS que deseja provisionar. Seu modelo pode especificar uma arquitetura inteira, parte de uma arquitetura ou uma pequena solução. Por exemplo, você pode especificar uma arquitetura para sua solução

do SimSpace Weaver que inclui buckets do Amazon S3, permissões do IAM, um banco de dados relacional de suporte no Amazon Relational Database Service ou no Amazon DynamoDB, e a `Simulation` de seu recurso. Em seguida, você pode usar o AWS CloudFormation para provisionar todos esses recursos como uma unidade e ao mesmo tempo.

Exemplo Modelo que cria recursos do IAM e inicia uma simulação

O modelo de exemplo a seguir cria permissões e um perfil do IAM que o SimSpace Weaver usará para realizar ações em sua conta. Os scripts do aplicativo do SDK do SimSpace Weaver criam a função e as permissões em uma área específica da Região da AWS ao criar um projeto, mas você pode usar um modelo do AWS CloudFormation para implantar a simulação em outra Região da AWS sem executar os scripts novamente. Por exemplo, você pode fazer isso para configurar uma simulação de backup para fins de recuperação de desastres.

Neste exemplo, o nome original da simulação é `MySimulation`. Já existe um bucket para o esquema no local da Região da AWS onde o AWS CloudFormation construirá a pilha. O bucket contém uma versão do esquema configurada adequadamente para executar a simulação naquela Região da AWS. Lembre-se de que o esquema especifica a localização dos arquivos `.zip` do seu aplicativo, que é um bucket do Amazon S3 na mesma Região da AWS da simulação. O bucket compactado do aplicativo e os arquivos já devem existir na Região da AWS quando o AWS CloudFormation criar a pilha. Caso contrário, a simulação não será iniciada. Observe que o nome do bucket neste exemplo inclui a Região da AWS, mas isso não determina onde o bucket está realmente localizado. Você deve se certificar de que o bucket está realmente naquela Região da AWS. Verifique as propriedades do bucket no console do Amazon S3, com as APIs do Amazon S3 ou com os comandos do Amazon S3 na AWS CLI.

Este exemplo usa algumas funções e parâmetros integrados no AWS CloudFormation para realizar a substituição de variáveis. Para mais informações, consulte [Referência da função intrínseca](#) e [Referência de pseudoparâmetros](#) no Guia do usuário do 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'

```

Usando snapshots com o AWS CloudFormation

Um [snapshot](#) é um backup de uma simulação. O exemplo a seguir inicia uma nova simulação a partir de um instantâneo em vez de um esquema. O instantâneo neste exemplo foi criado a partir de uma

simulação de projeto do aplicativo do SDK do SimSpace Weaver. O AWS CloudFormation cria o novo recurso de simulação e o inicializa com dados do snapshot. A nova simulação pode ter uma `MaximumDuration` diferente da simulação original.

Recomendamos que você faça e use uma cópia da função do aplicativo da simulação original. A função do aplicativo da simulação original pode ser excluída se você excluir a pilha do AWS CloudFormation dessa simulação.

```
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"
```

Snapshots

É possível criar um snapshot para fazer backup dos dados da entidade de simulação a qualquer momento. O SimSpace Weaver criará um arquivo `.zip` em um bucket do Amazon S3. Você pode criar uma nova simulação com o instantâneo. SimSpace Weaver inicializa o State Fabric de sua nova simulação com os dados da entidade armazenados no instantâneo, inicia os aplicativos espaciais e de serviços que estavam em execução quando o instantâneo foi criado e ajusta o relógio para a marca apropriada. SimSpace Weaver obtém a configuração da sua simulação a partir do instantâneo em vez de um arquivo de esquema. Os arquivos `.zip` do aplicativo devem estar no mesmo local no Amazon S3 em que estavam na simulação original. Você deve iniciar todos os aplicativos personalizados separadamente.

Tópicos

- [Casos de uso para snapshots](#)
- [Use o SimSpace Weaver console para trabalhar com instantâneos](#)

- [Use o AWS CLI para trabalhar com instantâneos](#)
- [Usando snapshots com o AWS CloudFormation](#)
- [Perguntas frequentes sobre snapshots](#)

Casos de uso para snapshots

Retornar a um estado anterior e explorar cenários de ramificação

É possível criar um snapshot da simulação para salvá-la em um estado específico. Em seguida, você pode criar várias novas simulações a partir desse snapshot e explorar diferentes cenários que podem se ramificar a partir desse estado.

Práticas recomendadas de recuperação de desastres e segurança

Recomendamos que você faça backup regular da simulação, especialmente para simulações que são executadas por mais de uma hora ou usam vários operadores. Os backups podem ajudar você a se recuperar de desastres e incidentes de segurança. Os snapshots fornecem uma maneira de você fazer backup da simulação. Os snapshots exigem que os arquivos .zip do aplicativo estejam no mesmo local no Amazon S3 de antes. Se você precisar mover os arquivos .zip do aplicativo para outro local, deverá usar uma solução de backup personalizada.

Para obter mais informações sobre as melhores práticas, consulte [Práticas recomendadas ao trabalhar com SimSpace Weaver](#) e [Práticas recomendadas de segurança para SimSpace Weaver](#).

Estender a duração da simulação

O recurso de simulação é a representação da simulação no SimSpace Weaver. Todos os recursos de simulação têm uma configuração de `MaximumDuration`. Um recurso de simulação é interrompido automaticamente quando atinge o `MaximumDuration`. O valor máximo de `MaximumDuration` é 14D (14 dias).

Se você precisar que sua simulação persista por mais tempo do que `MaximumDuration` do recurso de simulação, você pode criar um snapshot antes que o recurso de simulação alcance a `MaximumDuration`. É possível iniciar uma nova simulação (criar um novo recurso de simulação) com seu snapshot. O SimSpace Weaver inicializa os dados da entidade a partir do snapshot, inicia os mesmos aplicativos espaciais e de serviços executados antes e restaura o relógio. É possível iniciar aplicativos personalizados e realizar qualquer inicialização personalizada adicional. É possível definir a `MaximumDuration` do novo recurso de simulação com um valor diferente ao iniciá-lo.

Use o SimSpace Weaver console para trabalhar com instantâneos

Você pode usar o SimSpace Weaver console para criar um instantâneo da sua simulação.

Tópicos

- [Usar o console para criar um snapshot](#)
- [Usar o console para iniciar uma simulação a partir de um snapshot](#)

Usar o console para criar um snapshot

Para criar um snapshot

1. Faça login no AWS Management Console e conecte-se ao [SimSpace Weaver console](#).
2. No painel de navegação, escolha Simulações.
3. Selecione o botão de opção ao lado do nome da simulação. O status da simulação deve ser Iniciado.
4. Na parte superior da página, escolha Criar política.
5. Em Configurações do snapshot, em Destino do snapshot, insira o URI do Amazon S3 de um bucket ou bucket e pasta onde você SimSpace Weaver deseja criar seu snapshot. É possível escolher Procurar no S3 se preferir navegar pelos buckets disponíveis e selecionar um local.

Important

O bucket do Amazon S3 deve estar na mesma Região da AWS que a simulação.

Note

SimSpace Weaver cria uma snapshot pasta dentro do destino do snapshot selecionado. SimSpace Weaver cria o arquivo.zip de instantâneo nessa snapshot pasta.

6. Escolha Criar snapshot.

Usar o console para iniciar uma simulação a partir de um snapshot

Para iniciar uma simulação a partir de um snapshot, seu arquivo .zip do snapshot deve existir em um bucket do Amazon S3 que sua simulação possa acessar. Sua simulação usa permissões definidas na função do aplicativo que você selecionar ao iniciar a simulação. Todos os arquivos .zip do aplicativo da simulação original devem existir nos mesmos locais de quando o snapshot foi criado.

Como iniciar uma simulação a partir de um snapshot

1. Faça login no AWS Management Console e conecte-se ao [SimSpace Weaver console](#).
2. No painel de navegação, escolha Simulações.
3. Na parte superior da página, escolha Iniciar simulação.
4. Em Configurações de simulação, insira um nome e uma descrição opcional para sua simulação. O nome da simulação deve ser exclusivo na Conta da AWS.
5. Para o Método de início da simulação, escolha Usar um snapshot no Amazon S3.
6. Para o URI do Amazon S3 para snapshot, insira o URI do Amazon S3 do arquivo de snapshot ou escolha Procurar no S3 para navegar e selecionar o arquivo.

Important

O bucket do Amazon S3 deve estar na mesma Região da AWS que a simulação.

7. Para o Perfil do IAM, selecione a função do aplicativo que sua simulação usará.
8. Em Duração máxima, insira o tempo máximo pelo qual seu recurso de simulação deve ser executado. O valor máximo é 14D. Para mais informações sobre a duração máxima, consulte [Duração máxima](#).
9. Em Tags: opcional, escolha Adicionar nova tag se quiser adicionar uma tag.
10. Escolha Iniciar simulação.

Use o AWS CLI para trabalhar com instantâneos

Você pode usar o AWS CLI para chamar as SimSpace Weaver APIs a partir de um prompt de comando. Você deve ter o AWS CLI instalado e configurado corretamente. Para obter mais informações, consulte [Instalando ou atualizando a versão mais recente da AWS CLI](#) no Guia do AWS Command Line Interface usuário da versão 2.

Tópicos

- [Use o AWS CLI para criar um instantâneo](#)
- [Use o AWS CLI para iniciar uma simulação a partir de um instantâneo](#)

Use o AWS CLI para criar um instantâneo

Para criar um snapshot

- Em um prompt de comando, chame a API CreateSnapshot.

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

Parâmetros

simulation

O nome de uma simulação iniciada. É possível usar `aws simspaceweaver list-simulations` para ver os nomes e status das simulações.

destination

Uma string que especifica o bucket de destino do Amazon S3 e o prefixo de chave de objeto opcional para seu arquivo de snapshot. Seu prefixo de chave de objeto geralmente é uma pasta no seu bucket. SimSpace Weaver cria seu instantâneo dentro de uma snapshot pasta nesse destino.

Important

O bucket do Amazon S3 deve estar na mesma Região da AWS que a simulação.

Exemplo

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


Para obter mais informações sobre a CreateSnapshot API, consulte [CreateSnapshot](#) a Referência AWS SimSpace Weaver da API.

Use o AWS CLI para iniciar uma simulação a partir de um instantâneo

Como iniciar uma simulação a partir de um snapshot

- Em um prompt de comando, chame a API StartSimulation.

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

Parâmetros

name

Nome da nova simulação. O nome da simulação deve ser exclusivo em seu Conta da AWS. É possível usar `aws simspaceweaver list-simulations` para ver os nomes das simulações existentes.

role-arn

Nome do Recurso da Amazon (ARN) da função de aplicativo que sua simulação usará.

snapshot-s3-location

Uma string que especifica o bucket do Amazon S3 e a chave do objeto do arquivo de snapshot.

Important

O bucket do Amazon S3 deve estar na mesma Região da AWS que a simulação.

Exemplo

```
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 obter mais informações sobre a `StartSimulation` API, consulte [StartSimulation](#) Referência AWS SimSpace Weaver da API.

Perguntas frequentes sobre snapshots

A simulação continua sendo executada durante um snapshot?

Os recursos de simulação continuam sendo executados durante um snapshot e você continua recebendo cobranças de faturamento por esse período. O tempo conta para a duração máxima da simulação. Seus aplicativos não recebem marcações enquanto o snapshot está em andamento. Se o status do relógio era `STARTED` quando a criação do snapshot começou, seu relógio ainda indicará o status `STARTED`. Os aplicativos recebem marcações novamente após o término do snapshot. Se o status do relógio era `STOPPED`, o status do relógio permanecerá `STOPPED`. Observe que uma simulação com um status `STARTED` está sendo executada mesmo que o status do relógio seja `STOPPED`.

O que acontece se um snapshot estiver em andamento e a simulação atingir sua duração máxima?

A simulação finalizará o snapshot e será interrompida assim que o processo de captura do snapshot terminar (com ou sem sucesso). Recomendamos que você teste o processo de snapshot com antecedência para descobrir quanto tempo leva, o tamanho do arquivo de snapshot que você pode esperar e se ele será concluído com êxito.

O que acontece se eu interromper uma simulação que tem um snapshot em andamento?

Um snapshot em andamento é interrompido imediatamente quando você interrompe a simulação. Ele não criará um arquivo de snapshot.

Como posso interromper um snapshot em andamento?

A única maneira de interromper um snapshot em andamento é interromper a simulação. Não é possível reiniciar a simulação depois que ela for interrompida.

Quanto tempo vai levar para concluir o snapshot?

O tempo necessário para criar um snapshot depende da simulação. Recomendamos que você teste o processo de captura instantânea com antecedência para descobrir quanto tempo levará para sua simulação.

Qual será o tamanho do meu arquivo de snapshot?

O tamanho de um arquivo de snapshot depende da simulação. Recomendamos que você teste o processo de captura do snapshot com antecedência para descobrir o tamanho do arquivo para a simulação.

Sistema de mensagens

A API de mensagens simplifica a comunicação entre aplicativos na simulação. As APIs para enviar e receber mensagens fazem parte do SDK do SimSpace Weaver aplicativo. Atualmente, as mensagens usam a abordagem de melhor esforço para enviar e receber mensagens. SimSpace O Weaver tenta enviar/receber mensagens na próxima simulação, mas não há garantias de entrega, pedido ou horário de chegada.

Tópicos

- [Casos de uso para mensagens](#)
- [Usando as APIs de mensagens](#)
- [Quando usar mensagens](#)
- [Dicas para trabalhar com mensagens](#)
- [Erros de mensagens e solução de problemas](#)

Casos de uso para mensagens

Comunique-se entre aplicativos de simulação

Use a API de mensagens para se comunicar entre os aplicativos em sua simulação. Use-o para alterar o estado das entidades à distância, alterar o comportamento da entidade ou transmitir informações para toda a simulação.

Confirmar o recebimento de uma mensagem

As mensagens enviadas contêm informações sobre o remetente no cabeçalho da mensagem. Use essas informações para enviar de volta uma resposta de confirmação ao receber uma mensagem.

Encaminhar dados recebidos por um aplicativo personalizado para outros aplicativos dentro da simulação

As mensagens não substituem a forma como os clientes se conectam aos aplicativos personalizados em execução SimSpace Weaver. No entanto, as mensagens permitem aos usuários um método

de encaminhar dados de aplicativos personalizados que recebem dados do cliente para outros aplicativos que não têm uma conexão externa. O fluxo de mensagens também pode funcionar ao contrário, permitindo que aplicativos sem conexão externa encaminhem dados para um aplicativo personalizado e depois para um cliente.

Usando as APIs de mensagens

As APIs de mensagens estão contidas no SDK do SimSpace Weaver aplicativo (versão mínima 1.16.0). O sistema de mensagens é compatível com C++, Python e nossas integrações com o Unreal Engine 5 e o Unity.

Há duas funções que lidam com transações de mensagens: `SendMessage` e `ReceiveMessages`. Todas as mensagens enviadas contêm um destino e uma carga útil. A `ReceiveMessages` API retorna uma lista das mensagens atualmente na fila de mensagens de entrada de um aplicativo.

C++

Enviar mensagem

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

Receba mensagens

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

Python

Enviar mensagem

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

Receba mensagens

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

Tópicos

- [Enviar mensagens](#)
- [Recebimento de mensagens](#)
- [Respondendo ao remetente](#)

Enviar mensagens

As mensagens consistem em uma transação (semelhante a outras chamadas da API Weaver), uma carga útil e um destino.

Carga útil da mensagem

A carga útil da mensagem é uma estrutura de dados flexível de até 256 bytes. Recomendamos o seguinte como uma prática recomendada para criar suas cargas de mensagens.

Para criar a carga útil da mensagem

1. Crie uma estrutura de dados (como a `struct` em C++) que defina o conteúdo da mensagem.
2. Crie a carga útil da mensagem que contém os valores a serem enviados em sua mensagem.
3. Crie o `MessagePayload` objeto.

Destino da mensagem

O destino de uma mensagem é definido pelo `MessageEndpoint` objeto. Isso inclui um tipo de endpoint e uma ID de endpoint. O único tipo de endpoint atualmente suportado é `Partition`, que permite endereçar mensagens para outras partições na simulação. O ID do endpoint é o ID da partição do seu destino.

Você só pode fornecer 1 endereço de destino em uma mensagem. Crie e envie várias mensagens se quiser enviar mensagens para mais de uma partição ao mesmo tempo.

Para obter orientação sobre como resolver um endpoint de mensagem a partir de uma posição, consulte [Dicas para trabalhar com mensagens](#).

Envie a mensagem

Você pode usar a SendMessage API depois de criar os objetos de destino e de carga útil.

C++

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

Python

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

Exemplo completo de envio de mensagens

O exemplo a seguir demonstra como você pode criar e enviar uma mensagem genérica. Este exemplo envia 16 mensagens individuais. Cada mensagem contém uma carga útil com um valor entre 0 e 15 e a marca de simulação atual.

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
```


Recebimento de mensagens

SimSpace Weaver entrega mensagens na fila de mensagens de entrada de uma partição. Use a `ReceiveMessages` API para obter um `MessageList` objeto que contém as mensagens da fila. Processe cada mensagem com a `ExtractMessage` API para obter os dados da mensagem.

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
```

Respondendo ao remetente

Cada mensagem recebida contém um cabeçalho com informações sobre o remetente original da mensagem. Você pode usar o `message.header.source_endpoint` para enviar uma resposta.

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

```

Quando usar mensagens

O envio de mensagens SimSpace Weaver oferece outro padrão para a troca de informações entre aplicativos de simulação. As assinaturas fornecem um mecanismo de pull para ler dados de aplicativos ou áreas específicas da simulação; as mensagens fornecem um mecanismo push para enviar dados para aplicativos ou áreas específicas da simulação.

Abaixo estão dois casos de uso em que é mais útil enviar dados usando mensagens em vez de extrair ou ler dados por meio de uma assinatura.

Example 1: Enviar um comando para outro aplicativo para alterar a posição de uma entidade

```

// 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);
```

No lado receptor, o aplicativo atualiza a posição da entidade e a grava no 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: Enviando uma mensagem de criação de entidade para um aplicativo 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);
```

No lado receptor, o aplicativo cria uma nova entidade no State Fabric e atualiza sua posição.

```
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)));
}
```

Dicas para trabalhar com mensagens

Resolver um endpoint a partir de uma posição ou nome de aplicativo

Você pode usar a `AllPartitions` função para obter os limites espaciais e o ID de domínio necessários para determinar os IDs de partição e os destinos das mensagens. No entanto, se você souber a posição que deseja enviar uma mensagem, mas não o ID da partição, poderá usar a `MessageEndpointResolver` função.

```
/**
 * 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);
};
```

