Guida dell'utente per la versione 1.17.0

AWS SimSpace Weaver

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

AWS SimSpace Weaver: Guida dell'utente per la versione 1.17.0

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

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

Table of Contents

..... ix
Che cos'è SimSpace Weaver? 1
Concetti chiave 1
Come funziona SimSpace Weaver 2
Come si usa SimSpace Weaver 5
Schema di simulazione 6
Lavoratori e unità di risorse 6
Orologio di simulazione 6
Partizioni 7
Struttura statale 7
Entità 8
App 8
Casi d'uso di esempio 11
AWS SimSpace Weaver fine del supporto 12
Configurazione 13
Configura il tuo account 13
Iscriviti per un Account AWS 13
Crea un utente con accesso amministrativo 14
Aggiungi le autorizzazioni da utilizzare SimSpace Weaver 15
Configura il tuo ambiente locale 17
AL2 nel Docker 17
AL2 nel WSL 19
Uso di software concesso in licenza 23
Nozioni di base 24
Tutorial di avvio rapido 24
Passaggio 1: abilitare la registrazione (opzionale) 25
Passaggio 2: avvio rapido con il client della console (opzione 1) 25
Passo 2: Avvio rapido con il client Unreal Engine (opzione 2) 26
Interrompi ed elimina la simulazione 26
Risoluzione dei problemi 26
Tutorial dettagliato 27
Passaggio 1: abilitare la registrazione (opzionale) 28
Fase 2: Inizia la simulazione 28
Fase 3: Controlla i log (opzionale) 35

Fase 4: Visualizza la simulazione ... 37
Passaggio 5: interrompi ed elimina la simulazione ... 38
Risoluzione dei problemi ... 39
Lavorare con SimSpace Weaver ... 40
Configurazione della simulazione ... 40
Parametri di configurazione della simulazione ... 41
Versione SDK ... 42
Proprietà di simulazione ... 42
Worker ... 43
Orologio ... 44
Strategie di partizionamento ... 47
Domini ... 48
Durata massima ... 58
Valore massimo ... 58
Valore predefinito ... 58
Valore minimo ... 58
Avvio di una simulazione tramite la console ... 59
Lo stato di una simulazione che raggiunge la sua durata massima ... 59
Sviluppo di app ... 59
App spaziali ... 60
App personalizzate ... 60
Sviluppo di applicazioni client ... 61
Ottieni l'indirizzo IP e il numero di porta ... 63
Avvio del client di visualizzazione Unreal Engine ... 66
Risoluzione dei problemi ... 66
Sviluppo locale ... 67
Fase 1: Avvia la simulazione locale ... 68
Fase 2: Visualizza la simulazione locale ... 69
Passaggio 3: Interrompi la simulazione locale (opzionale su Windows) ... 70
Risoluzione dei problemi relativi allo sviluppo locale ... 71
SimSpace Weaver app SDK ... 71
I metodi API restituiscono un Result ... 72
Interazione con l'SDK dell'app al livello più alto ... 73
Gestione della simulazione ... 73
Sottoscrizioni ... 76
Entità ... 77

eventi dell'entità ... 89
Result e gestione degli errori ... 96
Generici e tipi di dominio ... 98
Operazioni varie dell'SDK dell'app ... 98
SimSpace Weaver quadro dimostrativo ... 101
Lavorare con le quote ... 102
Ottieni i limiti per un'app ... 102
Ottieni la quantità di risorse utilizzate da un'app ... 103
Reimposta le metriche ... 104
Superamento di un limite ... 104
Memoria a corto di memoria ... 105
Best practice ... 105
Simulazioni di debug ... 105
Utilizzo SimSpace Weaver Local e guarda l'output della console ... 106
Guarda i tuoi log in Amazon CloudWatch Logs ... 106
Utilizzo describe Chiamate API ... 106
Connect a client ... 107
Eseguire il debug di simulazioni locali ... 107
Contenitori personalizzati ... 108
Crea un contenitore personalizzato ... 109
Modifica un progetto per utilizzare un contenitore personalizzato ... 110
Domande frequenti ... 113
Risoluzione dei problemi ... 113
Uso di Python ... 114
Creare un progetto Python ... 115
Avvio di una simulazione in Python ... 116
Il client Python di esempio ... 117
Domande frequenti ... 118
Risoluzione dei problemi ... 118
Support per altri motori ... 119
Unità ... 120
Unreal Engine ... 120
Uso di software concesso in licenza ... 120
Gestire le risorse con AWS CloudFormation ... 120
Snapshot ... 123
Snapshot ... 124

Casi d'uso per le istantanee ... 124
SimSpace Weaver Console ... 125
AWS CLI ... 127
Domande frequenti ... 129
Messaggistica ... 130
Casi d'uso per la messaggistica ... 130
Utilizzo della messaggistica APIs ... 131
Quando usare la messaggistica ... 138
Suggerimenti per l'utilizzo della messaggistica ... 142
Errori di messaggistica e risoluzione dei problemi ... 143
Best practice ... 146
Imposta gli allarmi di fatturazione ... 146
Utilizzare SimSpace Weaver Local ... 146
Interrompi le simulazioni che non ti servono ... 147
Elimina le risorse che non ti servono ... 147
Avere dei backup ... 147
Sicurezza ... 149
Protezione dei dati ... 150
Crittografia dei dati a riposo ... 151
Crittografia in transito ... 151
Riservatezza del traffico Internet ... 152
Identity and Access Management ... 152
Destinatari ... 152
Autenticazione con identità ... 153
Gestione dell'accesso con policy ... 157
Come AWS SimSpace Weaver funziona con IAM ... 159
Esempi di policy basate su identità ... 166
Autorizzazioni create per te SimSpace Weaver ... 170
Prevenzione del problema "confused deputy" tra servizi ... 172
Risoluzione dei problemi ... 175
Registrazione e monitoraggio degli eventi di sicurezza ... 178
Convalida della conformità ... 179
Resilienza ... 180
Sicurezza dell'infrastruttura ... 180
Modello di sicurezza della connettività di rete ... 181
Analisi della configurazione e delle vulnerabilità ... 181

Best practice di sicurezza ... 182
Crittografa le comunicazioni tra le tue app e i relativi client ... 182
Effettua periodicamente il backup dello stato della simulazione ... 183
Gestisci le tue app e SDKs ... 183
Registrazione di log e monitoraggio ... 184
Effettua il login CloudWatch ... 184
Accedere ai tuoi registri SimSpace Weaver ... 184
SimSpace Weaver registri ... 185
Monitoraggio con CloudWatch ... 187
SimSpace Weaver metriche a livello di account ... 188
CloudTrail registri ... 188
SimSpace Weaver informazioni in CloudTrail ... 188
Comprensione delle SimSpace Weaver voci dei file di registro ... 189
Endpoint e quote di servizio ... 192
Endpoint del servizio ... 192
Quote del servizio ... 193
Quote di messaggistica ... 196
Frequenze di clock ... 196
Quote di servizio per SimSpace Weaver Local ... 196
Risoluzione dei problemi ... 198
AssumeRoleAccessDenied ... 198
InvalidBucketName ... 200
ServiceQuotaExceededException ... 201
TooManyBuckets ... 201
Autorizzazione negata durante l'avvio della simulazione ... 202
Problemi legati al tempo di utilizzo Docker ... 202
Il client della console non riesce a connettersi ... 203
No `simspaceweaver` nel AWS CLI ... 205
riferimento allo schema ... 206
Esempio di schema completo ... 206
Formato dello schema ... 208
Versione SDK ... 209
Proprietà di simulazione ... 209
Worker ... 211
Orologio ... 212
Strategie di partizionamento ... 212

Domini ... 214
Vincoli di posizionamento ... 224
Riferimenti API ... 226
SimSpace Weaver versioni ... 227
Versione più recente ... 227
Come trovare la tua versione attuale ... 227
Scarica la versione più recente ... 227
Risoluzione dei problemi relativi ai download dell'SDK delle app ... 228
Installa la versione più recente ... 229
Versioni del servizio ... 229
1.17.0 ... 244
Principali modifiche per 1.17.0 ... 244
Aggiorna un progetto alla versione 1.17.0 ... 245
Domande frequenti sulla versione 1.17.0 ... 246
1.15.1 ... 247
Aggiorna un progetto Python esistente a 1.15.1 ... 247
Risoluzione dei problemi per la versione 1.15.1 ... 248
Domande frequenti sulla versione 1.15.1 ... 248
Cronologia dei documenti ... 249
Glossario ... 258

Avviso di fine del supporto: il 20 maggio 2026, AWS terminerà il supporto per AWS SimSpace Weaver. Dopo il 20 maggio 2026, non potrai più accedere alla SimSpace Weaver console o SimSpace Weaver alle risorse. Per ulteriori informazioni, consulta AWS SimSpace Weaver Fine del supporto.

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.

Che cos'è AWS SimSpace Weaver?

AWS SimSpace Weaver è un servizio che è possibile utilizzare per creare ed eseguire simulazioni spaziali su larga scala in. Cloud AWS Ad esempio, puoi creare simulazioni di folla, grandi ambienti del mondo reale ed esperienze coinvolgenti e interattive.

Con SimSpace Weaver, puoi distribuire carichi di lavoro di simulazione su più istanze Amazon Elastic Compute Cloud (Amazon EC2). SimSpace Weaver implementa l' AWS infrastruttura sottostante per te e gestisce la gestione dei dati di simulazione e la comunicazione di rete tra le EC2 istanze Amazon che eseguono la simulazione.

Concetti chiave per SimSpace Weaver

Una simulazione o un gioco è limitato dal computer che lo esegue. Con l'aumentare delle dimensioni e della complessità del mondo virtuale, le prestazioni di elaborazione iniziano a peggiorare. I calcoli richiedono più tempo, i sistemi esauriscono la memoria e i frame rate dei client diminuiscono. Per le simulazioni che non richiedono prestazioni in tempo reale, questo potrebbe essere solo fastidioso. Oppure, potrebbe trattarsi di una situazione aziendale critica in cui l'aumento dei ritardi di elaborazione si traduce in un aumento dei costi. Se la simulazione o il gioco richiedono prestazioni in tempo reale, il degrado delle prestazioni è sicuramente un problema.

Una soluzione comune per una simulazione che raggiunge un limite di prestazioni è semplificare la simulazione. I giochi online con molti utenti spesso risolvono i problemi di scalabilità creando copie del loro mondo virtuale su server diversi e distribuendo gli utenti su di essi.

SimSpace Weaver risolve il problema di scalabilità dividendo spazialmente il mondo virtuale e distribuendo i pezzi su un cluster di istanze di calcolo eseguite in. Cloud AWS Le istanze di calcolo collaborano per elaborare l'intero mondo della simulazione in parallelo. Il tuo mondo di simulazione appare come un unico spazio integrato per tutto ciò che contiene e per tutti i client che vi si connettono. Non è più necessario semplificare una simulazione a causa di un limite di prestazioni hardware. Puoi invece aggiungere più capacità di elaborazione nel cloud.

Argomenti

- Come funziona SimSpace Weaver
- Come si usa SimSpace Weaver
- Schema di simulazione

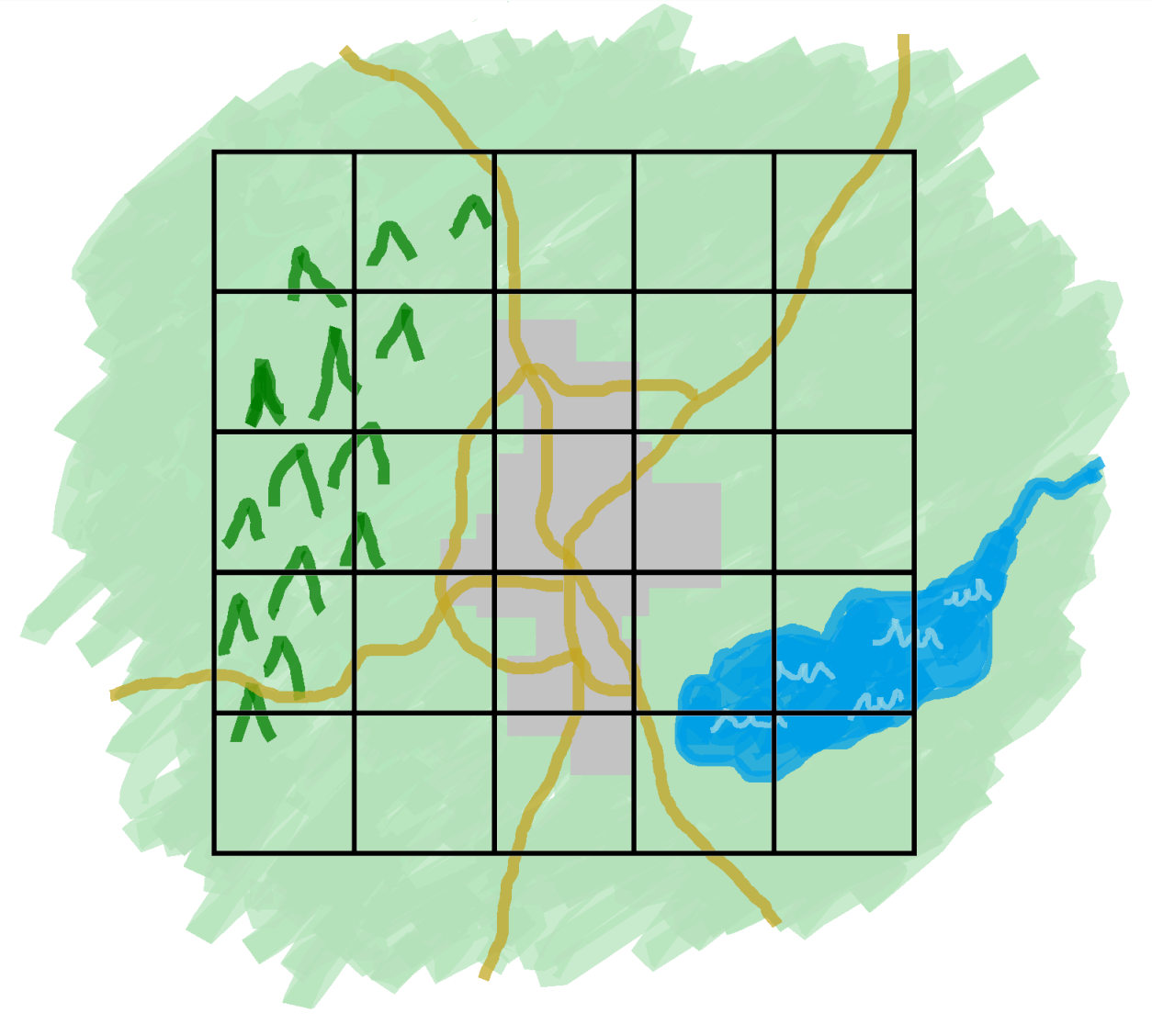
- Lavoratori e unità di risorse
- Orologio di simulazione
- Partizioni
- Struttura statale
- Entità
- App

Come funziona SimSpace Weaver

La tua simulazione consiste in un mondo con oggetti al suo interno. Alcuni oggetti (come persone e veicoli) si muovono e fanno cose. Altri oggetti (come alberi ed edifici) sono statici. In SimSpace Weaver, un'entità è un oggetto nel tuo mondo di simulazione.

Definisci i confini del tuo mondo di simulazione e lo dividi in una griglia. Invece di creare una logica di simulazione che funzioni sull'intera griglia, crei una logica di simulazione che opera su una cella della griglia. In SimSpace Weaver effetti, un'app spaziale è un programma scritto dall'utente che implementa la logica di simulazione per una cella della griglia. Ciò include la logica per tutte le entità in quella cella. L'area di proprietà di un'app spaziale è la cella della griglia controllata dall'app spaziale.

> **Note**
>
> In SimSpace Weaver, il termine «app» può riferirsi al codice di un'app o a un'istanza in esecuzione di quel codice.



Il tuo mondo di simulazione diviso in una griglia

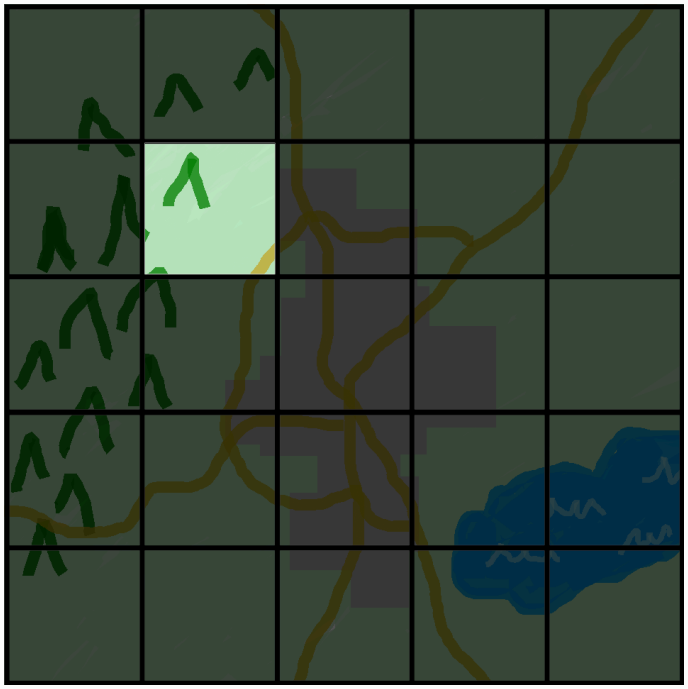
Dividi il tuo mondo di simulazione in una griglia. Ogni app spaziale implementa la logica di simulazione per una singola cella in quella griglia.

SimSpace Weaver esegue un'istanza del codice dell'app spaziale per ogni cella della griglia. Tutte le istanze dell'app spaziale vengono eseguite in parallelo. In sostanza, SimSpace Weaver divide la simulazione complessiva in più simulazioni più piccole. Ciascuna delle simulazioni più piccole gestisce una parte del mondo complessivo della simulazione. SimSpace Weaver può distribuire ed eseguire queste simulazioni più piccole su più istanze Amazon Elastic Compute Cloud (Amazon EC2) (chiamate worker) in. Cloud AWS Un singolo lavoratore può eseguire più app spaziali.

Le entità possono muoversi nel mondo della simulazione. Se un'entità entra nell'area di proprietà di un'altra app spaziale (un'altra cella nella griglia), il proprietario dell'app spaziale della nuova area assume il controllo dell'entità. Se la simulazione viene eseguita su più worker, un'entità potrebbe passare dal controllo di un'app spaziale su un worker a un'app spaziale su un altro worker. Quando un'entità passa a un altro worker, SimSpace Weaver gestisce la comunicazione di rete sottostante.

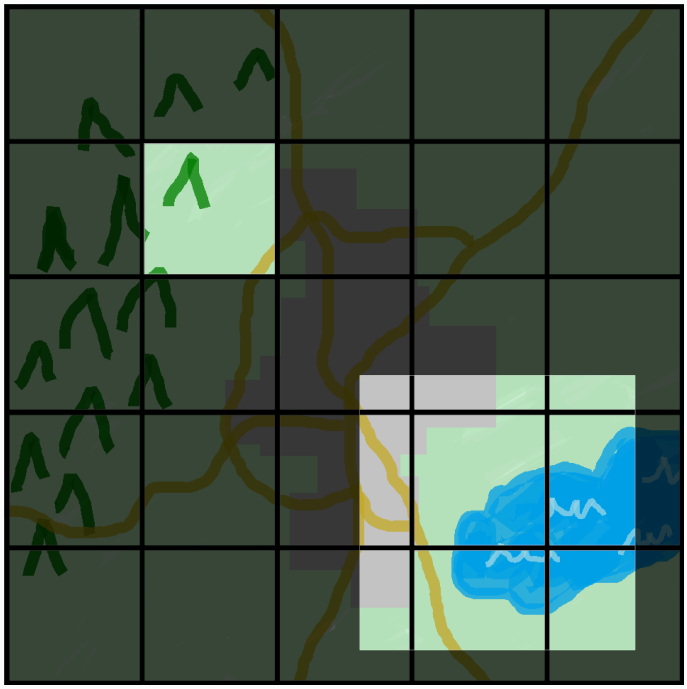
Sottoscrizioni

La visione del mondo di un'app spaziale è di sua proprietà. Per scoprire cosa sta succedendo in un'altra parte del mondo della simulazione, l'app spaziale crea un abbonamento. L'area di abbonamento è un sottoinsieme dell'area globale del mondo della simulazione. Un'area di abbonamento può includere parti di più aree di proprietà, inclusa l'area di proprietà dell'app spaziale. SimSpace Weaver notifica all'app spatial tutti gli eventi di entità (ad esempio, ingresso, uscita, creazione, aggiornamento ed eliminazione) che si verificano all'interno dell'area di sottoscrizione.



La visione del mondo offerta da un'app spaziale

La visione del mondo di un'app spaziale è la sua area di proprietà, che è costituit a da una cella nella griglia mondiale.



Visualizzazione di un'app spaziale con un'area di abbonamento aggiunta

Un'app spaziale utilizza un abbonamen to per scoprire cosa sta succedendo in un'altra parte del mondo della simulazio

	ne. L'area di sottoscrizione può contenere più celle della griglia e parti di celle.

Ad esempio, un'app che simula entità che interagiscono fisicamente potrebbe aver bisogno di conoscere le entità che si trovano appena oltre i confini spaziali della sua area di proprietà. A tale scopo, l'app può iscriversi alle aree che confinano con la sua area di proprietà. Dopo aver creato l'abbonamento, l'app riceve notifiche sugli eventi delle entità in quelle aree ed è in grado di leggere le entità. Un altro esempio è un veicolo autonomo che deve vedere tutte le entità a 200 metri di distanza, indipendentemente dall'app proprietaria dell'area. L'app per il veicolo può creare un abbonamento con un filtro come riquadro di delimitazione allineato all'asse (AABB) che copre l'area visualizzabile.

È possibile creare una logica di simulazione che non sia responsabile della gestione degli aspetti spaziali della simulazione. Un'app personalizzata è un programma eseguibile che viene eseguito su un singolo lavoratore. Sei tu a controllare il ciclo di vita (avvio e arresto) di un'app personalizzata. I client di simulazione possono connettersi a un'app personalizzata per visualizzare o interagire con la simulazione. Puoi anche creare un'app di servizio che funzioni su ogni lavoratore. SimSpace Weaver avvia un'istanza della tua app di servizio su ogni lavoratore che esegue la simulazione.

Le app personalizzate e le app di servizio creano abbonamenti per conoscere gli eventi delle entità e leggere le entità. Queste app non dispongono di aree di proprietà perché non sono spaziali. L'utilizzo di un abbonamento è l'unico modo per scoprire cosa sta succedendo nel mondo della simulazione.

Come si usa SimSpace Weaver

Quando si utilizza SimSpace Weaver, questi sono i passaggi principali da seguire:

1. Scrivi e costruisci C++ app che integrano l' SimSpace Weaver app SDK.
   a. Le tue app effettuano chiamate API per interagire con lo stato di simulazione.
2. Scrivi a clienti che visualizzano e interagiscono con la tua simulazione tramite alcune app.
3. Configura la simulazione in un file di testo.
4. Carica i pacchetti dell'app e la configurazione della simulazione sul servizio.
5. Inizia la simulazione.
6. Avvia e interrompi le tue app personalizzate in base alle tue esigenze.
7. Connect i client alle app personalizzate o di servizio per visualizzare o interagire con la simulazione.

8. Controlla i log delle simulazioni in Amazon CloudWatch Logs.
9. Interrompi la simulazione.
10. Ripulisci la simulazione.