Serializando e desserializando a carga útil da mensagem

Você pode usar as funções a seguir para criar e ler cargas de mensagens. Para obter mais informações, consulte `MessagingUtils.h` na biblioteca do SDK do aplicativo em seu 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());
}
```

Erros de mensagens e solução de problemas

Você pode enfrentar os seguintes erros ao usar as APIs de mensagens.

Erros de resolução de endpoints

Esses erros podem ocorrer antes que um aplicativo envie uma mensagem.

Verificação do nome de domínio

Enviar uma mensagem para um endpoint inválido resulta no seguinte erro:

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

Isso pode acontecer quando você tenta enviar uma mensagem para um aplicativo personalizado e esse aplicativo personalizado ainda não entrou na simulação. Use a `DescribeSimulation` API para garantir que seu aplicativo personalizado tenha sido lançado antes de você enviar uma mensagem para ele. Esse comportamento é o mesmo em SimSpace Weaver Local e Nuvem AWS o.

Verificação de posição

Tentar resolver um endpoint com um nome de domínio válido, mas com uma posição inválida, resulta no seguinte erro.


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

Sugerimos usar o `MessageEndpointResolver` na `MessageUtils` biblioteca contida no SDK do SimSpace Weaver aplicativo.

Erros de envio de mensagens

Os erros a seguir podem ocorrer quando um aplicativo envia uma mensagem.

Limite de envio de mensagens por aplicativo, por tick, excedido

O limite atual para o número de mensagens que podem ser enviadas por aplicativo por clique de simulação é 128. As chamadas subsequentes com o mesmo tick falharão com o seguinte erro:

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

SimSpace Weaver tenta enviar mensagens não enviadas na próxima marca. Diminua a frequência de envio para resolver esse problema. Combine cargas de mensagens menores que o limite de 256 bytes para reduzir o número de mensagens de saída.

Esse comportamento é o mesmo em SimSpace Weaver Local e no Nuvem AWS.

Limite de tamanho da carga útil da mensagem excedido

O limite atual para o tamanho da carga útil da mensagem é 256 bytes em ambos SimSpace Weaver Local e no Nuvem AWS. Enviar uma mensagem com uma carga útil maior que 256 bytes resulta no seguinte erro:

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

SimSpace Weaver verifica cada mensagem e rejeita somente aquelas que excedem o limite. Por exemplo, se seu aplicativo tentar enviar 10 mensagens e 1 falhar na verificação, somente essa mensagem será rejeitada. SimSpace Weaver envia as outras 9 mensagens.

Esse comportamento é o mesmo em SimSpace Weaver Local e Nuvem AWS o.

O destino é o mesmo que a origem

Os aplicativos não podem enviar mensagens para as partições de sua propriedade. Você receberá o erro a seguir se um aplicativo enviar uma mensagem para uma partição de sua propriedade.

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

Esse comportamento é o mesmo em SimSpace Weaver Local e Nuvem AWS o.

Mensagens de melhor esforço

SimSpace Weaver não garante a entrega da mensagem. O serviço tentará concluir a entrega das mensagens na simulação subsequente, mas as mensagens podem ser perdidas ou atrasadas.

Práticas recomendadas ao trabalhar com SimSpace Weaver

Recomendamos as seguintes melhores práticas ao trabalhar com SimSpace Weaver.

Tópicos

- [Configurar alarmes de faturamento](#)
- [Use SimSpace Weaver Local](#)
- [Interrompa simulações que você não precisa](#)
- [Exclua os recursos que você não precisa.](#)
- [Faça backups](#)

Configurar alarmes de faturamento

É fácil provisionar recursos AWS e deixá-los funcionando o tempo todo, mesmo quando não são mais necessários. Isso pode resultar em custos excessivos que podem ser uma surpresa quando você recebe sua fatura. Você pode configurar um alarme na Amazon CloudWatch que acionará e notificará você quando seus custos excederem um limite definido por você. Você pode examinar seus custos usando ferramentas de gerenciamento de custos. Para obter mais informações, consulte:

- [Crie um alarme de cobrança para monitorar suas cobranças estimadas AWS](#)
- [O que é AWS Cost Management](#)

Use SimSpace Weaver Local

Recomendamos que você use SimSpace Weaver Local para desenvolver e testar suas simulações antes de enviá-las para o SimSpace Weaver serviço no. Nuvem AWS Os benefícios de se desenvolver com SimSpace Weaver Local incluem:

- Não é necessário esperar por grandes carregamentos
- Não há limite para o número de simulações locais que você pode criar
- Você não é cobrado pelo tempo de computação em seu computador local
- Acesso direto à saída do console dos aplicativos

- Modifique, reconstrua e reinicie sua simulação local sem precisar recriá-la no Nuvem AWS

Interrompa simulações que você não precisa

Você recebe cobranças de faturamento de uma simulação enquanto ela está sendo executada. Você deve interromper uma simulação para parar de receber cobranças por ela. A execução de simulações também é computada em sua cota para o número máximo de simulações. Uma simulação em execução com o registro configurado também pode gerar grandes quantidades de logs, pelos quais você também recebe cobranças de faturamento. Você deve interromper qualquer simulação desnecessária para parar de receber cobranças adicionais.

Important

Parar o relógio da simulação não interrompe a simulação. O relógio simplesmente para de publicar marcações em nos aplicativos. Não é possível reiniciar uma simulação depois de interrompê-la.

Exclua os recursos que você não precisa.

Cada simulação que você cria SimSpace Weaver também cria recursos em outros AWS serviços. Você pode obter cobranças de faturamento por recursos e dados nesses outros serviços. Simulações em execução e com falha contam para sua cota para o número máximo de simulações. Você deve excluir simulações com falha desnecessárias para poder iniciar novas simulações. Quando você exclui uma simulação, os recursos para sua simulação que existem em outros AWS serviços podem não ser excluídos. Por exemplo, todos os dados de registro de simulação no Amazon CloudWatch Logs permanecerão lá até que você os exclua. Você receberá cobranças por esses dados de log. Limpe todos os recursos associados às simulações, caso não precise mais deles.

Faça backups

É uma boa ideia ter backups e planos de backup para tudo. Você não deve presumir que, só porque seus dados estão lá, você não precisa fazer backup deles. AWS Você deve criar seu próprio sistema se precisar fazer backup do estado da simulação. Considere usar várias Regiões da AWS e ter um plano em vigor para poder transferir rapidamente sua carga de trabalho de produção para outra,

Região da AWS se necessário. Para obter mais informações sobre Regiões da AWS esse suporte SimSpace Weaver, consulte [SimSpace Terminais e cotas do Weaver](#).

Segurança em AWS SimSpace Weaver

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de data centers e arquiteturas de rede criados para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O [modelo de responsabilidade compartilhada](#) descreve isso como segurança da nuvem e segurança na nuvem:

- **Segurança da nuvem** — AWS é responsável por proteger a infraestrutura que executa AWS os serviços no Nuvem AWS. AWS também fornece serviços que você pode usar com segurança. Auditores terceirizados testam e verificam regularmente a eficácia de nossa segurança como parte dos Programas de Conformidade Programas de [AWS](#) de . Para saber mais sobre os programas de conformidade aplicáveis AWS SimSpace Weaver, consulte [AWS Serviços no escopo do programa de conformidade AWS](#) .
- **Segurança na nuvem** — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar SimSpace Weaver. Os tópicos a seguir mostram como configurar para atender SimSpace Weaver aos seus objetivos de segurança e conformidade. Você também aprenderá a usar outros AWS serviços que ajudam a monitorar e proteger seus SimSpace Weaver recursos.

Tópicos

- [Proteção de dados em AWS SimSpace Weaver](#)
- [Identity and Access Management para AWS SimSpace Weaver](#)
- [Registro e monitoramento de eventos de segurança em AWS SimSpace Weaver](#)
- [Validação de conformidade para AWS SimSpace Weaver](#)
- [Resiliência em AWS SimSpace Weaver](#)
- [Segurança de infraestrutura em AWS SimSpace Weaver](#)
- [Análise de configuração e vulnerabilidade em AWS SimSpace Weaver](#)
- [Práticas recomendadas de segurança para SimSpace Weaver](#)

Proteção de dados em AWS SimSpace Weaver

O modelo de [responsabilidade AWS compartilhada modelo](#) se aplica à proteção de dados em AWS SimSpace Weaver. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre seu conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para obter mais informações sobre privacidade de dados, consulte [Privacidade de dados FAQ](#). Para obter informações sobre proteção de dados na Europa, consulte o [Modelo de Responsabilidade AWS Compartilhada e GDPR](#) a postagem no blog AWS de segurança.

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use a autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com AWS os recursos. Exigimos TLS 1,2 e recomendamos TLS 1,3.
- Configure API e registre as atividades do usuário com AWS CloudTrail. Para obter informações sobre o uso de CloudTrail trilhas para capturar AWS atividades, consulte Como [trabalhar com CloudTrail trilhas](#) no Guia AWS CloudTrail do usuário.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de FIPS 140-3 módulos criptográficos validados ao acessar AWS por meio de uma interface de linha de comando ou uma API, use um endpoint. FIPS Para obter mais informações sobre os FIPS endpoints disponíveis, consulte [Federal Information Processing Standard \(FIPS\) 140-3](#).

É altamente recomendável que nunca sejam colocadas informações de identificação confidenciais, como endereços de e-mail dos seus clientes, em marcações ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com SimSpace Weaver ou Serviços da AWS usa o console, API, AWS CLI, ou AWS SDKs. Quaisquer dados inseridos em tags ou campos de texto de

formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, é altamente recomendável que você não inclua informações de credenciais no URL para validar sua solicitação para esse servidor.

Criptografia em repouso

Os dados são considerados em repouso quando estão localizados em um armazenamento de dados não volátil (persistente), como um disco. Os dados localizados em armazenamentos de dados voláteis, como memória e logs, não são considerados em repouso.

Quando você usa SimSpace Weaver, os únicos dados em repouso são:

- Aplicativos e esquemas que você carrega para o Amazon Simple Storage Service (Amazon S3)
- Dados de registro de simulação armazenados na Amazon CloudWatch

Outros dados SimSpace Weaver usados internamente não persistem depois que você interrompe a simulação.

Para saber como criptografar seus dados em repouso, consulte:

- [Criptografar dados no Amazon S3](#)
- [Criptografar dados de log](#)

Criptografia em trânsito

Suas conexões com o SimSpace Weaver API AWS Command Line Interface (AWS CLI) e o SimSpace Weaver aplicativo SDK usam TLS criptografia com o [processo de assinatura Signature Version 4](#). AWS SDK AWS gerencia a autenticação usando as políticas IAM de acesso definidas para as credenciais de segurança que você usa para se conectar.

Internamente, SimSpace Weaver usa TLS para se conectar a outros AWS serviços que ele usa.

Important

As comunicações entre seus aplicativos e seus clientes não envolvem SimSpace Weaver. É sua responsabilidade criptografar as comunicações com clientes de simulação, se

necessário. Recomendamos que você crie uma solução para criptografar todos os dados em trânsito nas conexões do cliente.

Para saber mais sobre AWS os serviços que podem oferecer suporte às suas soluções de criptografia, consulte [o Blog AWS de segurança](#).

Privacidade do tráfego entre redes

SimSpace Weaver os recursos computacionais residem em 1 Amazon VPC compartilhado por todos os SimSpace Weaver clientes. Todo o tráfego interno do SimSpace Weaver serviço permanece na AWS rede e não viaja pela Internet. A comunicação entre clientes de simulação e os aplicativos viaja pela internet.

Identity and Access Management para AWS SimSpace Weaver

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. IAMos administradores controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) a usar SimSpace Weaver os recursos. IAMé um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Tópicos

- [Público](#)
- [Autenticando com identidades](#)
- [Gerenciando acesso usando políticas](#)
- [Como AWS SimSpace Weaver funciona com IAM](#)
- [Exemplos de políticas baseadas em identidade para AWS SimSpace Weaver](#)
- [Permissões que SimSpace Weaver criam para você](#)
- [Prevenção do problema do “confused deputy” entre serviços](#)
- [Solução de problemas AWS SimSpace Weaver de identidade e acesso](#)

Público

A forma como você usa AWS Identity and Access Management (IAM) difere, dependendo do trabalho que você faz SimSpace Weaver.

Usuário do serviço — Se você usar o SimSpace Weaver serviço para fazer seu trabalho, seu administrador fornecerá as credenciais e as permissões de que você precisa. À medida que você usa mais SimSpace Weaver recursos para fazer seu trabalho, talvez precise de permissões adicionais. Entender como o acesso é gerenciado pode ajudá-lo a solicitar as permissões corretas ao seu administrador. Se não for possível acessar um atributo no SimSpace Weaver, consulte [Solução de problemas AWS SimSpace Weaver de identidade e acesso](#).

Administrador de serviços — Se você é responsável pelos SimSpace Weaver recursos da sua empresa, provavelmente tem acesso total SimSpace Weaver a. É seu trabalho determinar quais SimSpace Weaver recursos e recursos seus usuários do serviço devem acessar. Em seguida, você deve enviar solicitações ao IAM administrador para alterar as permissões dos usuários do serviço. Revise as informações nesta página para entender os conceitos básicos do IAM. Para saber mais sobre como sua empresa pode usar IAM com SimSpace Weaver, consulte [Como AWS SimSpace Weaver funciona com IAM](#).

IAM administrador — Se você for IAM administrador, talvez queira saber detalhes sobre como criar políticas para gerenciar o acesso SimSpace Weaver. Para ver exemplos de políticas SimSpace Weaver baseadas em identidade que você pode usar em IAM, consulte [Exemplos de políticas baseadas em identidade para AWS SimSpace Weaver](#)

Autenticando com identidades

A autenticação é a forma como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado (conectado AWS) como IAM usuário ou assumindo uma IAM função. Usuário raiz da conta da AWS

Você pode entrar AWS como uma identidade federada usando credenciais fornecidas por meio de uma fonte de identidade. AWS IAM Identity Center Os usuários (do IAM Identity Center), a autenticação de login único da sua empresa e suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você entra como uma identidade federada, seu administrador configurou previamente a federação de identidades usando IAM funções. Ao acessar AWS usando a federação, você está assumindo indiretamente uma função.

Dependendo do tipo de usuário que você é, você pode entrar no AWS Management Console ou no portal de AWS acesso. Para obter mais informações sobre como fazer login em AWS, consulte [Como fazer login Conta da AWS](#) no Guia do Início de Sessão da AWS usuário.

Se você acessar AWS programaticamente, AWS fornece um kit de desenvolvimento de software (SDK) e uma interface de linha de comando (CLI) para assinar criptograficamente suas solicitações usando suas credenciais. Se você não usa AWS ferramentas, você mesmo deve assinar as solicitações. Para obter mais informações sobre como usar o método recomendado para você mesmo assinar solicitações, consulte [AWS Signature versão 4 para API solicitações](#) no Guia IAM do usuário.

Independente do método de autenticação usado, também pode ser exigido que você forneça informações adicionais de segurança. Por exemplo, AWS recomenda que você use a autenticação multifator (MFA) para aumentar a segurança da sua conta. Para saber mais, consulte [Autenticação multifator](#) no Guia do AWS IAM Identity Center usuário e [Autenticação AWS multifator IAM no Guia do IAM usuário](#).

Conta da AWS usuário root

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos Serviços da AWS os recursos da conta. Essa identidade é chamada de usuário Conta da AWS raiz e é acessada fazendo login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável não usar o usuário raiz para tarefas diárias. Proteja as credenciais do usuário raiz e use-as para executar as tarefas que somente ele puder executar. Para ver a lista completa de tarefas que exigem que você faça login como usuário raiz, consulte [Tarefas que exigem credenciais de usuário raiz](#) no Guia do IAM usuário.

Identidade federada

Como prática recomendada, exija que usuários humanos, incluindo usuários que precisam de acesso de administrador, usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório de usuários corporativo, de um provedor de identidade da web AWS Directory Service, do diretório do Identity Center ou de qualquer usuário que acesse usando credenciais fornecidas Serviços da AWS por meio de uma fonte de identidade. Quando as identidades federadas são acessadas Contas da AWS, elas assumem funções, e as funções fornecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos usar o AWS IAM Identity Center. Você pode criar usuários e grupos no IAM Identity Center ou pode se conectar e sincronizar com um conjunto de usuários e grupos em sua própria fonte de identidade para uso em todos os seus Contas da AWS aplicativos. Para obter informações sobre o IAM Identity Center, consulte [O que é o IAM Identity Center?](#) no Guia do AWS IAM Identity Center usuário.

Grupos e usuários do IAM

Um [IAMusuário](#) é uma identidade dentro da sua Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, recomendamos confiar em credenciais temporárias em vez de criar IAM usuários que tenham credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com IAM os usuários, recomendamos que você alterne as chaves de acesso. Para obter mais informações, consulte [Altere as chaves de acesso regularmente para casos de uso que exigem credenciais de longo prazo](#) no Guia do IAMusuário.

Um [IAMgrupo](#) é uma identidade que especifica uma coleção de IAM usuários. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdminse conceder a esse grupo permissões para administrar IAM recursos.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas um perfil pode ser assumido por qualquer pessoa que precisar dele. Os usuários têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte [Casos de uso para IAM usuários](#) no Guia IAM do usuário.

IAMfunções

Uma [IAMfunção](#) é uma identidade dentro da sua Conta da AWS que tem permissões específicas. É semelhante a um IAM usuário, mas não está associado a uma pessoa específica. Para assumir temporariamente uma IAM função no AWS Management Console, você pode [alternar de um usuário para uma IAM função \(console\)](#). Você pode assumir uma função chamando uma AWS API operação AWS CLI or ou usando uma personalizadaURL. Para obter mais informações sobre métodos de uso de funções, consulte [Métodos para assumir uma função](#) no Guia IAM do usuário.

IAMfunções com credenciais temporárias são úteis nas seguintes situações:

- Acesso de usuário federado: para atribuir permissões a identidades federadas, você pode criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa

identidade é associada ao perfil e recebe as permissões definidas pelo mesmo. Para obter informações sobre funções para federação, consulte [Criar uma função para um provedor de identidade terceirizado \(federação\)](#) no Guia IAM do usuário. Se você usa o IAM Identity Center, configura um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o IAM Identity Center correlaciona o conjunto de permissões a uma função em. IAM Para obter informações sobre conjuntos de permissões, consulte [Conjuntos de Permissões](#) no Manual do Usuário do AWS IAM Identity Center .

- Permissões temporárias IAM de IAM usuário — Um usuário ou função pode assumir uma IAM função para assumir temporariamente permissões diferentes para uma tarefa específica.
- Acesso entre contas — Você pode usar uma IAM função para permitir que alguém (um diretor confiável) em uma conta diferente acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, com alguns Serviços da AWS, você pode anexar uma política diretamente a um recurso (em vez de usar uma função como proxy). Para saber a diferença entre funções e políticas baseadas em recursos para acesso entre contas, consulte Acesso a [recursos entre contas IAM no Guia](#) do IAM usuário.
- Acesso entre serviços — Alguns Serviços da AWS usam recursos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicativos na Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões do principal de chamada, usando um perfil de serviço ou um perfil vinculado a um serviço.
- Sessões de acesso direto (FAS) — Quando você usa um IAM usuário ou uma função para realizar ações em AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. FAS as solicitações são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer FAS solicitações, consulte [Encaminhar sessões de acesso](#).
- Função de serviço — Uma função de serviço é uma [IAM função](#) que um serviço assume para realizar ações em seu nome. Um IAM administrador pode criar, modificar e excluir uma função de serviço internamente IAM. Para obter mais informações, consulte [Criar uma função para delegar permissões a um AWS service \(Serviço da AWS\)](#) no Guia do IAM usuário.
- Função vinculada ao serviço — Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode assumir o perfil para executar

uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um IAM administrador pode visualizar, mas não editar, as permissões das funções vinculadas ao serviço.

- Aplicativos em execução na Amazon EC2 — Você pode usar uma IAM função para gerenciar credenciais temporárias para aplicativos que estão sendo executados em uma EC2 instância e fazendo AWS CLI AWS API solicitações. Isso é preferível a armazenar chaves de acesso na EC2 instância. Para atribuir uma AWS função a uma EC2 instância e disponibilizá-la para todos os aplicativos, você cria um perfil de instância anexado à instância. Um perfil de instância contém a função e permite que os programas em execução na EC2 instância recebam credenciais temporárias. Para obter mais informações, consulte [Usar uma IAM função para conceder permissões a aplicativos executados em EC2 instâncias da Amazon](#) no Guia IAM do usuário.

Gerenciando acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política é um objeto AWS que, quando associada a uma identidade ou recurso, define suas permissões. AWS avalia essas políticas quando um principal (usuário, usuário raiz ou sessão de função) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas é armazenada AWS como JSON documentos. Para obter mais informações sobre a estrutura e o conteúdo dos documentos de JSON política, consulte [Visão geral das JSON políticas](#) no Guia IAM do usuário.

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

Por padrão, usuários e perfis não têm permissões. Para conceder permissão aos usuários para realizar ações nos recursos de que precisam, um IAM administrador pode criar IAM políticas. O administrador pode então adicionar as IAM políticas às funções e os usuários podem assumir as funções.

IAMas políticas definem permissões para uma ação, independentemente do método usado para realizar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de função do AWS Management Console AWS CLI, do ou do AWS API.

Políticas baseadas em identidade

Políticas baseadas em identidade são documentos de políticas de JSON permissões que você pode anexar a uma identidade, como um IAM usuário, grupo de usuários ou função. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte [Definir IAM permissões personalizadas com políticas gerenciadas pelo cliente no Guia](#) do IAMusuário.

As políticas baseadas em identidade podem ser categorizadas ainda adicionalmente como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou perfil. As políticas gerenciadas são políticas autônomas que você pode associar a vários usuários, grupos e funções em seu Conta da AWS. As políticas AWS gerenciadas incluem políticas gerenciadas e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte [Escolher entre políticas gerenciadas e políticas em linha no Guia](#) do IAMusuário.

Políticas baseadas no recurso

Políticas baseadas em recursos são documentos JSON de política que você anexa a um recurso. Exemplos de políticas baseadas em recursos são as políticas de confiança de IAM funções e as políticas de bucket do Amazon S3. Em serviços que suportem políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações um principal especificado pode executar nesse recurso e em que condições. Você deve [especificar uma entidade principal](#) em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas de uma política baseada IAM em recursos.

Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento JSON de política.

Amazon S3, AWS WAF, e Amazon VPC são exemplos de serviços que oferecem suporte. ACLs Para saber mais ACLs, consulte a [visão geral da lista de controle de acesso \(ACL\)](#) no Guia do desenvolvedor do Amazon Simple Storage Service.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- **Limites de permissões** — Um limite de permissões é um recurso avançado no qual você define as permissões máximas que uma política baseada em identidade pode conceder a uma IAM entidade (IAM usuário ou função). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade com seus limites de permissões. As políticas baseadas em recurso que especificam o usuário ou o perfil no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte [Limites de permissões para IAM entidades](#) no Guia IAM do usuário.
- **Políticas de controle de serviço (SCPs)** — SCPs são JSON políticas que especificam as permissões máximas para uma organização ou unidade organizacional (OU) em AWS Organizations. AWS Organizations é um serviço para agrupar e gerenciar centralmente vários Contas da AWS que sua empresa possui. Se você habilitar todos os recursos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as suas contas. Os SCP limites de permissões para entidades nas contas dos membros, incluindo cada uma Usuário raiz da conta da AWS. Para obter mais informações sobre Organizations e SCPs, consulte [Políticas de controle de serviços](#) no Guia AWS Organizations do Usuário.
- **Políticas de controle de recursos (RCPs)** — RCPs são JSON políticas que você pode usar para definir o máximo de permissões disponíveis para recursos em suas contas sem atualizar as IAM políticas anexadas a cada recurso que você possui. Isso RCP limita as permissões de recursos nas contas dos membros e pode afetar as permissões efetivas das identidades, incluindo a Usuário raiz da conta da AWS, independentemente de pertencerem à sua organização. Para obter mais informações sobre Organizations e RCPs, incluindo uma lista Serviços da AWS desse suporte RCPs, consulte [Políticas de controle de recursos \(RCPs\)](#) no Guia AWS Organizations do usuário.
- **Políticas de sessão:** são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para um perfil ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do usuário ou do perfil e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em atributo. Uma negação explícita em qualquer uma dessas políticas

substitui a permissão. Para obter mais informações, consulte [Políticas de sessão](#) no Guia IAM do usuário.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de política estão envolvidos, consulte [Lógica de avaliação](#) de políticas no Guia IAM do usuário.

Como AWS SimSpace Weaver funciona com IAM

Antes de usar IAM para gerenciar o acesso ao SimSpace Weaver, saiba quais IAM recursos estão disponíveis para uso SimSpace Weaver.

IAMrecursos que você pode usar com AWS SimSpace Weaver

IAMrecurso	SimSpace Weaver apoio
Políticas baseadas em identidade	Sim
Políticas baseadas em recursos	Não
Ações das políticas	Sim
Atributos de políticas	Sim
Chaves de condição de política (específicas do serviço)	Sim
ACLs	Não
ABAC(tags nas políticas)	Sim
Credenciais temporárias	Sim
Permissões de entidade principal	Sim
Perfis de serviço	Sim

IAMrecurso	SimSpace Weaver apoio
Perfis vinculados ao serviço	Não

Para obter uma visão geral de como SimSpace Weaver e outros AWS serviços funcionam com a maioria dos IAM recursos, consulte [AWS os serviços que funcionam com IAM](#) no Guia do IAM usuário.

Políticas baseadas em identidade para SimSpace Weaver

Compatível com políticas baseadas em identidade: Sim

Políticas baseadas em identidade são documentos de políticas de JSON permissões que você pode anexar a uma identidade, como um IAM usuário, grupo de usuários ou função. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte [Definir IAM permissões personalizadas com políticas gerenciadas pelo cliente no Guia](#) do IAM usuário.

Com políticas IAM baseadas em identidade, você pode especificar ações e recursos permitidos ou negados, bem como as condições sob as quais as ações são permitidas ou negadas. Você não pode especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou perfil ao qual ela está anexada. Para saber mais sobre todos os elementos que você pode usar em uma JSON política, consulte a [referência IAM JSON de elementos de política](#) no Guia IAM do usuário.

Exemplos de políticas baseadas em identidade para SimSpace Weaver

Para ver exemplos de políticas SimSpace Weaver baseadas em identidade, consulte. [Exemplos de políticas baseadas em identidade para AWS SimSpace Weaver](#)

Políticas baseadas em recursos dentro SimSpace Weaver

Suporte a políticas baseadas em recursos: não

Políticas baseadas em recursos são documentos JSON de política que você anexa a um recurso. Exemplos de políticas baseadas em recursos são as políticas de confiança de IAM funções e as políticas de bucket do Amazon S3. Em serviços que suportem políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações um principal especificado

pode executar nesse recurso e em que condições. Você deve [especificar uma entidade principal](#) em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Para habilitar o acesso entre contas, você pode especificar uma conta ou IAM entidades inteiras em outra conta como principal em uma política baseada em recursos. Adicionar uma entidade principal entre contas à política baseada em recurso é apenas metade da tarefa de estabelecimento da relação de confiança. Quando o principal e o recurso são diferentes Contas da AWS, um IAM administrador na conta confiável também deve conceder permissão à entidade principal (usuário ou função) para acessar o recurso. Eles concedem permissão ao anexar uma política baseada em identidade para a entidade. No entanto, se uma política baseada em recurso conceder acesso a uma entidade principal na mesma conta, nenhuma política baseada em identidade adicional será necessária. Para obter mais informações, [consulte Acesso a recursos entre contas IAM no](#) Guia do IAM usuário.

Ações políticas para SimSpace Weaver

Compatível com ações de políticas: Sim

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O `Action` elemento de uma JSON política descreve as ações que você pode usar para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome da AWS API operação associada. Há algumas exceções, como ações somente de permissão que não têm uma operação correspondente. API Algumas operações também exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista de SimSpace Weaver ações, consulte [Ações definidas por AWS SimSpace Weaver](#) na Referência de Autorização de Serviço.

As ações de política SimSpace Weaver usam o seguinte prefixo antes da ação:

```
simspaceweaver
```

Para especificar várias ações em uma única instrução, separe-as com vírgulas.

```
"Action": [
```

```
"simspaceweaver:action1",  
"simspaceweaver:action2"  
]
```

Para ver exemplos de políticas SimSpace Weaver baseadas em identidade, consulte [Exemplos de políticas baseadas em identidade para AWS SimSpace Weaver](#)

Recursos políticos para SimSpace Weaver

Compatível com recursos de políticas: Sim

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento Resource JSON de política especifica o objeto ou objetos aos quais a ação se aplica. As instruções devem incluir um elemento Resource ou NotResource. Como prática recomendada, especifique um recurso usando seu [Amazon Resource Name \(ARN\)](#). Isso pode ser feito para ações que oferecem compatibilidade com um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem compatibilidade com permissões em nível de recurso, como operações de listagem, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

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

Para ver uma lista dos tipos de SimSpace Weaver recursos e seus ARNs, consulte [Recursos definidos por AWS SimSpace Weaver](#) na Referência de Autorização de Serviço. Para saber com quais ações você pode especificar cada recurso, consulte [Ações definidas por AWS SimSpace Weaver](#). ARN

Para ver exemplos de políticas SimSpace Weaver baseadas em identidade, consulte [Exemplos de políticas baseadas em identidade para AWS SimSpace Weaver](#)

Chaves de condição de política para SimSpace Weaver

Compatível com chaves de condição de política específicas de serviço: Sim

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento Condition (ou bloco Condition) permite que você especifique condições nas quais uma instrução estiver em vigor. O elemento Condition é opcional. É possível criar expressões

condicionais que usem [agentes de condição](#), como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos `Condition` em uma instrução ou várias chaves em um único `Condition` elemento, a AWS os avaliará usando uma operação lógica AND. Se você especificar vários valores para uma única chave de condição, AWS avalia a condição usando uma OR operação lógica. Todas as condições devem ser atendidas antes que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar condições. Por exemplo, você pode conceder permissão a um IAM usuário para acessar um recurso somente se ele estiver marcado com o nome de IAM usuário. Para obter mais informações, consulte [elementos de IAM política: variáveis e tags](#) no Guia IAM do usuário.

AWS suporta chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as chaves de condição AWS globais, consulte as [chaves de contexto de condição AWS global](#) no Guia IAM do usuário.

Para ver uma lista de chaves de SimSpace Weaver condição, consulte [Chaves de condição AWS SimSpace Weaver](#) na Referência de autorização de serviço. Para saber com quais ações e recursos você pode usar uma chave de condição, consulte [Ações definidas por AWS SimSpace Weaver](#).

Para ver exemplos de políticas SimSpace Weaver baseadas em identidade, consulte. [Exemplos de políticas baseadas em identidade para AWS SimSpace Weaver](#)

Listas de controle de acesso (ACLs) em SimSpace Weaver

SuportesACLs: Não

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento JSON de política.

Controle de acesso baseado em atributos () ABAC com SimSpace Weaver

Suportes ABAC (tags nas políticas): Sim

O controle de acesso baseado em atributos (ABAC) é uma estratégia de autorização que define permissões com base em atributos. Em AWS, esses atributos são chamados de tags. Você pode anexar tags a IAM entidades (usuários ou funções) e a muitos AWS recursos. Marcar entidades e

recursos é a primeira etapa do ABAC. Em seguida, você cria ABAC políticas para permitir operações quando a tag do diretor corresponde à tag do recurso que ele está tentando acessar.

ABAC é útil em ambientes que estão crescendo rapidamente e ajuda em situações em que o gerenciamento de políticas se torna complicado.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no [elemento de condição](#) de uma política usando as `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou chaves de condição `aws:TagKeys`.

Se um serviço oferecer suporte às três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço oferecer suporte às três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para obter mais informações sobre ABAC, consulte [Definir permissões com ABAC autorização](#) no Guia IAM do usuário. Para ver um tutorial com etapas de configuração ABAC, consulte [Usar controle de acesso baseado em atributos \(ABAC\) no Guia](#) do IAM usuário.

Usando credenciais temporárias com SimSpace Weaver

Compatível com credenciais temporárias: Sim

Alguns Serviços da AWS não funcionam quando você faz login usando credenciais temporárias. Para obter informações adicionais, incluindo quais Serviços da AWS funcionam com credenciais temporárias, consulte [Serviços da AWS esse trabalho IAM](#) no Guia do IAM usuário.

Você está usando credenciais temporárias se fizer login AWS Management Console usando qualquer método, exceto um nome de usuário e senha. Por exemplo, quando você acessa AWS usando o link de login único (SSO) da sua empresa, esse processo cria automaticamente credenciais temporárias. Você também cria automaticamente credenciais temporárias quando faz login no console como usuário e, em seguida, alterna perfis. Para obter mais informações sobre a troca de funções, consulte [Alternar de um usuário para uma IAM função \(console\)](#) no Guia IAM do usuário.

Você pode criar manualmente credenciais temporárias usando o AWS CLI ou AWS API. Em seguida, você pode usar essas credenciais temporárias para acessar AWS. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para obter mais informações, consulte [Credenciais de segurança temporárias em IAM](#).

Permissões principais entre serviços para SimSpace Weaver

Suporta sessões de acesso direto (FAS): Sim

Quando você usa um IAM usuário ou uma função para realizar ações em AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. FASusa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. FASas solicitações são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer FAS solicitações, consulte [Encaminhar sessões de acesso](#).

Funções de serviço para SimSpace Weaver

Compatível com perfis de serviço: Sim

Uma função de serviço é uma [IAMfunção](#) que um serviço assume para realizar ações em seu nome. Um IAM administrador pode criar, modificar e excluir uma função de serviço internamenteIAM. Para obter mais informações, consulte [Criar uma função para delegar permissões a um AWS service \(Serviço da AWS\)](#) no Guia do IAM usuário.

Warning

Alterar as permissões de uma função de serviço pode interromper SimSpace Weaver a funcionalidade. Edite as funções de serviço somente quando SimSpace Weaver fornecer orientação para fazer isso.

Os SDK scripts do SimSpace Weaver aplicativo usam um AWS CloudFormation modelo para criar recursos em outros AWS serviços para apoiar sua simulação. Um desses recursos é a função do aplicativo para sua simulação. SimSpace Weaver assume a função do aplicativo para realizar ações Conta da AWS em seu nome, como gravar dados de registro no CloudWatch Logs. Para mais informações sobre a função do aplicativo, consulte [Permissões que SimSpace Weaver cria para você](#).

Funções vinculadas a serviços para SimSpace Weaver

Compatível com perfis vinculados ao serviço: Não

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode assumir o perfil para executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um IAM administrador pode visualizar, mas não editar, as permissões das funções vinculadas ao serviço.

Para obter detalhes sobre como criar ou gerenciar funções vinculadas a serviços, consulte [AWS serviços que funcionam](#) com IAM. Encontre um serviço na tabela que inclua um Yes na coluna Função vinculada ao serviço. Escolha o link Sim para visualizar a documentação do perfil vinculado ao serviço desse serviço.

Exemplos de políticas baseadas em identidade para AWS SimSpace Weaver

Por padrão, usuários e perfis não têm permissão para criar ou modificar recursos do SimSpace Weaver. Eles também não podem realizar tarefas usando o AWS Management Console, AWS Command Line Interface (AWS CLI) ou AWS API. Para conceder permissão aos usuários para realizar ações nos recursos de que precisam, um IAM administrador pode criar IAM políticas. O administrador pode então adicionar as IAM políticas às funções e os usuários podem assumir as funções.

Para saber como criar uma política IAM baseada em identidade usando esses exemplos de documentos de JSON política, consulte [Criar IAM políticas \(console\) no Guia](#) do IAM usuário.

Para obter detalhes sobre ações e tipos de recursos definidos por SimSpace Weaver, incluindo o formato do ARNs para cada um dos tipos de recursos, consulte [Ações, recursos e chaves de condição AWS SimSpace Weaver na Referência de Autorização de Serviço](#).

Tópicos

- [Melhores práticas de política](#)
- [Usar o console do SimSpace Weaver](#)
- [Permitir que usuários visualizem suas próprias permissões](#)
- [Permitir que os usuários criem e executem simulações](#)

Melhores práticas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir SimSpace Weaver recursos em sua conta. Essas ações podem incorrer em custos para sua Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão

disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para obter mais informações, consulte [políticas AWS gerenciadas](#) ou [políticas AWS gerenciadas para funções de trabalho](#) no Guia IAM do usuário.

- Aplique permissões com privilégios mínimos — Ao definir permissões com IAM políticas, conceda somente as permissões necessárias para realizar uma tarefa. Você faz isso definindo as ações que podem ser executadas em atributos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre IAM como usar para aplicar permissões, consulte [Políticas e permissões IAM no](#) Guia IAM do usuário.
- Use condições nas IAM políticas para restringir ainda mais o acesso — Você pode adicionar uma condição às suas políticas para limitar o acesso a ações e recursos. Por exemplo, você pode escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como AWS CloudFormation. Para obter mais informações, consulte [Elementos IAM JSON da política: Condição](#) no Guia IAM do usuário.
- Use o IAM Access Analyzer para validar suas IAM políticas e garantir permissões seguras e funcionais — o IAM Access Analyzer valida políticas novas e existentes para que as políticas sigam a linguagem da IAM política (JSON) e as melhores práticas. IAM IAMO Access Analyzer fornece mais de 100 verificações de políticas e recomendações práticas para ajudá-lo a criar políticas seguras e funcionais. Para obter mais informações, consulte [Validar políticas com o IAM Access Analyzer](#) no Guia do IAM Usuário.
- Exigir autenticação multifatorial (MFA) — Se você tiver um cenário que exija IAM usuários ou um usuário root Conta da AWS, ative MFA para obter segurança adicional. Para exigir MFA quando API as operações são chamadas, adicione MFA condições às suas políticas. Para obter mais informações, consulte [API Acesso seguro com MFA](#) no Guia IAM do usuário.

Para obter mais informações sobre as melhores práticas em IAM, consulte [as melhores práticas de segurança IAM no](#) Guia IAM do usuário.

Usar o console do SimSpace Weaver

Para acessar o AWS SimSpace Weaver console, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os SimSpace Weaver recursos em seu Conta da AWS. Caso crie uma política baseada em identidade mais restritiva que

as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis) com essa política.

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para AWS CLI o. ou AWS API o. Em vez disso, permita o acesso somente às ações que correspondam à API operação que eles estão tentando realizar.

Para garantir que usuários e funções ainda possam usar o SimSpace Weaver console, anexe também a política SimSpace Weaver *ConsoleAccess* ou a política *ReadOnly* AWS gerenciada às entidades. Para obter mais informações, consulte [Adicionar permissões a um usuário](#) no Guia do IAM usuário.

Permitir que usuários visualizem suas próprias permissões

Este exemplo mostra como você pode criar uma política que permita IAM aos usuários visualizar as políticas embutidas e gerenciadas que estão anexadas à identidade do usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando o AWS CLI ou. AWS API

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

```

```

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

```

Permitir que os usuários criem e executem simulações

Este exemplo IAM de política fornece as permissões básicas necessárias para criar e executar simulações em 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"
      }
    }
  }
]
}

```

Permissões que SimSpace Weaver criam para você

Quando você cria um SimSpace Weaver projeto, o serviço cria uma função AWS Identity and Access Management (IAM) com o nome `weaver-project-name-app-role` e uma política de IAM confiança. A política de confiança SimSpace Weaver permite assumir a função para que ela possa realizar operações para você.

Política de permissões da função do aplicativo

A função do aplicativo de simulação utiliza a seguinte política de permissões.

```

{
  "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 confiança da função do aplicativo

SimSpace Weaver adiciona uma relação de confiança à função do aplicativo de simulação como uma [política de confiança](#). SimSpace Weaver cria uma política de confiança para cada simulação, semelhante ao exemplo a seguir.

```

{
  "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

Neste exemplo, 111122223333 é o número da conta e MySimName é o nome. Esses valores são diferentes em suas políticas de confiança.

Prevenção do problema do “confused deputy” entre serviços

O [problema confused deputy](#) é um problema de segurança em que uma entidade que não tem permissão para executar uma ação pode coagir uma entidade mais privilegiada a executá-la. Em AWS, a falsificação de identidade entre serviços pode resultar em um problema confuso de delegado. A personificação entre serviços pode ocorrer quando um serviço (o serviço de chamada) chama outro serviço (o serviço chamado). O serviço de chamada pode ser manipulado de modo a usar suas permissões para atuar nos recursos de outro cliente de uma forma na qual ele não deveria ter permissão para acessar. Para evitar isso, a AWS fornece ferramentas que ajudam você a proteger seus dados para todos os serviços com entidades principais de serviço que receberam acesso aos recursos em sua conta.

Recomendamos usar as chaves de contexto de condição [aws:SourceAccount](#) global [aws:SourceArn](#) as chaves de contexto nas políticas de recursos para limitar as permissões que AWS SimSpace Weaver concedem outro serviço ao recurso. Se o `aws:SourceArn` valor não contiver o ID da conta, como um bucket do Amazon S3 Amazon Resource Name (ARN), você deverá usar as duas chaves de contexto de condição global para limitar as permissões. Se você utilizar ambas as chaves de contexto de condição global, e o valor `aws:SourceArn` contiver o ID da conta, o valor `aws:SourceAccount` e a conta no valor `aws:SourceArn` deverão utilizar o mesmo ID de conta quando utilizados na mesma declaração da política. Use `aws:SourceArn` se quiser apenas um recurso associado a acessibilidade de serviço. Use `aws:SourceAccount` se quiser permitir que qualquer recurso nessa conta seja associado ao uso entre serviços.

O valor de `aws:SourceArn` deve usar a extensão ARN.

A maneira mais eficaz de se proteger contra o confuso problema do deputado é usar a chave de contexto ARN de condição `aws:SourceArn` global com todo o recurso. Se você não souber a extensão completa ARN ou se estiver especificando várias extensões, use a chave de condição de contexto `aws:SourceArn` global com curingas (*) para as partes desconhecidas do ARN. Por exemplo, `arn:aws:simspaceweaver:*:111122223333:*`.

O exemplo a seguir mostra como você pode usar as chaves de contexto de condição `aws:SourceAccount` global `aws:SourceArn` e as chaves de contexto SimSpace Weaver para evitar o confuso problema substituto. Essa política só permitirá assumir SimSpace Weaver a função quando a solicitação vier da conta de origem especificada e for fornecida com o especificadoARN. Nesse caso, só SimSpace Weaver pode assumir a função para solicitações de simulações na própria conta do solicitante (`111122223333`) e somente na região 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/*"
        }
      }
    }
  ]
}
```

Uma maneira mais segura de escrever essa política é incluir o nome da simulação no `aws:SourceArn`, conforme mostrado no exemplo a seguir, o que restringe a política a uma simulação chamada `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"
        }
      }
    }
  ]
}
```

```

    ]
  },
  "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"
    }
  }
}
]
}

```

Quando você inclui `aws:SourceArn` explicitamente um número de conta, você pode omitir o teste de `Condition` elemento para o `aws:SourceAccount` (consulte o [Guia IAM do usuário](#) para obter mais informações), como na seguinte 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"
        }
      }
    }
  ]
}

```


Solução de problemas AWS SimSpace Weaver de identidade e acesso

Use as informações a seguir para ajudá-lo a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com SimSpace Weaver e IAM.

Tópicos

- [Não estou autorizado a realizar uma ação em SimSpace Weaver](#)
- [Não estou autorizado a realizar iam: PassRole](#)
- [Quero visualizar minhas chaves de acesso](#)
- [Sou administrador e quero permitir que outras pessoas acessem SimSpace Weaver](#)
- [Quero permitir que pessoas fora da minha Conta da AWS acessem meus SimSpace Weaver recursos](#)

Não estou autorizado a realizar uma ação em SimSpace Weaver

Se isso AWS Management Console indicar que você não está autorizado a realizar uma ação, entre em contato com o administrador para obter ajuda. O administrador é a pessoa que forneceu o seu nome de usuário e senha.

O exemplo de erro a seguir ocorre quando o mateojackson IAM usuário tenta usar o console para ver detalhes sobre um *my-example-widget* recurso fictício, mas não tem as permissões fictíciassimspaceweaver: *GetWidget*.

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

Neste caso, Mateo pede ao administrador para atualizar suas políticas para permitir a ele o acesso ao recurso *my-example-widget* usando a ação *simspaceweaver: GetWidget*.

Não estou autorizado a realizar iam: PassRole

Se você receber uma mensagem de erro informando que não está autorizado a executar a ação *iam: PassRole*, as suas políticas devem ser atualizadas para permitir que você passe uma função para o SimSpace Weaver.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando um IAM usuário chamado `marymajor` tenta usar o console para realizar uma ação no SimSpace Weaver. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

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

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero visualizar minhas chaves de acesso

Depois de criar suas chaves de acesso de IAM usuário, você pode ver sua ID de chave de acesso a qualquer momento. No entanto, você não pode visualizar sua chave de acesso secreta novamente. Se você perder sua chave secreta, crie um novo par de chaves de acesso.

As chaves de acesso consistem em duas partes: um ID de chave de acesso (por exemplo, `AKIAIOSFODNN7EXAMPLE`) e uma chave de acesso secreta (por exemplo, `wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY`). Como um nome de usuário e uma senha, você deve usar o ID da chave de acesso e a chave de acesso secreta em conjunto para autenticar suas solicitações. Gerencie suas chaves de acesso de forma tão segura quanto você gerencia seu nome de usuário e sua senha.

Important

Não forneça as chaves de acesso a terceiros, mesmo que seja para ajudar a [encontrar o ID de usuário canônico](#). Ao fazer isso, você pode dar a alguém acesso permanente ao seu Conta da AWS.

Ao criar um par de chaves de acesso, você é solicitado a guardar o ID da chave de acesso e a chave de acesso secreta em um local seguro. A chave de acesso secreta só está disponível no momento em que é criada. Se você perder sua chave de acesso secreta, deverá adicionar novas chaves de acesso ao seu IAM usuário. Você pode ter no máximo duas chaves de acesso. Se você já tiver duas,

você deverá excluir um par de chaves para poder criar um novo. Para ver as instruções, consulte [Gerenciamento de chaves de acesso](#) no Guia IAM do usuário.

Sou administrador e quero permitir que outras pessoas acessem SimSpace Weaver

Para permitir que outras pessoas acessem SimSpace Weaver, você deve conceder permissão às pessoas ou aplicativos que precisam de acesso. Se você estiver usando o AWS IAM Identity Center para gerenciar pessoas e aplicações, você atribui conjuntos de permissões a usuários ou grupos para definir seu nível de acesso. Os conjuntos de permissões criam e atribuem IAM políticas automaticamente às IAM funções associadas à pessoa ou ao aplicativo. Para obter mais informações, consulte [Conjuntos de permissões](#) no Guia do usuário do AWS IAM Identity Center .

Se você não estiver usando o IAM Identity Center, deverá criar IAM entidades (usuários ou funções) para as pessoas ou aplicativos que precisam de acesso. Você deve anexar uma política à entidade que concede a eles as permissões corretas no SimSpace Weaver. Depois que as permissões forem concedidas, forneça as credenciais ao usuário ou desenvolvedor da aplicação. Eles usarão essas credenciais para acessar a AWS. Para saber mais sobre a criação de IAM usuários, grupos, políticas e permissões, consulte [IAMIdenticidades, políticas e permissões IAM no](#) Guia do IAM usuário.

Quero permitir que pessoas fora da minha Conta da AWS acessem meus SimSpace Weaver recursos

Você pode criar um perfil que os usuários de outras contas ou pessoas fora da sua organização podem usar para acessar seus recursos. Você pode especificar quem é confiável para assumir o perfil. Para serviços que oferecem suporte a políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se é SimSpace Weaver compatível com esses recursos, consulte [Como AWS SimSpace Weaver funciona com IAM](#).
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte [Fornecer acesso a um IAM usuário em outro Conta da AWS de sua propriedade](#) no Guia do IAM usuário.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte [Fornecer Contas da AWS acesso a terceiros](#) no Guia do IAM usuário.
- Para saber como fornecer acesso por meio da federação de identidades, consulte [Fornecendo acesso a usuários autenticados externamente \(federação de identidades\)](#) no Guia do IAM usuário.

- Para saber a diferença entre usar funções e políticas baseadas em recursos para acesso entre contas, consulte Acesso a [recursos entre contas IAM no Guia](#) do IAM usuário.

Registro e monitoramento de eventos de segurança em AWS SimSpace Weaver

O monitoramento é uma parte importante da manutenção da confiabilidade, disponibilidade e desempenho de SimSpace Weaver suas AWS soluções. Você deve coletar dados de monitoramento de todas as partes da sua AWS solução para poder depurar com mais facilidade uma falha multiponto, caso ocorra.

AWS e SimSpace Weaver forneça várias ferramentas para monitorar seus recursos de simulação e responder a possíveis incidentes.

Logs na Amazon CloudWatch

SimSpace Weaver armazena seus logs CloudWatch. Você pode usar esses logs para monitorar eventos na simulação (como iniciar e interromper aplicativos), bem como para depuração. Para obter mais informações, consulte [SimSpace Weaver loga no Amazon CloudWatch Logs](#).

CloudWatch Alarmes da Amazon

Usando CloudWatch os alarmes da Amazon, você observa uma única métrica durante um período de tempo especificado por você. Se a métrica exceder um determinado limite, uma notificação será enviada para um SNS tópico da Amazon ou para uma política de AWS Auto Scaling. CloudWatch os alarmes são acionados quando seu estado muda e são mantidos por um determinado número de períodos, não por estarem em um estado específico. Para obter mais informações, consulte [Monitoramento SimSpace Weaver com a Amazon CloudWatch](#).

AWS CloudTrail Registros

CloudTrail fornece um registro das ações realizadas por um usuário, função ou AWS serviço em SimSpace Weaver. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita SimSpace Weaver, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais. Para obter mais informações, consulte [Registrando AWS SimSpace Weaver API chamadas usando AWS CloudTrail](#).

Validação de conformidade para AWS SimSpace Weaver

SimSpace Weaver não está no escopo de nenhum programa de AWS conformidade.

Audidores terceirizados avaliam a segurança e a conformidade de outros AWS serviços como parte de vários programas de AWS conformidade. Isso inclui SOC, PCI RAMPHIPAA, Fed e outros.

Para saber se um AWS service (Serviço da AWS) está dentro do escopo de programas de conformidade específicos, consulte [Serviços da AWS Escopo por Programa de Conformidade](#) [Serviços da AWS](#) e escolha o programa de conformidade em que você está interessado. Para obter informações gerais, consulte Programas de [AWS conformidade Programas AWS](#) de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte [Baixar relatórios em AWS Artifact](#) .

Sua responsabilidade de conformidade ao usar Serviços da AWS é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentações aplicáveis. AWS fornece os seguintes recursos para ajudar na conformidade:

- [Conformidade e governança de segurança](#) — Esses guias de implementação de soluções discutem considerações arquitetônicas e fornecem etapas para a implantação de recursos de segurança e conformidade.
- [Arquitetura para HIPAA segurança e conformidade na Amazon Web Services](#) — Este whitepaper descreve como as empresas podem usar AWS para criar HIPAA aplicativos qualificados.

Note

Nem todos Serviços da AWS são HIPAA elegíveis. Para obter mais informações, consulte a [Referência de serviços HIPAA elegíveis](#).

- AWS Recursos de <https://aws.amazon.com/compliance/resources/> de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- [AWS Guias de conformidade do cliente](#) — Entenda o modelo de responsabilidade compartilhada sob a ótica da conformidade. Os guias resumem as melhores práticas de proteção Serviços da AWS e mapeiam as diretrizes para controles de segurança em várias estruturas (incluindo o Instituto Nacional de Padrões e Tecnologia (NIST), o Conselho de Padrões de Segurança do Setor de Cartões de Pagamento (PCI) e a Organização Internacional de Padronização ()). ISO

- [Avaliação de recursos com regras](#) no Guia do AWS Config desenvolvedor — O AWS Config serviço avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- [AWS Security Hub](#)— Isso AWS service (Serviço da AWS) fornece uma visão abrangente do seu estado de segurança interno AWS. O Security Hub usa controles de segurança para avaliar os recursos da AWS e verificar a conformidade com os padrões e as práticas recomendadas do setor de segurança. Para obter uma lista dos serviços e controles aceitos, consulte a [Referência de controles do Security Hub](#).
- [Amazon GuardDuty](#) — Isso AWS service (Serviço da AWS) detecta possíveis ameaças às suas cargas de trabalho Contas da AWS, contêineres e dados monitorando seu ambiente em busca de atividades suspeitas e maliciosas. GuardDuty pode ajudá-lo a atender a vários requisitos de conformidade, por exemplo PCIDSS, atendendo aos requisitos de detecção de intrusões exigidos por determinadas estruturas de conformidade.
- [AWS Audit Manager](#)— Isso AWS service (Serviço da AWS) ajuda você a auditar continuamente seu AWS uso para simplificar a forma como você gerencia o risco e a conformidade com as regulamentações e os padrões do setor.

Resiliência em AWS SimSpace Weaver

A infraestrutura AWS global é construída em torno Regiões da AWS de zonas de disponibilidade. Regiões da AWS fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e alta redundância. Com as zonas de disponibilidade, é possível projetar e operar aplicativos e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre zonas de disponibilidade Regiões da AWS e zonas de disponibilidade, consulte [Infraestrutura AWS global](#).

Segurança de infraestrutura em AWS SimSpace Weaver

Como serviço gerenciado, AWS SimSpace Weaver é protegido pela segurança de rede AWS global. Para obter informações sobre serviços AWS de segurança e como AWS proteger a infraestrutura, consulte [AWS Cloud Security](#). Para projetar seu AWS ambiente usando as melhores práticas

de segurança de infraestrutura, consulte [Proteção](#) de infraestrutura no Security Pillar AWS Well-Architected Framework.

Você usa API chamadas AWS publicadas para acessar SimSpace Weaver pela rede. Os clientes devem oferecer suporte para:

- Segurança da camada de transporte (TLS). Exigimos TLS 1,2 e recomendamos TLS 1,3.
- Suítes de criptografia com sigilo direto perfeito (), como (Ephemeral PFS Diffie-Hellman) ou DHE (Elliptic Curve Ephemeral Diffie-Hellman). ECDHE A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas usando uma ID de chave de acesso e uma chave de acesso secreta associada a um IAM principal. Ou é possível usar o [AWS Security Token Service](#) (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Modelo de segurança de conectividade de rede

Suas simulações são executadas em instâncias computacionais dentro de uma Amazon VPC localizada em uma AWS região que você seleciona. Uma Amazon VPC é uma rede virtual na AWS nuvem, que isola a infraestrutura por carga de trabalho ou entidade organizacional. As comunicações entre instâncias computacionais na Amazon VPC permanecem dentro da AWS rede e não trafegam pela Internet. Algumas comunicações internas de serviços cruzam a Internet e são criptografadas. Simulações para todos os clientes que operam na mesma AWS região compartilham a mesma AmazonVPC. Simulações para clientes diferentes usam instâncias de computação separadas dentro da mesma Amazon. VPC

Comunicações entre seus clientes de simulação e suas simulações executadas em SimSpace Weaver viagens pela Internet. SimSpace Weaver não lida com essas conexões. É sua responsabilidade proteger as conexões de seus clientes.

Suas conexões com o SimSpace Weaver serviço atravessam a Internet e são criptografadas. Isso inclui conexões usando o AWS Management Console, AWS Command Line Interface (AWS CLI), kits AWS de desenvolvimento de software (SDK) e o SimSpace Weaver aplicativoSDK.

Análise de configuração e vulnerabilidade em AWS SimSpace Weaver

A configuração e os controles de TI são uma responsabilidade compartilhada entre você AWS e você. Para obter mais informações, consulte o modelo de [responsabilidade AWS compartilhada modelo](#). AWS lida com tarefas básicas de segurança para a infraestrutura subjacente, como corrigir o sistema operacional em instâncias de computação, configuração de firewall e recuperação de desastres da AWS infraestrutura. Esses procedimentos foram revisados e certificados por terceiros certificados. Para obter mais detalhes, consulte [Práticas recomendadas de segurança, identidade e conformidade](#).

Você é responsável pela segurança do software de simulação:

- Mantenha o código do aplicativo, incluindo atualizações e patches de segurança.
- Autentique e criptografe a comunicação entre seus clientes de simulação e os aplicativos aos quais eles se conectam.
- Atualize suas simulações para usar as SDK versões mais recentes, incluindo o SimSpace Weaver aplicativo AWS SDK SDK e.

Note

SimSpace Weaver não oferece suporte a atualizações de aplicativos em uma simulação em execução. Se precisar atualizar os aplicativos, interrompa e exclua a simulação e, em seguida, crie uma nova simulação com o código atualizado do aplicativo. Recomendamos que você salve o estado da simulação em um armazenamento de dados externo para poder restaurá-lo se precisar recriar a simulação.

Práticas recomendadas de segurança para SimSpace Weaver

Esta seção descreve as melhores práticas de segurança que são específicas do SimSpace Weaver. Para saber mais sobre as melhores práticas de segurança em AWS, consulte [Melhores práticas de segurança, identidade e conformidade](#).

Tópicos

- [Criptografe as comunicações entre os aplicativos e seus clientes](#)

- [Faça backup periódico do estado da simulação](#)
- [Faça a manutenção de seus aplicativos e SDKs](#)

Criptografe as comunicações entre os aplicativos e seus clientes

SimSpace Weaver não gerencia as comunicações entre seus aplicativos e seus clientes. Você deve implementar alguma forma de autenticação e criptografia para as sessões do cliente.


Faça backup periódico do estado da simulação

SimSpace Weaver não salva seu estado de simulação. As simulações que são interrompidas (como resultado de uma API chamada, opção de console ou falha do sistema) não salvam seu estado e não têm uma maneira inerente de recuperá-las. As simulações interrompidas não podem ser reiniciadas. A única maneira de realizar o equivalente a uma reinicialização é recriar sua simulação usando a mesma configuração e os mesmos dados. Você pode usar backups do estado de simulação para inicializar a nova simulação. A AWS oferece serviços de [banco de dados](#) e [armazenamento em nuvem](#) altamente confiáveis e disponíveis que você pode usar para salvar seu estado de simulação.

Faça a manutenção de seus aplicativos e SDKs

Mantenha seus aplicativos, instalações locais dos kits de desenvolvimento de AWS software (SDKs) e o SimSpace Weaver aplicativoSDK. Você pode baixar e instalar novas versões do AWS SDKs. Teste novas versões do SimSpace Weaver aplicativo SDK com versões de aplicativos que não são de produção para garantir que seus aplicativos continuem funcionando conforme o esperado. Não é possível atualizar os aplicativos em uma simulação em execução. Para atualizar os aplicativos:

1. Atualize e teste o código do aplicativo localmente (ou em um ambiente de teste).
2. Pare de alterar o estado da simulação e salve-a (se necessário).
3. Interrompa a simulação (uma vez interrompida, ela não pode ser reiniciada).
4. Exclua a simulação (simulações interrompidas que não foram excluídas contam para seus limites de serviço).
5. Recrie a simulação com a mesma configuração e o código atualizado do aplicativo.
6. Inicialize a simulação usando dados de estado salvos (se disponíveis).
7. Inicie a nova simulação.

 **Note**

Uma nova simulação criada com a mesma configuração estará separada da simulação antiga. Ele terá um novo ID de simulação e enviará registros para um novo fluxo de registros na Amazon CloudWatch.

Registro e monitoramento em SimSpace Weaver

O monitoramento é uma parte importante da manutenção da confiabilidade, disponibilidade e desempenho de SimSpace Weaver suas outras AWS soluções. AWS fornece as seguintes ferramentas de monitoramento para observar SimSpace Weaver, relatar quando algo está errado e realizar ações automáticas quando apropriado:

- A Amazon CloudWatch monitora seus AWS recursos e os aplicativos em que você executa AWS em tempo real. É possível coletar e rastrear métricas, criar painéis personalizados e definir alarmes que o notificam ou que realizam ações quando uma métrica especificada atinge um limite definido. Para obter mais informações, consulte o [Guia CloudWatch do usuário da Amazon](#).
- O Amazon CloudWatch Logs permite que você monitore, armazene e acesse seus dados de log de seus SimSpace Weaver funcionários e de outras fontes. CloudTrail CloudWatch Os registros podem monitorar as informações nos dados do registro e notificá-lo quando determinados limites forem atingidos. É possível também arquivar seus dados de log em armazenamento resiliente. Para obter mais informações, consulte o [Guia do usuário do Amazon CloudWatch Logs](#).
- AWS CloudTrail captura API chamadas e eventos relacionados feitos por você ou em seu nome Conta da AWS e entrega os arquivos de log em um bucket do Amazon S3 que você especificar. Você pode identificar quais usuários e contas chamaram a AWS, o endereço IP de origem no qual as chamadas foram feitas e quando elas ocorreram. Para obter mais informações, consulte o [Guia do usuário do AWS CloudTrail](#).

Tópicos

- [SimSpace Weaver loga no Amazon CloudWatch Logs](#)
- [Monitoramento SimSpace Weaver com a Amazon CloudWatch](#)
- [Registrando AWS SimSpace Weaver API chamadas usando AWS CloudTrail](#)

SimSpace Weaver loga no Amazon CloudWatch Logs

Acessar os logs no SimSpace Weaver

Todos os registros gerados a partir de suas SimSpace Weaver simulações são armazenados no Amazon CloudWatch Logs. Para acessar seus registros, você pode usar o botão de CloudWatch registros no painel Visão geral da sua simulação no SimSpace Weaver console, que o levará diretamente aos registros dessa simulação específica.

The screenshot shows the AWS SimSpace Weaver console interface. On the left, there is a sidebar with the 'AWS SimSpace Weaver' logo and navigation options like 'Simulations', 'User guide', and 'Download app SDK'. The main content area displays the details for a simulation named 'MyProjectSimulation_22-11-24_10_31_07'. The 'Overview' section includes fields for ARN, Execution id, Description, Target status (Started), Role ARN, and Schema s3 location. A yellow arrow points to a 'CloudWatch logs' button located in the top right corner of the overview section.

Você também pode acessar os registros pelo CloudWatch console. Você precisará do nome da simulação para pesquisar os logs.

This screenshot is similar to the previous one, showing the same simulation details. A yellow arrow points to the simulation name 'MyProjectSimulation_22-11-24_10_31_07' in the breadcrumb navigation at the top of the main content area.

SimSpace Weaver troncos

SimSpace Weaver grava mensagens de gerenciamento de simulação e a saída do console de seus aplicativos no Amazon CloudWatch Logs. Para obter mais informações sobre como trabalhar com registros, consulte [Trabalho com grupos de registros e fluxos de registros](#) no Guia do usuário do Amazon CloudWatch Logs.

Cada simulação que você cria tem seu próprio grupo de CloudWatch registros em Registros. O nome do grupo de logs é especificado no esquema de simulação. No trecho de esquema a seguir, o valor de `log_destination_service` é `logs`. Isso significa que o valor de

`log_destination_resource_name` é o nome de um grupo de logs. Nesse caso, o grupo de logs é `MySimulationLogs`.

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

Você também pode usar o `DescribeSimulation` API para encontrar o nome do grupo de registros para simulação depois de iniciá-lo.

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

O exemplo a seguir mostra a parte da saída de `DescribeSimulation` que descreve a configuração de registro. O nome do grupo de logs é mostrado no final do `LogGroupArn`.