Schema di simulazione

Lo schema (o schema) di simulazione è un YAML-file di testo in formato che contiene informazioni di configurazione per la simulazione. SimSpace Weaver utilizza lo schema quando avvia una simulazione. Il pacchetto distribuibile SimSpace Weaver dell'app SDK include uno schema per un progetto di esempio. Puoi usarlo come punto di partenza per il tuo schema. Per ulteriori informazioni sullo schema di simulazione, vedereSimSpace Weaver riferimento allo schema di simulazione.

Lavoratori e unità di risorse

Un worker è un' EC2 istanza Amazon che esegue la tua simulazione. Specificate un tipo di lavoratore nello schema di simulazione. SimSpace Weaver mappa il tipo di lavoratore su un tipo di EC2 istanza Amazon specifico utilizzato dal servizio. SimSpace Weaver avvia e arresta i tuoi lavoratori per te e gestisce le comunicazioni di rete tra i lavoratori. SimSpace Weaver avvia un set di lavoratori per ogni simulazione. Simulazioni diverse utilizzano lavoratori diversi.

La capacità di calcolo (processore e memoria) disponibile su un worker è suddivisa in unità logiche denominate unità di risorse di calcolo (o unità di risorse). Un'unità di risorsa rappresenta una quantità fissa di capacità di processore e memoria.

> **Note**
>
> In precedenza ci riferivamo a un'unità di risorse di calcolo come a uno slot. Potresti ancora vedere questo termine precedente nella nostra documentazione.

Orologio di simulazione

Ogni simulazione ha il suo orologio. È possibile avviare e arrestare l'orologio utilizzando le chiamate API o la SimSpace Weaver console. La simulazione si aggiorna solo quando l'orologio è in esecuzione. Tutte le operazioni della simulazione avvengono all'interno di segmenti temporali denominati tick. L'orologio comunica l'ora di inizio di ogni segno di spunta a tutti i lavoratori.

La frequenza (o tick-rate) è il numero di tick al secondo (hertz o Hz) che l'orologio annuncia. La frequenza di clock desiderata per una simulazione fa parte dello schema di simulazione. Tutte le operazioni relative a un segno di spunta devono essere completate prima dell'inizio del segno di spunta successivo. Per questo motivo, la frequenza di clock effettiva può essere inferiore alla frequenza di clock desiderata. La frequenza di clock effettiva non sarà superiore alla frequenza di clock desiderata.

Partizioni

Una partizione è un segmento della memoria condivisa di un worker. Ogni partizione contiene parte dei dati sullo stato della simulazione.

Una partizione per un'app spaziale (chiamata anche partizione spaziale dell'app o partizione spaziale) contiene tutte le entità nell'area di proprietà di un'app spaziale. SimSpace Weaver inserisce le entità nelle partizioni spaziali dell'app in base alla posizione spaziale di ciascuna entità. Ciò significa che SimSpace Weaver cerca di collocare entità spazialmente vicine l'una all'altra sullo stesso worker. Ciò riduce al minimo la quantità di conoscenza che un'app richiede alle entità di cui non è proprietaria per simulare le entità di cui è proprietaria.

Struttura statale

State Fabric è il sistema di memoria condivisa (la raccolta di tutte le partizioni) su tutti i lavoratori. Contiene tutti i dati di stato per la simulazione.

State Fabric utilizza un formato binario personalizzato che descrive un'entità come un insieme di dati iniziali e un registro di aggiornamento, per ogni campo di dati di quell'entità. Con questo formato, è possibile accedere allo stato di un'entità in un momento precedente del tempo di simulazione e mapparla fino a un punto nel mondo reale. Il buffer ha una dimensione limitata e non è possibile tornare indietro nel tempo oltre il contenuto del buffer. SimSpace Weaver utilizza un puntatore all'offset corrente nel registro degli aggiornamenti per ogni campo e aggiorna un puntatore come parte di un aggiornamento del campo. SimSpace Weaver mappa questi log di aggiornamento nello spazio di processo di un'app utilizzando la memoria condivisa.

Questo formato di oggetto comporta un basso sovraccarico e nessun costo di serializzazione. SimSpace Weaver utilizza questo formato di oggetto anche per analizzare e identificare i campi indice (come la posizione dell'entità).

Entità

Un'entità è l'elemento costitutivo più piccolo di dati nella simulazione. Esempi di entità includono attori (come persone e veicoli) e oggetti statici (come edifici e ostacoli). Le entità hanno proprietà (come posizione e orientamento) in cui è possibile archiviare come dati persistenti SimSpace Weaver. Le entità esistono all'interno delle partizioni.

App

Un' SimSpace Weaver app è un software scritto dall'utente che contiene una logica personalizzata che esegue ogni segno di simulazione. Lo scopo della maggior parte delle app è aggiornare le entità durante l'esecuzione della simulazione. Le app richiamano APIs l'SDK SimSpace Weaver dell'app per eseguire azioni (come la lettura e l'aggiornamento) sulle entità della simulazione.

Impacchettate le app e le relative risorse necessarie (come le librerie) come file.zip e le caricate su. SimSpace Weaver Un'app viene eseguita in un contenitore Docker su un worker. SimSpace Weaver assegna a ciascuna app un numero fisso di unità di risorse sul lavoratore.

SimSpace Weaver assegna la proprietà di una (e solo una) partizione a ciascuna app. Un'app e la relativa partizione si trovano sullo stesso worker. Ogni partizione ha un solo proprietario dell'app. Un'app può creare, leggere, aggiornare ed eliminare entità nella sua partizione. Un'app possiede tutte le entità nella sua partizione.

Esistono tre tipi di app: app spaziali, app personalizzate e app di servizio. Differiscono in base ai casi d'uso e ai cicli di vita.

> **Note**
>
> Nel SimSpace Weaver, il termine «app» può riferirsi al codice di un'app o a un'istanza in esecuzione di quel codice.

App spaziali

Le app spaziali aggiornano lo stato delle entità che esistono spazialmente nella simulazione. Ad esempio, potreste definire un'`Physics`app responsabile dello spostamento e della collisione delle entità per ogni segno di spunta in base alla loro velocità, forma e dimensione. In questo caso, SimSpace Weaver esegue più istanze dell'`Physics`app in parallelo per gestire le dimensioni del carico di lavoro.

SimSpace Weaver gestisce il ciclo di vita delle app spaziali. Specificate una disposizione delle partizioni spaziali delle app nello schema di simulazione. Quando avvii la simulazione, SimSpace Weaver avvia un'app spaziale per ogni partizione spaziale dell'app. Quando interrompi la simulazione, SimSpace Weaver spegne le app spaziali.

Altri tipi di app possono creare entità, ma solo le app spaziali possono aggiornare le entità. Altri tipi di app devono trasferire le entità che creano in un dominio spaziale. SimSpace Weaver utilizza la posizione spaziale di un'entità per spostarla nella partizione di un'app spaziale. Ciò trasferisce la proprietà dell'entità all'app spaziale.

App personalizzate

Utilizzi app personalizzate per interagire con la tua simulazione. Un'app personalizzata legge i dati delle entità utilizzando gli abbonamenti. Un'app personalizzata può creare entità. Tuttavia, l'app deve trasferire un'entità a un'app spaziale per includerla nella simulazione e aggiornarla. È possibile SimSpace Weaver assegnare un endpoint di rete a un'app personalizzata. I client di simulazione possono connettersi all'endpoint di rete per interagire con la simulazione. L'utente definisce le app personalizzate nello schema di simulazione, ma ha la responsabilità di avviarle e interromperle (utilizzando SimSpace Weaver le chiamate API). Dopo aver avviato un'istanza di app personalizzata su un worker, SimSpace Weaver non trasferisce l'istanza a un altro worker.

App di servizio

Puoi utilizzare un'app di servizio quando hai bisogno di un processo di sola lettura in esecuzione su ogni lavoratore. Ad esempio, puoi utilizzare un'app di servizio se hai una simulazione di grandi dimensioni e hai bisogno di un client di visualizzazione che si muova attraverso la simulazione e mostri all'utente solo le entità visibili. In questo caso, una singola istanza dell'app personalizzata non può elaborare tutte le entità della simulazione. È possibile configurare un'app di servizio da avviare su ogni lavoratore. Ciascuna di queste app di servizio può quindi filtrare le entità sul lavoratore assegnato e inviare solo le entità pertinenti ai client connessi. Il client di visualizzazione può quindi connettersi a diverse app di servizio mentre si muove nello spazio di simulazione. Le app di servizio vengono configurate nel proprio schema di simulazione. SimSpace Weaver avvia e arresta le app di servizio per te.

Riepilogo dell'app

La tabella seguente riassume le caratteristiche dei diversi tipi di SimSpace Weaver app.

	App spaziali	App personalizzate	App di servizio
Leggi le entità	Sì	Sì	Sì
Aggiorna entità	Sì	No	No
Creare entità	Sì	Sì*	Sì*
Ciclo di vita	Gestito (lo SimSpace Weaver controlla).	Non gestito (lo controlli tu).	Gestito (lo SimSpace Weaver controlla).
Metodo di avvio	SimSpace Weaver avvia un'istanza dell'app per ogni partizione spaziale, come specificato nello schema.	Si avvia ogni istanza dell'app.	SimSpace Weaver avvia una o più istanze dell'app su ogni lavoratore, come specificato nello schema.
I client possono connettersi	No	Sì	Sì

* Quando un'app o un'app di servizio personalizzata crea un'entità, l'app deve trasferire la proprietà dell'entità a un'app spaziale in modo che l'app spaziale possa aggiornare lo stato dell'entità.

Domini

Un SimSpace Weaver dominio è una raccolta di istanze di app che eseguono lo stesso codice eseguibile dell'app e hanno le stesse opzioni e comandi di avvio. Ci riferiamo ai domini in base ai tipi di app che contengono: domini spaziali, domini personalizzati e domini di servizio. Puoi configurare le tue app all'interno dei domini.

Abbonamenti e repliche

Un'app crea un abbonamento a un'area spaziale per apprendere gli eventi delle entità (ad esempio, immissione, uscita, creazione, aggiornamento ed eliminazione) in quell'area. Un'app elabora gli eventi di entità da un abbonamento prima di leggere i dati per le entità in partizioni di cui non è proprietaria.

Una partizione può esistere sullo stesso worker dell'app (si chiama partizione locale), ma un'altra app può possedere la partizione. Una partizione può esistere anche su un altro worker (si chiama

partizione remota). Se l'abbonamento riguarda una partizione remota, l'operatore crea una copia locale della partizione remota tramite un processo chiamato replica. Il lavoratore legge quindi la copia locale (partizione remota replicata). Se un'altra app sul worker deve leggere da quella partizione con lo stesso segno di spunta, l'operatore legge la stessa copia locale.

Esempi di casi d'uso per SimSpace Weaver

È possibile utilizzarlo SimSpace Weaver per modelli basati su agenti e simulazioni discrete di fasi temporali con un componente spaziale.

Crea simulazioni di grandi affollamenti

Puoi usarle SimSpace Weaver per simulare la folla in ambienti reali. SimSpace Weaver consente di adattare le simulazioni a milioni di oggetti dinamici con comportamenti propri.

Crea ambienti su scala urbana

Utilizzalo SimSpace Weaver per creare un gemello digitale di un'intera città. Crea simulazioni per la pianificazione urbana, per progettare il percorso del traffico e per pianificare la risposta ai rischi ambientali. Puoi utilizzare le tue fonti di dati geospaziali come elementi costitutivi dei tuoi ambienti.

Crea esperienze coinvolgenti e interattive

Crea esperienze di simulazione a cui più utenti possano partecipare e interagire. Usa strumenti di sviluppo popolari come Unreal Engine e Unity per creare mondi virtuali tridimensionali (3D). Personalizza la tua esperienza 3D con i tuoi contenuti e comportamenti.

AWS SimSpace Weaver fine del supporto

Dopo un'attenta valutazione, abbiamo deciso di interrompere il supporto AWS SimSpace Weaver a partire dal 20 maggio 2026. AWS SimSpace Weaver non accetteremo più nuovi clienti a partire dal 20 maggio 2025. In qualità di cliente esistente con un account registrato al servizio prima del 20 maggio 2025, puoi continuare a utilizzare AWS SimSpace Weaver le funzionalità. Dopo il 20 maggio 2026, non potrai più utilizzarle. AWS SimSpace Weaver

Per ulteriori informazioni sulla transizione AWS Batch verso l'esecuzione di simulazioni containerizzate, consulta questo post del blog.

Configurazione per SimSpace Weaver

Per configurarlo per l'utilizzo SimSpace Weaver per la prima volta, è necessario configurare il proprio ambiente Account AWS e quello locale. Una volta completate queste attività, sarete pronti per i tutorial introduttivi.

Attività di configurazione

1. Configura il tuo Account AWS da usare SimSpace Weaver.
2. Configura il tuo ambiente locale per SimSpace Weaver.

Configura il tuo Account AWS da usare SimSpace Weaver

Completa le seguenti attività per Account AWS configurarlo da utilizzare SimSpace Weaver.

Iscriviti per un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la https://portal.aws.amazon.com/billing/registrazione.
2. Segui le istruzioni online.

   Parte della procedura di registrazione prevede la ricezione di una telefonata o di un messaggio di testo e l'immissione di un codice di verifica sulla tastiera del telefono.

   Quando ti iscrivi a un Account AWS, Utente root dell'account AWSviene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come best practice di sicurezza, assegna l'accesso amministrativo a un utente e utilizza solo l'utente root per eseguire attività che richiedono l'accesso di un utente root.

AWS ti invia un'email di conferma dopo il completamento della procedura di registrazione. In qualsiasi momento, puoi visualizzare l'attività corrente del tuo account e gestirlo accedendo a https://aws.amazon.com/e scegliendo Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi AWS Management Consolecome proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso utilizzando un utente root, consulta la pagina Signing in as the root user della Guida per l'utente di Accedi ad AWS .

2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

   Per istruzioni, consulta Abilitare un dispositivo MFA virtuale per l'utente Account AWS root (console) nella Guida per l'utente IAM.

Crea un utente con accesso amministrativo

1. Abilita Centro identità IAM.

   Per istruzioni, consulta Abilitazione di AWS IAM Identity Center nella Guida per l'utente di AWS IAM Identity Center .

2. In IAM Identity Center, assegna l'accesso amministrativo a un utente.

   Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta Configurare l'accesso utente con l'impostazione predefinita IAM Identity Center directory nella Guida per l'AWS IAM Identity Center utente.

Accesso come utente amministratore

- Per accedere con l'utente IAM Identity Center, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.

  Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta AWS Accedere al portale di accesso nella Guida per l'Accedi ad AWS utente.

Assegna l'accesso a ulteriori utenti

1. In IAM Identity Center, crea un set di autorizzazioni conforme alla best practice dell'applicazione di autorizzazioni con il privilegio minimo.

   Segui le istruzioni riportate nella pagina Creazione di un set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .
2. Assegna al gruppo prima gli utenti e poi l'accesso con autenticazione unica (Single Sign-On).

   Per istruzioni, consulta Aggiungere gruppi nella Guida per l'utente di AWS IAM Identity Center .

Aggiungi le autorizzazioni da utilizzare SimSpace Weaver

Per fornire l'accesso, aggiungi autorizzazioni agli utenti, gruppi o ruoli:

- Utenti e gruppi in AWS IAM Identity Center:

  Crea un set di autorizzazioni. Segui le istruzioni riportate nella pagina Create a permission set (Creazione di un set di autorizzazioni) nella Guida per l'utente di AWS IAM Identity Center .
- Utenti gestiti in IAM tramite un provider di identità:

  Crea un ruolo per la federazione delle identità. Segui le istruzioni riportate nella pagina Create a role for a third-party identity provider (federation) della Guida per l'utente IAM.
- Utenti IAM:
  - Crea un ruolo che l'utente possa assumere. Segui le istruzioni riportate nella pagina Create a role for an IAM user della Guida per l'utente IAM.
  - (Non consigliato) Collega una policy direttamente a un utente o aggiungi un utente a un gruppo di utenti. Segui le istruzioni riportate nella pagina Aggiunta di autorizzazioni a un utente (console) nella Guida per l'utente IAM.

Example Politica IAM per la concessione delle autorizzazioni di utilizzo 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"
                }
            }
        }
    ]
}
```

Configura il tuo ambiente locale per SimSpace Weaver

SimSpace Weaver le simulazioni vengono eseguite in container Amazon Linux 2 () ambientiAL2. È necessario disporre di un AL2 ambiente per compilare e collegare le app all'SDK dell' SimSpace Weaver app. L'ambiente di sviluppo locale standard è un contenitore in AL2 Docker. Se scegli di non usare Docker, forniamo istruzioni alternative per eseguire un AL2 ambiente in Windows Subsystem for Linux (WSL). È inoltre possibile utilizzare un metodo personalizzato per creare un AL2 ambiente locale. Per altri modi di esecuzione AL2 locale, consulta la EC2 documentazione di Amazon.

> **Important**
>
> Docker on Microsoft Windows è l'ambiente di sviluppo standard. Per comodità, suggeriamo altri modi per configurare l'ambiente di sviluppo locale, ma non sono standard e non sono supportati.

Argomenti

- Configurare il pacchetto SimSpace Weaver di distribuzione per Amazon Linux 2 (AL2) in Docker
- Configurare il pacchetto SimSpace Weaver di distribuzione per Amazon Linux 2 (AL2) in Windows Subsystem for Linux (WSL)

Configurare il pacchetto SimSpace Weaver di distribuzione per Amazon Linux 2 (AL2) in Docker

Questa sezione fornisce istruzioni per configurare il file zip di SimSpace Weaver distribuzione locale con un AL2 ambiente in Docker. Per istruzioni sulla configurazione con AL2 in Windows Subsystem for Linux (WSL), consulta Configurare il pacchetto SimSpace Weaver di distribuzione per Amazon Linux 2 (AL2) in Windows Subsystem for Linux (WSL).

Requisiti

- Microsoft Windows 10 o versione successiva o un sistema Linux compatibile
- Microsoft Visual Studio 2019o più tardi, con Desktop development with C++carico di lavoro installato
- CMake3
- Git

- Docker Desktop
- AWS CLI
- Python 3.9

Per configurare lo zip di SimSpace Weaver distribuzione con in AL2 Docker

1. Se non hai ancora configurato AWS le tue credenziali per AWS CLI, segui queste istruzioni: Configurazione dell'AWS CLI.
2. Scarica il pacchetto distribuibile SDK SimSpace Weaver dell'app. Contiene i seguenti dati:
   - Binari e librerie per lo sviluppo di app SimSpace Weaver
   - Script di supporto che automatizzano parti del flusso di lavoro di sviluppo
   - Esempi di applicazioni che illustrano i concetti SimSpace Weaver
3. Decomprimi il file in un *sdk-folder* file a tua scelta.
4. Vai a *sdk-folder*.
5. Immettete il seguente comando per installare i pacchetti Python richiesti:

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

6. Immettete il seguente comando per configurare la SimSpace Weaver distribuzione con un'immagine Docker.

   ```
   python setup.py
   ```

   Questo comando funziona nel modo seguente:

   - Crea un'immagine AL2 docker con tutti i requisiti per la creazione di SimSpace Weaver progetti installati.
   - Crea le CloudFormation risorse necessarie per avviare una simulazione.
     - Il modello di CloudFormation stack di esempio è disponibile in *sdk-folder*/`PackagingTools/sample-stack-template.yaml`
   - Configura i progetti di esempio forniti con i percorsi corretti per il sistema locale.

Risoluzione dei problemi

- Docker sembra bloccato
  - Se l'output della console sembra bloccato dopo la chiamata dei comandi Docker, prova a riavviare il motore Docker. Se il problema persiste, riavvia il computer.

Configurare il pacchetto SimSpace Weaver di distribuzione per Amazon Linux 2 (AL2) in Windows Subsystem for Linux (WSL)

Questa sezione fornisce istruzioni per configurare il file zip di SimSpace Weaver distribuzione con un AL2 ambiente in Windows Subsystem for Linux (WSL). Per istruzioni sulla configurazione AL2 in Docker, consulta Configurare il pacchetto SimSpace Weaver di distribuzione per Amazon Linux 2 (AL2) in Docker.

> **Important**
>
> Questa sezione descrive una soluzione che utilizza una versione non posseduta, sviluppata o supportata da Amazon. AL2 Questa soluzione viene fornita solo per comodità dell'utente, se si sceglie di non utilizzarla Docker. Amazon e non AWS si assumono alcuna responsabilità se scegli di utilizzare questa soluzione.

Requisiti

- Hyper-V on Windows 10
- Windows Subsystem for Linux (WSL)
- AL2 Distribuzione open source di terze parti per WSL (scarica la versione 2.0.20200722.0-update.2) (vedi le istruzioni)

> **Important**
>
> Il nostro WSL le istruzioni utilizzano la versione 2.0.20200722.0-update.2 della distribuzione per AL2 WSL. È possibile che si verifichino degli errori se si utilizza un'altra versione.

Per configurare lo zip di SimSpace Weaver distribuzione con in AL2 WSL

1. Al prompt dei comandi di Windows, avvia l'ambiente in AL2 WSL.

```
wsl -d Amazon2
```

> **Important**
>
> Mentre stai correndo WSL, includi l'`--al2`opzione quando esegui uno degli script di supporto in `quick-start.py` Python che si trovano in. `sdky-folder/Samples/sample-name/tools/cloud/quick-start.py`

2. Al prompt della shell Linux, aggiorna il tuo gestore di pacchetti yum.

```
yum update -y
```

> **Important**
>
> Se questo passaggio scade, potrebbe essere necessario passare a WSL1 e riprova queste procedure. Esci dal tuo WSL AL2 sessione e inserisci quanto segue al prompt dei comandi di Windows:
>
> ```
> wsl --set-version Amazon2 1
> ```

3. Installa lo strumento di decompressione.

```
yum install -y unzip
```

4. Rimuovi tutto AWS CLI ciò che è `yum` stato installato. Prova entrambi i seguenti comandi se non sei sicuro di aver `yum` installato un AWS CLI.

```
yum remove awscli
```

```
yum remove aws-cli
```

5. Crea una cartella temporanea e accedi ad essa.

```
mkdir ~/temp
```

```
cd ~/temp
```

6. Scarica e installa AWS CLI:

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

7. È possibile rimuovere la cartella temporanea.

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

8. Riavvia la sessione di shell per aggiornare il percorso nell'ambiente.

```
exec
```

9. Configura AWS le tue credenziali per il AWS CLI tuo AL2 ambiente. Per ulteriori informazioni, consulta Configurazione di . Se lo utilizzi AWS IAM Identity Center, consulta Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'AWS Command Line Interface utente.

```
aws configure
```

10. Installa Git.

```
yum install -y git
```

11. Installare `wget`.

```
yum install -y wget
```

12. Crea una cartella per l'SDK dell' SimSpace Weaver app.

```
mkdir sdk-folder
```

13. Vai alla tua cartella SDK.

```
cd sdk-folder
```

14. Scarica il pacchetto distribuibile SDK SimSpace Weaver dell'app. Contiene i seguenti dati:

- Binari e librerie per lo sviluppo di app SimSpace Weaver
- Script di supporto che automatizzano parti del flusso di lavoro di sviluppo
- Esempi di applicazioni che illustrano i concetti SimSpace Weaver

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

15. Decomprimere il file.