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

Cada grupo de logs de simulação contém vários fluxos de logs:

- Fluxo de registros de gerenciamento — mensagens de gerenciamento de simulação produzidas pelo SimSpace Weaver serviço.

```
/sim/management
```

- Fluxo de registro de erros — mensagens de erro produzidas pelo SimSpace Weaver serviço. Esse fluxo de log só existe se houver erros. SimSpace Weaver armazena erros gravados por seus aplicativos em seus próprios fluxos de registro de aplicativos (consulte os fluxos de registro a seguir).

```
/sim/errors
```

- Fluxos de logs de aplicativos espaciais (um para cada aplicativo espacial em cada operador): saída do console produzida por aplicativos espaciais. Cada aplicativo espacial grava em seu próprio fluxo de logs. A ferramenta *spatial-app-id* são todos os caracteres após a barra final no final do *worker-id*.

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

- Fluxos de logs de aplicativos personalizados (um para cada instância de aplicativo personalizado): saída de console produzida por aplicativos personalizados. Cada instância do aplicativo personalizado grava em seu próprio fluxo de logs.

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

- Fluxos de logs do aplicativo de serviço (um para cada instância do aplicativo de serviço): saída do console produzida pelos aplicativos de serviço. Cada aplicativo de serviço grava em seu próprio fluxo de logs. A ferramenta *service-app-id* são todos os caracteres após a barra final no final do *service-app-name*.

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

Monitoramento SimSpace Weaver com a Amazon CloudWatch

Você pode monitorar SimSpace Weaver usando a Amazon CloudWatch, que coleta dados brutos e os processa em métricas legíveis, quase em tempo real. Essas estatísticas são mantidas por 15 meses, de maneira que você possa acessar informações históricas e ter uma perspectiva melhor de como o aplicativo web ou o serviço está se saindo. Você também pode definir alarmes que observam determinados limites e enviam notificações ou realizam ações quando esses limites são atingidos. Para obter mais informações, consulte o [Guia CloudWatch do usuário da Amazon](#).

O SimSpace Weaver serviço relata as seguintes métricas no `AWS/simspaceweaver` namespace.

SimSpace Weaver métricas no nível da conta

O SimSpace Weaver namespace inclui as seguintes métricas relacionadas à atividade no nível da AWS conta.

Métrica	Descrição
SimulationCount	O número de simulações para a conta atual. Unidades: contagem Dimensões: nenhuma Estatísticas válidas: média, mínimo, máximo

Registrando AWS SimSpace Weaver API chamadas usando AWS CloudTrail

AWS SimSpace Weaver é integrado com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço em SimSpace Weaver. CloudTrail captura todas as API chamadas para SimSpace Weaver eventos. As chamadas capturadas incluem chamadas do SimSpace Weaver console e chamadas de código para as SimSpace Weaver API operações. Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para SimSpace Weaver. Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console em Event history. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita em SimSpace Weaver, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais sobre isso CloudTrail, consulte o [Guia AWS CloudTrail do usuário](#).

SimSpace Weaver informações em CloudTrail

CloudTrail é ativado no seu Conta da AWS quando você cria a conta. Quando a atividade ocorre em SimSpace Weaver, essa atividade é registrada em um CloudTrail evento junto com outros eventos AWS de serviço no Event history. Você pode visualizar, pesquisar e baixar eventos recentes no seu Conta da AWS. Para obter mais informações, consulte [Visualização de eventos com histórico de CloudTrail eventos](#).

Para um registro contínuo dos eventos em sua Conta da AWS, incluindo eventos para SimSpace Weaver, crie uma trilha. Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as Regiões da AWS. A trilha registra eventos de todas as regiões na AWS partição e entrega os arquivos de log ao bucket do Amazon S3 que você especificar. Além disso, você pode configurar outros AWS serviços para analisar e agir com base nos dados de eventos coletados nos CloudTrail registros. Para obter mais informações, consulte as informações a seguir.

- [Visão geral da criação de uma trilha](#)
- [CloudTrail serviços e integrações suportados](#)
- [Configurando SNS notificações da Amazon para CloudTrail](#)
- [Recebendo arquivos de CloudTrail log de várias regiões](#) e [Recebendo arquivos de CloudTrail log de várias contas](#)

Todas SimSpace Weaver as ações são registradas CloudTrail e documentadas na [AWS SimSpace Weaver APIReferência](#). Por exemplo, chamadas para o `ListSimulations` `DescribeSimulation` e `DeleteSimulation` as ações geram entradas nos arquivos de CloudTrail log.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar o seguinte:

- Se a solicitação foi feita com credenciais de usuário root ou AWS Identity and Access Management (IAM).
- Se a solicitação foi feita com credenciais de segurança temporárias de um perfil ou de um usuário federado.
- Se a solicitação foi feita por outro AWS serviço.

Para obter mais informações, consulte o [CloudTrail userIdentityelemento](#).

Entendendo as entradas do arquivo de SimSpace Weaver log

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3 que você especificar. CloudTrail os arquivos de log contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a ação solicitada, como data e hora da ação, parâmetros da solicitação e outros detalhes. CloudTrail os arquivos de log não são um rastreamento de pilha ordenado das API chamadas públicas, portanto, eles não aparecem em nenhuma ordem específica.

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a `ListSimulations` ação.

```
{
  "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 Terminais e cotas do Weaver

As tabelas a seguir descrevem os endpoints e as cotas de serviço do Weaver. SimSpace As cotas de serviço, também chamadas de limites, correspondem ao número máximo de recursos ou operações de serviço para sua Conta da AWS. Para obter mais informações, consulte [Cotas de serviço da AWS](#), na Referência geral da AWS.

Service endpoints

Nome da região	Região	Endpoint	Protocolo
Leste dos EUA (Norte da Virgínia)	us-east-1	simspaceweaver.us-east-1.amazonaws.com	HTTPS
Leste dos EUA (Ohio)	us-east-2	simspaceweaver.us-east-2.amazonaws.com	HTTPS
Oeste dos EUA (Oregon)	us-west-2	simspaceweaver.us-west-2.amazonaws.com	HTTPS
Ásia-Pacífico (Singapura)	ap-southeast-1	simspaceweaver.ap-southeast-1.amazonaws.com	HTTPS
Ásia-Pacífico (Sydney)	ap-southeast-2	simspaceweaver.ap-southeast-2.amazonaws.com	HTTPS
Europa (Estocolmo)	eu-north-1	simspaceweaver.eu-north-1.amazonaws.com	HTTPS

Nome da região	Região	Endpoint	Protocolo
Europa (Frankfurt)	eu-central-1	simspaceweaver.eu-central-1.amazonaws.com	HTTPS
Europa (Irlanda)	eu-west-1	simspaceweaver.eu-west-1.amazonaws.com	HTTPS
AWS GovCloud (Leste dos EUA)	us-gov-east-1	simspaceweaver.us-gov-east-1.amazonaws.com	HTTPS
AWS GovCloud (Oeste dos EUA)	us-gov-west-1	simspaceweaver.us-gov-west-1.amazonaws.com	HTTPS

Cotas de serviço

Nome	Padrão	Ajuste	Descrição
Unidades de recursos de computação para cada aplicativo	Cada região compatível: 4	Não	Número máximo de unidades de recursos computacionais que você pode alocar para cada aplicativo.
Unidades de recursos de computação para cada operador	Cada região compatível: 17	Não	Número de unidades de recursos computacionais disponíveis para cada operador.
Campos de dados para cada entidade	Cada região compatível: 17	Não	Número máximo de campos de dados (não indexados) que uma entidade pode ter.

Nome	Padrão	Ajuste	Descrição
Entidades em uma partição	Cada região compatível: 8.192	Não	Número máximo de entidades em uma partição.
Tamanho do campo de dados da entidade	Cada região compatível: 1.024 bytes	Não	Tamanho máximo de um campo de dados (não indexado) de uma entidade.
Transferências de entidades entre operadores	Cada região compatível: 25	Não	Número máximo de transferências de entidades entre operadores, para cada partição e cada marcação.
Transferências de entidades no mesmo operador	Cada região com suporte: 500	Não	Número máximo de transferências de entidades no mesmo operador, para cada partição e cada marcação.
Campos de índice para cada entidade	Cada região compatível: 1	Não	Número máximo de campos de índice que uma entidade pode ter.

Nome	Padrão	Ajuste	Descrição
Maior duração máxima (em dias) para uma simulação	Cada região compatível: 14	Não	Maior número de dias especificados como a duração máxima de uma simulação. Todas as simulações têm uma duração máxima, mesmo que você não especifique o valor. Uma simulação é interrompida automaticamente quando atinge a duração máxima.
Memória para cada unidade de recurso computacional	Todas as regiões compatíveis: 1 gigabyte	Não	A quantidade de memória de acesso aleatório (RAM) que um aplicativo obtém para cada unidade de recurso computacional.
Assinaturas remotas para cada operador	Cada região compatível: 24	Não	Número máximo de assinaturas remotas para cada funcionário.
Contagem de simulações	Cada região compatível: 2	Sim	Número máximo de simulações com um status de destino de STARTED em sua conta. É possível solicitar um aumento da cota até 10.

Nome	Padrão	Ajusté	Descrição
Operadores para uma simulação	Cada região compatível: 2	Sim	O número máximo de operadores que você pode atribuir a uma simulação. É possível solicitar um aumento da cota até 10.
vCPUs para cada unidade de recursos computacionais	Cada região compatível: 2	Não	O número de unidades de processamento centrais virtuais (vCPUs) que um aplicativo obtém para cada unidade de recurso de computação.

Cotas de mensagens

As cotas a seguir se aplicam às mensagens de aplicativo para aplicativo para SimSpace Weaver Local e noNuvem AWS.

Nome	Padrão	Ajustável	Descrição
Tamanho máximo de mensagem	Cada região suportada: 256 bytes	Não	O tamanho máximo da carga útil de uma mensagem.
Taxa máxima de envio de mensagens	Cada região compatível: 128	Não	O número máximo de mensagens que cada aplicativo pode enviar por tick.

Taxas de clock

O esquema de simulação especifica a taxa de clock (também chamada de taxa de atualização) para uma simulação. A tabela a seguir especifica as taxas de clock válidas que você pode usar.

Nome	Valores válidos	Descrição
Taxa de clock	Cada região suportada: "10", "15", "30", "ilimitada"	Taxas de clock válidas para uma simulação.
Taxa de clock (versões 1.13 e 1.12)	Cada região suportada: 10, 15, 30	Taxas de clock válidas para uma simulação.

Cotas de serviço para o SimSpace Weaver Local

As cotas de serviço a seguir se aplicam somente ao SimSpace Weaver Local. Todas as outras cotas também se aplicam ao SimSpace Weaver Local.

Nome	Padrão	Ajustável	Descrição
Máximo de partições	SimSpace Weaver Local: 24	Não	O número máximo de partições para uma simulação.
Máximo de aplicações	SimSpace Weaver Local: 24	Não	O número total máximo de aplicações (de qualquer tipo) para uma simulação.
Máximo de domínios	SimSpace Weaver Local: 24	Não	O número total máximo de domínios (de qualquer tipo) para uma simulação.
Entidades por partição	SimSpace Weaver Local: 4.096	Não	Número máximo de entidades em cada partição.
Campos por entidade	SimSpace Weaver Local: 8	Não	O número máximo de campos para cada entidade.

Nome	Padrão	Ajustável	Descrição
Tamanho do campo	SimSpace Weaver Local: 1.024 bytes	Não	O tamanho máximo de um campo de entidade.

Solucionar problemas no SimSpace Weaver

Tópicos

- [AssumeRoleAccessDenied](#)
- [InvalidBucketName](#)
- [ServiceQuotaExceededException](#)
- [TooManyBuckets](#)
- [Permissão negada durante o início da simulação](#)
- [Problemas relacionados ao tempo de uso do Docker](#)
- [PathfindingSample o cliente do console não consegue se conectar](#)
- [A AWS CLI não reconhece o simspaceweaver](#)

AssumeRoleAccessDenied

Você pode receber o seguinte erro se sua simulação falhar ao iniciar:

```
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
```

Você pode receber esse erro se uma das seguintes afirmações for verdadeira sobre a função AWS Identity and Access Management (IAM) da sua simulação:

- O nome do recurso da Amazon (ARN) se refere a uma função do IAM que não existe.
- A política de confiança para a função do IAM não permite que o nome da nova simulação assuma a função.

Verifique se a função existe. Se a função existir, verifique a política de confiança para ela. O exemplo `aws:SourceArn` de política de confiança a seguir permite somente uma simulação (na conta 111122223333) cujo nome comece com `MySimulation` para assumir a função.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": "arn:aws:iam::111122223333:role/MySimulation",  
      "Action": "sts:AssumeRole",  
      "Resource": "arn:aws:iam::111122223333:role/MySimulation"    }  
  ]  
}
```

```

    {
      "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 outra simulação cujo nome comece com `MyOtherSimulation` assuma a função, a política de confiança deve ser modificada como no exemplo editado a seguir:

```

{
  "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

Você pode receber o seguinte erro ao criar um projeto:

```
An error occurred (InvalidBucketName) when calling the CreateBucket operation: The specified bucket is not valid.
```

Você recebeu esse erro porque o nome que o SimSpace Weaver transmitiu ao Amazon Simple Storage Service (Amazon S3) violou as regras de nomenclatura do bucket. Para obter mais informações, consulte [Regras de nomenclatura do bucket](#) no Guia do usuário do Amazon Simple Storage Service.

O script `create-project` no SDK do aplicativo do SimSpace Weaver cria nomes de bucket usando o nome do projeto que você fornecer ao script. Os nomes dos buckets seguem os seguintes formatos.

- Versão 1.13.x ou posterior
 - `weaver-lowercase-project-name-account-number-region`
- Versão 1.12.x
 - `weaver-lowercase-project-name-account-number-app-zips-region`
 - `weaver-lowercase-project-name-account-number-schemas-region`

Por exemplo, dadas as seguintes propriedades do projeto:

- Nome do projeto: `MyProject`
- Número da Conta da AWS: `111122223333`
- Região da AWS: `us-west-2`

O projeto teria os seguintes buckets:

- Versão 1.13.x ou posterior
 - `weaver-myproject-111122223333-us-west-2`
- Versão 1.12.x
 - `weaver-myproject-111122223333-app-zips-us-west-2`
 - `weaver-myproject-111122223333-schemas-us-west-2`

O nome do seu projeto não deve violar as regras de nomenclatura do Amazon S3. Você também deve usar um nome de projeto que seja curto o suficiente para que os nomes de bucket criados pelo script `create-project` não excedam o limite de tamanho de nome para buckets do Amazon S3.

ServiceQuotaExceededException

Você pode receber o seguinte erro ao iniciar uma simulação:

```
An error occurred (ServiceQuotaExceededException) when calling the StartSimulation operation: Failed to start simulation due to: simulation quota has already been reached.
```

Você receberá esse erro se tentar iniciar uma nova simulação, mas sua conta atualmente tiver o número máximo de simulações com um status-alvo de `STARTED`. Isso inclui simulações em execução, com falha e que pararam porque atingiram a duração máxima. É possível excluir uma simulação interrompida ou com falha para iniciar uma nova simulação. Se todas as simulações estiverem em execução, você poderá interromper e excluir uma simulação que está sendo executada. É possível solicitar um aumento das Service Quotas se ainda não estiverem no limite de solicitações.

TooManyBuckets

Você pode receber o seguinte erro ao criar um projeto:

```
An error occurred (TooManyBuckets) when calling the CreateBucket operation: You have attempted to create more buckets than allowed.
```

O Amazon Simple Storage Service (Amazon S3) tem um limite no número de buckets que você pode ter em AWS sua conta (para obter mais informações, [consulte Restrições e limitações de bucket](#) no Guia do usuário do Amazon Simple Storage Service).

É necessário realizar uma das seguintes ações antes de continuar:

- Excluir dois ou mais buckets do Amazon S3 que você não precisa.
- Solicite um aumento de limite do Amazon S3. Para obter mais informações, consulte [Restrições e limitações de bucket](#) no Guia do usuário do Amazon Simple Storage Service.
- Use uma AWS conta diferente.

Note

A `DeleteSimulation` API SimSpace Weaver não exclui os recursos do Amazon S3 associados à sua simulação. É recomendável que você remova todos os recursos associados às simulações quando eles não são mais necessários.

Permissão negada durante o início da simulação

Ao iniciar uma simulação, você pode receber uma mensagem de erro indicando que a permissão foi negada ou que houve um erro ao acessar os artefatos do seu aplicativo. Esse problema pode ocorrer quando você especifica buckets do Amazon S3 para sua simulação que o SimSpace Weaver não criou para você, seja por meio do console ou dos scripts do aplicativo do SDK do SimSpace Weaver.

As seguintes situações são as causas mais prováveis:

- O serviço não tem permissão para acessar um ou mais dos buckets do Amazon S3 que você especificou em seu esquema de simulação. Verifique sua política de permissões de função de aplicativo, políticas de bucket do Amazon S3 e permissões de bucket do Amazon S3 para garantir que o `sinspaceweaver.amazonaws.com` tenha as permissões corretas para acessar seus buckets. Para obter mais informações sobre como editar uma política de permissões de função, consulte [Permissões que SimSpace Weaver criam para você](#).
- Seu bucket do Amazon S3 pode estar em um local da Região da AWS diferente da sua simulação. Seus buckets do Amazon S3 para seus artefatos de simulação devem estar na mesma Região da AWS da simulação. Verifique seu console Amazon S3 para ver em que Região da AWS seu bucket está. Se o seu bucket do Amazon S3 estiver em outra Região da AWS, selecione um bucket que esteja na mesma Região da AWS que a simulação.

Problemas relacionados ao tempo de uso do Docker

Se você estiver usando o Docker e receber erros relacionados ao tempo ao executar scripts do aplicativo do SDK do SimSpace Weaver, a causa pode ser que o relógio da máquina virtual do Docker esteja incorreto. Isso pode acontecer se o computador estava executando o Docker antes de voltar do modo de suspensão ou hibernação.

Soluções possíveis

- Reinicie o Docker.

- Desative e reative a sincronização de horário no Windows PowerShell:

```
Get-VMIntegrationService -VMName DockerDesktopVM -Name "Time Synchronization" |  
  Disable-VMIntegrationService  
Get-VMIntegrationService -VMName DockerDesktopVM -Name "Time Synchronization" |  
  Enable-VMIntegrationService
```

PathfindingSample o cliente do console não consegue se conectar

Você pode receber o seguinte erro do cliente do console ao se conectar à PathfindingSample simulação descrita nos tutoriais em [Começando com SimSpace Weaver](#). Esse erro ocorre porque o cliente não consegue abrir uma conexão de rede com o ViewApp no endereço IP combinado e no número da porta fornecido.

```
Fatal error in function nng_dial. Error code: 268435577. Error message: no link
```

Para uma simulação no Nuvem AWS

- Sua conexão de rede está funcionando corretamente? Verifique se você pode se conectar a outros endereços IP ou sites que devem funcionar. Certifique-se de que seu navegador não esteja carregando um site a partir do cache.
- Sua simulação está sendo executada? Você pode usar a ListSimulationsAPI para obter o status da sua simulação. Para ter mais informações, consulte [Obtenha o endereço IP e o número da porta de um aplicativo personalizado](#). Também é possível usar o [console do SimSpace Weaver](#) para conferir o status das simulações.
- Seus aplicativos estão em execução? Você pode usar a DescribeAppAPI para obter o status dos seus aplicativos. Para ter mais informações, consulte [Obtenha o endereço IP e o número da porta de um aplicativo personalizado](#). Também é possível usar o [console do SimSpace Weaver](#) para conferir o status das simulações.
- Seus aplicativos estão em execução? Você pode usar a DescribeAppAPI para obter o status dos seus aplicativos. Para ter mais informações, consulte [Obtenha o endereço IP e o número da porta de um aplicativo personalizado](#). Também é possível usar o [console do SimSpace Weaver](#) para conferir o status das simulações.
- Você usou o endereço IP e o número da porta corretos? Ao se conectar à internet, você deve usar o endereço IP e o número da porta Actual do ViewApp. Você pode encontrar o IP Address e o número da Actual porta no EndpointInfo bloco da saída da DescribeAppAPI. Você também

pode usar o [console do SimSpace Weaver](#) para encontrar o endereço IP (URI) e o número da porta (porta de entrada) para o ViewApp na página de detalhes do MyViewDomain.

- Sua conexão de rede passa por um firewall? Seu firewall pode bloquear a conexão com o endereço IP ou o número da porta, ou ambos. Verifique as configurações do firewall ou consulte o administrador dele.