```
unzip *.zip
```

16. Esegui lo script di configurazione WSL.

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

17. Immettete il seguente comando per installare i pacchetti Python richiesti:

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

18. Esegui lo script di configurazione dello zip di SimSpace Weaver distribuzione:

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

Questo comando funziona nel modo seguente:

- Crea le CloudFormation risorse necessarie per avviare una simulazione.
  - Il modello di CloudFormation stack di esempio è disponibile in *sdk-folder*/`PackagingTools/sample-stack-template.yaml`
- Configura i progetti di esempio forniti con i percorsi corretti per il sistema locale.

> **Note**
> È sufficiente eseguire questa operazione una sola volta per l' AL2 ambiente in WSL.

Utilizzo di software concesso in licenza con AWS SimSpace Weaver

AWS SimSpace Weaver ti consente di creare simulazioni con il motore di simulazione e i contenuti che preferisci. In relazione all'utilizzo di SimSpace Weaver, l'utente è responsabile dell'ottenimento, della manutenzione e del rispetto delle condizioni di licenza di qualsiasi software o contenuto utilizzato nelle simulazioni. Verifica che il contratto di licenza ti consenta di distribuire software e contenuti in un ambiente ospitato virtuale.

Guida introduttiva con SimSpace Weaver

Questa sezione fornisce tutorial per aiutarti a iniziare. SimSpace Weaver Questi tutorial introducono al flusso di lavoro generale con cui creare simulazioni. SimSpace Weaver Questi tutorial mostrano come creare, implementare ed eseguire simulazioni in. SimSpace Weaver Ti consigliamo di iniziare con il tutorial di avvio rapido per avviare una simulazione in pochi minuti. Segui gli altri tutorial successivi per saperne di più.

Questi tutorial utilizzano un'applicazione di esempio (`PathfindingSample`) inclusa nel file.zip dell' SimSpace Weaver app SDK che hai scaricato durante le procedure di configurazione. L'applicazione di esempio dimostra i concetti condivisi da tutte le SimSpace Weaver simulazioni, tra cui il partizionamento spaziale, il trasferimento di entità tra partizioni, le app e gli abbonamenti.

Nei tutorial, creerai una simulazione con quattro partizioni spaziali. Un'istanza separata dell'app `PathfindingSample` spaziale gestisce ogni singola partizione. Le app spaziali creano entità nelle proprie partizioni. Le entità si spostano in una posizione particolare nel mondo della simulazione, evitando gli ostacoli mentre si muovono. È possibile utilizzare un'applicazione client separata (inclusa nell'SDK dell' SimSpace Weaver app) per visualizzare la simulazione.

Argomenti

- Tutorial di avvio rapido per SimSpace Weaver
- Tutorial dettagliato: scopri i dettagli durante la creazione dell'applicazione di esempio

Tutorial di avvio rapido per SimSpace Weaver

Questo tutorial ti guida attraverso il processo per creare ed eseguire una simulazione SimSpace Weaver in pochi minuti. Ti consigliamo di iniziare con questo tutorial e poi passare al tutorial dettagliato in seguito.

Requisiti

Prima di iniziare, assicurati di aver completato i passaggi indicatiConfigurazione per SimSpace Weaver.

> **Note**
>
> Gli script utilizzati qui vengono forniti per comodità dell'utente e NON sono obbligatori. Consulta il tutorial dettagliato per scoprire come eseguire questi passaggi manualmente.

Passaggio 1: abilitare la registrazione (opzionale)

Per attivare la registrazione

1. Vai a:

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

2. Apri il file dello schema in un editor di testo:

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

3. Trova la `simulation_properties:` sezione all'inizio del file:

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

4. Inserisci le seguenti 2 righe dopo la riga`simulation_properties:`:

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

5. Conferma che la `simulation_properties:` sezione sia uguale alla seguente:

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

6. Salva il file ed esci dall'editor di testo.

Passaggio 2: avvio rapido con il client della console (opzione 1)

Vai a:

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

Eseguire uno dei seguenti comandi:

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

Per impostazione predefinita, verrà avviata una simulazione con una singola partizione su un singolo lavoratore. Altre configurazioni possono essere avviate passando il file `--schema {file name}.yaml` dalla cartella. `/Samples/PathfindingSample/tools/`

> **Note**
>
> Vedi Tutorial dettagliato: scopri i dettagli durante la creazione dell'applicazione di esempio per una spiegazione approfondita di ciò che fa questo script.

Passo 2: Avvio rapido con il client Unreal Engine (opzione 2)

Per informazioni, consulta Avvio del client di visualizzazione Unreal Engine.

Interrompi ed elimina la simulazione

Vai a:

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

Trova i nomi delle tue simulazioni:

```
aws simspaceweaver list-simulations
```

Interrompi ed elimina la simulazione

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

Risoluzione dei problemi

- FileNotFoundError: cmake

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

  - Risoluzione: lo script non riesce a trovare il comando`cmake`. Assicurati di avere installata la CMake versione minima consigliata e che possa essere richiamata con il `cmake` comando in PATH. Usa il comando `cmake -version` per verificare.
- ImportError: caricamento della DLL non riuscito durante l'importazione di libweaver_app_sdk_python_v1: il modulo specificato non è stato trovato.
  - Risoluzione: questo errore si verifica quando Python 3.9 non viene utilizzato per avviare Weaver Python SDK. Assicurati che la versione python associata al comando «python» sia Python 3.9. Puoi verificare eseguendo il comando. `python --version`
- Lo script di avvio rapido appare bloccato dopo l'avvio di Docker Build.
  - Risoluzione: a volte Docker ha bisogno di alcuni minuti per riscaldarsi. Se il problema persiste per più di ~5 minuti, riavvia Docker o il sistema.
- target_compile_features nessuna funzionalità nota per il compilatore CXX «GNU»:
  - Risoluzione: svuota la cache di Docker, elimina l'immagine Docker di weaverappbuilder, elimina gli artefatti di compilazione del progetto e riesegui. `setup.py` Questo dovrebbe ripristinare l'ambiente Docker e risolvere l'errore.

Tutorial dettagliato: scopri i dettagli durante la creazione dell'applicazione di esempio

Il tutorial di avvio rapido ha spiegato come creare, avviare, interrompere ed eliminare una simulazione di esempio utilizzando `quick-start.py` `estop-and-delete.py`. Questo tutorial illustrerà in dettaglio come funzionano questi script e i parametri aggiuntivi che questi script possono utilizzare per massimizzare la flessibilità delle simulazioni Weaver personalizzate.

Requisiti

Prima di iniziare, assicurati di aver completato i passaggi indicati in. Configurazione per SimSpace Weaver

Passaggio 1: abilitare la registrazione (opzionale)

Per attivare la registrazione

1. Vai a:

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

2. Apri il file di schema in un editor di testo:

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

3. Trova la `simulation_properties:` sezione all'inizio del file:

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

4. Inserisci le seguenti 2 righe dopo la riga`simulation_properties:`:

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

5. Conferma che la `simulation_properties:` sezione sia uguale alla seguente:

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

6. Salva il file ed esci dall'editor di testo.

Fase 2: Inizia la simulazione

Come illustrato nel tutorial di avvio rapido, i passaggi fondamentali per avviare una simulazione di esempio sono:

1. Vai a:

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

2. Esegui uno dei seguenti comandi:

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

Questo script automatizza i comandi terminali più comuni, che possono essere eseguiti tutti manualmente utilizzando. AWS CLI Questi passaggi sono:

1. Carica lo schema Weaver su S3.
   - SimSpace Weaver utilizza uno schema per configurare la simulazione. Lo schema è un file di testo semplice in formato YAML. Per ulteriori informazioni, consulta Configurazione della simulazione.
2. Crea e carica un contenitore personalizzato (opzionale).
   - Se lo schema definisce un contenitore personalizzato, lo script di avvio rapido creerà l'immagine docker e la caricherà su Amazon ECR. Per ulteriori informazioni, consulta Contenitori personalizzati. Consulta lo `PythonBubblesSample` schema per un esempio di questa funzionalità.
3. Compilare il progetto.
   - `quick-start.py`chiama la `build_project` funzione definita in`build.py`. Questo passaggio varierà a seconda del progetto. Per il PathfindingSample, CMake viene utilizzato. Il comando CMake and Docker per cui è disponibile in`build.py`.
4. Carica gli artefatti della build su S3.
   - Puoi controllare i tuoi bucket S3 per assicurarti che tutti i caricamenti siano stati eseguiti correttamente. Per ulteriori informazioni sull'uso di Amazon S3, consulta Creazione, configurazione e utilizzo dei bucket Amazon S3 nella Guida per l'utente di Amazon Simple Storage Service.
   - Gli zip dell'applicazione di esempio e il bucket S3 utilizzano il seguente formato di nome:
     - `weaver-sample-bucket-account-number-region`
     - App spaziale: `ProjectNameSpatial.zip`
     - Visualizza l'app (personalizzata): `ProjectNameView.zip`
5. Avvia la simulazione.
   - Questo è un riepilogo della chiamata. `aws simspaceweaver start-simulation` AWS CLI Per ulteriori informazioni, vedere il AWS CLI Command Reference for. SimSpace Weaver

- Lo script verrà ripetuto fino a quando lo stato della simulazione non sarà `STARTED` o`FAILED`. L'avvio di una simulazione può richiedere alcuni minuti.

6. Ottieni i dettagli della simulazione.

   - L'DescribeSimulationAPI fornisce dettagli sulla simulazione, incluso lo stato. Una simulazione può trovarsi in uno dei seguenti stati:

     Stati del ciclo di vita della simulazione

     1. **`STARTING`**— Stato iniziale dopo la chiamata StartSimulation
     2. **`STARTED`**— tutte le app spaziali sono avviate e funzionanti
     3. **`STOPPING`**— Stato iniziale dopo la chiamata StopSimulation
     4. **`STOPPED`**— Tutte le risorse di elaborazione vengono interrotte
     5. **`DELETING`**— Stato iniziale dopo la chiamata DeleteSimulation
     6. **`DELETED`**— Tutte le risorse assegnate alla simulazione vengono eliminate
     7. **`FAILED`**— La simulazione ha avuto un errore/fallimento critico e si è interrotta
     8. **`SNAPSHOT_IN_PROGRESS`**— È in corso un'istantanea

     Per ottenere i dettagli della simulazione

     1. Chiamata dell'API ListSimulations.

        ```
        aws simspaceweaver list-simulations
        ```

        Lo script dovrebbe mostrare dettagli su ciascuna delle simulazioni, simili ai seguenti:

        ```
        {
            "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. Chiama DescribeSimulation per avere i dettagli della simulazione. Sostituiscilo *`simulation-name`* Name con la simulazione dell'output del passaggio precedente.

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

Lo script dovrebbe mostrare maggiori dettagli sulla simulazione specificata, simili ai seguenti:

```
{
    "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. Avvia app personalizzate.

   - SimSpace Weaver non gestisce il ciclo di vita delle app personalizzate. È necessario avviare le app personalizzate. È consigliabile avviare le app personalizzate prima di avviare l'orologio di simulazione, ma è possibile avviare le app personalizzate dopo aver avviato l'orologio.

     Puoi chiamare l'StartAppAPI per avviare le tue app personalizzate.

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

     La chiamata StartApp API creerà e avvierà una nuova istanza dell'app personalizzata utilizzando il nome fornito. Se fornisci il nome di un'app già esistente, riceverai un errore. Se desideri riavviare una particolare app (istanza), devi prima arrestare quell'app ed eliminarla.

     > **Note**
     >
     > Lo stato della simulazione deve essere quello `STARTED` precedente all'avvio di app personalizzate.

     L'applicazione di esempio fornisce l'app `ViewApp` personalizzata per visualizzare la simulazione. Questa app fornisce un indirizzo IP statico e un numero di porta per connettere i client di simulazione (lo farai in un passaggio successivo di questo tutorial). Puoi pensare a

una domain classe di app che hanno lo stesso codice eseguibile e le stesse opzioni di avvio. app nameIdentifica l'istanza dell'app. Per ulteriori informazioni sui SimSpace Weaver concetti, vedereConcetti chiave per SimSpace Weaver.

Puoi utilizzare l'DescribeAppAPI per verificare lo stato di un'app personalizzata dopo averla avviata.

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

Per avviare l'app di visualizzazione in questo tutorial

1. Chiama StartApp per`ViewApp`.

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

2. Chiama DescribeApp per verificare lo stato della tua app personalizzata.

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

Dopo lo stato dell'app personalizzata (istanza)`STARTED`, l'output di DescribeApp includerà l'indirizzo IP e il numero di porta dell'app personalizzata (istanza). Nell'output di esempio seguente, l'indirizzo IP è il valore di `Address` e il numero di porta è il valore di `Actual` nel `EndpointInfo` blocco.

```
{
    "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**
>
> Il valore di `Declared` è il numero di porta a cui deve essere associato il codice dell'app. Il valore di `Actual` è il numero di porta che SimSpace Weaver i client possono connettersi alla tua app. SimSpace Weaver mappa la `Declared` porta alla `Actual` porta.

> **Note**
>
> È possibile utilizzare la procedura descritta in Ottieni l'indirizzo IP e il numero di porta di un'app personalizzata per ottenere l'indirizzo IP e il numero di porta di qualsiasi app personalizzata avviata.

8. Avvia l'orologio.

   - Quando crei la simulazione per la prima volta, c'è un orologio ma l'orologio non è in funzione. Quando l'orologio non è in funzione, la simulazione non aggiornerà il suo stato. Dopo aver avviato l'orologio, inizierà a inviare segni di spunta alle tue app. Ogni segno di spunta, le tue app spaziali esaminano le entità che possiedono e a cui inviano i risultati SimSpace Weaver

   > **Note**
   >
   > L'avvio dell'orologio può richiedere 30-60 secondi.

   Chiamata dell'API StartClock.

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

> **Note**
>
> L'StartClockAPI utilizza il tuo*simulation-name*, che puoi trovare utilizzando l'ListSimulationsAPI:
>
> ```
> aws simspaceweaver list-simulations
> ```

parametri di avvio rapido

- -h, --help
  - Elenca questi parametri.
- --clean
  - Elimina il contenuto della cartella di compilazione prima della compilazione.
- --al2
  - Si basa direttamente sulla macchina nativa anziché su Docker. Usalo solo se esegui in un ambiente Amazon Linux 2, come WSL.
- --upload only
  - Carica solo lo schema e gli zip dell'app su Amazon S3, non avviare la simulazione.
- --nessuna build
  - Salta la ricostruzione del progetto.
- - nessun contenitore
  - Salta la ricostruzione del contenitore di simulazione elencato nello schema.
- --consoleclient
  - Crea e connetti automaticamente il client della console elencato in config.py.
- SCHEMA --schema
  - Quale schema utilizzerà questa invocazione. Il valore predefinito è 'SCHEMA' in config.py.
- --name NAME
  - Che nome avrà la simulazione. Il valore predefinito è 'Project_name'-date-time in config.py.

Fase 3: Controlla i log (opzionale)

SimSpace Weaver scrive i messaggi di gestione della simulazione e l'output della console dalle tue app su Amazon CloudWatch Logs. Per ulteriori informazioni sull'utilizzo dei log, consulta Working with log groups and log stream nella Amazon CloudWatch Logs User Guide.

Ogni simulazione che crei ha il proprio gruppo di log in Logs. CloudWatch Il nome del gruppo di log è specificato nello schema di simulazione. Nel frammento di schema seguente, il valore di è. `log_destination_service` `logs` Ciò significa che il valore di `log_destination_resource_name` è il nome di un gruppo di log. In questo caso, il gruppo di log è`MySimulationLogs`.

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

Puoi anche utilizzare l'DescribeSimulationAPI per trovare il nome del gruppo di log per la simulazione dopo averlo avviato.

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

L'esempio seguente mostra la parte dell'output DescribeSimulation che descrive la configurazione di registrazione. Il nome del gruppo di log viene visualizzato alla fine di`LogGroupArn`.

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

Ogni gruppo di log di simulazione contiene diversi flussi di log:

- Flusso di log di gestione: messaggi di gestione della simulazione prodotti dal servizio. SimSpace Weaver

```
/sim/management
```

- Flusso di registro degli errori: messaggi di errore prodotti dal SimSpace Weaver servizio. Questo flusso di log esiste solo in presenza di errori. SimSpace Weaver archivia gli errori scritti dalle tue app nei rispettivi flussi di log delle app (vedi i seguenti flussi di log).

```
/sim/errors
```

- Stream di log spaziali delle app (1 per ogni app spaziale su ciascun worker): output della console prodotto dalle app spaziali. Ogni app spaziale scrive nel proprio flusso di log. Sono *spatial-app-id* tutti i caratteri dopo la barra finale alla fine di. *worker-id*

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

- Stream di log delle app personalizzati (1 per ogni istanza di app personalizzata): output della console prodotto da app personalizzate. Ogni istanza dell'app personalizzata scrive nel proprio flusso di log.

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

- Stream di log delle app di servizio (1 per ogni istanza dell'app di servizio): output della console prodotto dalle app di servizio. Ogni app di servizio scrive nel proprio flusso di log. Sono *service-app-id* tutti i caratteri dopo la barra finale alla fine di. *service-app-name*

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

> **Note**
>
> L'applicazione di esempio non dispone di app di servizio.

Fase 4: Visualizza la simulazione

L'SDK SimSpace Weaver dell'app offre diverse opzioni per visualizzare l'applicazione di esempio. Puoi utilizzare il client della console di esempio se non disponi di alcun supporto locale per lo sviluppo di Unreal Engine. Le istruzioni per il client Unreal Engine presuppongono che tu stia utilizzando Windows.

Il client della console visualizza un elenco di eventi di entità man mano che si verificano. Il client ottiene le informazioni sugli eventi dell'entità da`ViewApp`. Se il client della console visualizza l'elenco degli eventi, conferma la connettività di rete con l'attività `ViewApp` e della simulazione.

La `PathfindingSample` simulazione crea entità fisse e mobili su un piano bidimensionale. Le entità mobili si muovono attorno alle entità fisse. Il client Unreal Engine fornisce una visualizzazione degli eventi dell'entità.

Client della console

Il client della console può essere creato e connesso automaticamente all'avvio di un esempio, `quick-start.py` se si include l'`--consoleclient`opzione. Per creare e connettere il client della console dopo `quick-start.py` che è già stato chiamato, procedi come segue:

Vai a:

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

Esegui lo script per creare e connettere il client:

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

Lo script eseguirà le seguenti operazioni:

1. Crea il client della console con CMake.
2. Avvia l'eseguibile integrato con l'indirizzo IP e il numero di porta forniti.

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

Client Unreal Engine

Consultare Avvio del client di visualizzazione Unreal Engine.

Passaggio 5: interrompi ed elimina la simulazione

Vai a:

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

Trova i nomi delle tue simulazioni:

```
aws simspaceweaver list-simulations
```

Interrompi ed elimina la simulazione:

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

Lo script `stop-and-delete.py` eseguirà le seguenti operazioni:

1. Chiama il AWS CLI comando per interrompere una simulazione.
   - `aws simspaceweaver stop-simulation`
   - Per ulteriori informazioni, vedere AWS CLI Command Reference for SimSpace Weaver.
2. Chiama il AWS CLI comando per eliminare una simulazione.
   - `aws simpaceweaver delete-simulation`
   - Per ulteriori informazioni, vedere AWS CLI Command Reference for SimSpace Weaver.

stop-and-delete parametri

- -h, --help
  - Elenca questi parametri.
- --simulazione di simulazione
  - Il nome della simulazione di stop-and-delete
- --stop
  - Interrompi solo la simulazione. Non lo elimina.
- --delete
  - Elimina solo una simulazione. Funzionerà solo se la simulazione è uno dei due`STOPPED`. `FAILED`

Risoluzione dei problemi

Vedi Risoluzione dei problemi nel tutorial di avvio rapido.

Lavorare con SimSpace Weaver

Questo capitolo fornisce informazioni e linee guida per aiutarti a creare le tue applicazioni in SimSpace Weaver.

Argomenti

- Configurazione della simulazione
- Durata massima di una simulazione
- Sviluppo di app
- Sviluppo di applicazioni client
- Ottieni l'indirizzo IP e il numero di porta di un'app personalizzata
- Avvio del client di visualizzazione Unreal Engine
- Sviluppo locale in SimSpace Weaver
- AWS SimSpace Weaver SDK dell'app
- AWS SimSpace Weaver quadro dimostrativo
- Utilizzo delle quote di servizio
- Simulazioni di debug
- Contenitori personalizzati
- Uso di Python
- Support per altri motori
- Utilizzo di software concesso in licenza con AWS SimSpace Weaver
- Gestisci le tue risorse con AWS CloudFormation
- Snapshot
- Messaggistica

Configurazione della simulazione

Uno schema (o schema) di simulazione è YAML-file di testo in formato che specifica la configurazione per una simulazione. È possibile utilizzare lo stesso schema per avviare più simulazioni. Il file dello schema si trova nella cartella del progetto per la simulazione. È possibile utilizzare qualsiasi editor di testo per modificare il file. SimSpace Weaver legge lo schema solo

quando avvia la simulazione. Tutte le modifiche apportate a un file di schema influiscono solo sulle nuove simulazioni avviate dopo le modifiche.

Per configurare la simulazione, modifica il file dello schema di simulazione (usa il separatore di percorso appropriato per il tuo sistema operativo):

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

Lo schema di simulazione viene caricato quando si crea una nuova simulazione. Lo script di supporto per l'avvio rapido del progetto caricherà lo schema come parte del processo di creazione della simulazione:

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

Per ulteriori informazioni sull'esecuzione dello script di avvio rapido, consulta il Nozioni di base capitolo di Tutorial dettagliato questa guida.

Parametri di configurazione della simulazione

Lo schema di simulazione contiene informazioni di bootstrap, tra cui:

- Proprietà di simulazione: versione SDK e configurazione di calcolo (tipo e numero di lavoratori)
- Orologi: frequenza di ticchettio e tolleranze
- Strategie di partizionamento spaziale: topologia spaziale (come una griglia), limiti e gruppi di posizionamento (raggruppamento delle partizioni spaziali sui lavoratori)
- Domini e relative app: app bucket, path e comandi di avvio

SimSpace Weaver utilizza la configurazione dello schema per configurare e disporre le partizioni spaziali, avviare app e far avanzare la simulazione alla frequenza specificata.

> **Note**
>
> Lo script di creazione del progetto nell'SDK dell' SimSpace Weaver app genererà automaticamente uno schema di simulazione basato sull'applicazione di esempio.

I seguenti argomenti descrivono i parametri dello schema di simulazione. Per una descrizione completa dello schema di simulazione, vedere. SimSpace Weaver riferimento allo schema di simulazione

Argomenti

- Versione SDK
- Proprietà di simulazione
- Worker
- Orologio
- Strategie di partizionamento
- Domini

Versione SDK

Il `sdk_version` campo specifica la versione per SimSpace Weaver cui è formattato lo schema. Valori validi: `1.17`, `1.16`, `1.15`, `1.14`, `1.13`, `1.12`

> **Important**
>
> Il valore di include `sdk_version` solo il numero della versione principale e il numero della prima versione secondaria. Ad esempio, il valore `1.12` specifica tutte le versioni`1.12.x`, ad esempio `1.12.0`1.12.1, e`1.12.2`.

Proprietà di simulazione

La `simulation_properties` sezione dello schema specifica la configurazione di registrazione e un tipo di dati per il campo indice (in genere la posizione spaziale) delle entità.

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

Il valore di `log_destination_service` determina l'interpretazione del valore di. `log_destination_resource_name` Attualmente, l'unico valore supportato è `logs`. Ciò significa

che il valore di `log_destination_resource_name` è il nome di un gruppo di log in Amazon CloudWatch Logs

> **Note**
>
> La registrazione è facoltativa. Se non configuri le proprietà di destinazione dei log, la simulazione non produrrà log.

È obbligatoria solo la proprietà `default_entity_index_key_type`. L'unico valore valido è `Vector3<f32>`.

Worker

La `workers` sezione specifica il tipo e il numero di lavoratori che desideri per la simulazione. SimSpace Weaver utilizza i propri tipi di worker mappati ai tipi di EC2 istanze Amazon.

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

Abilitazione di simulazioni con più lavoratori

È possibile creare una simulazione che utilizzi più di un lavoratore. Per impostazione predefinita, le simulazioni utilizzano 1 lavoratore. È necessario modificare lo schema di simulazione prima di iniziare la simulazione.

> **Note**
>
> Non è possibile modificare una simulazione già iniziata. Se desideri abilitare multiworker per una simulazione in esecuzione, devi prima interrompere ed eliminare la simulazione.

Per utilizzare più di un worker, imposta il `desired` numero di istanze di calcolo su un valore maggiore di 1. È previsto un numero massimo di app per ogni lavoratore. Per ulteriori informazioni, vedereSimSpace Endpoint e quote Weaver. SimSpace Weaver utilizzerà più di 1 worker solo quando

il numero di app su un worker supera questo limite. SimSpace Weaver può inserire un'app su uno qualsiasi dei lavoratori disponibili. Il posizionamento dell'app su un lavoratore specifico non è garantito.

Il seguente frammento di schema mostra una configurazione per una simulazione che richiede 2 lavoratori. SimSpace Weaver tenterà di assegnare il secondo lavoratore se il numero di app supera il numero massimo di app per 1 lavoratore.

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

Orologio

La `clock` sezione specifica le proprietà dell'orologio di simulazione. Attualmente, puoi solo configurare la frequenza dei tick (il numero di tick al secondo che l'orologio invia alle app). Il tick-rate è il tasso massimo. La percentuale di tick effettiva potrebbe essere inferiore perché tutte le operazioni (come gli aggiornamenti delle entità) relative a un segno di spunta devono essere completate prima che possa iniziare il tick successivo. Il tick rate è anche chiamato clock rate.

I valori validi per `tick_rate` dipendono da quelli `sdk_version` specificati nello schema.

Valori validi per il tick-rate

- Versioni precedenti a`"1.14"`:
  - `10`
  - `15`
  - `30`
- Versione `"1.14"` o successiva:
  - `"10"`
  - `"15"`
  - `"30"`
  - `"unlimited"`

    Per ulteriori informazioni, consulta Tick rate illimitato.

> **Important**
>
> - Per gli schemi `"1.14"` il cui valore `sdk_version` precedente a `tick_rate` è un numero intero, ad esempio. `30`
> - Per gli schemi con un `sdk_version` `"1.14"` o successivo, il valore di `tick_rate` è una stringa, ad esempio. `"30"` Il valore deve includere le virgolette doppie.
>
> Se si converte una versione `"1.12"` o `"1.13"` uno schema in una versione `"1.14"` o successiva, è necessario racchiudere il valore di `tick_rate` tra virgolette doppie.

Tick rate illimitato

Puoi impostare to per consentire `tick_rate` `"unlimited"` alla simulazione di funzionare alla stessa velocità di esecuzione del codice. Con una frequenza di selezione illimitata, SimSpace Weaver invia il segno di spunta successivo immediatamente dopo che tutte le app hanno completato i commit per il segno di spunta corrente.

> **Important**
>
> La frequenza di selezione illimitata non è supportata nelle versioni precedenti alla 1.14.0. SimSpace Weaver Il valore minimo di `sdk_version` nello schema è. `"1.14"`

Tick rate illimitato in SimSpace Weaver Local

SimSpace Weaver Local implementa `"unlimited"` come se lo schema specificasse una frequenza di tick di 10 kHz (10000). L'effetto è lo stesso di una frequenza di tick illimitata in. Cloud AWS Lo specifichi ancora `tick_rate: "unlimited"` nel tuo schema. Per ulteriori informazioni sull' SimSpace Weaver Local, consulta Sviluppo locale in SimSpace Weaver.

Domande frequenti sull'orologio

D1. Posso modificare una simulazione STARTED per utilizzare un tick rate diverso?

Non è possibile modificare il tick rate di una simulazione già esistente in qualsiasi Cloud AWS fase del suo ciclo di vita. Inoltre, non è possibile modificare il tick rate di una simulazione in esecuzione in SimSpace Weaver Local. È possibile impostarlo `tick_rate` nello schema e iniziare una nuova simulazione da quello schema.

D2. Posso eseguire la mia simulazione con un tick rate illimitato in una versione precedente alla 1.14?

No, il tick rate illimitato non è supportato nelle versioni precedenti alla 1.14.0.

Risoluzione degli errori dell'orologio

Se la simulazione non si avvia, puoi controllare il valore di `"StartError"` nell'output dell'DescribeSimulationAPI. Un `tick_rate` valore non valido nello schema produrrà i seguenti errori.

> **Note**
>
> L'output di errore mostrato qui viene visualizzato su più righe per migliorare la leggibilità. L'output di errore effettivo è una singola riga.

- `sdk_version`è precedente a `"1.14"` e il valore di `tick_rate` è un numero intero non valido. Valori validi: `10`, `15`, `30`

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

- `sdk_version`è precedente a `"1.14"` e il valore di `tick_rate` è una stringa. Valori validi: `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\"}]"
```

- `sdk_version`è `"1.14"` o successivo e il valore di `tick_rate` è una stringa non valida. Valori validi: `"10"`, `"15"`, `"30"`, `"unlimited"`

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

- `sdk_version`È `"1.14"` o successivo e il valore di `tick_rate` è un numero intero. Valori validi: `"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\"}]"
```

Strategie di partizionamento

La `partitioning_strategies` sezione specifica le proprietà di configurazione per le partizioni delle app spaziali. È possibile fornire il proprio nome per una strategia di partizionamento (un insieme di proprietà in questa sezione) e utilizzarlo nella configurazione dell'app spaziale.

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

La `topology` proprietà specifica il tipo di sistema di coordinate utilizzato dalla simulazione. Il valore `Grid` specifica una griglia bidimensionale (2D).

Per una `Grid` topologia, lo spazio di simulazione è modellato come un riquadro di delimitazione allineato agli assi (AABB). Specificate i limiti delle coordinate per ogni asse del vostro AABB nella proprietà. `aabb_bounds` Tutte le entità che esistono spazialmente nella simulazione devono avere una posizione all'interno dell'AABB.

Gruppi di posizionamento sulla griglia

Un gruppo di collocamento è una raccolta di partizioni spaziali di app che si desidera SimSpace Weaver posizionare sullo stesso lavoratore. È possibile specificare il numero e la disposizione dei gruppi di collocamento (in una griglia) nella `grid_placement_groups` proprietà. SimSpace Weaver tenterà di distribuire uniformemente le partizioni tra i gruppi di posizionamento. Le aree di proprietà delle app spaziali con partizioni nello stesso gruppo di posizionamento saranno spazialmente adiacenti.

Si consiglia che x * y sia uguale al numero di lavoratori desiderato. Se non è uguale, SimSpace Weaver cercheremo di bilanciare i gruppi di collocamento tra i lavoratori disponibili.

Se non specifichi una configurazione del gruppo di collocamento, ne SimSpace Weaver calcolerà una per te.

Domini

Fornisci un nome per un set di proprietà di configurazione per un dominio. L'impostazione di avvio per le app in un dominio determina il tipo di dominio:

- **`launch_apps_via_start_app_call`**— dominio personalizzato
- **`launch_apps_by_partitioning_strategy`**— dominio spaziale
- **`launch_apps_per_worker`**(non incluso nell'applicazione di esempio) — dominio di servizio

> **Important**
>
> SimSpace Weaver supporta fino a 5 domini per ogni simulazione. Ciò include tutti i domini spaziali, personalizzati e di servizio.

```
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 I progetti app SDK versione 1.12.x utilizzano bucket separati per i file.zip dell'app e lo schema:
>
> - weaver- - -app-zips- *lowercase-project-name account-number region*
> - weaver- - -schemi- *lowercase-project-name account-number region*

Argomenti

- Configurazione dell'app
- Configurazione dei domini spaziali
- Endpoint di rete
- Configurazione dei domini di servizio

Configurazione dell'app

Specificate la configurazione di un'app (`app_config`) come parte della configurazione per il relativo dominio. Tutti i tipi di domini utilizzano le stesse proprietà di configurazione dell'app.

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

> **Note**
>
> SimSpace Weaver I progetti app SDK versione 1.12.x utilizzano bucket separati per i file.zip dell'app e lo schema:
>
> - weaver- - -app-zips- *lowercase-project-name account-number region*

- weaver- - -schemi- *lowercase-project-name account-number region*

La `package` proprietà specifica l'URI S3 di un file zip in un bucket S3. Il file zip contiene l'eseguibile dell'app (chiamato anche binario) e tutte le altre risorse di cui ha bisogno (come le librerie). Ogni istanza dell'eseguibile dell'app viene eseguita in un Docker contenitore su un lavoratore.

La `launch_command` proprietà specifica il nome dell'eseguibile e le eventuali opzioni della riga di comando per eseguire l'app. Il valore di `launch_command` è un array. Ogni token dell'intera stringa di comando di avvio è un elemento dell'array.

Esempio

- Per il comando di avvio: `MyTestApp --option1 value1`
- Specificare: `launch_command: ["MyTestApp", "-option1", "value1"]`

La `required_resource_units` proprietà specifica il numero di unità di risorse di calcolo da SimSpace Weaver allocare a questa app. Un'unità di risorse di calcolo è una quantità fissa di capacità di elaborazione (vCPU) e memoria (RAM) su un operaio. È possibile aumentare questo valore per aumentare la quantità di potenza di calcolo disponibile per l'app quando viene eseguita su un worker. Esiste un numero limitato di unità di risorse di calcolo per ogni lavoratore. Per ulteriori informazioni, consulta SimSpace Endpoint e quote Weaver.

Configurazione dei domini spaziali

Per i domini spaziali, è necessario specificare un. `partitioning_strategy` Il valore di questa proprietà è il nome assegnato a una strategia di partizionamento definita in un'altra parte dello schema.

```
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 I progetti app SDK versione 1.12.x utilizzano bucket separati per i file.zip dell'app e lo schema:
>
> - weaver- - -app-zips- *lowercase-project-name account-number region*
> - weaver- - -schemi- *lowercase-project-name account-number region*

Una strategia di partizionamento con una `Grid` topologia (l'unica topologia supportata in questa versione) consente di disporre le partizioni spaziali delle app di questo SimSpace Weaver dominio in una griglia. La `grid_partition` proprietà specifica il numero di righe e colonne della griglia di partizione.

SimSpace Weaver avvierà 1 istanza dell'app spaziale per ogni cella nella griglia di partizione. Ad esempio, se un dominio spaziale ha `grid_partition` valori `x: 2` e `y: 2` ci sono 2 * 2 = 4 partizioni nel dominio spaziale. SimSpace Weaver avvierà 4 istanze dell'app configurate nel dominio spaziale e assegnerà 1 partizione a ciascuna istanza dell'app.

Argomenti

- Requisiti di risorse per i domini spaziali
- Domini spaziali multipli
- Domande frequenti sui domini spaziali
- Risoluzione dei problemi relativi ai domini spaziali

Requisiti di risorse per i domini spaziali

È possibile assegnare fino a 17 unità di risorse di calcolo per ogni lavoratore. Specificate il numero di unità di risorse di calcolo utilizzate da ogni app spaziale nella `app_config` sezione del vostro dominio spaziale.

Example frammento di schema che mostra le unità di risorse di calcolo per un'app spaziale

```
  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
```

Per calcolare il numero di unità di risorse di calcolo richieste da un dominio, moltiplica il numero di celle nella griglia (nel tuo`grid_partition`, `x` *`y`) per il numero di unità di risorse di calcolo assegnate alle app spaziali.

Nell'esempio precedente, il dominio specifica: `MySpatialDomain`

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

La griglia per `MySpatialDomain` ha 2* 2 = 4 celle. Il dominio spaziale richiede 4* 1 = 4 unità di risorse di calcolo.

Il numero totale di unità di risorse di calcolo per tutti i domini specificati nello schema deve essere inferiore o uguale al numero di worker moltiplicato per il `desired` numero massimo di unità di risorse di calcolo per ogni lavoratore (17).

Domini spaziali multipli

È possibile configurare la simulazione per utilizzare più di un dominio spaziale. Ad esempio, puoi utilizzare 1 dominio spaziale per controllare gli attori principali di una simulazione (come persone e automobili) e un dominio spaziale diverso per controllare l'ambiente.

Puoi anche utilizzare più domini spaziali per assegnare risorse diverse a diverse parti della simulazione. Ad esempio, se la simulazione ha un tipo di entità che ha 10 volte più istanze di entità rispetto a un altro tipo, è possibile creare domini diversi per gestire ciascun tipo di entità e allocare più risorse per il dominio con più entità.

> **Important**
>
> SimSpace Weaver le versioni precedenti alla 1.14.0 non supportano più domini spaziali.

> **Important**
>
> AWS SimSpace Weaver Local attualmente non supporta più domini spaziali. Per ulteriori informazioni sull' SimSpace Weaver Local, consulta Sviluppo locale in SimSpace Weaver.

> **Important**
>
> SimSpace Weaver supporta fino a 5 domini per ogni simulazione. Ciò include tutti i domini spaziali, personalizzati e di servizio.

Configura più domini spaziali

Per configurare più di un dominio spaziale, aggiungi le altre definizioni di dominio spaziale come sezioni denominate separate nello schema. Ogni dominio deve specificare la chiave. `launch_apps_by_partitioning_strategy` Vedi lo schema di esempio seguente.

```
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
```

Collocazione di domini spaziali

In alcuni scenari, potresti voler posizionare le partizioni per un dominio spaziale sui worker accanto alle partizioni di un altro dominio. Ciò può migliorare le caratteristiche prestazionali se tali partizioni creano sottoscrizioni tra domini diversi.

Aggiungi la chiave `placement_constraints` di primo livello allo schema per specificare quali domini SimSpace Weaver devono essere messi insieme. La `on_workers` chiave richiesta deve fare riferimento a una `workers` configurazione denominata nello schema.

Example frammento di schema che mostra i domini spaziali messi insieme

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

> **Important**
>
> - Se si utilizzano gruppi di collocamento:
> - Assicurati che x * y sia un multiplo del numero di lavoratori.
> - Assicurati che i valori del gruppo di posizionamento siano divisori comuni per le dimensioni della griglia dei domini che metti insieme.
> - Se non utilizzi i gruppi di collocamento:
> - Assicurati che 1 asse delle griglie del dominio spaziale abbia un divisore comune uguale al numero di lavoratori.
>
> Per ulteriori informazioni sui gruppi di collocamento, vedere. Strategie di partizionamento

Domande frequenti sui domini spaziali

D1. Come posso aggiungere un altro dominio spaziale a una simulazione esistente?

- Per una simulazione in esecuzione: non è possibile modificare la configurazione di una simulazione di esecuzione. Modifica la configurazione del dominio nello schema, carica lo schema e gli zip dell'app e avvia una nuova simulazione.
- Per una nuova simulazione: aggiungi la configurazione del dominio allo schema, carica lo schema e gli zip dell'app e avvia la nuova simulazione.

Risoluzione dei problemi relativi ai domini spaziali

È possibile che venga visualizzato il seguente errore quando si tenta di avviare la simulazione ma la configurazione del dominio non è valida.

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

Cause potenziali

- Lo schema alloca per le app un numero di unità di risorse di calcolo superiore a quello disponibile per i worker.
- SimSpace Weaver non è in grado di stabilire una disposizione per l'assegnazione dei domini ai lavoratori. Ciò accade quando si specificano più domini spaziali ma non esiste un divisore o un multiplo comune tra le griglie di dominio, ad esempio tra una griglia 2x4 e una griglia 3x5).

Endpoint di rete

Le app personalizzate e di servizio possono avere endpoint di rete a cui i client esterni possono connettersi. Si specifica un elenco di numeri di porta come valore per `ingress_ports` in. `endpoint_config` Questi numeri di porta sono sia TCP che UDP. L'app personalizzata o di servizio deve essere associata ai numeri di porta specificati in. `ingress_ports` SimSpace Weaver alloca dinamicamente i numeri di porta in fase di esecuzione e associa queste porte alle porte dinamiche. Puoi chiamare il app di descrizione API dopo che le app hanno iniziato a trovare i numeri di porta dinamici (effettivi). Per ulteriori informazioni, Ottieni l'indirizzo IP e il numero di porta di un'app personalizzata consulta il tutorial di avvio rapido.

```
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 I progetti app SDK versione 1.12.x utilizzano bucket separati per i file.zip dell'app e lo schema:
>
> - weaver- - -app-zips- *lowercase-project-name account-number region*
> - weaver- - -schemi- *lowercase-project-name account-number region*

> **Note**
>
> `endpoint_config`è una proprietà opzionale per app personalizzate e app di servizio. Se non specifichi un`endpoint_config`, l'app non avrà un endpoint di rete.

Configurazione dei domini di servizio

La presenza di `launch_apps_per_worker:` in una configurazione di dominio indica che si tratta di un dominio di servizio con app di servizio. SimSpace Weaver avvia e arresta automaticamente le app di servizio. Quando SimSpace Weaver avvia e arresta un'app, si considera che l'app abbia un ciclo di vita gestito. SimSpace Weaver attualmente supporta l'avvio di 1 o 2 app di servizio su ogni lavoratore.

Example Esempio di dominio configurato per avviare 1 app di servizio su ogni lavoratore

```
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 Esempio di dominio configurato per avviare 2 app di servizio su ciascun lavoratore

```
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
```

Durata massima di una simulazione

Ogni simulazione AWS SimSpace Weaver ha un'impostazione di durata massima che specifica il tempo massimo di esecuzione della simulazione. La durata massima viene fornita come parametro quando si avvia una simulazione. L'interfaccia di programmazione dell'`StartSimulation`applicazione (API) ha un parametro `MaximumDuration` opzionale. Il valore del parametro è un numero di minuti (m o M), ore (h o H) o giorni (d o D). Ad esempio, `1h` o `1H` significa 1 ora. SimSpace Weaver interrompe la simulazione quando raggiunge questo limite.

Valore massimo

Il valore più alto valido per `MaximumDuration` è`14D`, o il suo equivalente in ore (`336H`) o minuti (`20160M`).

Valore predefinito

Il parametro `MaximumDuration` è facoltativo. Se non fornisci un valore, SimSpace Weaver utilizza un valore di`14D`.

Valore minimo

Il valore più basso valido per `MaximumDuration` è un valore numericamente equivalente a`0`. Ad esempio, i valori`0M`, e `0H0D`, sono tutti equivalenti numericamente a. `0`

Se si fornisce il valore minimo per la durata massima, la simulazione passa immediatamente allo `STOPPING` stato non appena raggiunge lo stato. `STARTED`

Avvio di una simulazione tramite la console

È possibile fornire un valore per la durata massima quando si avvia una simulazione nella SimSpace Weaver console. Inserisci il valore nel campo Durata massima del modulo Impostazioni di simulazione quando scegli Avvia simulazione.

> **Important**
>
> Se non fornite un valore per Durata massima, SimSpace Weaver utilizza il valore predefinito ()`14D`.

Lo stato di una simulazione che raggiunge la sua durata massima

Quando interrompe SimSpace Weaver automaticamente una simulazione che raggiunge la sua durata massima, lo stato della simulazione è `STOPPING` (se in corso) o. `STOPPED` Nella SimSpace Weaver console, lo stato di destinazione della simulazione è fisso`STARTED`, poiché quello era l'ultimo stato richiesto da un utente.

Sviluppo di app

SimSpace Weaver lo sviluppo richiede un Amazon Linux 2 (AL2) ambiente in cui creare app perché le tue simulazioni vengono eseguite su Amazon Linux nel AWS Cloud. Se stai usando Windows, puoi utilizzare gli script nell'SDK dell' SimSpace Weaver app per creare e avviare un Docker contenitore che funziona AL2 con le dipendenze necessarie per creare SimSpace Weaver app. Puoi anche lanciare un AL2 ambiente utilizzando Windows Subsystem for Linux (WSL)o usa un nativo AL2 sistema. Per ulteriori informazioni, consulta Configura il tuo ambiente locale per SimSpace Weaver.

> **Note**
>
> Indipendentemente da come configuri il tuo ambiente di sviluppo locale, le tue app vengono eseguite in Docker contenitori quando li carichi per eseguirli in Cloud AWS. Le tue app non hanno accesso diretto al sistema operativo host.

Flusso generale di un' SimSpace Weaver app

1. Crea un'applicazione di .

2. Ciclo continuo:
   a. Inizia l'aggiornamento creando un`Transaction`.
      - Esci dal loop se la simulazione si sta chiudendo.
   b. Elabora gli eventi delle entità di sottoscrizione e proprietà.
   c. Aggiorna la simulazione.
   d. Impegnati `Transaction` a terminare l'aggiornamento.
3. Distruggi l'applicazione.

App spaziali

Ogni app spaziale ha un'area di proprietà che è una regione spaziale del mondo della simulazione. Le entità situate nell'area di proprietà di un'app spaziale vengono archiviate nella partizione assegnata all'app. La singola app spaziale ha la piena proprietà (autorizzazioni di lettura e scrittura) su tutte le entità all'interno della partizione assegnata. Nessun'altra app può scrivere su tali entità. L'app spaziale fa avanzare lo stato delle sue entità. Ogni app spaziale possiede solo 1 partizione. SimSpace Weaver utilizza la posizione spaziale di un'entità per indicizzarla e assegnarla a una partizione spaziale dell'app.

L'SDK dell' SimSpace Weaver app fornisce un'applicazione di esempio. Puoi trovare il codice sorgente per l'app spaziale dell'applicazione di esempio nella seguente cartella (usa il separatore di percorso corretto per il tuo sistema operativo):

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

App personalizzate

Puoi creare e utilizzare app personalizzate per interagire con la simulazione.

Le app personalizzate possono

- Creare entità
- Sottoscrivi ad altre partizioni
- Conferma le modifiche

Flusso generale di un'app personalizzata

1. Crea un'applicazione di .
2. Iscriviti a una regione specifica della simulazione:
   a. Crea un file `Transaction` per iniziare il primo aggiornamento.
   b. Crea un abbonamento per la regione specifica.
   c. `Transaction`Impegnati a terminare il primo aggiornamento.
3. Ciclo continuo:
   a. Crea un `Transaction` file per iniziare l'aggiornamento.
      - Esci dal loop se la simulazione si sta chiudendo.
   b. Modifiche allo stato del processo.
   c. `Transaction`Impegnati a terminare l'aggiornamento.
4. Distruggi l'applicazione.

Dopo che un'app personalizzata ha creato un'entità, deve trasferirla in un dominio spaziale affinché l'entità esista spazialmente all'interno della simulazione. SimSpace Weaver utilizza la posizione spaziale dell'entità per posizionarla nella partizione spaziale appropriata dell'app. L'app personalizzata che ha creato l'entità non può aggiornare o eliminare l'entità dopo averla trasferita in un dominio spaziale.

L'SDK SimSpace Weaver dell'app fornisce un'applicazione di esempio. Puoi utilizzare le app personalizzate incluse nell'applicazione di esempio come modelli per le tue app personalizzate. Puoi trovare il codice sorgente per l'app view (un'app personalizzata) dell'applicazione di esempio nella seguente cartella (usa il separatore di percorso corretto per il tuo sistema operativo):

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

Sviluppo di applicazioni client

Alcuni dei motivi per cui potresti voler connettere un client a una simulazione includono:

- Inserisci informazioni sul traffico in tempo reale in una simulazione su scala urbana.
- Crea human-in-the-loopsimulazioni, in cui un operatore umano controlla alcuni aspetti della simulazione.

- Consenti agli utenti di interagire con la simulazione, ad esempio per una simulazione di addestramento.

Le app personalizzate in questi esempi fungono da interfaccia tra lo stato di simulazione e il mondo esterno. I client si connettono alle app personalizzate per interagire con la simulazione.

SimSpace Weaver non gestisce le applicazioni client e la loro comunicazione con le app personalizzate. Sei responsabile della progettazione, della creazione, del funzionamento e della sicurezza delle tue applicazioni client e della loro comunicazione con le tue app personalizzate. SimSpace Weaver espone solo un indirizzo IP e un numero di porta per ciascuna delle tue app personalizzate in modo che i client possano connettersi ad esse.

L'SDK SimSpace Weaver dell'app fornisce client per la sua applicazione di esempio. È possibile utilizzare questi client come modelli per le proprie applicazioni client. È possibile trovare il codice sorgente per i client applicativi di esempio nella seguente cartella:

Docker

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

WSL

> **Important**
>
> Forniamo queste istruzioni per facilitarvi l'uso. Sono destinate all'uso con Windows Subsystem for Linux (WSL)e non sono supportati. Per ulteriori informazioni, consulta Configura il tuo ambiente locale per SimSpace Weaver.

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

Per ulteriori informazioni sulla creazione e l'utilizzo dei client applicativi di esempio, consulta i tutorial in. Guida introduttiva con SimSpace Weaver

Ottieni l'indirizzo IP e il numero di porta di un'app personalizzata

Per visualizzare la simulazione, devi creare un'app personalizzata e connetterti ad essa con un client. Per ulteriori informazioni, consulta i tutorial in. Guida introduttiva con SimSpace Weaver È possibile utilizzare la procedura seguente per ottenere l'indirizzo IP e il numero di porta dell'app personalizzata. Utilizzate il separatore di percorso appropriato per il vostro sistema operativo (ad esempio, \ in Windows e / Linux).

Per ottenere l'indirizzo IP e il numero di porta

1. Usa l' ListSimulationsAPI per ottenere il nome della tua simulazione.