Para uma simulação local

- Você consegue se conectar ao seu endereço de loopback (127.0.0.1)? Se você tiver a ferramenta de linha de comando ping no Windows, poderá abrir uma janela do prompt de comando e tentar executar o ping em 127.0.0.1. Pressione Ctrl-C para finalizar o ping.

```
ping 127.0.0.1
```

Example saída do ping

```
C:\>ping 127.0.0.1

Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time=1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
Control-C
^C
C:\>
```

Se o ping indicar que pacotes foram perdidos, pode haver outro software (como um firewall local, configurações de segurança ou programas antimalware) que esteja bloqueando sua conexão.

- Seus aplicativos estão em execução? A simulação local é executada como janelas separadas para cada aplicativo. Certifique-se de que as janelas de seus aplicativos espaciais e para o ViewApp estejam abertas. Para ter mais informações, consulte [Desenvolvimento local em SimSpace Weaver](#).

- Você usou o endereço IP e o número da porta corretos? Você deve usar `tcp://127.0.0.1:7000` ao se conectar a uma simulação local. Para ter mais informações, consulte [Desenvolvimento local em SimSpace Weaver](#).
- Você tem um software de segurança local que pode bloquear sua conexão? Verifique suas configurações de segurança, firewall local ou programas antimalware para ver se eles estão bloqueando sua conexão com `127.0.0.1` em uma porta TCP `7000`.

A AWS CLI não reconhece o **simspaceweaver**

Se a AWS CLI apresentar erros que sugiram que ela não reconhece o SimSpace Weaver, execute o comando a seguir.

```
aws simspaceweaver help
```

Se você receber um erro que começa com as linhas a seguir e lista todas as opções disponíveis, a AWS CLI talvez seja de uma versão mais antiga.

```
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:
```

Para verificar a versão da AWS CLI, execute o seguinte comando:

```
aws --version
```

Se a versão for anterior a 2.9.19, você deve atualizar a AWS CLI. Observe que a versão atual da AWS CLI é posterior à 2.9.19.

Para obter informações sobre como atualizar a AWS CLI, consulte [Instalar ou atualizar a versão mais recente da AWS CLI](#) no Guia do usuário da AWS Command Line Interface versão 2.

SimSpace Weaver referência do esquema de simulação

SimSpace Weaver usa um YAML arquivo para configurar as propriedades de uma simulação. Esse arquivo é chamado de esquema de simulação (ou apenas esquema). A simulação de amostra incluída no SimSpace Weaver aplicativo SDK inclui um esquema que você pode copiar e editar para sua própria simulação.

Tópicos

- [Exemplo de um esquema completo](#)
- [Formato do esquema](#)

Exemplo de um esquema completo

O exemplo a seguir mostra o YAML-arquivo de texto formatado que descreve uma SimSpace Weaver simulação. Este exemplo inclui valores fictícios para as propriedades. O formato do arquivo varia com base no valor para `sdk_version` especificado nele. Consulte [Formato do esquema](#) para obter uma descrição completa das propriedades e valores válidos.

```
sdk_version: "1.17"
simulation_properties:
  log_destination_resource_name: "MySimulationLogs"
  log_destination_service: "logs"
  default_entity_index_key_type: "Vector3<f32>"
  default_image: "111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-
repository:latest"
workers:
  MyComputeWorkers:
    type: "sim.c5.24xlarge"
    desired: 3
clock:
  tick_rate: "30"
partitioning_strategies:
  MyGridPartitioning:
    topology: "Grid"
    aabb_bounds:
      x: [-1000, 1000]
      y: [-1000, 1000]
    grid_placement_groups:
      x: 3
```

```
  y: 3
domains:
  MyCustomDomain:
    launch_apps_via_start_app_call: {}
    app_config:
      package: "s3://weaver-myproject-111122223333-us-west-2/MyViewApp.zip"
      launch_command: ["MyViewApp"]
      required_resource_units:
        compute: 1
      endpoint_config:
        ingress_ports: [9000, 9001]
  MyServiceDomain:
    launch_apps_per_worker:
      count: 1
    app_config:
      package: "s3://weaver-myproject-111122223333-us-west-2/
MyConnectionServiceApp.zip"
      launch_command: ["MyConnectionServiceApp"]
      required_resource_units:
        compute: 1
      endpoint_config:
        ingress_ports:
          - 9000
          - 9001
  MySpatialDomain:
    launch_apps_by_partitioning_strategy:
      partitioning_strategy: "MyGridPartitioning"
      grid_partition:
        x: 6
        y: 6
    app_config:
      package: "s3://weaver-myproject-111122223333-us-west-2/MySpatialApp.zip"
      launch_command: ["MySpatialApp"]
      required_resource_units:
        compute: 1
  MySpatialDomainWithCustomContainer:
    launch_apps_by_partitioning_strategy:
      partitioning_strategy: "MyGridPartitioning"
      grid_partition:
        x: 6
        y: 6
    app_config:
      package: "s3://weaver-myproject-111122223333-us-west-2/MySpatialApp2.zip"
      launch_command: ["MySpatialApp2"]
```

```
required_resource_units:
  compute: 1
  image: "111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest"
placement_constraints:
  - placed_together: ["MySpatialDomain", "MySpatialDomainWithCustomContainer"]
  on_workers: ["MyComputeWorkers"]
```

Formato do esquema

O exemplo a seguir mostra a estrutura geral de um esquema. A ordem das propriedades em cada nível do esquema não importa, desde que as relações pai-filho sejam as mesmas. A ordem é importante para os elementos em uma 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
```

Seções

- [SDKversão](#)
- [Propriedades da simulação](#)
- [Operadores](#)
- [Relógio](#)
- [Estratégias de particionamento](#)
- [Domínios](#)
- [Limitações de posicionamento](#)

SDKversão

A `sdk_version` seção (obrigatória) identifica a versão do SimSpace Weaver aplicativo compatível com SDK esse esquema. Valores válidos: 1.17, 1.16, 1.15, 1.14, 1.13, 1.12

Important

O valor de `sdk_version` inclui somente o número da versão principal e o número da primeira versão secundária. Por exemplo, o valor 1.12 especifica todas as versões 1.12.x, como 1.12.0, 1.12.1, e 1.12.2.

```
sdk_version: "1.17"
```

Propriedades da simulação

A seção `simulation_properties` (obrigatória) especifica várias propriedades da simulação. Use esta seção para configurar o registro e especificar uma imagem de contêiner padrão. Esta seção é obrigatória mesmo que você não configure o registro em log ou opte por especificar uma imagem de contêiner padrão.

```
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"
```

Properties

`log_destination_resource_name`

Especifica o recurso no qual os registros SimSpace Weaver serão gravados.

Obrigatório: não. Se essa propriedade não estiver incluída, o SimSpace Weaver não gravará logs para a simulação.

Type: string

Valores válidos:

- O nome de um grupo de CloudWatch registros de registros (por exemplo, MySimulationLogs)
- O nome do recurso Amazon (ARN) de um grupo de CloudWatch registros de registros (por exemplo, arn:aws:logs:us-west-2:111122223333:log-group/MySimulationLogs)

Note

SimSpace Weaver suporta apenas um destino de log na mesma conta e Região da AWS na simulação.

log_destination_service

Indica o tipo de recurso de destino do registro quando você especifica um `logging_destination_resource_name` que não é um ARN.

Obrigatório: você deve especificar essa propriedade se `log_destination_resource_name` for especificada e não for uma ARN. Você não pode especificar essa propriedade se a `log_destination_resource_name` não for especificada ou for uma ARN.

Type: string

Valores válidos:

- `logs`: o recurso de destino do log é um grupo de logs.

default_entity_index_key_type

Especifica o tipo de dados para o campo da chave de índice das entidades de simulação.

Obrigatório: Sim

Type: string

Valores válidos: `Vector3<f32>`

default_image

Especifica a imagem padrão do contêiner para a simulação (não compatível com as versões 1.13 e 1.12). Se essa propriedade for especificada, os domínios que não especificam `image` podem usar `default_image`.

Obrigatório: não

Type: string

Valores válidos:

- O URI de um repositório no Amazon Elastic Container Registry (Amazon ECR) (por exemplo, `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`)

Operadores

A seção `workers` (obrigatória) especifica configurações para grupos de operadores. O SimSpace Weaver usa essas informações junto com `placement_constraints` para configurar a infraestrutura subjacente da simulação. Somente um grupo de operadores tem suporte atualmente.

Para especificar as propriedades de um grupo de trabalhadores, substitua *worker-group-name* com um nome de sua escolha. O nome deve ter de 3 a 64 caracteres e pode conter os caracteres A-Z, a-z, 0-9, e `_` (hífen). Especifique as propriedades do grupo de operadores após o nome.

```
workers:  
  worker-group-name:  
    type: "sim.c5.24xlarge"  
    desired: number-of-workers
```

Properties

type

Especifique o tipo de operador.

Obrigatório: Sim

Type: string

Valores válidos: `sim.c5.24xlarge`

desired

Especifica o número desejado de operadores para esse grupo de operadores.

Obrigatório: Sim

Tipo: inteiro

Valores válidos: 1-3. A cota de serviço (limite) para o número de operadores nas simulações determina o valor máximo dessa propriedade. Por exemplo, se a cota de serviço for 2, o valor máximo dessa propriedade será 2. Também é possível solicitar um aumento em suas cotas. Para obter mais informações, consulte [SimSpace Terminais e cotas do Weaver](#).

Relógio

A seção `clock` (obrigatória) especifica as propriedades do relógio da simulação.

```
clock:  
  tick_rate: tick-rate
```

Properties

`tick_rate`

Especifica o número de tiques por segundo que o relógio publica nos aplicativos.

Obrigatório: Sim

Type:

- Versão 1.14 e 1.15: string
- Versão 1.13 e 1.12: inteiro

Valores válidos:

- Versão 1.14 e 1.15: "10" | "15" | "30" | "unlimited"
 - "unlimited": o relógio envia a próxima marca assim que todos os aplicativos concluírem suas operações de confirmação para a marca atual.
- Versão 1.13 e 1.12: 10 | 15 | 30

Estratégias de particionamento

A seção `partitioning_strategies` (obrigatória) especifica a organização das partições para um domínio espacial.

Note

SimSpace Weaver suporta apenas 1 estratégia de particionamento.

Para especificar as propriedades de uma estratégia de particionamento, substitua *partitioning-strategy-name* com um nome de sua escolha. O nome deve ter de 3 a 64 caracteres e pode conter os caracteres A-Z, a-z, 0-9, e _- (hífen). Especifique as propriedades da estratégia de particionamento após o nome.

```
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
```

Properties

topology

Especifique a topologia (esquema de organização de partições) para essa estratégia de particionamento.

Obrigatório: Sim

Type: string

Valores válidos: "Grid"

aabb_bounds

Especifica os limites da caixa delimitadora alinhada ao eixo principal (AABB) para sua simulação. Você especifica os limites como matrizes ordenadas de 2 elementos que descrevem os valores mínimo e máximo (nessa ordem) para cada eixo (x e y).

Obrigatório: condicional. Essa propriedade é obrigatória e só pode ser especificada se a topologia estiver definida como "Grid".

Tipo: matriz Float (para cada eixo)

Valores válidos: $-3.4028235e38$ - $3.4028235e38$

`grid_placement_groups`

Especifica o número de grupos de posicionamento ao longo de cada eixo (x e y) em uma topologia de grade. Um grupo de posicionamento é uma coleção de partições no mesmo domínio, espacialmente adjacentes.

Obrigatório: condicional. Essa propriedade é obrigatória e só pode ser especificada se a topologia estiver definida como "Grid". Se você não especificar uma configuração de grupos de posicionamento, o SimSpace Weaver calculará uma para você. Qualquer domínio que usa uma estratégia de particionamento sem uma configuração de grupos de posicionamento deve especificar uma `grid_partition` (consulte [Estratégia de particionamento de domínio espacial](#)).

Tipo: inteiro (para cada eixo)

Valores válidos: 1-20. Recomendamos que $x * y$ seja igual ao número desejado de operadores. Caso contrário, SimSpace Weaver tentará equilibrar seus grupos de colocação entre os trabalhadores disponíveis.

Domínios

A seção `domains` (obrigatória) especifica as propriedades de cada um dos seus domínios. Todas as simulações devem ter pelo menos uma seção para um domínio espacial. Você pode criar várias seções para domínios adicionais. Cada tipo de domínio tem seu próprio formato de configuração.

Important

As versões 1.13 e 1.12 não oferecem suporte a vários domínios espaciais.

Important

SimSpace Weaver suporta até 5 domínios para cada simulação. Isso inclui todos os domínios espaciais, personalizados e de serviço.

```
domains:
```

```

domain-name:
  domain-configuration
domain-name:
  domain-configuration
...

```

Configuração do domínio

- [Configuração de domínios espaciais](#)
- [Configuração de domínios personalizados](#)
- [Configuração do domínio do serviço](#)

Configuração de domínios espaciais

Para especificar as propriedades de um domínio espacial, substitua *spatial-domain-name* com um nome de sua escolha. O nome deve ter de 3 a 64 caracteres e pode conter os caracteres A-Z, a-z, 0-9, e _- (hífen). Especifique as propriedades do domínio espacial após o nome.

```

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"

```

Estratégia de particionamento de domínio espacial

A seção `launch_apps_by_partitioning_strategy` (obrigatória) especifica a estratégia de particionamento e as dimensões (em número de partições) do espaço de simulação.

```

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

```

Properties

partitioning_strategy

Especifica a estratégia de particionamento para esse domínio espacial.

Obrigatório: Sim

Type: string

Valores válidos: o valor dessa propriedade deve corresponder ao nome de uma estratégia de particionamento definida na seção `partitioning_strategies`. Para obter mais informações, consulte [Estratégias de particionamento](#).

grid_partition

Especifica o número de partições ao longo de cada eixo (x e y) em uma topologia de grade. Essas dimensões descrevem o espaço total de simulação para esse domínio.

Obrigatório: condicional. Pode ser especificada somente se a topologia estiver definida como "Grid". Essa propriedade depende da propriedade `grid_placement_groups` da estratégia de particionamento especificada para esse domínio:

- Essa propriedade é necessária se a estratégia de particionamento desse domínio não especificar uma configuração para `grid_placement_groups`.
- Se houver uma configuração `grid_placement_groups`, mas você não especificar `grid_partition`, o SimSpace Weaver usará as mesmas dimensões de configuração da `grid_placement_groups`.
- Se você especificar ambos `grid_placement_groups` e `grid_partition`, as dimensões de `grid_partition` devem ser múltiplos das dimensões de `grid_placement_groups` (por exemplo, se suas dimensões de `grid_placement_groups` forem 2 x 2, algumas dimensões válidas para `grid_partition` são 2 x 2, 4 x 4, 6 x 6, 8 x 8, 10 x 10).

Tipo: inteiro (para cada eixo)

Valores válidos: 1-20

Configuração do aplicativo espacial

A seção `app_config` (obrigatório) especifica o pacote, a configuração de inicialização e os requisitos de recursos para aplicativos nesse domínio.

```
app_config:  
  package: "app-package-s3-uri"  
  launch_command: ["app-launch-command", "parameter1", ...]  
  required_resource_units:  
    compute: app-resource-units
```

Properties

package

Especifica o pacote (arquivo .zip) que contém o executável/binário do aplicativo. O pacote deve ser armazenado em um bucket do Amazon S3. Somente o formato de arquivo .zip é compatível.

Obrigatório: Sim

Type: string

Valores válidos: O Amazon S3 URI do pacote em um bucket do Amazon S3. Por exemplo, `s3://weaver-myproject-111122223333-app-zips-us-west-2/MySpatialApp.zip`.

launch_command

Especifica o nome do arquivo executável/binário e os parâmetros da linha de comando para iniciar o aplicativo. Cada token de string de linha de comando é um elemento na matriz.

Obrigatório: Sim

Tipo: matriz de strings

required_resource_units

Especifica o número de unidades de recursos que o SimSpace Weaver deve alocar para cada instância desse aplicativo. Uma unidade de recurso é uma quantidade fixa de unidades de processamento central virtual (vCPUs) e memória de acesso aleatório (RAM) em um trabalhador. Para obter mais informações sobre unidades de recursos, consulte [Endpoints e cotas de serviço](#). A propriedade `compute` especifica uma alocação de unidades de recursos para a família de operadores `compute` e atualmente é o único tipo válido de alocação.

Obrigatório: Sim

Tipo: inteiro

Valores válidos: 1-4

Imagem de contêiner personalizada

A `image` propriedade (opcional) especifica a localização de uma imagem de contêiner SimSpace Weaver usada para executar aplicativos nesse domínio (não suportada nas versões 1.13 e 1.12). Forneça o URI para um repositório no Amazon Elastic Container Registry (Amazon ECR) que contém a imagem. Se essa propriedade não for especificada, mas uma `default_image` estiver especificada na seção de nível superior `simulation_properties`, os aplicativos nesse domínio usarão a `default_image`. Para obter mais informações, consulte [Contêineres personalizados](#).

```
image: "ecr-repository-uri"
```

Properties

image

Especifica a localização de uma imagem de contêiner para executar aplicativos nesse domínio.

Obrigatório: não

Type: string

Valores válidos:

- O URI de um repositório no Amazon Elastic Container Registry (Amazon ECR) (por exemplo, `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`)

Configuração de domínios personalizados

Para especificar as propriedades de um domínio personalizado, substitua `custom-domain-name` com um nome de sua escolha. O nome deve ter de 3 a 64 caracteres e pode conter os caracteres A-Z, a-z, 0-9, e `_` (hífen). Especifique as propriedades do domínio personalizado após o nome. Repita esse processo para cada domínio 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"
```

Properties

launch_apps_via_start_app_call

Essa propriedade é necessária para iniciar seus aplicativos personalizados usando o StartApp API.

Obrigatório: Sim

Tipo: N/A

Valores válidos: {}

Configuração do aplicativo personalizado

O `app_config` section (obrigatório) especifica o pacote, a configuração de inicialização, os requisitos de recursos e as portas de rede para aplicativos nesse domínio 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, ...]
```

Properties

package

Especifica o pacote (arquivo .zip) que contém o executável/binário do aplicativo. O pacote deve ser armazenado em um bucket do Amazon S3. Somente o formato de arquivo .zip é compatível.

Obrigatório: Sim

Type: string

Valores válidos: O Amazon S3 URI do pacote em um bucket do Amazon S3. Por exemplo, `s3://weaver-myproject-111122223333-app-zips-us-west-2/MyCustomApp.zip`.

`launch_command`

Especifica o nome do arquivo executável/binário e os parâmetros da linha de comando para iniciar o aplicativo. Cada token de string de linha de comando é um elemento na matriz.

Obrigatório: Sim

Tipo: matriz de strings

`required_resource_units`

Especifica o número de unidades de recursos que o SimSpace Weaver deve alocar para cada instância desse aplicativo. Uma unidade de recurso é uma quantidade fixa de unidades de processamento central virtual (vCPUs) e memória de acesso aleatório (RAM) em um trabalhador. Para obter mais informações sobre unidades de recursos, consulte [Endpoints e cotas de serviço](#). A propriedade `compute` especifica uma alocação de unidades de recursos para a família de operadores `compute` e atualmente é o único tipo válido de alocação.

Obrigatório: Sim

Tipo: inteiro

Valores válidos: 1-4

`endpoint_config`

Especifica os endpoints de rede para aplicativos nesse domínio. O valor de `ingress_ports` especifica as portas às quais seus aplicativos personalizados se vinculam para conexões de entrada de clientes. SimSpace Weaver mapeia portas alocadas dinamicamente para suas portas de entrada especificadas. As portas de entrada são ambas TCP e UDP Use o comando `DescribeApp API` para encontrar o número real da porta para conectar seus clientes.

Obrigatório: não. Se você não especificar a configuração do endpoint, seus aplicativos personalizados nesse domínio não terão endpoints de rede.

Tipo: matriz de inteiros

Valores válidos: 1024-49152. Os valores devem ser exclusivos.

Imagem de contêiner personalizada

A `image` propriedade (opcional) especifica a localização de uma imagem de contêiner SimSpace Weaver usada para executar aplicativos nesse domínio (não suportada nas versões 1.13 e 1.12). Forneça o URI para um repositório no Amazon Elastic Container Registry (Amazon ECR) que contém a imagem. Se essa propriedade não for especificada, mas uma `default_image` estiver especificada na seção de nível superior `simulation_properties`, os aplicativos nesse domínio usarão a `default_image`. Para obter mais informações, consulte [Contêineres personalizados](#).

```
image: "ecr-repository-uri"
```

Properties

image

Especifica a localização de uma imagem de contêiner para executar aplicativos nesse domínio.

Obrigatório: não

Type: string

Valores válidos:

- O URI de um repositório no Amazon Elastic Container Registry (Amazon ECR) (por exemplo, `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`)

Configuração do domínio do serviço

Para especificar as propriedades de um domínio de serviço, substitua `service-domain-name` com um nome de sua escolha. O nome deve ter de 3 a 64 caracteres e pode conter os caracteres A-Z, a-z, 0-9, e `_` (hífen). Especifique as propriedades do domínio do serviço após o nome. Repita esse processo para cada domínio de serviço.