```
aws simspaceweaver list-simulations
```

Output di esempio:

```
{
    "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. Usa l' DescribeSimulationAPI per ottenere un elenco di domini nella tua simulazione.

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

Cerca la `Domains` sezione nella `LiveSimulationState` sezione dell'output.

Output di esempio:

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

3. Usa l' ListAppsAPI per ottenere un elenco di app personalizzate in un dominio. Ad esempio, il nome di dominio per l'app di visualizzazione (personalizzata) nel progetto di esempio è`MyViewDomain`. Cerca il nome dell'app nell'output.

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

Output di esempio:

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

4. Usa l' DescribeAppAPI per ottenere l'indirizzo IP e il numero di porta. Per il progetto di esempio, il nome di dominio è `MyViewDomain` e il nome dell'app è`ViewApp`.

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

L'indirizzo IP e il numero di porta si trovano nel `EndpointInfo` blocco dell'output. L'indirizzo IP è il valore di `Address` e il numero di porta è il valore di`Actual`.

Output di esempio:

```
{
    "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**
>
> Il valore di `Declared` è il numero di porta a cui deve essere associato il codice dell'app. Il valore di `Actual` è il numero di porta che SimSpace Weaver i client possono connettersi alla tua app. SimSpace Weaver mappa la `Declared` porta alla `Actual` porta.

Avvio del client di visualizzazione Unreal Engine

Vai a:

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

1. Eseguire uno dei seguenti comandi:
   - Docker: `python quick-start.py`
   - WSL: `python quick-start.py --al2`
2. Ottieni l'indirizzo IP e il numero di porta «effettivo». Questi saranno presenti nell'output della console dopo l'esecuzione di quick-start.py, oppure li otterrete seguendo le procedure riportate inOttieni l'indirizzo IP e il numero di porta di un'app personalizzata.
3. Vai a:

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

4. Esegui i seguenti comandi per creare la libreria NNG:

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

5. In un editor di testo, apri`view_app_url.txt`.
6. Aggiorna l'URL con l'indirizzo IP e il numero di porta dell'app View: `tcp://ip-address:actual-port-number` (dovrebbe apparire così`tcp://198.51.100.135:1234`).
7. Nell'editor di Unreal, scegli play.

Risoluzione dei problemi

- La fase di CMake installazione di NNG fallisce con «Forse sono necessari i privilegi amministrativi «:

  ```
  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)
```

- Risoluzione: se `nng.lib` o `nng.so` esiste nella directory UnrealClient /lib, questo errore può essere tranquillamente ignorato. In caso contrario, prova a eseguire i comandi cmake build in un terminale con privilegi di amministratore.

- «CMake per trovare un file di configurazione del pacchetto fornito da 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.
```

- Risoluzione: CMake ha problemi a trovare il `Findnng.cmake` file. Quando costruisci con CMake, aggiungi l'argomento`-DTHIRD_PARTY_LIB_PATH sdk-folder/ThirdParty`. Assicurati che il `Findnng.cmake` file sia ancora nella `ThirdParty` directory prima di eseguire nuovamente la build. CMake

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

Sviluppo locale in SimSpace Weaver

Puoi distribuire SimSpace Weaver le tue applicazioni localmente per test e debug rapidi.

Requisiti

- Completa le fasi descritte in Configurazione per SimSpace Weaver.

Argomenti

- Fase 1: Avvia la simulazione locale
- Fase 2: Visualizza la simulazione locale
- Passaggio 3: Interrompi la simulazione locale (opzionale su Windows)
- Risoluzione dei problemi di sviluppo locale in SimSpace Weaver

Fase 1: Avvia la simulazione locale

1. Vai a

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

2. Esegui il seguente comando per creare e avviare la simulazione localmente.

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

Questo script eseguirà le seguenti operazioni:

1. Compilare il progetto.
   - `quick-start.py`chiama la `build_project` funzione definita in build.py. Questo passaggio varierà a seconda del progetto. Per il PathfindingSample, CMake viene utilizzato. Il comando CMake and Docker per cui è disponibile in build.py.
2. Avvia la tua simulazione locale
   - Lo script avvierà un processo locale per ogni partizione spaziale definita nello schema.
   - Lo script avvierà un processo per ogni app personalizzata definita nello schema.
   - Le app spaziali verranno avviate per prime, seguite dalle app personalizzate, ciascuna nell'ordine in cui appaiono nello schema.

> **Important**
>
> Quando si avvia in un ambiente che non supporta la GUI, ad esempio una sessione SSH della console, utilizza l'`--noappwindow`opzione per reindirizzare tutto l'output al terminale corrente.

> **Important**
>
> Per gli utenti Linux, lo script presuppone che il sistema disponga del comando. `xterm` Se la vostra distribuzione Linux non dispone del `xterm` comando, utilizzate l'`--noappwindow`opzione per reindirizzare tutto l'output al terminale corrente.

- -h, --help
  - Elenca questi parametri.
- --clean
  - Elimina il contenuto della cartella di compilazione prima della compilazione.
- --nobuild
  - Salta la ricostruzione del progetto.
- --nessuna finestra dell'app
  - Non aprire una nuova finestra per ogni app. Invece, reindirizza lo stdout al terminale corrente.
- --logfile
  - Scrive l'output della console in un file di registro.
- --consoleclient
  - Connetti automaticamente il client della console elencato nella configurazione.
- SCHEMA --schema
  - Quale schema utilizzerà questa invocazione. Il valore predefinito è 'SCHEMA' in config.py.

Fase 2: Visualizza la simulazione locale

Per visualizzare la simulazione locale, puoi utilizzare uno qualsiasi dei client inclusi in. SimSpaceWeaverAppSdkDistributable Per ulteriori informazioni sulla creazione e l'utilizzo dei client di esempio, consulta i tutorial in. Guida introduttiva con SimSpace Weaver

È necessario aggiornare l'indirizzo IP e il numero di porta nel client per connettersi all'app View per la simulazione locale. Usa sempre i seguenti valori con SimSpace Weaver Local:

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

A seconda del client selezionato, è possibile aggiornare l'indirizzo IP e il numero di porta come segue:

- Unreal: modifica l'URL sulla riga 1 di `view_app_url.txt`
- Console: avvia il client con l'indirizzo IP e il numero di porta (URL) come parametro

Passaggio 3: Interrompi la simulazione locale (opzionale su Windows)

> **Note**
>
> Questo passaggio è obbligatorio su Linux ma facoltativo su Windows.

1. Vai a:

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

2. Esegui il comando seguente per interrompere la simulazione locale ed eliminare tutte le risorse di memoria condivisa.

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

   Questo script eseguirà le seguenti operazioni:

   - Arresta i processi locali.
   - Elimina l'oggetto di memoria condivisa (necessario solo su Linux).

stop-and-deleteparametri.py

- -h, --help
  - Elenca questi parametri.
- --stop
  - Tenta solo di fermare i processi.
- --delete
  - Tenta solo di eliminare le risorse di memoria condivise.
- --process
  - Il nome del processo da interrompere. Utilizzatelo se il nome del processo non corrisponde al nome del pacchetto nello schema.
- SCHEMA --schema
  - Quale schema utilizzerà questa invocazione. Il valore predefinito è 'SCHEMA' in config.py.

Risoluzione dei problemi di sviluppo locale in SimSpace Weaver

- Linux: comando xterm non trovato/impossibile aprire
  - Gli script locali presuppongono che il comando xterm esista quando viene eseguito su Linux. Se non hai il comando xterm o sei in esecuzione in un ambiente che non supporta la GUI, usa l'`--noappwindow`opzione quando esegui lo script di avvio rapido.
- Nessuna finestra dell'app si apre!
  - Ciò accade quando la simulazione locale si blocca immediatamente. Per vedere l'output della console dopo l'arresto anomalo, usa le `--logfile` opzioni `--noappwindow` o quando esegui lo script di avvio rapido.
- La simulazione non funziona dopo l'avvio dell'app di visualizzazione o la visualizzazione delle connessioni del client!
  - L'esecuzione con l'`–noappwindow`opzione in genere risolve questo tipo di problemi. In caso contrario, anche il riavvio alcune volte ha esito positivo (sebbene a una velocità molto inferiore).

AWS SimSpace Weaver SDK dell'app

L'SDK SimSpace Weaver dell'app consente APIs di controllare le entità della simulazione e rispondere agli eventi. SimSpace Weaver Include il seguente spazio dei nomi:

- API: definizioni principali dell'API e del suo utilizzo

Collegamento con la seguente libreria:

- `libweaver_app_sdk_cxx_v1_full.so`

> **Important**
>
> La libreria è disponibile per il collegamento dinamico quando esegui le app in. Cloud AWS Non è necessario caricarla con le app.

> **Note**
>
> L'SDK SimSpace Weaver dell'app APIs controlla i dati all'interno della simulazione. Questi APIs sono separati dal SimSpace Weaver servizio APIs, che controlla le risorse SimSpace Weaver del servizio (come simulazioni, app e orologi). AWS Per ulteriori informazioni, consulta SimSpace Weaver Riferimenti API.

Argomenti

- I metodi API restituiscono un Result
- Interazione con l'SDK dell'app al livello più alto
- Gestione della simulazione
- Sottoscrizioni
- Entità
- eventi dell'entità
- Result e gestione degli errori
- Generici e tipi di dominio
- Operazioni varie dell'SDK dell'app

I metodi API restituiscono un Result

La maggior parte delle funzioni SimSpace Weaver API ha un tipo di ritornoAws::WeaverRuntime::Result<T>. Se la funzione è stata eseguita correttamente, Result contieneT. Altrimenti, Result contiene un file Aws::WeaverRuntime::ErrorCode che rappresenta un codice di errore proveniente da Rust App SDK.

Example Esempio

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

Questo metodo:

- Restituisce Transaction se BeginUpdate() viene eseguito correttamente.
- Restituisce Aws::WeaverRuntime::ErrorCode se BeginUpdate() fallisce.

Interazione con l'SDK dell'app al livello più alto

Ciclo di vita

- L'SDK SimSpace Weaver dell'app gestisce il ciclo di vita dell'app. Non è necessario leggere o scrivere lo stato del ciclo di vita di un'app.

Partizioni

- Usalo `Result <PartitionSet> AssignedPartitions(Transaction& txn);` per ottenere partizioni di proprietà.
- Usa `Result <PartitionSet> AllPartitions(Transaction& txn);` per ottenere tutte le partizioni della simulazione.

Gestione della simulazione

Questa sezione descrive le soluzioni per le attività più comuni di gestione della simulazione.

Argomenti

- Avvia una simulazione
- Aggiorna una simulazione
- Termina una simulazione

Avvia una simulazione

Utilizzalo `CreateApplication()` per creare un'app.

Example Esempio

```
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()));
```

Aggiorna una simulazione

Utilizza le seguenti `BeginUpdate` funzioni per aggiornare l'app:

- `Result<Transaction> BeginUpdate(Application& app)`
- `Result<bool> BeginUpdateWillBlock(Application& app)`— ti dice se `BeginUpdate()` bloccherà o non bloccherà.

Si usa `Result<void> Commit(Transaction& txn)` per confermare le modifiche.

Example Esempio

```
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();
}
```

Termina una simulazione

Utilizzare `Result<void> DestroyApplication(Application&& app)` per terminare l'app e la simulazione.

Altre app scoprono che la simulazione si interrompe quando ricevono `ErrorCode::ShuttingDown` chiamate verso o. `BeginUpdateWillBlock()` `BeginUpdate()` Quando un'app riceve`ErrorCode::ShuttingDown`, può chiamare `Result<void> DestroyApplication(Application&& app)` per terminarsi da sola.

Example Esempio

```
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**
>
> Chiama solo `Result<void> DestroyApplication(Application&& app)` dopo`Api::Commit()`. La distruzione di un'applicazione durante un aggiornamento può causare un comportamento indefinito.

> **Important**
>
> È necessario effettuare una chiamata `DestroyApplication()` prima della chiusura del programma per assicurarsi che l'applicazione riporti la chiusura con successo.
> La mancata chiamata alla `DestroyApplication()` chiusura del programma farà sì che lo stato venga considerato uguale. `FATAL`

Sottoscrizioni

Crei un abbonamento con un'area di abbonamento e un ID di dominio. L'ID di dominio rappresenta il dominio che possiede quell'area di sottoscrizione. A `BoundingBox2F32` descrive l'area di sottoscrizione. Utilizza la seguente funzione per creare un abbonamento:

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

Example Esempio

```
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();
}
```

È possibile utilizzare il comando `Api::SubscriptionHandle` restituito da `CreateSubscriptionBoundingBox2F32()` per modificare l'abbonamento. Lo si passa come argomento alle seguenti funzioni:

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

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

Entità

L'`Store`evento di cambio `Api:Entity` di proprietà `Result<Api::Entity>` restituito o `Load` APIs utilizzato viene richiamato quando un'entità entra nell'area di sottoscrizione dell'app (per ulteriori informazioni, consultaeventi dell'entità). `CreateEntity()` Ti consigliamo di tracciare i tuoi `Api::Entity` oggetti in modo da poterli utilizzare con questi APIs.

Argomenti

- Crea entità
- Trasferisci un'entità in un dominio spaziale

- Scrivi e leggi i dati dei campi dell'entità
- Memorizza la posizione di un'entità
- Carica la posizione di un'entità

Crea entità

Usa `CreateEntity()` per creare un'entità. Siete voi a definire il significato di `Api::TypeId` ciò che passate a questa funzione.

```
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**
>
> I valori 0-511 per `Api::BuiltinTypeId` sono riservati. La tua entità TypeID (`k_entityTypeId`in questo esempio) deve avere un valore pari o superiore a 512.

Trasferisci un'entità in un dominio spaziale

Dopo che un'app o un'app di servizio personalizzata ha creato un'entità, l'app deve trasferire l'entità in un dominio spaziale affinché l'entità esista spazialmente nella simulazione. Le entità in un dominio spaziale possono essere lette da altre app e aggiornate da un'app spaziale. Usa l'`ModifyEntityDomain()`API per trasferire un'entità in un dominio spaziale.

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

Se `DomainId` non corrisponde a quello assegnato `Partition` dall'app chiamante, `DomainId` deve essere per un `DomainType::SpatialDomain`. Il trasferimento della proprietà alla nuova `Domain` avviene durante il`Commit(Transaction&&)`.

Parametri

`txn`

La corrente`Transaction`.

`entity`

L'obiettivo `Entity` per il cambio di`Domain`.

`domainId`

`DomainId`La destinazione `Domain` per`Entity`.

Questa API restituisce `Success` se il dominio dell'entità è stato modificato con successo.

Scrivi e leggi i dati dei campi dell'entità

Tutti i campi di dati delle entità sono di tipo blob. È possibile scrivere fino a 1.024 byte di dati su un'entità. Ti consigliamo di mantenere i blob il più piccoli possibile perché dimensioni maggiori ridurranno le prestazioni. Quando si scrive su un blob, si passa SimSpace Weaver un puntatore ai dati e alla loro lunghezza. Quando si legge da un blob, SimSpace Weaver fornisce un puntatore e una lunghezza da leggere. Tutte le letture devono essere completate prima che l'app effettui una chiamata. `Commit()` I puntatori restituiti da una chiamata di lettura vengono invalidati quando l'app chiama. `Commit()`

> **Important**
>
> - La lettura da un puntatore blob memorizzato nella cache dopo a non `Commit()` è supportata e può causare il fallimento della simulazione.
> - La scrittura su un puntatore blob restituito da una chiamata di lettura non è supportata e può causare il fallimento della simulazione.

Argomenti

- Memorizza i dati del campo di un'entità

- Carica i dati del campo di un'entità
- Caricamento dei dati di campo delle entità rimosse

Memorizza i dati del campo di un'entità

Gli esempi seguenti mostrano come è possibile archiviare (scrivere nello State Fabric) i dati del campo di un'entità di proprietà dell'app. Questi esempi utilizzano la seguente funzione:

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

Il `Api::TypeId keyTypeId` parametro rappresenta il tipo di dati dei dati trasmessi.

Il `Api::TypeId keyTypeId` parametro deve ricevere il modulo corrispondente `Api::TypeIdApi::BuiltinTypeId`. Se non esiste una conversione appropriata, puoi usare`Api::BuiltinTypeId::Dynamic`.

Per tipi di dati complessi, usa`Api::BuiltInTypeId::Dynamic`.

> **Note**
>
> Il valore di `FieldIndex index` deve essere maggiore di 0. Il valore 0 è riservato alla chiave dell'indice (vedi`StoreEntityIndexKey()`).

Example Esempio di utilizzo di tipi di dati primitivi

```
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 Esempio di utilizzo di un struct per contenere i dati

```
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));
}
```

Carica i dati del campo di un'entità

Gli esempi seguenti mostrano come caricare (leggere dallo state fabric) i dati del campo di un'entità. Questi esempi utilizzano la seguente funzione:

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

Il `Api::TypeId keyTypeId` parametro dovrebbe ricevere il corrispondente `Api::TypeId` da`Api::BuiltinTypeId`. Se non esiste una conversione appropriata, puoi usare`Api::BuiltinTypeId::Dynamic`.

> **Note**
>
> Il valore dell'`FieldIndex`indice deve essere maggiore di 0. Il valore 0 è riservato alla chiave dell'indice (vedi`StoreEntityIndexKey()`).

Example Esempio di utilizzo di tipi di dati primitivi

```
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 Esempio di utilizzo di un struct per contenere i dati

```
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);
}
```

Caricamento dei dati di campo delle entità rimosse

Non puoi caricare (leggere dallo State Fabric) i dati dei campi di entità per le entità che sono state rimosse dalle aree di proprietà e abbonamento dell'app. L'esempio seguente

genera un errore perché richiama `Api::LoadIndexKey()` un'entità come risultato di un`Api::ChangeListAction::Remove`. Il secondo esempio mostra un modo corretto per archiviare e caricare i dati delle entità direttamente nell'app.

Example Esempio di codice errato

```
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 Esempio di un modo corretto per archiviare e caricare i dati delle entità nell'app

```
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;
            }
        }
    }

    /* ... */
}
```

Memorizza la posizione di un'entità

È possibile memorizzare (scrivere nel tessuto statale) la posizione di un'entità utilizzando una struttura di dati intera. Questi esempi utilizzano la seguente funzione:

```
Result<void> StoreEntityIndexKey(
    Transaction& txn,
    const Entity& entity,
```

```
    TypeId keyTypeId,
    std::int8_t* src,
    std::size_t length)
```

> **Note**
>
> È necessario fornire `Api::BuiltinTypeId::Vector3F32` a`Api::StoreEntityIndexKey()`, come illustrato negli esempi seguenti.

Example Esempio di utilizzo di un array per rappresentare la posizione

```
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 Esempio di utilizzo di un struct per rappresentare la posizione

```
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));
}
```

Carica la posizione di un'entità

È possibile caricare (leggere dalla struttura degli stati) la posizione di un'entità utilizzando una struttura di dati intera. Questi esempi utilizzano la seguente funzione:

> **Note**
>
> È necessario fornire `Api::BuiltinTypeId::Vector3F32` a`Api::LoadEntityIndexKey()`, come illustrato negli esempi seguenti.

Example Esempio di utilizzo di un array per rappresentare la posizione

```
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 Esempio di utilizzo di un struct per rappresentare la posizione

```
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);
}
```

eventi dell'entità

Puoi utilizzare le seguenti funzioni nell'SDK dell' SimSpace Weaver app per ottenere tutti gli eventi di proprietà e abbonamento:

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

È possibile utilizzare il framework SimSpace Weaver demo se è necessaria l'elaborazione di eventi di entità basata su callback. Per ulteriori informazioni, consulta il seguente file di intestazione:

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

Puoi anche creare l'elaborazione degli eventi della tua entità.

Argomenti

- Esegui iterazioni tra gli eventi per le entità di proprietà
- Esegui un'iterazione tra gli eventi per le entità sottoscritte
- Esegui un'iterazione degli eventi di cambio di proprietà delle entità

Esegui iterazioni tra gli eventi per le entità di proprietà

`OwnershipChanges()`Utilizzalo per ottenere un elenco di eventi per le entità di proprietà (entità nell'area di proprietà dell'app). La funzione ha la seguente firma:

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

Quindi esegui un'iterazione tra le entità con un ciclo, come illustrato nell'esempio seguente.

Example Esempio

```
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;
    }
```

```
}
```

Event types (Tipi di evento)

- `None`— L'entità si trova nell'area e i dati relativi alla posizione e al campo non sono stati modificati.
- `Remove`— L'entità è stata rimossa dall'area.
- `Add`— L'entità è stata aggiunta all'area.
- `Update`— L'entità si trova nell'area ed è stata modificata.
- `Reject`— L'app non è riuscita a rimuovere l'entità dall'area.

> **Note**
>
> In caso di `Reject` evento, l'app tenterà nuovamente il trasferimento con il segno di spunta successivo.

Esegui un'iterazione tra gli eventi per le entità sottoscritte

Utilizzalo `AllSubscriptionEvents()` per ottenere un elenco di eventi per le entità sottoscritte (entità nell'area di iscrizione dell'app). La funzione ha la seguente firma:

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

Quindi esegui un'iterazione tra le entità con un ciclo, come illustrato nell'esempio seguente.

Example Esempio

```
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;
    }
}
```

Event types (Tipi di evento)

- `None`— L'entità si trova nell'area e i dati relativi alla posizione e al campo non sono stati modificati.
- `Remove`— L'entità è stata rimossa dall'area.
- `Add`— L'entità è stata aggiunta all'area.
- `Update`— L'entità si trova nell'area ed è stata modificata.
- `Reject`— L'app non è riuscita a rimuovere l'entità dall'area.

> **Note**
>
> In caso di `Reject` evento, l'app tenterà nuovamente il trasferimento con il segno di spunta successivo.

Esegui un'iterazione degli eventi di cambio di proprietà delle entità

Per visualizzare gli eventi in cui un'entità si sposta tra un'area di proprietà e un'area di sottoscrizione, confronta le modifiche tra gli eventi di proprietà e sottoscrizione dell'entità correnti e precedenti.

Puoi gestire questi eventi leggendo:

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

È quindi possibile confrontare le modifiche con i dati precedentemente memorizzati.

L'esempio seguente mostra come gestire gli eventi di modifica della proprietà delle entità. Questo esempio presuppone che per le entità che passano dall'essere entità sottoscritte a entità possedute (in entrambe le direzioni), l'remove/add event occurs first followed by the subscription remove/addevento di proprietà sia visualizzato nel segno di spunta successivo.

Example Esempio

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

    static EntityIdsByAction m_entityIdsByPreviousOwnershipAction;

    EntityIdSetByAction entityIdSetByAction;

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

    for (const Api::SubscriptionEvent& event : subscriptionEvents.changes)
    {
        const Api::ChangeListAction action = event.action;

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

            {
                entityIdSetByAction[action].insert(
                    event.entity.descriptor->id);
                break;
            }
```

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

    EntityIdsByAction entityIdsByAction;

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

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

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

        switch (action)
        {
        case Api::ChangeListAction::Add:
        case Api::ChangeListAction::Remove:
            {
                entityIdsByAction[action].push_back(
                    event.entity.descriptor->id);
                break;
            }
        case Api::ChangeListAction::None:
        case Api::ChangeListAction::Update:
        case Api::ChangeListAction::Reject:
            {
                break;
            }
        }

    }

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

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


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

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

    m_entityIdsByPreviousOwnershipAction = entityIdsByOwnershipAction;
```

```
    return Success();
}
```

Result e gestione degli errori

La `Aws::WeaverRuntime::Result<T>` classe utilizza una `Outcome` libreria di terze parti. È possibile utilizzare lo schema seguente per verificare gli errori `Result` e catch restituiti dalle chiamate API.

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

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

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

Result istruzione di controllo, macro

All'interno di una funzione con un tipo restituito`Aws::WeaverRuntime::Result<T>`, è possibile utilizzare la `WEAVERRUNTIME_TRY` macro anziché lo schema di codice precedente. La macro eseguirà la funzione che le è stata passata. Se la funzione passata fallisce, la macro farà in modo che la funzione di inclusione restituisca un errore. Se la funzione passata ha esito positivo, l'esecuzione passa alla riga successiva. L'esempio seguente mostra una riscrittura della funzione precedente. `DoBeginUpdate()` Questa versione utilizza la

`WEAVERRUNTIME_TRY` macro anziché if-else struttura di controllo. Nota che il tipo restituito dalla funzione è`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()` fallisce, la macro `DoBeginUpdate()` restituisce anticipatamente con un errore. È possibile utilizzare la `WEAVERRUNTIME_EXPECT_ERROR` macro per ottenere il file `Aws::WeaverRuntime::ErrorCode` da`BeginUpdate()`. L'esempio seguente mostra come la `Update()` funzione chiama `DoBeginUpdate()` e ottiene il codice di errore in caso di errore.