```
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"
```

Inicie aplicativos por operador

A seção `launch_apps_per_worker` (obrigatória) indica que essa é uma configuração de domínio de serviço e especifica o número de aplicativos de serviço a serem iniciados por operador.

```
launch_apps_per_worker:  
  count: number-of-apps-to-launch
```

Properties

count

Essa propriedade especifica o número de aplicativos de serviço a serem iniciados por operador.

Obrigatório: Sim

Tipo: inteiro

Valores válidos: {} | 1 | 2. O valor de {} especifica o valor padrão de 1.

Configuração do aplicativo de serviço

O `app_config` section (obrigatório) especifica o pacote, a configuração de inicialização, os requisitos de recursos e as portas de rede para aplicativos nesse domínio de serviço.

```
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, ...]
```

Properties

package

Especifica o pacote (arquivo .zip) que contém o executável/binário do aplicativo. O pacote deve ser armazenado em um bucket do Amazon S3. Somente o formato de arquivo .zip é compatível.

Obrigatório: Sim

Tipo: string

Valores válidos: O Amazon S3 URI do pacote em um bucket do Amazon S3. Por exemplo, `s3://weaver-myproject-111122223333-app-zips-us-west-2/MyServiceApp.zip`.

launch_command

Especifica o nome do arquivo executável/binário e os parâmetros da linha de comando para iniciar o aplicativo. Cada token de string de linha de comando é um elemento na matriz.

Obrigatório: Sim

Tipo: matriz de strings

required_resource_units

Especifica o número de unidades de recursos que o SimSpace Weaver deve alocar para cada instância desse aplicativo. Uma unidade de recurso é uma quantidade fixa de unidades de processamento central virtual (vCPUs) e memória de acesso aleatório (RAM) em um trabalhador. Para obter mais informações sobre unidades de recursos, consulte [Endpoints e cotas de serviço](#). A propriedade `compute` especifica uma alocação de unidades de recursos para a família de operadores `compute` e atualmente é o único tipo válido de alocação.

Obrigatório: Sim

Tipo: inteiro

Valores válidos: 1-4

endpoint_config

Especifica os endpoints de rede para aplicativos nesse domínio. O valor de `ingress_ports` especifica as portas às quais seus aplicativos de serviço se vinculam para conexões de entrada de clientes. O SimSpace Weaver mapeia portas alocadas dinamicamente para suas portas de entrada especificadas. As portas de entrada são ambas TCP e UDP Use o comando `DescribeApp` API para encontrar o número real da porta para conectar seus clientes.

Obrigatório: não. Se você não especificar a configuração do endpoint, seus aplicativos de serviço nesse domínio não terão endpoints de rede.

Tipo: matriz de inteiros

Valores válidos: 1024-49152. Os valores devem ser exclusivos.

Imagem de contêiner personalizada

A `image` propriedade (opcional) especifica a localização de uma imagem de contêiner SimSpace Weaver usada para executar aplicativos nesse domínio (não suportada nas versões 1.13 e 1.12). Forneça o URI para um repositório no Amazon Elastic Container Registry (Amazon ECR) que contém a imagem. Se essa propriedade não for especificada, mas uma `default_image` estiver especificada na seção de nível superior `simulation_properties`, os aplicativos nesse domínio usarão a `default_image`. Para obter mais informações, consulte [Contêineres personalizados](#).

```
image: "ecr-repository-uri"
```

Properties

image

Especifica a localização de uma imagem de contêiner para executar aplicativos nesse domínio.

Obrigatório: não

Type: string

Valores válidos:

- O URI de um repositório no Amazon Elastic Container Registry (Amazon ECR) (por exemplo, `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`)

Limitações de posicionamento

A `placement_constraints` seção (opcional) especifica quais domínios espaciais SimSpace Weaver devem ser posicionados juntos nos mesmos operadores. Para obter mais informações, consulte [Configurar domínios espaciais](#).

⚠ Important

As versões 1.13 e 1.12 não suportam `placement_constraints`.

```
placement_constraints:  
- placed_together: ["spatial-domain-name", "spatial-domain-name", ...]  
  on_workers: ["worker-group-name"]
```

Properties

`placed_together`

Especifica os domínios espaciais que SimSpace Weaver devem ser colocados juntos.

Obrigatório: Sim

Tipo: matriz de strings

Valores válidos: nomes de domínios espaciais especificados no esquema

`on_workers`

Especifica o grupo de trabalhadores no qual os domínios SimSpace Weaver devem ser colocados.

Obrigatório: Sim

Tipo: matriz de strings de um elemento

Valores válidos: nome de um grupo de operadores especificado no esquema

SimSpace Weaver Referências de API

SimSpace Weaver tem 2 conjuntos diferentes de interfaces de programação de aplicativos (APIs):

- APIs de serviço: essas APIs controlam o serviço e os recursos do serviço, como simulações, relógios e aplicativos. Eles fazem parte do kit principal de desenvolvimento de AWS software (SDK) e você pode usar a Interface de Linha de AWS Comando (CLI) para chamá-los. Para obter mais informações sobre como usar as APIs, consulte a [Referência da API do SimSpace Weaver](#).
- APIs do SDK do aplicativo: essas APIs controlam os dados na simulação. Você as usa no código do aplicativo para fazer coisas como ler e gravar dados do campo da entidade, trabalhar com assinaturas e monitorar eventos na simulação. Para obter mais informações, consulte a documentação do SDK do SimSpace Weaver aplicativo na pasta do SDK do aplicativo: `sdk-folder\SimSpaceWeaverAppSdk\documentation`

Note

`sdk-folder` é a pasta em que você descompactou o pacote.
SimSpaceWeaverAppSdkDistributable

AWS SimSpace Weaver versões

Melhoramos continuamente o AWS SimSpace Weaver. Você deve baixar o SDK do SimSpace Weaver aplicativo mais recente quando lançarmos uma nova versão se quiser aproveitar os novos recursos e as atualizações de recursos. Para executar uma simulação existente com uma versão mais recente, talvez seja necessário atualizar o esquema e o código e, em seguida, iniciar uma nova instância da simulação. Não é necessário fazer o upgrade e é possível continuar executando simulações existentes com versões anteriores. Consulte esta página para ver o que há de diferente entre as versões. Atualmente, todas as versões são compatíveis.

Important

A versão mais recente do [Guia AWS SimSpace Weaver do Usuário](#) abrange somente a versão mais recente do serviço. Você pode encontrar a documentação das versões anteriores no [Catálogo do AWS SimSpace Weaver Guide](#), disponível na [página inicial da documentação principal](#).

Versão mais recente

A última versão é: 1.17.0

Como encontrar a sua versão atual

Se você criou uma simulação com o SDK do SimSpace Weaver aplicativo, o `create-project` script baixou uma versão das bibliotecas do SDK em um subdiretório no seu `sdk-folder`. O subdiretório que contém as bibliotecas do SDK tem um nome que inclui o número da versão do SDK: `SimSpaceWeaverAppSdk-sdk-version`. Por exemplo, as bibliotecas da versão 1.16.0 estão em `SimSpaceWeaverAppSdk-1.16.0`.

Você também pode encontrar a versão do pacote distribuível do SDK do SimSpace Weaver aplicativo no arquivo de texto `app_sdk_distributable_version.txt` em seu `sdk-folder`.

Baixe a versão mais recente.

Use os links a seguir para baixar a versão mais recente.

- [Pacote completo do SDK do aplicativo distribuível](#)
- [Somente as bibliotecas do SDK do aplicativo](#)

Você também pode baixar o pacote distribuível completo do SDK do SimSpace Weaver aplicativo no [SimSpace Weaver console](#) no. AWS Management Console No painel de navegação, selecione Baixe o SDK do aplicativo.

Warning

Não use o AWS CLI para baixar nada que pareça ser o pacote distribuível do SDK do SimSpace Weaver aplicativo. Use somente os links de download nesta página ou o link de download no console. Qualquer outro método ou local de download não é compatível e pode conter código obsoleto, incorreto ou malicioso.

Solução de problemas de downloads do SDK do aplicativo

Usamos a Amazon CloudFront (CloudFront) para distribuir os arquivos.zip do SDK do aplicativo. Algumas das seguintes situações podem ocorrer.

- O pacote baixado não é a versão mais recente
 - Se o arquivo.zip que você baixou não contiver a versão mais recente, é possível que o cache no seu ponto de CloudFront presença ainda não tenha sido atualizado. Baixe novamente após 24 horas.
- Há um erro HTTP 4xx ou 5xx usando um link de download
 - Tente novamente após 24 horas. Se o mesmo erro ocorrer, use o link Feedback na parte inferior do [Console do SimSpace Weaver](#) para relatar o problema. Selecione Relatar um problema como Tipo de feedback.
- O navegador informa que não consegue carregar a página
 - Pode haver um problema de configuração da rede local ou do navegador. Verifique se pode carregar outras páginas. Limpe o cache do navegador e tente novamente. Verifique se há alguma regra de firewall que possa bloquear a URL de download.
- Há um erro ao tentar salvar o arquivo
 - Verifique as permissões do sistema de arquivos local para confirmar que tem as permissões corretas para salvar o arquivo.

- Seu navegador é exibido AccessDenied
 - Se você inseriu a URL manualmente no navegador, verifique se está escrita corretamente. Se você usou um link para download, verifique se algo não interferiu com a URL do seu navegador e use o link novamente.

Instalar a versão mais recente

Para instalar a versão mais recente

1. [Baixe a versão mais recente.](#)
2. Descompacte o SimSpaceWeaverAppSdkDistributable arquivo.zip em uma pasta.
3. Execute `python setup.py` a partir da pasta SDK do SimSpace Weaver aplicativo da versão mais recente descompactada.
4. Use a pasta SDK do SimSpace Weaver aplicativo da versão mais recente descompactada em vez da versão anterior.

Versões do serviço

Version (Versão)	Observações	Data de lançamento	Documentação	Baixar o SDK do aplicativo
1.17.0	<p>Principais mudanças no pacote distribuível do SDK do SimSpace Weaver aplicativo</p> <ul style="list-style-type: none"> • Substituímos os scripts em lote do Windows e do Linux Bash por scripts baseados em Python. Portanto, agora 	17 de abril de 2024	Este guia	<ul style="list-style-type: none"> • Pacote completo • Somente bibliotecas <p>Consulte Solução de problemas.</p>

Version (Versão)	Observações	Data de lançamento	Documentação	Baixar o SDK do aplicativo
	<p>é necessário usar os scripts e amostras do Python 3.9, mesmo que você não use (ou pretenda usar) o SDK do Python.</p> <ul style="list-style-type: none">• Esta versão aumenta o suporte para o Amazon Linux 2.• Corrigimos vários bugs no SimSpace Weaver Local. <p>Para obter mais informações, consulte as notas de lançamento.</p> <p>Correção de bugs</p> <ul style="list-style-type: none">• Corrigimos um bug que fazia com que entidades ficassem sem dono se não concluíssem a transferência			

Version (Versão)	Observações	Data de lançamento	Documentação	Baixar o SDK do aplicativo
	entre trabalhadores remotos.			
1.16.0	<p>Novo recurso</p> <ul style="list-style-type: none">• Agora você pode usar APIs de mensagens no SDK do SimSpace Weaver aplicativo para enviar e receber mensagens entre seus aplicativos. Esse recurso está disponível para C++, Python e as integrações Unity e Unreal Engine 5.	12 de fevereiro de 2024	Consulte o catálogo do AWS SimSpace Weaver guia .	<ul style="list-style-type: none">• Pacote completo• Somente bibliotecas <p>Consulte Solução de problemas.</p>

Version (Versão)	Observações	Data de lançamento	Documentação	Baixar o SDK do aplicativo
1.15.3	<p>Atualização de SimSpace Weaver Local:</p> <ul style="list-style-type: none">Alteramos SimSpace Weaver Local para alinhá-lo mais estreitamente com o desenvolvimento para a Nuvem AWS. Essas alterações afetam projetos e fluxos de trabalho em C++, Python, Unity e Unreal Engine para SimSpace Weaver Local.	4 de dezembro de 2023	Consulte o catálogo do AWS SimSpace Weaver guia .	Não disponível para download.

Version (Versão)	Observações	Data de lançamento	Documentação	Baixar o SDK do aplicativo
1.15.2	<p>Atualização do pacote distribuível do SDK do aplicativo:</p> <ul style="list-style-type: none">• Atualizamos o <code>Dockerfile</code> para usar a versão específica necessária do <code>cmake</code>. As compilações de contêineres do Docker podem falhar sem essa alteração.	2 de novembro de 2023	Consulte o catálogo do AWS SimSpace Weaver guia .	Não disponível para download.
1.15.1	<p>Atualização de recursos:</p> <ul style="list-style-type: none">• SDK do Python: esta versão corrige um problema que fazia com que as simulações baseadas em Python falhassem na Nuvem AWS.	22 de setembro de 2023	Consulte o catálogo do AWS SimSpace Weaver guia .	Não disponível para download.

Version (Versão)	Observações	Data de lançamento	Documentação	Baixar o SDK do aplicativo
1.15.0	<p>Novo recurso</p> <ul style="list-style-type: none">Python do SDK: agora é possível desenvolver simulações em Python. O pacote distribuível do SDK do aplicativo o SimSpace Weaver inclui um modelo para um projeto de exemplo em Python e seu cliente de visualização em Python.	31 de agosto de 2023	Consulte o catálogo do AWS SimSpace Weaver guia .	Não disponível para download.

Version (Versão)	Observações	Data de lançamento	Documentação	Baixar o SDK do aplicativo
1.14.0	<p>Novos atributos:</p> <ul style="list-style-type: none">• Contêineres personalizados: crie sua própria imagem de contêiner baseada no Amazon Linux 2 (AL2), armazene-a no Amazon Elastic Container Registry (Amazon ECR) e use-a para executar seus aplicativos SimSpace Weaver na Nuvem AWS.• Vários domínios espaciais: crie mais de um domínio espacial em uma simulação. Separe a lógica de simulação em vez de combinar tudo em um único aplicativo espacial. Aloque recursos	26 de julho de 2023	Consulte o catálogo do AWS SimSpace Weaver guia .	Não disponível para download.

Version (Versão)	Observações	Data de lançamento	Documentação	Baixar o SDK do aplicativo
	<p>diferentes para domínios espaciais com base em seus requisitos.</p> <ul style="list-style-type: none"> Taxa de atualização ilimitada: permita que a simulação seja executada tão rápido quanto seu código seja executado. Defina o relógio da simulação para que ele envie a próxima marca assim que todos os aplicativos concluírem suas operações de confirmação para a marca atual. <p>SimSpace Weaver SDK do aplicativo:</p> <ul style="list-style-type: none"> O valor da <code>tick_rate</code> agora é uma string. O valor deve incluir 			

Version (Versão)	Observações	Data de lançamento	Documentação	Baixar o SDK do aplicativo
	aspas duplas ("). A taxa de atualização das versões anteriores ainda é um número inteiro.			
1.13.1	<p>SimSpace Weaver SDK do aplicativo:</p> <ul style="list-style-type: none">• Atualização do recurso: a criação do projeto agora funciona corretamente com o modelo PathfindingSampleUnreal .	7 de junho de 2023	Consulte o catálogo do AWS SimSpace Weaver guia .	Não disponível para download.

Version (Versão)	Observações	Data de lançamento	Documentação	Baixar o SDK do aplicativo
1.13.0	<p>SimSpace Weaver APIs de serviço:</p> <ul style="list-style-type: none">• Nova ação CreateSnapshot• Altera a ação StartSimulation :<ul style="list-style-type: none">• Foi adicionado um parâmetro <code>SnapshotS3Location</code> para iniciar a partir de um snapshot.• O parâmetro <code>SchemaS3Location</code> agora é opcional.• Alterações na saída de DescribeSimulation :<ul style="list-style-type: none">• <code>SchemaError</code> está obsoleto.	29 de abril de 2023	Consulte o catálogo do AWS SimSpace Weaver guia.	Não disponível para download.

Version (Versão)	Observações	Data de lançamento	Documentação	Baixar o SDK do aplicativo
	<ul style="list-style-type: none"> • Adicionado o campo <code>StartError</code>. • Adicionado o campo <code>SnapshotS3Location</code>. • Adicionado o estado de simulação <code>SNAPSHOT_IN_PROGRESS</code>. • Novo tipo de dados S3Destination. <p>SimSpace Weaver console:</p> <ul style="list-style-type: none"> • Nova funcionalidade para criar snapshots. • Nova funcionalidade para iniciar uma simulação a partir de um snapshot. 			

Version (Versão)	Observações	Data de lançamento	Documentação	Baixar o SDK do aplicativo
	<p>SimSpace Weaver SDK do aplicativo:</p> <ul style="list-style-type: none"> • Novos scripts para oferecer suporte a snapshots • <code>create-snapshot- <i>project-name</i> .bat</code> • <code>start-from-snapshot- <i>project-name</i> .bat</code> • <code>quick-start-from-snapshot- <i>project-name</i> -cli.bat</code> • <code>list-snapshots- <i>project-name</i> .bat</code> • Os projetos agora usam um único bucket do Amazon S3 por projeto: <code>weaver-<i>lowercase-project-name</i> -<i>account-number</i> -<i>region</i></code> 			

Version (Versão)	Observações	Data de lançamento	Documentação	Baixar o SDK do aplicativo
1.12.3	<p>SimSpace Weaver SDK do aplicativo:</p> <ul style="list-style-type: none"> Os scripts a seguir agora oferecem suporte ao parâmetro <code>--maximum-duration</code> : <code>quick-start-<i>project-name</i> - cli.bat</code> <code>quick-start-<i>project-name</i> - cli.sh</code> <code>start-simulation-<i>project-name</i> .bat</code> <code>start-simulation-<i>project-name</i> .sh</code> <code>run-<i>project-name</i> .bat</code> <code>run-<i>project-name</i> .sh</code> 	27 de março de 2023	Consulte o catálogo do AWS SimSpace Weaver guia .	Não disponível para download.

Version (Versão)	Observações	Data de lançamento	Documentação	Baixar o SDK do aplicativo
1.12.2	SimSpace Weaver SDK do aplicativo: <ul style="list-style-type: none">• Correção de erro: o <code>docker-create-image.bat</code> agora funciona corretamente.	1 de março de 2023	Consulte o catálogo do AWS SimSpace Weaver guia .	Não disponível para download.

Version (Versão)	Observações	Data de lançamento	Documentação	Baixar o SDK do aplicativo
1.12.1	<p>SimSpace Weaver SDK do aplicativo:</p> <ul style="list-style-type: none">Os scripts agora aceitam um AWS CLI perfil para uso na AWS autenticação.Os scripts agora oferecem suporte AWS IAM Identity Center para AWS autenticação. <p>SimSpace Weaver Local:</p> <ul style="list-style-type: none">Correção de erro: o <code>Api::BeginUpdateWillBlock</code> agora retorna corretamente <code>true</code> se todos os aplicativos espaciais não tiverem entrado na simulação.	28 de fevereiro de 2023	Consulte o catálogo do AWS SimSpace Weaver guia .	Não disponível para download.

Version (Versão)	Observações	Data de lançamento	Documentação	Baixar o SDK do aplicativo
1.12.0	Lançamento da versão disponível ao público (GA)	29 de novembro de 2022	Consulte o catálogo do AWS SimSpace Weaver guia .	Não disponível para download.

AWS SimSpace Weaver versão 1.17.0

Esta versão é uma revisão do pacote distribuível do SDK do SimSpace Weaver aplicativo. Substituímos scripts desatualizados do Windows batch e Linux Bash por scripts baseados em Python.

Important

O Python 3.9 agora é um requisito para usar os scripts e amostras, não apenas para o SDK do Python.

Sumário

- [Principais mudanças para 1.17.0](#)
- [Atualizar um projeto para 1.17.0](#)
- [Perguntas frequentes sobre a versão 1.17.0](#)

Principais mudanças para 1.17.0

- Criação simplificada de projetos
 - Depois de executar `setup.py`, você pode criar seu próprio projeto simplesmente copiando e colando uma amostra.
- Amostras em 1 clique
 - O arquivo zip de distribuição agora contém ready-to-use amostras que funcionam após a configuração da distribuição.
- Cada SDK agora existe em seu próprio diretório: `cplusplus`, `unity`, e `unity`. Talvez seja necessário atualizar seus caminhos dependendo de qual SDK você usa.

- Melhorias nos scripts auxiliares.
 - Os scripts agora contêm várias AWS CLI opções para maximizar sua flexibilidade.
 - Lançamento e conexão integrados do cliente do console como parte do início rápido.
 - Saída de console aprimorada.
 - A criação de amostras do Unreal e do Unity agora funciona `quick-start`, sem a necessidade de mais etapas manuais.
 - SimSpace Weaver Local agora funciona apenas ligando `quick-start`, chega de construir e lançar manualmente.
 - SimSpace Weaver Local `quick-start` tem suporte integrado para registrar a saída do aplicativo.
 - SimSpace Weaver Local agora pode ser iniciado em um ambiente sem interface gráfica, como em uma sessão ssh.
 - O recurso de “contêiner personalizado” agora está integrado ao `quick-start` script.
 - Maior suporte ao Amazon Linux 2 (AL2): os fluxos de trabalho de scripts para Windows e AL2 agora são comparáveis. Anteriormente, os projetos de AL2 exigiam mais etapas manuais e SimSpace Weaver Local não eram compatíveis com o AL2.
- Os plug-ins Unreal Engine e Unity agora estão incluídos como parte do pacote distribuível do SDK do SimSpace Weaver aplicativo.
- Correções de bugs para SimSpace Weaver Local
 - Corrigido um bug em que entidades podiam ser atribuídas ao mesmo ID de entidade.
 - Corrigido um erro em que duas partições podiam ser atribuídas ao mesmo ID de partição.
 - Corrigido um bug relacionado à tentativa de aplicativos de gravar em entidades que não possuíam.
 - Resolveu um problema de vazamento de memória.

Atualizar um projeto para 1.17.0

1. Configure a distribuição 1.17.0: siga os procedimentos de configuração novamente porque os alteramos para a 1.17.0. Para ter mais informações, consulte [Configuração do SimSpace Weaver](#).
2. Cada SDK do aplicativo Weaver agora existe em seu próprio diretório. Atualize seus caminhos de construção para refletir isso.

- a. Diretório C++: `SimSpaceWeaverAppSdk/cpp`
 - O SDK do aplicativo SimSpace Weaver C++ agora usa um `FindSimSpaceWeaverAppSdk.cmake` arquivo. Esse arquivo configura um `weaver` destino que é vinculado e inclui correções de erros importantes ao criar para o Weaver no Nuvem AWS. Você deve usar isso em vez de vincular diretamente aos binários.
 - b. Diretório Python: `SimSpaceWeaverAppSdk/python`
 - c. Plug-in Unity: `SimSpaceWeaverAppSdk/unity`
 - d. Plugin Unreal Engine: `SimSpaceWeaverAppSdk/unreal`
3. Os `tools` scripts anteriores não funcionarão com a nova SimSpace Weaver distribuição. Para usar os novos `tools` scripts com seu projeto:
- a. Exclua seus `tools/local` diretórios e antigos `tools/windows`, `tools/linux`
 - b. Copie o `tools` diretório de um projeto de amostra que usa o mesmo SDK do SimSpace Weaver aplicativo do seu projeto. Certifique-se de ter executado `setup.py` antes de copiar esse diretório.

Important

É garantido que os scripts de ferramentas funcionem apenas com os projetos de amostra. Talvez seja necessário editar esses scripts, especialmente o `build.py` script, para trabalhar com seu projeto. Todas as edições serão exclusivas do seu projeto, portanto, não podemos fornecer nenhuma orientação.

Perguntas frequentes sobre a versão 1.17.0

Preciso atualizar para a versão 1.17.0?

Essa não é uma atualização obrigatória porque não há alterações na SimSpace Weaver API ou no SDK do SimSpace Weaver aplicativo. Você deve atualizar para 1.17.0 se quiser usar o 1.17.0 SimSpace Weaver Local, que contém várias correções de erros.

Qual é a versão mínima do Python necessária?

Python 3.9 é a versão mínima.

Qual é a versão mínima do CMake necessária?

A versão 3.13 do CMake é a mínima.

Qual é a versão mínima do Unreal Engine necessária?

O Unreal Engine 5.0 é o mínimo.

Qual é a versão mínima do Unity necessária?

A versão 2022.3.19.F1 do Unity é a mínima.

AWS SimSpace Weaver version 1.15.1

Essa versão é uma atualização necessária para o SDK do Python que foi originalmente lançado na versão 1.15.0 do SimSpace Weaver. Ela corrige um problema de incompatibilidade de versão que fazia com que as simulações baseadas em Python falhassem no Nuvem AWS. Use essa versão em vez da 1.15.0.

Atualizar um projeto Python existente para o 1.15.1

Se você já tiver um projeto Python criado com a versão 1.15.0 do SDK do Python, execute as etapas a seguir para atualizá-lo para a versão 1.15.1 para que ele possa ser executado no Nuvem AWS.

Em vez de seguir esse procedimento, você também pode criar um novo projeto em Python com o SDK 1.15.1 para Python e mover seu código personalizado para o novo projeto.

Como atualizar um projeto Python 1.15.0 para 1.15.1

1. Vá para a pasta do seu projeto em Python.
2. Na `src/PythonBubblesSample/bin/run-python`, altere a seguinte linha:

```
export PYTHONPATH=$PYTHONPATH:/roapp/lib
```

Para o seguinte:

```
export PYTHONPATH=$PYTHONPATH:$LD_LIBRARY_PATH:/roapp/lib
```

3. Em `CMakeLists.txt`, exclua as seguintes linhas:

- ```
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)
```

Solução de problemas para a versão 1.15.1

Depois de atualizar uma simulação do Python 1.15.0, ela falha ao iniciar no Nuvem AWS

Sintomas: após aproximadamente 5 a 10 minutos do início da simulação, o log de gerenciamento da simulação relata um `internal error` e o status da simulação é `FAILED`.

Isso pode acontecer se um arquivo de biblioteca do SDK do Python 1.15.0 for incluído em um arquivo `.zip` do aplicativo. Certifique-se de ter concluído as etapas para atualizar o projeto e veja se o `libweaver_app_sdk_python_v1.so` não está nos arquivos `.zip` ou referenciado de alguma forma.

Perguntas frequentes sobre a versão 1.15.1

Essa versão afeta algo além do SDK do Python?

Não.

Preciso atualizar para a versão 1.15.1?

Você não precisa atualizar para a versão 1.15.1 se não pretende usar o Python em seus aplicativos espaciais. Se você atualizou para 1.15.0, suas simulações baseadas em Python não serão executadas no Nuvem AWS. Recomendamos que você atualize para 1.15.1 se você usar o 1.15.0.

O que é o `$LD_LIBRARY_PATH`?

É a localização do SDK do Python quando sua simulação é executada no Nuvem AWS. Isso é novo para a 1.15.1. Fizemos essa alteração para evitar problemas com a versão do Python no futuro. A vinculação a esse diretório é funcionalmente igual à da versão 1.15.0 no `libweaver_app_sdk_python_v1.so`.

Histórico do documento para AWS SimSpace Weaver

A tabela a seguir descreve mudanças importantes na SimSpace Weaver documentação.

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
17 de abril de 2024	Conteúdo atualizado	Atualizado em todo o guia do usuário para a versão 1.17.0. Principais mudanças no Configuração capítulo e nos Conceitos básicos tutoriais. Consulte as notas de lançamento para obter mais informações.	N/D
12 de fevereiro de 2024	Conteúdo atualizado	Atualizou o AWS SimSpace Weaver versões capítulo para a versão 1.16.0.	N/D
12 de fevereiro de 2024	Novo conteúdo	A Sistema de mensagens seção foi adicionada como parte da versão 1.16.0. Esta seção descreve as APIs de mensagens adicionadas ao SDK do SimSpace Weaver aplicativo. Você pode usar essas APIs para enviar e receber mensagens entre seus aplicativos.	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
12 de fevereiro de 2024	Conteúdo atualizado	Atualizado o SimSpace Weaver referência do esquema de simulação capítulo para a versão 1.16.0.	N/D
12 de fevereiro de 2024	Conteúdo atualizado	Foram adicionadas cotas de serviço para mensagens ao SimSpace Terminais e cotas do Weaver capítulo.	N/D
12 de fevereiro de 2024	Novos guias	Divida o conteúdo das versões anteriores à 1.16.0 em um guia separado. Foi adicionado o Catálogo do AWS SimSpace Weaver Guia (disponível na página inicial da documentação principal) para acessar os guias das versões anteriores.	N/D
4 de dezembro de 2023	Conteúdo atualizado	Atualização do capítulo AWS SimSpace Weaver versões para o lançamento da versão 1.15.3.	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
4 de dezembro de 2023	Conteúdo atualizado	Atualização do capítulo AWS SimSpace Weaver versões para incluir instruções de instalação para a versão mais recente.	N/D
4 de dezembro de 2023	Conteúdo atualizado	Atualização do Cotas de serviço para o SimSpace Weaver Local .	N/D
4 de dezembro de 2023	Conteúdo novo e atualizado	Reestruturação da seção Desenvolvimento local em SimSpace Weaver e adição de uma nova página que descreve as diferenças do SimSpace Weaver Local introduzidas na versão 1.15.3.	N/D
7 de novembro de 2023	Conteúdo atualizado	As instruções de configuração do Docker e do WSL foram atualizadas para usar o link/URL de download direto para o SDK do aplicativo. Para ter mais informações, consulte Configurar o ambiente local para o SimSpace Weaver .	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
2 de novembro de 2023	Conteúdo atualizado	Atualizou a página de versões do serviço para a versão 1.15.2. Para ter mais informações, consulte Versões do serviço .	N/D
23 de outubro de 2023	Conteúdo atualizado	Atualizou a página de versões do serviço com novas instruções para baixar o pacote distribuível do SDK do aplicativo. Agora, os clientes devem usar apenas um dos nossos links de download direto aprovados e não usar o AWS CLI para baixar o pacote distribuível do SDK do aplicativo. Para ter mais informações, consulte Baixe a versão mais recente .	N/D
22 de setembro de 2023	Novo conteúdo	Foi adicionada uma página de notas da versão com instruções de atualização para a versão 1.15.1. Para ter mais informações, consulte AWS SimSpace Weaver version 1.15.1 .	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
10 de setembro de 2023	Novo conteúdo	Foi adicionada uma seção de solução de problemas para situações em que AWS CLI não reconhece SimSpace Weaver. Para ter mais informações, consulte AWS CLI não reconhece o simspaceweaver .	N/D
10 de setembro de 2023	Conteúdo atualizado	Instruções de instalação atualizadas para o AWS CLI no WSL. Para ter mais informações, consulte Configure o pacote SimSpace Weaver de distribuição para Amazon Linux 2 (AL2) em Windows Subsystem for Linux (WSL) .	N/D
7 de setembro de 2023	Atualização da API	BucketName e agora ObjectKey são necessários para o tipo de dados S3Location . BucketName e agora é necessário para o tipo de dados S3Destination .	AWS SDK: 2023-09-07
31 de agosto de 2023	Novo conteúdo	Foi adicionada uma nova seção para a versão 1.15.0: Como trabalhar com Python .	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
15 de agosto de 2023	Conteúdo atualizado	Instruções de download atualizadas no AWS SimSpace Weaver versões para listar somente os buckets oficiais do SimSpace Weaver no Amazon S3. Outros locais de download não são controlados AWS e podem conter códigos maliciosos.	N/D
26 de julho de 2023	Conteúdo atualizado	Atualizado Relógio .	N/D
26 de julho de 2023	Conteúdo atualizado	Atualizado Configurar domínios espaciais .	N/D
26 de julho de 2023	Novo conteúdo	Adição de uma nova seção: Contêineres personalizados .	N/D
26 de julho de 2023	Conteúdo atualizado	Atualização do AWS SimSpace Weaver versões para a versão 1.14.0.	N/D
6 de julho de 2023	Novo conteúdo	Adição de uma nova seção: Pathfindi ngSample o cliente do console não consegue se conectar .	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
7 de junho de 2023	Conteúdo atualizado	Atualização do AWS SimSpace Weaver versões para a versão 1.13.1.	N/D
15 de maio de 2023	Novo conteúdo	Adição de uma nova seção: Usando snapshots com o AWS CloudFormation .	N/D
29 de abril de 2023	Novo conteúdo	Conteúdo adicionado para a versão 1.13.0. Para ter mais informações, consulte AWS SimSpace Weaver versões .	AWS SDK: 2023-04-28
27 de março de 2023	Novo conteúdo	Foi adicionada uma seção que descreve a duração máxima das simulações. Foram adicionadas notas nos tutoriais da versão 1.12.3, que agregaram suporte ao parâmetro <code>--maximum-duration</code> aos scripts do SDK do aplicativo SimSpace Weaver .	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
9 de março de 2023	Conteúdo alterado	Esclarecido que fornecemos apenas instruções para o Docker no Windows e no Windows Subsystem for Linux (WSL), e que o WSL (e qualquer outro ambiente Linux) não é compatível.	N/D
28 de fevereiro de 2023	Novo conteúdo	Foi adicionado um capítulo que descreve SimSpace Weaver as versões.	N/D
28 de fevereiro de 2023	Conteúdo alterado	Conteúdo alterado sobre autenticação para incluir o uso AWS IAM Identity Center e os perfis nomeados para o AWS Command Line Interface (AWS CLI).	N/D
17 de fevereiro de 2023	Novo conteúdo	Foi adicionada uma seção sobre como gerenciar seus recursos com AWS CloudFormation.	N/D
23 de janeiro de 2023	Novo conteúdo	Instruções adicionadas para depurar simulações locais.	N/D

Data	Alteração	Atualizações feitas na documentação	Versões de API atualizadas
29 de novembro de 2022	Inicialização do serviço	O Guia do usuário e a Referência de API foram lançados para o SimSpace Weaver.	AWS SDK: 2022-11-29

Glossário

Este glossário define termos que são específicos de AWS SimSpace Weaver.

Para obter a AWS terminologia mais recente, consulte o [AWS glossário](#) na Referência AWS Geral.

A

Aplicativo	Código executável (também chamado de binários) que você cria. O termo aplicativo pode se referir ao código ou a uma instância em execução desse código. Um aplicativo encapsula o comportamento da simulação. Os aplicativos criam, excluem, leem e atualizam entidades .
SDK do aplicativo	Um kit de desenvolvimento de software (SDK) que você usa para integrar um aplicativo no SimSpace Weaver. O SDK fornece APIs para ler e gravar dados de entidades e monitorar o tempo de simulação. Para ter mais informações, consulte Aplicativo do SDK do SimSpace Weaver .

C

Cliente	Processos (ou suas definições) que existem fora da simulação SimSpace Weaver e interagem com ela por meio de um aplicativo personalizado ou aplicativo de serviço . Você pode usar um cliente para visualizar ou alterar o estado da simulação.
Relógio	Uma abstração dos processos internos SimSpace Weaver de agendamento. O relógio publica marcações nos aplicativos para manter a sincronização do tempo. Cada simulação tem seu próprio relógio.
Taxa de clock	O número de marcações por segundo que o relógio publica nos aplicativos . Para mais informações sobre taxas de clock compatíveis, consulte SimSpace Terminais e cotas do Weaver .
Taxa de atualização do relógio	Consulte taxa de clock .
Unidade de recursos computacionais	Uma unidade de recursos computacionais (processador e memória) em um operador . Normalmente, uma única instância de um aplicativo recebe

uma unidade de recurso computacional. Você pode alocar mais de uma unidade de recursos computacionais para cada aplicativo.

Aplicativo
personalizado

Um tipo de [aplicativo](#) usado para ler e interagir com o estado da simulação. Aplicativos personalizados podem criar entidades na simulação, mas não são proprietários delas. Quando um aplicativo personalizado cria uma entidade, ele precisa transferir a entidade para o [domínio espacial](#). Você controla o ciclo de vida de um aplicativo personalizado usando as APIs do aplicativo. Para obter mais informações sobre as SimSpace Weaver APIs, consulte [SimSpace Weaver Referências de API](#).

Domínio
personalizado

Um [domínio](#) que contém [aplicativos personalizados](#).

Partição personalizada

A [partição](#) de um [aplicativo personalizado](#).

D

Prazo

Um [horário real](#) em que uma operação (como o processamento de uma [marcação](#)) deve ser concluída.

Domínio

Um grupo de instâncias do [aplicativo](#) que executa o mesmo código executável (binário do aplicativo) e tem as mesmas opções de lançamento.

E

Endpoint (serviço)

Um nome de domínio totalmente qualificado (FQDN) que programas (como o AWS Command Line Interface) usam para se conectar ao serviço. SimSpace Weaver

Endpoint (simulação)

Um endereço IP e número de porta que os clientes usam para se conectar a uma simulação. Você pode configurar endpoints em [aplicativos personalizados](#) e [aplicativos de serviço](#).

Entidade

Objetos de dados do cliente (ou suas definições). As entidades podem ser estáticas (permanecer em um local) ou dinâmicas (mover-se pelo espaço de simulação). Por exemplo, pessoas e edifícios em uma simulação.

I

Índice (simulação) Uma descrição das propriedades espaciais de uma simulação, incluindo os limites espaciais e sistema de coordenadas.

L

Ciclo de vida (de um aplicativo) Uma descrição das etapas lógicas pelas quais um [aplicativo](#) deve passar durante uma simulação. Os ciclos de vida são gerenciados (SimSpace Weaver inicia e interrompe o aplicativo) ou não gerenciados (você inicia e interrompe o aplicativo).

Carregar (dados do campo da entidade) Leia os dados do campo da [entidade](#) do [State Fabric](#).

P

Partição Um segmento de memória compartilhada em um [operador](#). Cada partição contém um subconjunto discreto de [entidades](#) dentro de um [domínio](#). Cada [aplicativo](#) tem uma partição atribuída. Um aplicativo possui todas as entidades em sua partição. Quando um aplicativo cria uma entidade, ele a cria em sua partição. Quando as entidades se movem de uma partição para outra, a propriedade é transferida do aplicativo da partição de origem para o aplicativo da partição de destino.

R

Unidade de recurso Consulte [???](#).

S

Esquema Um documento YAML ou JSON que descreve a configuração de uma simulação. O SimSpace Weaver usa um esquema para criar um [recurso de simulação](#).

Aplicativo de serviço Um tipo de [aplicativo](#) usado para ler e interagir com o estado da simulação. Os aplicativos de serviço podem criar entidades na simulação,

mas devem transferi-las para o [domínio espacial](#). O SimSpace Weaver gerencia o [ciclo de vida](#) de um aplicativo de serviço e inicia um (ou mais, conforme especificado no [esquema](#)) em cada [operador](#) na simulação.

Domínio de serviço	Um domínio que contém aplicativos de serviço .
Partição de serviço	A partição de um aplicativo de serviço .
Simulação (recurso)	A abstração de um cluster de computação que executa um espaço virtual simulado. É possível ter vários tipos de simulação. Uma simulação é configurada usando um esquema .
Aplicativo espacial	Um tipo de aplicativo que encapsula a lógica central da simulação. Cada aplicativo espacial possui uma partição (somente uma).
Domínio espacial	Um domínio que contém aplicativos espaciais .
Partição espacial	A partição de um aplicativo espacial .
State Fabric	SimSpace Weaver Banco de dados em memória. Ele State Fabric armazena o estado das simulações, incluindo entidades e SimSpace Weaver dados internos.
Armazenamento (dados do campo da entidade)	Grava os dados do campo da entidade no State Fabric .
Assinatura	Uma solicitação de longa duração para que uma instância específica do aplicativo receba dados de uma área de assinatura . O aplicativo de assinatura usa uma assinatura para descobrir alterações nas entidades dentro da área de assinatura.
Área de assinatura	Uma região bidimensional do espaço de simulação. Uma assinatura se refere a uma área de assinatura. Uma área de assinatura pode abranger mais de uma partição e também incluir partes de partições. Uma área de assinatura é contínua dentro de seus limites definidos.

T

Marcação	Um valor discreto para o tempo (hora do relógio ou hora da simulação). Os aplicativos podem iterar mais rápido do que a duração da marcação,
----------	--

mas espera-se que gravem marcações específicas dentro de prazos específicos. Todas as operações de todos os aplicativos de uma determinada marcação devem ser concluídas antes que a próxima marcação possa começar.

Taxa de atualização	Consulte taxa de clock.
Hora (real)	O tempo atual do ponto de vista da realidade. SimSpace Weaver usa um timestamp POSIX de 64 bits, que é o número de nanossegundos desde a época Unix (January 1, 1970, 00:00:00 UTC).
Hora (simulação)	A hora atual do ponto de vista da simulação. SimSpace Weaver usa um contador lógico de números inteiros de 64 bits, que pode não corresponder diretamente à hora real.

W

Operador	Uma instância do Amazon Elastic Compute Cloud (Amazon EC2) que executa o código da simulação.
----------	---

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.