```
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);
```

```
    }
}
```

È possibile rendere `BeginUpdate()` disponibile il codice di errore per una funzione che chiama `Update()` modificando il tipo restituito da `Update()` to`Aws::WeaverRuntime::Result<void>`. È possibile ripetere questo processo per continuare a inviare il codice di errore più in basso nello stack di chiamate.

Generici e tipi di dominio

L'SDK SimSpace Weaver dell'app fornisce i tipi di dati a precisione singola `Api::Vector2F32` e`Api::BoundingBox2F32`, a doppia precisione e. `Api::Vector2F64` `Api::BoundingBox2F64` Questi tipi di dati sono strutture di dati passive prive di metodi pratici. Tieni presente che l'API utilizza solo `Api::Vector2F32` e`Api::BoundingBox2F32`. Puoi utilizzare questi tipi di dati per creare e modificare abbonamenti.

Il framework SimSpace Weaver demo fornisce una versione minima di AzCore libreria matematica, che contiene `Vector3` e`Aabb`. Per ulteriori informazioni, consulta i file di intestazione in:

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

Operazioni varie dell'SDK dell'app

Argomenti

- AllSubscriptionEvents e OwnershipChanges contengono gli eventi dell'ultima chiamata
- Rilascia i blocchi di lettura dopo l'elaborazione SubscriptionChangeList
- Crea un'istanza di app autonoma per il test

AllSubscriptionEvents e OwnershipChanges contengono gli eventi dell'ultima chiamata

I valori restituiti dalle chiamate a `Api::AllSubscriptionEvents()` e `Api::OwnershipChanges()` contengono gli eventi dell'ultima chiamata, non l'ultimo segno di spunta. Nell'esempio seguente, `secondSubscriptionEvents` e `secondOwnershipChangeList` sono vuoti perché le relative funzioni vengono richiamate immediatamente dopo le prime chiamate.

Se attendi 10 tick e poi chiami `Api::AllSubscriptionEvents()` and`Api::OwnershipChanges()`, i risultati conterranno sia gli eventi che le modifiche rispetto agli ultimi 10 tick (non all'ultimo segno di spunta).

Example Esempio

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

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

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

> **Note**
>
> La funzione **`AllSubscriptionEvents()`** è implementata ma la funzione non **`SubscriptionEvents()`** è implementata.

Rilascia i blocchi di lettura dopo l'elaborazione SubscriptionChangeList

Quando si avvia un aggiornamento, sono presenti segmenti di memoria condivisa per i dati salvati in altre partizioni relative al segno di spunta precedente. Questi segmenti di memoria condivisa potrebbero essere bloccati dai lettori. Un'app non può eseguire il commit completo finché tutti i lettori non hanno rilasciato i lucchetti. Come ottimizzazione, un'app dovrebbe chiamare `Api::ReleaseReadLeases()` per rilasciare i blocchi dopo aver

`Api::SubscriptionChangelist` elaborato gli elementi. Ciò riduce le contese al momento del commit. `Api::Commit()`rilascia i lease di lettura per impostazione predefinita, ma è consigliabile rilasciarli manualmente dopo l'elaborazione degli aggiornamenti dell'abbonamento.

Example Esempio

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

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

    WEAVERRUNTIME_EXPECT(Api::ReleaseReadLeases(transaction));

    ...
}
```

Crea un'istanza di app autonoma per il test

Puoi utilizzarla `Api::CreateStandaloneApplication()` per creare un'app autonoma per testare la logica dell'app prima di eseguire il codice in una simulazione effettiva.

Example Esempio

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

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

    ...
}
```

AWS SimSpace Weaver quadro dimostrativo

Il framework AWS SimSpace Weaver demo (framework demo) è una libreria di utilità che puoi utilizzare per sviluppare SimSpace Weaver app.

Il framework demo fornisce

- Esempi di codice e modelli di programmazione da utilizzare ed esaminare
- Astrazioni e funzioni di utilità che semplificano lo sviluppo di app semplici
- Un modo più semplice per testare le funzionalità sperimentali dell'SDK dell'app SimSpace Weaver

Abbiamo progettato l'SDK SimSpace Weaver dell'app con accesso a basso livello per offrire SimSpace Weaver APIs prestazioni più elevate. Al contrario, abbiamo progettato il framework demo per fornire astrazioni di livello superiore e un accesso a APIs ciò più facile da usare. SimSpace Weaver Il costo della facilità d'uso è un livello di prestazioni inferiore rispetto all'utilizzo diretto dell'SDK dell' SimSpace Weaver app. Le simulazioni in grado di tollerare prestazioni inferiori (ad esempio quelle senza requisiti di prestazioni in tempo reale) potrebbero essere buone candidate per utilizzare il framework demo. Ti consigliamo di utilizzare la funzionalità nativa dell'SDK dell' SimSpace Weaver app per applicazioni complesse, poiché il framework demo non è un toolkit completo.

Il framework demo include

- Esempi di codice funzionanti che supportano e dimostrano:
  - Gestione del flusso delle app
  - Elaborazione di eventi di entità basata su callback
- Un set di librerie di utilità di terze parti:
  - spdlog (una libreria di registrazione)
  - Una versione minimale di AZCore (una libreria matematica) che contiene solo:
    - Vector3
    - Aabb
  - cxxopts (una libreria di analisi delle opzioni a riga di comando)
- Funzioni di utilità specifiche per SimSpace Weaver

Il framework demo è composto da una libreria, file sorgente e CMakeLists. I file sono inclusi nel pacchetto distribuibile dell' SimSpace Weaver app SDK.

Utilizzo delle quote di servizio

Questa sezione descrive come utilizzare le quote di servizio per. SimSpace WeaverLe quote sono anche chiamate limiti. Per un elenco delle quote di servizio, vedere. SimSpace Endpoint e quote Weaver I APIs contenuti in questa sezione provengono dal set di app APIs. APIsLe app sono diverse dal servizio APIs. Le app APIs fanno parte dell'SDK dell' SimSpace Weaver app. Puoi trovare la documentazione per l'app APIs nella cartella SDK dell'app sul tuo sistema locale:

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

Argomenti

- Ottieni i limiti per un'app
- Ottieni la quantità di risorse utilizzate da un'app
- Reimposta le metriche
- Superamento di un limite
- Memoria a corto di memoria
- Best practice

Ottieni i limiti per un'app

Puoi utilizzare il plugin RuntimeLimits API dell'app per interrogare i limiti di un'app.

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

Parametri

Application& app

Un riferimento all'app.

LimitType tipo

Un enum con i seguenti tipi di limite:

```
enum LimitType {
    Unset = 0,
```

```
    EntitiesPerPartition = 1,
    RemoteEntityTransfers = 2,
    LocalEntityTransfers = 3
};
```

L'esempio seguente interroga il limite di conteggio delle entità.

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

Ottieni la quantità di risorse utilizzate da un'app

Puoi chiamare il RuntimeMetrics API dell'app per ottenere la quantità di risorse utilizzate da un'app:

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

Parametri

Application& app

Un riferimento all'app.

L'API restituisce un riferimento a struct che contiene le metriche. Una metrica contatore contiene un valore totale corrente e non fa che aumentare. Una metrica dell'indicatore contiene un valore che può aumentare o diminuire. Il runtime dell'applicazione aggiorna un contatore ogni volta che un evento aumenta il valore. Il runtime aggiorna gli indicatori solo quando si chiama l'API. SimSpace Weaver garantisce che il riferimento sia valido per tutta la durata dell'app. Le chiamate ripetute all'API non modificheranno il riferimento.

```
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
}
```

Reimposta le metriche

Il ResetRuntimeMetrics L'API dell'app reimposta i valori in `AppRuntimeMetrics` struct.

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

L'esempio seguente mostra come è possibile chiamare ResetRuntimeMetrics nella tua 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));
}
```

Superamento di un limite

Una chiamata all'API dell'app che supera un limite restituirà un`ErrorCode::CapacityExceeded`, ad eccezione dei trasferimenti di entità. SimSpace Weaver gestisce i trasferimenti di entità in modo asincrono come parte di Commit e BeginUpdate operazioni dell'API dell'app, quindi non esiste un'operazione specifica che restituisca un errore se un trasferimento non riesce a causa del limite di trasferimento dell'entità. Per rilevare gli errori di trasferimento, puoi confrontare i valori correnti di `rejected_incoming_transfer_counter` e `rejected_outgoing_transfer_counter` (nel `AppRuntimeMetrics` struct) con i valori precedenti. Le entità rifiutate non saranno presenti nella partizione, ma l'app può comunque simularle.

Memoria a corto di memoria

SimSpace Weaver utilizza un processo Garbage Collector per ripulire e liberare la memoria liberata. È possibile scrivere dati più velocemente di quanto il Garbage Collector riesca a liberare memoria. In tal caso, le operazioni di scrittura potrebbero superare il limite di memoria riservata dell'app. SimSpace Weaver restituirà un errore interno con un messaggio che contiene `OutOfMemory` (e dettagli aggiuntivi). Per ulteriori informazioni, consulta Diffondi le scritture nel tempo.

Best practice

Le seguenti best practice sono linee guida generali per progettare le app in modo da evitare il superamento dei limiti. Potrebbero non essere applicabili al design specifico della tua app.

Monitora frequentemente e rallenta

È necessario monitorare frequentemente le metriche e rallentare le operazioni che stanno per raggiungere un limite.

Evita di superare i limiti di abbonamento e i limiti di trasferimento

Se possibile, progetta la simulazione per ridurre il numero di abbonamenti remoti e trasferimenti di entità. È possibile utilizzare i gruppi di collocamento per posizionare più partizioni sullo stesso lavoratore e ridurre la necessità di trasferimenti di entità remote tra lavoratori.

Diffondi le scritture nel tempo

Il numero e la dimensione degli aggiornamenti in un colpo di spunta possono avere un impatto significativo sul tempo e sulla memoria necessari per eseguire una transazione. I requisiti di memoria elevati possono causare l'esaurimento della memoria nel runtime dell'applicazione. È possibile distribuire le scritture nel tempo per ridurre la dimensione totale media degli aggiornamenti per segno di spunta. Questo può aiutare a migliorare le prestazioni ed evitare il superamento dei limiti. Ti consigliamo di non scrivere più di una media di 12 MB per segno di spunta o 1,5 KB per ogni entità.

Simulazioni di debug

È possibile utilizzare i seguenti metodi per ottenere informazioni sulle simulazioni.

Argomenti

- Utilizzo SimSpace Weaver Local e guarda l'output della console

- Guarda i tuoi log in Amazon CloudWatch Logs
- Utilizzo describe Chiamate API
- Connect a client

Utilizzo SimSpace Weaver Local e guarda l'output della console

Ti consigliamo di sviluppare prima le simulazioni localmente e poi di eseguirle Cloud AWS in. È possibile visualizzare l'output della console direttamente quando si esegue con SimSpace Weaver Local. Per ulteriori informazioni, vedereSviluppo locale in SimSpace Weaver.

Guarda i tuoi log in Amazon CloudWatch Logs

Quando esegui la simulazione nella console, Cloud AWS l'output delle tue app viene inviato ai flussi di log in Amazon CloudWatch Logs. La tua simulazione scrive anche altri dati di log. È necessario abilitare la registrazione nello schema di simulazione se si desidera che la simulazione scriva dati di registro. Per ulteriori informazioni, consulta SimSpace Weaver accessi in Amazon CloudWatch Logs.

> **Warning**
>
> La simulazione può produrre grandi quantità di dati di registro. I dati di registro possono crescere molto rapidamente. È necessario monitorare attentamente i log e interrompere le simulazioni quando non è più necessario eseguirle. La registrazione può comportare costi elevati.

Utilizzo describe Chiamate API

È possibile utilizzare il seguente servizio APIs per ottenere informazioni sulle simulazioni in. Cloud AWS

- ListSimulations— ottieni un elenco di tutte le tue simulazioni in. Cloud AWS

  Example Esempio

  ```
  aws simspaceweaver list-simulations
  ```

- DescribeSimulation— ottieni dettagli su una simulazione.

Example Esempio

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

- DescribeApp— ottieni dettagli su un'app.

  Example Esempio

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

Per ulteriori informazioni su SimSpace Weaver APIs, vedereSimSpace Weaver Riferimenti API.

Connect a client

Puoi connettere un client a un'app personalizzata o di servizio `endpoint_config` in esecuzione definita con uno schema di simulazione. L'SDK SimSpace Weaver dell'app include client di esempio che puoi utilizzare per visualizzare l'applicazione di esempio. Puoi esaminare il codice sorgente di questi client di esempio e l'applicazione di esempio per vedere come creare i tuoi client. Per ulteriori informazioni su come creare ed eseguire i client di esempio, consulta i tutorial in. Guida introduttiva con SimSpace Weaver

Puoi trovare il codice sorgente per i client di esempio nella seguente cartella:

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

Eseguire il debug di simulazioni locali

Puoi eseguire il debug delle tue SimSpace Weaver Local app con. Microsoft Visual Studio Per ulteriori informazioni su come eseguire il debug conVisual Studio, consulta. Microsoft Visual Studio documentation

Per eseguire il debug della simulazione locale

1. Assicurati che `schema.yaml` sia nella tua directory di lavoro.
2. In Visual Studio, apri il menu contestuale per ogni app di cui desideri eseguire il debug (ad esempio `PathfindingSampleLocalSpatial` o`PathfindingSampleLocalView`) e imposta la directory di lavoro nella sezione di debug.

3. Apri il menu contestuale dell'app di cui desideri eseguire il debug e seleziona Imposta come progetto di avvio.
4. Scegli F5 di avviare il debug dell'app.

I requisiti per eseguire il debug di una simulazione sono gli stessi necessari per eseguire una simulazione normalmente. È necessario avviare il numero di app spaziali specificato nello schema. Ad esempio, se lo schema specifica una griglia 2x2 e si avvia un'app spaziale in modalità debug, la simulazione non verrà eseguita finché non si avviano altre 3 app spaziali (in modalità debug o non in modalità debug).

Per eseguire il debug di un'app personalizzata, devi prima avviare le app spaziali e quindi avviare l'app personalizzata nel debugger.

Tieni presente che la simulazione viene eseguita in blocco. Non appena un'app raggiunge un punto di interruzione, tutte le altre app verranno messe in pausa. Dopo aver proseguito da quel punto di interruzione, le altre app continueranno.

Contenitori personalizzati

AWS SimSpace Weaver le app vengono eseguite in contenitori Amazon Linux 2 (AL2) ambienti. Nel Cloud AWS, SimSpace Weaver esegue le simulazioni in contenitori Docker creati a partire da un'`amazonlinux:2`immagine fornita da Amazon Elastic Container Registry (Amazon ECR). Puoi creare un'immagine Docker personalizzata, archiviarla in Amazon ECR e utilizzare quell'immagine per la tua simulazione anziché l'immagine Docker predefinita che forniamo.

Puoi utilizzare un contenitore personalizzato per gestire le dipendenze software e includere componenti software aggiuntivi che non sono presenti nell'immagine Docker standard. Ad esempio, puoi aggiungere al contenitore le librerie software disponibili pubblicamente utilizzate dall'app e inserire il codice personalizzato solo nel file zip dell'app.

> **Important**
>
> Supportiamo solo immagini AL2 Docker ospitate nei repository Amazon ECR, in Amazon ECR Public Gallery o nel tuo registro Amazon ECR privato. Non supportiamo immagini Docker ospitate al di fuori di Amazon ECR. Per ulteriori informazioni su Amazon ECR, consulta la documentazione di Amazon Elastic Container Registry.

Argomenti

- Crea un contenitore personalizzato
- Modifica un progetto per utilizzare un contenitore personalizzato
- Domande frequenti sui contenitori personalizzati
- Risoluzione dei problemi relativi ai contenitori personalizzati

Crea un contenitore personalizzato

Queste istruzioni presuppongono che tu sappia usare Docker e Amazon Elastic Container Registry (Amazon ECR). Per ulteriori informazioni su Amazon ECR, consulta la Guida per l'utente di Amazon ECR.

Prerequisiti

- L'identità IAM (uso o ruolo) che usi per eseguire queste azioni dispone delle autorizzazioni corrette per utilizzare Amazon ECR
- Docker è installato sul tuo sistema locale

Per creare un contenitore personalizzato

1. Crea un `Dockerfile`.

   Una AWS SimSpace Weaver app `Dockerfile` per eseguire inizia con Amazon Linux 2 immagine in Amazon ECR.

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

2. Costruisci il tuo. `Dockerfile`
3. Carica l'immagine del contenitore su Amazon ECR.
   - Usa il. AWS Management Console
   - Usa il AWS Command Line Interface.

   > **Note**
   >
   > Se ricevi un `AccessDeniedException` errore quando tenti di caricare l'immagine del contenitore su Amazon ECR, la tua identità IAM (utente o ruolo) potrebbe non disporre

delle autorizzazioni necessarie per utilizzare Amazon ECR. Puoi collegare la policy `AmazonEC2ContainerRegistryPowerUser` AWS gestita alla tua identità IAM e riprovare. Per ulteriori informazioni su come allegare una policy, consulta Aggiungere e rimuovere le autorizzazioni di identità IAM nella Guida per l'AWS Identity and Access Management utente.

Modifica un progetto per utilizzare un contenitore personalizzato

Queste istruzioni presuppongono che tu sappia già come usarle AWS SimSpace Weaver e che desideri rendere Cloud AWS più efficienti i flussi di lavoro di archiviazione e sviluppo delle app.

Prerequisiti

- Hai un contenitore personalizzato in Amazon Elastic Container Registry (Amazon ECR). Per ulteriori informazioni sulla creazione di un contenitore personalizzato, consultaCrea un contenitore personalizzato.

Per modificare il progetto in modo da utilizzare un contenitore personalizzato

1. Aggiungi le autorizzazioni al ruolo dell'app di simulazione del tuo progetto per utilizzare Amazon ECR.

   a. Se non disponi già di una policy IAM con le seguenti autorizzazioni, crea la policy. Ti suggeriamo il nome `simspaceweaver-ecr` della policy. Per ulteriori informazioni su come creare una policy IAM, consulta Creazione di policy IAM nella Guida per l'AWS Identity and Access Management utente.

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

```
        }
    ]
}
```

b. Trova il nome del ruolo dell'app di simulazione del tuo progetto:

   i. In un editor di testo, apri il AWS CloudFormation modello:

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

   ii. Trova la `RoleName` proprietà sotto`WeaverAppRole`. Il valore è il nome del ruolo dell'app di simulazione del progetto.

   Example

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

c. Allega la `simspaceweaver-ecr` policy al ruolo dell'app di simulazione del progetto. Per ulteriori informazioni su come allegare una policy, consulta Aggiungere e rimuovere le autorizzazioni di identità IAM nella Guida per l'AWS Identity and Access Management utente.

d. Vai al seguente comando *`sdk-folder`* ed esegui il seguente comando per aggiornare lo SimSpace Weaver stack di esempio:

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

2. Specificate le immagini del contenitore nello schema di simulazione del progetto.

   - Puoi aggiungere la `default_image` proprietà opzionale in `simulation_properties` per specificare un'immagine contenitore personalizzata predefinita per tutti i domini.

- Aggiungi la `image` proprietà nel campo `app_config` per il dominio in cui desideri utilizzare un'immagine contenitore personalizzata. Specificare l'URI del repository Amazon ECR come valore. Puoi specificare un'immagine diversa per ogni dominio.
  - Se `image` non è specificato per un dominio e `default_image` viene specificato, le app di quel dominio utilizzano l'immagine predefinita.
  - Se `image` non è specificato per un dominio e `default_image` non è specificato, le app di quel dominio vengono eseguite in un SimSpace Weaver contenitore standard.

Example Frammento di schema che include impostazioni personalizzate del contenitore

```
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. Crea e carica il tuo progetto come al solito.

Domande frequenti sui contenitori personalizzati

D1. Cosa devo fare se voglio modificare il contenuto del mio contenitore?

- Per una simulazione di esecuzione: non è possibile cambiare il contenitore per una simulazione di esecuzione. È necessario creare un nuovo contenitore e avviare una nuova simulazione che utilizzi quel contenitore.
- Per una nuova simulazione: crea un nuovo contenitore, caricalo su Amazon Elastic Container Registry (Amazon ECR) e avvia una nuova simulazione che utilizzi quel contenitore.

D2. Come posso modificare l'immagine del contenitore per la mia simulazione?

- Per una simulazione in esecuzione: non è possibile modificare il contenitore per una simulazione in esecuzione. È necessario avviare una nuova simulazione che utilizzi il nuovo contenitore.
- Per una nuova simulazione: specifica la nuova immagine del contenitore nello schema di simulazione del progetto. Per ulteriori informazioni, consulta Modifica un progetto per utilizzare un contenitore personalizzato.

Risoluzione dei problemi relativi ai contenitori personalizzati

Argomenti

- AccessDeniedException quando carichi l'immagine su Amazon Elastic Container Registry (Amazon ECR)
- Una simulazione che utilizza un contenitore personalizzato non riesce ad avviarsi

AccessDeniedException quando carichi l'immagine su Amazon Elastic Container Registry (Amazon ECR)

Se ricevi un `AccessDeniedException` errore quando tenti di caricare l'immagine del contenitore su Amazon ECR, la tua identità IAM (utente o ruolo) potrebbe non disporre delle autorizzazioni necessarie per utilizzare Amazon ECR. Puoi collegare la policy

`AmazonEC2ContainerRegistryPowerUser` AWS gestita alla tua identità IAM e riprovare. Per ulteriori informazioni su come allegare una policy, consulta Aggiungere e rimuovere le autorizzazioni di identità IAM nella Guida per l'AWS Identity and Access Management utente.

Una simulazione che utilizza un contenitore personalizzato non riesce ad avviarsi

Suggerimenti per la risoluzione dei problemi

- Se la registrazione è abilitata per la simulazione, controlla i log degli errori.
- Testa la tua simulazione senza un contenitore personalizzato.
- Testa la tua simulazione localmente. Per ulteriori informazioni, consulta Sviluppo locale in SimSpace Weaver.

Uso di Python

Puoi usare Python per le tue SimSpace Weaver app e il tuo client. Il kit di sviluppo software Python (Python SDK) è incluso come parte del pacchetto distribuibile SDK per app standard SimSpace Weaver . Lo sviluppo con Python funziona in modo simile allo sviluppo negli altri linguaggi supportati.

> **Important**
>
> SimSpace Weaver supporta solo la versione 3.9 di Python.

> **Important**
>
> SimSpace Weaver il supporto per Python richiede la SimSpace Weaver versione 1.15.0 o successiva.

Argomenti

- Creare un progetto Python
- Avvio di una simulazione in Python
- Il client Python di esempio
- Domande frequenti sull'uso di Python
- Risoluzione dei problemi relativi a Python

Creare un progetto Python

Contenitore personalizzato Python

Per eseguire la SimSpace Weaver simulazione basata su Python in Cloud AWS, puoi creare un contenitore personalizzato che includa le dipendenze necessarie. Per ulteriori informazioni, consulta Contenitori personalizzati.

Un contenitore personalizzato in Python deve includere quanto segue:

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

Se usi il `PythonBubblesSample` modello per creare il tuo progetto, puoi eseguire lo `quick-start.py` script (che si trova nella `tools` cartella del progetto) per creare un'immagine Docker con le dipendenze necessarie. Lo script carica l'immagine su Amazon Elastic Container Registry (Amazon ECR).

Lo `quick-start.py` script utilizza quanto segue: `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
```

Puoi aggiungere le tue dipendenze a: `Dockerfile`

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

Il `requirements.txt` file contiene un elenco di pacchetti Python necessari per la simulazione di `PythonBubblesSample` esempio:

```
Flask==2.1.1
```

Puoi aggiungere le tue dipendenze dei pacchetti Python a: `requirements.txt`

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

I `Dockerfile` e `requirements.txt` si trovano nella `tools` cartella del tuo progetto.

> **Important**
>
> Tecnicamente non è necessario utilizzare un contenitore personalizzato con la simulazione Python, ma consigliamo vivamente di utilizzare un contenitore personalizzato. Il contenitore Amazon Linux 2 (AL2) standard che forniamo non ha Python. Pertanto, se non utilizzi un contenitore personalizzato con Python, devi includere Python e le dipendenze richieste in ogni file zip dell'app in cui carichi. SimSpace Weaver

Avvio di una simulazione in Python

Puoi iniziare la tua simulazione basata su Python allo stesso modo di una normale SimSpace Weaver simulazione, sia in SimSpace Weaver Local e in. SimSpace Weaver Cloud AWS Per ulteriori informazioni, consulta i tutorial in. Guida introduttiva con SimSpace Weaver

`PythonBubblesSample`Include il proprio client di esempio Python. Per ulteriori informazioni, consulta Il client Python di esempio.

Il client Python di esempio

Se usi il `PythonBubblesSample` modello per creare un progetto, il progetto contiene un client di esempio Python. È possibile utilizzare il client di esempio per visualizzare la `PythonBubblesSample` simulazione. Puoi anche usare il client di esempio come punto di partenza per creare il tuo client Python.

La procedura seguente presuppone che tu abbia creato un `PythonBubblesSample` progetto e ne abbia avviato la simulazione.

Per avviare il client Python

1. In una finestra del prompt dei comandi, vai alla cartella del progetto `PyBubbleClient` di esempio.

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

2. Esegui il client Python.

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

   Parametri

   host

   L'indirizzo IP della simulazione. Per una simulazione iniziata in Cloud AWS, puoi trovare l'indirizzo IP della simulazione nella SimSpace Weaver console o utilizzare la procedura descritta nel tutorial Ottieni l'indirizzo IP e il numero di porta di un'app personalizzata di avvio rapido. Per una simulazione locale, usa `127.0.0.1` come indirizzo IP.

   port

   Il numero di porta della simulazione. Per una simulazione iniziata in Cloud AWS, questo è il numero di `Actual` porta. Puoi trovare il numero di porta della simulazione nella SimSpace Weaver console o utilizzare la procedura Ottieni l'indirizzo IP e il numero di porta di un'app personalizzata del tutorial di avvio rapido. Per una simulazione locale, usa `7000` come numero di porta.

   simsize

   Il numero massimo di entità da visualizzare nel client.

Domande frequenti sull'uso di Python

D1. Quali versioni di Python sono supportate?

SimSpace Weaver supporta solo la versione 3.9 di Python.

Risoluzione dei problemi relativi a Python

Argomenti

- Errore durante la creazione di contenitori personalizzati
- La tua simulazione in Python non si avvia
- Un client di simulazione o visualizzazione in Python genera un errore ModuleNotFound

Errore durante la creazione di contenitori personalizzati

Se ricevi un errore `no basic auth credentials` dopo l'esecuzione`quick-start.py`, potrebbe esserci un problema con le tue credenziali temporanee per Amazon ECR. Esegui il comando seguente con il tuo Regione AWS ID e il numero di AWS account:

```
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**
>
> Assicurati che Regione AWS quello specificato sia lo stesso che usi per la simulazione. Usa uno dei SimSpace Weaver supporti Regioni AWS supportati. Per ulteriori informazioni, consulta SimSpace Endpoint e quote Weaver.

Dopo aver eseguito il `aws ecr` comando, esegui `quick-start.py` di nuovo.

Altre risorse per la risoluzione dei problemi da verificare

- Risoluzione dei problemi relativi ai contenitori personalizzati

- Risoluzione dei problemi di Amazon ECR nella Amazon ECR User Guide
- Configurazione con Amazon ECR nella Amazon ECR User Guide

La tua simulazione in Python non si avvia

Potresti visualizzare un `Unable to start app` errore nel registro di gestione della simulazione. Questo può accadere se la creazione del contenitore personalizzato non è riuscita. Per ulteriori informazioni, consulta Errore durante la creazione di contenitori personalizzati. Per ulteriori informazioni sui log, consulta SimSpace Weaver accessi in Amazon CloudWatch Logs.

Se sei sicuro che non ci sia nulla di sbagliato nel tuo contenitore, controlla il codice sorgente Python dell'app. È possibile utilizzare… SimSpace Weaver Local per testare la tua app. Per ulteriori informazioni, consulta Sviluppo locale in SimSpace Weaver.

Un client di simulazione o visualizzazione in Python genera un errore ModuleNotFound

Python genera un `ModuleNotFound` errore quando non riesce a trovare un pacchetto Python richiesto.

Se la tua simulazione è in Cloud AWS, assicurati che il tuo contenitore personalizzato abbia tutte le dipendenze richieste elencate nel tuo. `requirements.txt` Ricordati di eseguirlo di `quick-start.py` nuovo se apporti modifiche. `requirements.txt`

Se ricevi l'errore relativo al `PythonBubblesSample` client, usa `pip` per installare il pacchetto indicato:

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

Support per altri motori

Puoi usare il tuo personalizzato C++ motore con SimSpace Weaver. Attualmente stiamo sviluppando il supporto per i seguenti motori. Esiste una documentazione separata per ciascuno di questi motori.

> **Important**
>
> Le integrazioni con i motori elencati qui sono sperimentali. Sono disponibili in anteprima.

Motori

- Unità(versione minima 2022.3.19.F1)
- Unreal Engine(versione minima 5.0)

Unità

È necessario disporre del Unity ambiente di sviluppo già installato prima di creare SimSpace Weaver simulazioni con Unity. Per ulteriori informazioni, consulta le istruzioni separate:

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

Unreal Engine

È necessario creare un Unreal Engine server dedicato dal codice sorgente. SimSpaceWeaverAppSdkDistributable Include una versione PathfindingSample del Unreal Engine. Per ulteriori informazioni, consulta le istruzioni separate:

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

Utilizzo di software concesso in licenza con AWS SimSpace Weaver

AWS SimSpace Weaver ti consente di creare simulazioni con il motore di simulazione e i contenuti che preferisci. In relazione all'utilizzo di SimSpace Weaver, l'utente è responsabile dell'ottenimento, della manutenzione e del rispetto delle condizioni di licenza di qualsiasi software o contenuto utilizzato nelle simulazioni. Verifica che il contratto di licenza ti consenta di distribuire software e contenuti in un ambiente ospitato virtuale.

Gestisci le tue risorse con AWS CloudFormation

Puoi usarlo AWS CloudFormation per gestire le tue AWS SimSpace Weaver risorse. AWS CloudFormation è un AWS servizio separato che consente di specificare, fornire e gestire l' AWS infrastruttura come codice. Con AWS CloudFormation te crei un file JSON o YAML, chiamato modello. Il modello specifica i dettagli della tua infrastruttura. AWS CloudFormation utilizza il modello per effettuare il provisioning dell'infrastruttura come una singola unità, denominata stack. Quando

elimini lo stack, puoi fare in modo che AWS CloudFormation tutti gli elementi dello stack vengano eliminati contemporaneamente. Puoi gestire il tuo modello utilizzando processi standard di gestione del codice sorgente (ad esempio, tracciandolo in un sistema di controllo della versione come Git). Per ulteriori informazioni in merito AWS CloudFormation, consulta la Guida AWS CloudFormation per l'utente.

La tua risorsa di simulazione

In AWS, una risorsa è un'entità con cui puoi lavorare. Gli esempi includono un' EC2 istanza Amazon, un bucket Amazon S3 o un ruolo IAM. La tua SimSpace Weaver simulazione è una risorsa. Nelle configurazioni, di solito si specifica una AWS risorsa nel modulo. `AWS::service::resource` Infatti SimSpace Weaver, si specifica la risorsa di simulazione come. `AWS::SimSpaceWeaver::Simulation` Per ulteriori informazioni sulla risorsa di simulazione in AWS CloudFormation, consulta la SimSpace Weaversezione della Guida per l'AWS CloudFormation utente.

Come posso usare AWS CloudFormation con SimSpace Weaver?

È possibile creare un AWS CloudFormation modello che specifichi le AWS risorse che si desidera fornire. Il modello può specificare un'intera architettura, parte di un'architettura o una piccola soluzione. Ad esempio, puoi specificare un'architettura per la tua SimSpace Weaver soluzione che includa bucket Amazon S3, autorizzazioni IAM, un database di supporto in Amazon Relational Database Service o Amazon DynamoDB e la tua risorsa. `Simulation` Puoi quindi utilizzarle AWS CloudFormation per fornire tutte queste risorse come unità e contemporaneamente.

Example modello che crea risorse IAM e avvia una simulazione

Il seguente modello di esempio crea un ruolo IAM e le autorizzazioni SimSpace Weaver necessarie per eseguire azioni nel tuo account. Gli script SDK dell' SimSpace Weaver app creano il ruolo e le autorizzazioni in uno specifico Regione AWS quando crei un progetto, ma puoi utilizzare un AWS CloudFormation modello per distribuire la simulazione su un altro Regione AWS senza eseguire nuovamente gli script. Ad esempio, puoi farlo per configurare una simulazione di backup a scopo di disaster recovery.

In questo esempio, il nome originale della simulazione è. `MySimulation` Un bucket per lo schema esiste già nel file Regione AWS dove AWS CloudFormation verrà creato lo stack. Il bucket contiene una versione dello schema configurata correttamente per eseguire la simulazione in esso. Regione AWS Ricorda che lo schema specifica la posizione dei file zip dell'app, che è un bucket Amazon S3 Regione AWS uguale alla simulazione. Il bucket e i file dell'app zip devono già esistere nello stack Regione AWS when AWS CloudFormation build the stack, altrimenti la simulazione non verrà avviata.

Nota che il nome del bucket in questo esempio include Regione AWS, ma ciò non determina dove si trovi effettivamente il bucket. Devi assicurarti che il bucket si trovi effettivamente al suo interno Regione AWS (puoi controllare le proprietà del bucket nella console Amazon S3, con Amazon S3 o con i APIs comandi Amazon S3 in). AWS CLI

Questo esempio utilizza alcune funzioni e parametri incorporati per eseguire la sostituzione delle variabili. AWS CloudFormation Per ulteriori informazioni, vedete Riferimento alla funzione intrinseca e Riferimento agli pseudo parametri nella Guida per l'utente.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'
```

Utilizzo di istantanee con AWS CloudFormation

Un'istantanea è il backup di una simulazione. L'esempio seguente avvia una nuova simulazione da un'istantanea anziché da uno schema. L'istantanea in questo esempio è stata creata da una simulazione di progetto SDK di un' SimSpace Weaver app. AWS CloudFormation crea la nuova risorsa di simulazione e la inizializza con i dati dell'istantanea. La nuova simulazione può avere una simulazione diversa `MaximumDuration` da quella originale.

Ti consigliamo di creare e utilizzare una copia del ruolo dell'app della simulazione originale. Il ruolo dell'app della simulazione originale potrebbe essere eliminato se elimini lo stack di quella simulazione. AWS CloudFormation

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

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

Snapshot

È possibile creare un'istantanea per eseguire il backup dei dati dell'entità di simulazione in qualsiasi momento. SimSpace Weaver crea un file.zip in un bucket Amazon S3. Puoi creare una nuova simulazione con lo snapshot. SimSpace Weaver inizializza lo State Fabric della nuova simulazione con i dati dell'entità memorizzati nell'istantanea, avvia le app spaziali e di servizio che erano in esecuzione al momento della creazione dell'istantanea e imposta l'orologio sul segno di spunta appropriato. SimSpace Weaver ottiene la configurazione della simulazione dall'istantanea anziché da un file di schema. I file.zip dell'app devono trovarsi nella stessa posizione in Amazon S3 in cui si trovavano nella simulazione originale. Tutte le app personalizzate devono essere avviate separatamente.

Argomenti

- Casi d'uso per le istantanee
- Usa la SimSpace Weaver console per lavorare con le istantanee
- Usa il AWS CLI per lavorare con le istantanee
- Utilizzo di istantanee con AWS CloudFormation
- Domande frequenti sulle istantanee

Casi d'uso per le istantanee

Ritorna allo stato precedente ed esplora gli scenari ramificati

Puoi creare un'istantanea della simulazione per salvarla in uno stato specifico. È quindi possibile creare più nuove simulazioni da quell'istantanea ed esplorare diversi scenari che potrebbero derivare da quello stato.

Migliori pratiche di disaster recovery e sicurezza

Si consiglia di eseguire regolarmente il backup della simulazione, in particolare per le simulazioni che durano più di un'ora o che utilizzano più lavoratori. I backup possono aiutarti a riprenderti da disastri e incidenti di sicurezza. Le istantanee consentono di eseguire il backup della simulazione. Le istantanee richiedono che i file.zip dell'app esistano nella stessa posizione in Amazon S3 in cui si trovavano in precedenza. Se devi poter spostare i file.zip dell'app in un'altra posizione, devi utilizzare una soluzione di backup personalizzata.

Per ulteriori informazioni su altre best practice, consulta Procedure consigliate quando si lavora con SimSpace Weaver eLe migliori pratiche di sicurezza per SimSpace Weaver.

Estendi la durata della simulazione

La tua risorsa di simulazione è la rappresentazione della tua simulazione in. SimSpace Weaver Tutte le risorse di simulazione hanno un'impostazione. `MaximumDuration` Una risorsa di simulazione si interrompe automaticamente quando raggiunge il suo valore. `MaximumDuration` Il valore massimo di `MaximumDuration` è `14D` (14 giorni).

Se è necessario che la simulazione persista più a lungo `MaximumDuration` della relativa risorsa di simulazione, è possibile creare un'istantanea prima che la risorsa di simulazione raggiunga il suo valore. `MaximumDuration` Puoi iniziare una nuova simulazione (creare una nuova risorsa di simulazione) con la tua istantanea. SimSpace Weaver inizializza i dati dell'entità dall'istantanea, avvia le stesse app spaziali e di servizio eseguite in precedenza e ripristina l'orologio. Puoi avviare le tue app personalizzate ed eseguire qualsiasi inizializzazione personalizzata aggiuntiva. È possibile impostare la nuova risorsa `MaximumDuration` di simulazione su un valore diverso all'avvio.

Usa la SimSpace Weaver console per lavorare con le istantanee

È possibile utilizzare la SimSpace Weaver console per creare un'istantanea della simulazione.

Argomenti

- Usa la console per creare un'istantanea
- Usa la console per avviare una simulazione da un'istantanea

Usa la console per creare un'istantanea

Per creare una snapshot

1. Accedi a AWS Management Console e connettiti alla SimSpace Weaver console.
2. Scegli Simulazioni dal pannello di navigazione.
3. Seleziona il pulsante radio accanto al nome della simulazione. Lo stato della simulazione deve essere Avviato.
4. Nella parte superiore della pagina, scegli Crea istantanea.
5. In Impostazioni snapshot, per destinazione Snapshot, inserisci l'URI Amazon S3 di un bucket o di un bucket e della cartella in cui desideri SimSpace Weaver creare la tua istantanea. Puoi scegliere Browse S3 se preferisci sfogliare i bucket disponibili e selezionare una posizione.

> **Important**
>
> Il bucket Amazon S3 deve essere Regione AWS uguale alla simulazione.

> **Note**
>
> SimSpace Weaver crea una `snapshot` cartella all'interno della destinazione dello snapshot selezionato. SimSpace Weaver crea il file snapshot .zip in quella cartella. `snapshot`

6. Scegli Create snapshot (Crea snapshot).

Usa la console per avviare una simulazione da un'istantanea

Per avviare una simulazione da uno snapshot, il file snapshot .zip deve esistere in un bucket Amazon S3 a cui la simulazione può accedere. La simulazione utilizza le autorizzazioni definite nel ruolo dell'app selezionato all'avvio della simulazione. Tutti i file.zip dell'app della simulazione originale devono esistere nelle stesse posizioni in cui è stata creata l'istantanea.

Per avviare una simulazione da un'istantanea

1. Accedi a AWS Management Console e connettiti alla SimSpace Weaver console.
2. Scegli Simulazioni dal pannello di navigazione.
3. Nella parte superiore della pagina, scegli Avvia simulazione.
4. In Impostazioni di simulazione, inserisci un nome e una descrizione opzionale per la simulazione. Il nome della simulazione deve essere univoco nel tuo. Account AWS
5. Per il metodo di avvio della simulazione, scegli Usa uno snapshot in Amazon S3.
6. Per l'URI di Amazon S3 per snapshot, inserisci l'URI Amazon S3 del tuo file di snapshot oppure scegli Browse S3 per sfogliare e selezionare il file.

> **Important**
>
> Il bucket Amazon S3 deve essere Regione AWS uguale alla simulazione.

7. Per il ruolo IAM, seleziona il ruolo dell'app che verrà utilizzato dalla simulazione.

8. Per Durata massima, inserisci il periodo di tempo massimo per cui la risorsa di simulazione deve essere eseguita. Il valore massimo è `14D`. Per ulteriori informazioni sulla durata massima, vedere.
9. In Tag, facoltativo, scegli Aggiungi nuovo tag se desideri aggiungere un tag.
10. Scegli Avvia simulazione.

Usa il AWS CLI per lavorare con le istantanee

È possibile utilizzare il AWS CLI per chiamarli SimSpace Weaver APIs da un prompt dei comandi. È necessario che AWS CLI sia installato e configurato correttamente. Per ulteriori informazioni, consulta Installazione o aggiornamento della versione più recente della AWS CLI nella Guida per l'AWS Command Line Interface utente per la versione 2.

Argomenti

- Utilizzate il AWS CLI per creare un'istantanea
- Utilizzate il AWS CLI per avviare una simulazione da un'istantanea

Utilizzate il AWS CLI per creare un'istantanea

Per creare una snapshot

- Al prompt dei comandi, richiama l'`CreateSnapshot`API.

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

Parametri

simulazione

Il nome di una simulazione iniziata. Puoi usarlo `aws simspaceweaver list-simulations` per vedere i nomi e gli stati delle tue simulazioni.

destinazione

Una stringa che specifica il bucket Amazon S3 di destinazione e il prefisso opzionale della chiave oggetto per il file di snapshot. Il prefisso della chiave oggetto è in genere una cartella

nel bucket. SimSpace Weaver crea la tua istantanea all'interno di una `snapshot` cartella in questa destinazione.

> **Important**
>
> Il bucket Amazon S3 deve essere Regione AWS uguale alla simulazione.

Esempio

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

Per ulteriori informazioni sull'*`CreateSnapshot`*API, consulta l'API CreateSnapshotReference.AWS SimSpace Weaver

Utilizzate il AWS CLI per avviare una simulazione da un'istantanea

Per avviare una simulazione da un'istantanea

- Al prompt dei comandi, richiama l'API. `StartSimulation`

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

Parametri

name

Il nome della nuova simulazione. Il nome della simulazione deve essere univoco nel tuo. Account AWS Puoi usarlo `aws simspaceweaver list-simulations` per vedere i nomi delle tue simulazioni esistenti.

ruolo-arn

L'Amazon Resource Name (ARN) del ruolo dell'app che verrà utilizzato dalla simulazione.

snapshot-s3-location

Una stringa che specifica il bucket Amazon S3 e la chiave oggetto del file snapshot.

> **Important**
>
> Il bucket Amazon S3 deve essere Regione AWS uguale alla simulazione.

Esempio

```
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
```

Per ulteriori informazioni sull'*StartSimulation*API, consulta l'API StartSimulationReference.AWS SimSpace Weaver

Domande frequenti sulle istantanee

La mia simulazione continua a essere eseguita durante un'istantanea?

Le tue risorse di simulazione continuano a funzionare durante un'istantanea e continui a ricevere addebiti di fatturazione per quel periodo. Il tempo viene conteggiato ai fini della durata massima della simulazione. Le tue app non ricevono segni di spunta mentre l'istantanea è in corso. Se lo stato dell'orologio era `STARTED` quando è iniziata la creazione dell'istantanea, l'orologio continuerà a indicare lo stato. `STARTED` Le tue app ricevono nuovamente i segni di spunta al termine dell'istantanea. Se lo stato dell'orologio era, `STOPPED` lo stato dell'orologio rimarrà invariato. `STOPPED` Tieni presente che una simulazione con uno `STARTED` stato è in esecuzione anche se lo stato dell'orologio è`STOPPED`.

Cosa succede se è in corso un'istantanea e la mia simulazione raggiunge la sua durata massima?

La simulazione completerà l'istantanea e poi si interromperà non appena il processo di istantanea termina (con successo o meno). Ti consigliamo di testare prima il processo di istantanea per scoprire quanto tempo impiega, la dimensione del file di istantanea che puoi aspettarti e se il processo deve essere completato correttamente.

Cosa succede se interrompo una simulazione in cui è in corso un'istantanea?

Un'istantanea in corso si interrompe immediatamente quando si interrompe la simulazione. Non creerà un file di istantanea.

Come posso interrompere un'istantanea in corso?

L'unico modo per fermare un'istantanea in corso è interrompere la simulazione. Non è possibile riavviare una simulazione dopo averla interrotta.

Quanto tempo ci vorrà per completare la mia istantanea?

Il tempo necessario per creare un'istantanea dipende dalla simulazione. Ti consigliamo di testare prima il processo di istantanea per scoprire quanto tempo impiegherà per la simulazione.

Quanto sarà grande il mio file di istantanea?

La dimensione di un file di istantanea dipende dalla simulazione. Ti consigliamo di testare prima il processo di istantanea per scoprire quanto potrebbe essere grande il file per la tua simulazione.

Messaggistica

L'API di messaggistica semplifica la comunicazione tra applicazioni all'interno della simulazione. APIs per inviare e ricevere messaggi fa parte dell'SDK dell' SimSpace Weaver app. La messaggistica attualmente utilizza un approccio ottimale per inviare e ricevere messaggi. SimSpace Weaver tenta di inviare/ricevere messaggi alla prossima simulazione, ma non ci sono garanzie sull'orario di consegna, ordinazione o arrivo.

Argomenti

- Casi d'uso per la messaggistica
- Utilizzo della messaggistica APIs
- Quando usare la messaggistica
- Suggerimenti per l'utilizzo della messaggistica
- Errori di messaggistica e risoluzione dei problemi

Casi d'uso per la messaggistica

Comunica tra applicazioni di simulazione

Utilizza l'API di messaggistica per comunicare tra le applicazioni della simulazione. Usala per modificare lo stato delle entità a distanza, modificare il comportamento delle entità o trasmettere informazioni all'intera simulazione.

Conferma della ricezione di un messaggio

I messaggi inviati contengono informazioni sul mittente nell'intestazione del messaggio. Utilizza queste informazioni per inviare una risposta di conferma alla ricezione di un messaggio.

Inoltra i dati ricevuti da un'app personalizzata ad altre app all'interno della simulazione

La messaggistica non sostituisce il modo in cui i client si connettono alle app personalizzate in esecuzione in SimSpace Weaver. Tuttavia, la messaggistica consente agli utenti un metodo per inoltrare i dati dalle app personalizzate che ricevono i dati dei client ad altre app che non dispongono di una connessione esterna. Il flusso di messaggi può funzionare anche in senso inverso, consentendo alle app senza connessione esterna di inoltrare i dati a un'app personalizzata e quindi a un client.

Utilizzo della messaggistica APIs

I messaggi APIs sono contenuti nell'SDK dell' SimSpace Weaver app (versione minima 1.16.0). La messaggistica è supportata in C++, Python e nelle nostre integrazioni con Unreal Engine 5 e Unity.

Esistono due funzioni che gestiscono le transazioni di messaggi: e. `SendMessage` `ReceiveMessages` Tutti i messaggi inviati contengono una destinazione e un payload. L'`ReceiveMessages`API restituisce un elenco di messaggi attualmente presenti nella coda dei messaggi in entrata di un'app.

C++

Invia messaggio

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

Ricevere messaggi

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

Python

Inviare un messaggio

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

Ricevere messaggi

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

Argomenti

- Invio di messaggi
- Ricezione di messaggi
- Rispondere al mittente

Invio di messaggi

I messaggi sono costituiti da una transazione (simile ad altre chiamate dell'API Weaver), un payload e una destinazione.

Payload del messaggio

Il payload dei messaggi è una struttura dati flessibile fino a 256 byte. Consigliamo quanto segue come best practice per creare i payload dei messaggi.

Per creare il payload dei messaggi

1. Crea una struttura di dati (ad esempio `struct` in C++) che definisca il contenuto del messaggio.
2. Crea il payload del messaggio che contiene i valori da inviare nel messaggio.
3. Crea l'`MessagePayload`oggetto.

Destinazione del messaggio

La destinazione di un messaggio è definita dall'`MessageEndpoint`oggetto. Ciò include sia un tipo di endpoint che un ID di endpoint. L'unico tipo di endpoint attualmente supportato è`Partition`, che consente di indirizzare messaggi ad altre partizioni nella simulazione. L'ID endpoint è l'ID di partizione della destinazione di destinazione.

Puoi fornire solo 1 indirizzo di destinazione in un messaggio. Crea e invia più messaggi se desideri inviare messaggi a più di 1 partizione contemporaneamente.

Per indicazioni su come risolvere un endpoint di messaggio da una posizione, vedere. Suggerimenti per l'utilizzo della messaggistica

Inviare il messaggio

È possibile utilizzare l'`SendMessage`API dopo aver creato gli oggetti di destinazione e di payload.

C++

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

Python

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

Esempio completo di invio di messaggi

L'esempio seguente mostra come è possibile costruire e inviare un messaggio generico. Questo esempio invia 16 messaggi singoli. Ogni messaggio contiene un payload con un valore compreso tra 0 e 15 e il segno di spunta di simulazione corrente.

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
```

Ricezione di messaggi

SimSpace Weaver consegna i messaggi nella coda dei messaggi in entrata di una partizione. Utilizza l'`ReceiveMessages`API per ottenere un `MessageList` oggetto che contiene i messaggi dalla coda. Elabora ogni messaggio con l'`ExtractMessage`API per ottenere i dati del messaggio.

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
```

Rispondere al mittente

Ogni messaggio ricevuto contiene un'intestazione del messaggio con informazioni sul mittente originale del messaggio. Puoi usare message.header.source_endpoint per inviare una risposta.

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 usare la messaggistica

La messaggistica in SimSpace Weaver offre un altro modello per lo scambio di informazioni tra applicazioni di simulazione. Gli abbonamenti forniscono un meccanismo pull per leggere i dati da applicazioni o aree specifiche della simulazione; i messaggi forniscono un meccanismo push per inviare dati ad applicazioni o aree specifiche della simulazione.

Di seguito sono riportati due casi d'uso in cui è più utile inviare dati utilizzando la messaggistica piuttosto che estrarre o leggere i dati tramite un abbonamento.

Example 1: Invio di un comando a un'altra app per modificare la posizione di un'entità

```
// 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);
```

Sul lato ricevente, l'app aggiorna la posizione dell'entità e la scrive nello 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: Invio di un messaggio di creazione dell'entità a un'app spaziale

```
struct WeaverMessage
{
    const Aws::WeaverRuntime::Api::TypeId messageTypeId;
};

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

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


CreateEntityMessage messageData {
    k_createEntityMessageTypeId,
    Vector3{ position.GetX(), position.GetY(), position.GetZ() },
```

```
    Api::TypeId { 0 }
}

WEAVERRUNTIME_TRY(Api::MessageEndpoint destination,
 Api::Utils::MessageEndpointResolver::ResolveFromPosition(
    transaction, "MySpatialDomain", DemoFramework::ToVector2F32(position)
));

Api::MessagePayload payload = Api::Utils::CreateMessagePayload(
    reinterpret_cast<const uint8_t*>(&messageData),
    sizeof(CreateEntityMessage));

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

Sul lato ricevente, l'app crea una nuova entità nello State Fabric e ne aggiorna la posizione.

```
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)));
}
```

Suggerimenti per l'utilizzo della messaggistica

Risolvi un endpoint da una posizione o dal nome dell'app

È possibile utilizzare la `AllPartitions` funzione per ottenere i limiti spaziali e l'ID di dominio necessari per determinare la partizione IDs e le destinazioni dei messaggi. Tuttavia, se conosci la posizione in cui desideri inviare il messaggio, ma non il relativo ID di partizione, puoi utilizzare la funzione. MessageEndpointResolver

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

Serializzazione e deserializzazione del payload del messaggio

È possibile utilizzare le seguenti funzioni per creare e leggere i payload dei messaggi. Per ulteriori informazioni, consulta MessagingUtils .h nella libreria SDK dell'app sul tuo sistema locale.

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

Errori di messaggistica e risoluzione dei problemi

È possibile che si verifichino i seguenti errori quando si utilizza la messaggistica APIs.

Errori di risoluzione degli endpoint

Questi errori possono verificarsi prima che un'app invii un messaggio.

Controllo del nome di dominio

L'invio di un messaggio a un dispositivo non valido genera il seguente errore:

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

Questo può accadere quando si tenta di inviare un messaggio a un'app personalizzata e tale app personalizzata non ha ancora partecipato alla simulazione. Usa l'`DescribeSimulation`API per assicurarti che l'app personalizzata sia stata avviata prima di inviarle un messaggio. Questo comportamento è lo stesso in SimSpace Weaver Local e il Cloud AWS.

Controllo della posizione

Il tentativo di risolvere un endpoint con un nome di dominio valido ma una posizione non valida genera il seguente errore.

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

Ti consigliamo di utilizzarlo `MessageEndpointResolver` nella `MessageUtils` libreria contenuta nell'SDK dell' SimSpace Weaver app.

Errori di invio dei messaggi

I seguenti errori possono verificarsi quando un'app invia un messaggio.

È stato superato il limite di invio di messaggi per app, per segno di spunta

Il limite attuale per il numero di messaggi che possono essere inviati per app per segno di simulazione è 128. Le chiamate successive con lo stesso segno di spunta falliranno con il seguente errore:

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

SimSpace Weaver tenta di inviare messaggi non inviati con il segno di spunta successivo. Riduci la frequenza di invio per risolvere il problema. Combina payload di messaggi inferiori al limite di 256 byte per ridurre il numero di messaggi in uscita.

Questo comportamento è lo stesso in SimSpace Weaver Local e in Cloud AWS.

Il limite di dimensione del payload dei messaggi è stato superato

Il limite attuale per la dimensione del payload dei messaggi è di 256 byte in entrambi SimSpace Weaver Local e in. Cloud AWS L'invio di un messaggio con un payload superiore a 256 byte genera il seguente errore:

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

SimSpace Weaver controlla ogni messaggio e rifiuta solo quelli che superano il limite. Ad esempio, se l'app tenta di inviare 10 messaggi e 1 non supera il controllo, solo quel messaggio viene rifiutato. SimSpace Weaver invia gli altri 9 messaggi.

Questo comportamento è lo stesso in SimSpace Weaver Local e il Cloud AWS.

La destinazione è la stessa della fonte

Le app non possono inviare messaggi alle partizioni di cui sono proprietarie. Se un'app invia un messaggio a una partizione di sua proprietà, viene visualizzato il seguente errore.

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

Questo comportamento è lo stesso in SimSpace Weaver Local e il Cloud AWS.

Messaggistica con il massimo impegno

SimSpace Weaver non garantisce la consegna dei messaggi. Il servizio cercherà di completare la consegna dei messaggi al successivo segno di spunta di simulazione, ma i messaggi potrebbero andare persi o subire ritardi.

Procedure consigliate quando si lavora con SimSpace Weaver

Consigliamo le seguenti best practice quando si lavora con SimSpace Weaver.

Argomenti

- Imposta gli allarmi di fatturazione
- Utilizzare SimSpace Weaver Local
- Interrompi le simulazioni che non ti servono
- Elimina le risorse che non ti servono
- Avere dei backup

Imposta gli allarmi di fatturazione

È facile allocare le risorse AWS e lasciarle sempre attive, anche quando non sono più necessarie. Ciò può comportare costi eccessivi che possono sorprendere una volta ricevuta la fattura. Puoi configurare un allarme in Amazon CloudWatch che si attiverà e ti avviserà quando i costi superano una soglia da te impostata. Puoi esaminare i costi utilizzando strumenti di gestione dei costi. Per ulteriori informazioni, consultare:

- Crea un allarme di fatturazione per monitorare gli addebiti stimati AWS
- Che cos'è AWS Cost Management

Utilizzare SimSpace Weaver Local

Ti consigliamo di utilizzare SimSpace Weaver Local per sviluppare e testare le tue simulazioni prima di caricarle sul SimSpace Weaver servizio in. Cloud AWS I vantaggi dello sviluppo con SimSpace Weaver Local includono:

- Non c'è bisogno di aspettare caricamenti di grandi dimensioni
- Nessun limite al numero di simulazioni locali che puoi creare
- Non ti viene addebitato alcun costo per il tempo di elaborazione sul tuo computer locale
- Accesso diretto all'output della console dalle tue app

- Modifica, ricostruisci e riavvia la simulazione locale senza doverla ricreare nel Cloud AWS

Interrompi le simulazioni che non ti servono

Ti verranno addebitati i costi di fatturazione per una simulazione mentre è in esecuzione. È necessario interrompere una simulazione per non ricevere più addebiti. Le simulazioni in esecuzione vengono conteggiate anche ai fini della quota per il numero massimo di simulazioni. Una simulazione in esecuzione con registrazione configurata può anche generare grandi quantità di log, per i quali vengono addebitati anche costi di fatturazione. È necessario interrompere qualsiasi simulazione non necessaria per evitare costi aggiuntivi.

> **Important**
>
> L'interruzione dell'orologio di simulazione non interrompe la simulazione, l'orologio interrompe semplicemente la pubblicazione di segni di spunta nelle tue app. Non puoi riavviare una simulazione dopo averla interrotta.

Elimina le risorse che non ti servono

Ogni simulazione creata crea SimSpace Weaver anche risorse in altri AWS servizi. Puoi ottenere addebiti di fatturazione per risorse e dati in questi altri servizi. Le simulazioni in esecuzione e quelle con esito negativo vengono conteggiate ai fini della quota per il numero massimo di simulazioni. È necessario eliminare le simulazioni non riuscite non necessarie in modo da poter avviare nuove simulazioni. Quando si elimina una simulazione, è possibile che le risorse per la simulazione presenti in altri AWS servizi non vengano eliminate. Ad esempio, tutti i dati di log di simulazione in Amazon CloudWatch Logs rimarranno lì finché non li elimini. Ti verranno addebitati i costi di fatturazione per quei dati di log. Dovresti pulire tutte le risorse associate per le tue simulazioni se non ne hai più bisogno.

Avere dei backup

È una buona idea disporre di backup e piani di backup per tutto. Non dovresti dare per scontato che solo perché ci sono i tuoi dati non AWS sia necessario eseguirne il backup. È necessario creare il proprio sistema se è necessario eseguire il backup dello stato di simulazione. Valuta la possibilità di utilizzarne più di uno Regioni AWS e di predisporre un piano per poter passare rapidamente il carico

di lavoro di produzione a un altro, Regione AWS se necessario. Per ulteriori informazioni su Regioni AWS tale supporto SimSpace Weaver, consultaSimSpace Endpoint e quote Weaver.

Sicurezza in AWS SimSpace Weaver

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di data center e architetture di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra te e te. AWS Il modello di responsabilità condivisa descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi in Cloud AWS. AWS fornisce inoltre servizi che è possibile utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei AWS Programmi di AWS conformità dei Programmi di conformità dei di . Per ulteriori informazioni sui programmi di conformità applicabili AWS SimSpace Weaver, consulta AWS Servizi nell'ambito del programma di conformitàAWS .
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della tua azienda e le leggi e normative vigenti.

Questa documentazione ti aiuta a capire come applicare il modello di responsabilità condivisa durante l'utilizzo SimSpace Weaver. I seguenti argomenti mostrano come eseguire la configurazione SimSpace Weaver per soddisfare gli obiettivi di sicurezza e conformità. Imparerai anche a utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere SimSpace Weaver le tue risorse.

Argomenti

- Protezione dei dati in AWS SimSpace Weaver
- Identity and Access Management per AWS SimSpace Weaver
- Registrazione e monitoraggio degli eventi di sicurezza AWS SimSpace Weaver
- Convalida della conformità per AWS SimSpace Weaver
- Resilienza in AWS SimSpace Weaver
- Sicurezza dell'infrastruttura in AWS SimSpace Weaver
- Analisi della configurazione e della vulnerabilità in AWS SimSpace Weaver
- Le migliori pratiche di sicurezza per SimSpace Weaver

Protezione dei dati in AWS SimSpace Weaver

Il modello di responsabilità AWS condivisa modello di di si applica alla protezione dei dati in AWS SimSpace Weaver. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, vedi le Domande frequenti sulla privacy dei dati. Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al Modello di responsabilità condivisa AWS e GDPR nel Blog sulla sicurezza AWS .

Ai fini della protezione dei dati, consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- SSL/TLS Da utilizzare per comunicare con AWS le risorse. È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail. Per informazioni sull'utilizzo dei CloudTrail percorsi per acquisire AWS le attività, consulta Lavorare con i CloudTrail percorsi nella Guida per l'AWS CloudTrail utente.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-3 per accedere AWS tramite un'interfaccia a riga di comando o un'API, usa un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il Federal Information Processing Standard (FIPS) 140-3.

Ti consigliamo di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori SimSpace Weaver o Servizi AWS utilizzi la console, l'API o. AWS CLI AWS SDKs I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Quando fornisci un URL a un server esterno, ti suggeriamo

vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta al server.

Crittografia dei dati a riposo

I dati sono considerati inattivi quando si trovano in un archivio di dati non volatile (persistente), ad esempio un disco. I dati che si trovano in archivi di dati volatili, come memoria e registri, non sono considerati inattivi.

Quando si utilizza SimSpace Weaver, gli unici dati inattivi sono:

- App e schemi caricati su Amazon Simple Storage Service (Amazon S3)
- Dati del registro di simulazione archiviati in Amazon CloudWatch

Gli altri dati SimSpace Weaver utilizzati internamente non persistono dopo l'interruzione della simulazione.

Per informazioni su come crittografare i dati inattivi, consulta:

- Crittografa i tuoi dati in Amazon S3
- Crittografa i dati di registro

Crittografia in transito

Le tue connessioni all' SimSpace Weaver API tramite AWS Command Line Interface (AWS CLI), AWS SDK e SimSpace Weaver app SDK utilizzano la crittografia TLS con il processo di firma Signature Version 4. AWS gestisce l'autenticazione utilizzando le politiche di accesso definite da IAM per le credenziali di sicurezza utilizzate per la connessione.

Internamente, SimSpace Weaver utilizza TLS per connettersi agli altri AWS servizi che utilizza.

> **Important**
>
> Le comunicazioni tra le tue app e i loro clienti non implicano. SimSpace WeaverÈ tua responsabilità crittografare le comunicazioni con i client di simulazione, se necessario. Ti consigliamo di creare una soluzione per crittografare tutti i dati in transito tra le connessioni client.

Per saperne di più sui AWS servizi in grado di supportare le tue soluzioni di crittografia, consulta il AWS Security Blog.

Riservatezza del traffico Internet

SimSpace Weaver le risorse di calcolo risiedono all'interno di 1 Amazon VPC condiviso da tutti i clienti. SimSpace Weaver Tutto il traffico interno dei SimSpace Weaver servizi rimane all'interno della AWS rete e non viaggia su Internet. La comunicazione tra i client di simulazione e le tue app viaggia su Internet.

Identity and Access Management per AWS SimSpace Weaver

AWS Identity and Access Management (IAM) è un software Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. Gli amministratori IAM controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse. SimSpace Weaver IAM è uno Servizio AWS strumento che puoi utilizzare senza costi aggiuntivi.

Argomenti

- Destinatari
- Autenticazione con identità
- Gestione dell'accesso con policy
- Come AWS SimSpace Weaver funziona con IAM
- Esempi di policy basate sull'identità per AWS SimSpace Weaver
- Autorizzazioni create per te SimSpace Weaver
- Prevenzione del problema "confused deputy" tra servizi
- Risoluzione dei problemi di AWS SimSpace Weaver identità e accesso

Destinatari

Il modo in cui usi AWS Identity and Access Management (IAM) varia a seconda del lavoro che SimSpace Weaver svolgi.

Utente del servizio: se utilizzi il SimSpace Weaver servizio per svolgere il tuo lavoro, l'amministratore ti fornisce le credenziali e le autorizzazioni necessarie. Man mano che utilizzi più SimSpace

Weaver funzionalità per svolgere il tuo lavoro, potresti aver bisogno di autorizzazioni aggiuntive. La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una funzionalità di SimSpace Weaver, consulta Risoluzione dei problemi di AWS SimSpace Weaver identità e accesso.

Amministratore del servizio: se sei responsabile delle SimSpace Weaver risorse della tua azienda, probabilmente hai pieno accesso a SimSpace Weaver. È tuo compito determinare a quali SimSpace Weaver funzionalità e risorse devono accedere gli utenti del servizio. Devi inviare le richieste all'amministratore IAM per cambiare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a IAM. Per saperne di più su come la tua azienda può utilizzare IAM con SimSpace Weaver, consultaCome AWS SimSpace Weaver funziona con IAM.

Amministratore IAM: un amministratore IAM potrebbe essere interessato a ottenere dei dettagli su come scrivere policy per gestire l'accesso a SimSpace Weaver. Per visualizzare esempi di policy SimSpace Weaver basate sull'identità che puoi utilizzare in IAM, consulta. Esempi di policy basate sull'identità per AWS SimSpace Weaver

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi essere autenticato (aver effettuato l' Utente root dell'account AWS accesso AWS) come utente IAM o assumendo un ruolo IAM.

Puoi accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAM Identity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Se accedi come identità federata, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Quando accedi AWS utilizzando la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente.

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le tue richieste utilizzando le tue credenziali. Se non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sul metodo

consigliato per la firma delle richieste, consulta Signature Version 4 AWS per le richieste API nella Guida per l'utente IAM.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, ti AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consulta Autenticazione a più fattori nella Guida per l'utente di AWS IAM Identity Center e Utilizzo dell'autenticazione a più fattori (MFA)AWS in IAM nella Guida per l'utente IAM.

Account AWS utente root

Quando si crea un account Account AWS, si inizia con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conserva le credenziali dell'utente root e utilizzale per eseguire le operazioni che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono l'accesso come utente root, consulta la sezione Attività che richiedono le credenziali dell'utente root nella Guida per l'utente IAM.

Identità federata

Come procedura consigliata, richiedi agli utenti umani, compresi gli utenti che richiedono l'accesso come amministratore, di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente dell'elenco utenti aziendale, un provider di identità Web AWS Directory Service, la directory Identity Center o qualsiasi utente che accede Servizi AWS utilizzando credenziali fornite tramite un'origine di identità. Quando le identità federate accedono Account AWS, assumono ruoli e i ruoli forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. Puoi creare utenti e gruppi in IAM Identity Center oppure puoi connetterti e sincronizzarti con un set di utenti e gruppi nella tua fonte di identità per utilizzarli su tutte le tue applicazioni. Account AWS Per ulteriori informazioni su IAM Identity Center, consulta Cos'è IAM Identity Center? nella Guida per l'utente di AWS IAM Identity Center .

Utenti e gruppi IAM

Un utente IAM è un'identità interna Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ove possibile, consigliamo di fare affidamento a credenziali

temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso. Tuttavia, se si hanno casi d'uso specifici che richiedono credenziali a lungo termine con utenti IAM, si consiglia di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta la pagina Rotazione periodica delle chiavi di accesso per casi d'uso che richiedono credenziali a lungo termine nella Guida per l'utente IAM.

Un gruppo IAM è un'identità che specifica un insieme di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, potresti avere un gruppo denominato IAMAdminse concedere a quel gruppo le autorizzazioni per amministrare le risorse IAM.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per ulteriori informazioni, consulta Casi d'uso per utenti IAM nella Guida per l'utente IAM.

Ruoli IAM

Un ruolo IAM è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. Per assumere temporaneamente un ruolo IAM in AWS Management Console, puoi passare da un ruolo utente a un ruolo IAM (console). Puoi assumere un ruolo chiamando un'operazione AWS CLI o AWS API o utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta Utilizzo di ruoli IAM nella Guida per l'utente IAM.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- Accesso utente federato: per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per ulteriori informazioni sulla federazione dei ruoli, consulta Create a role for a third-party identity provider (federation) nella Guida per l'utente IAM. Se utilizzi IAM Identity Center, configura un set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per informazioni sui set di autorizzazioni, consulta Set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .

- Autorizzazioni utente IAM temporanee: un utente IAM o un ruolo può assumere un ruolo IAM per ottenere temporaneamente autorizzazioni diverse per un'attività specifica.
- Accesso multi-account: è possibile utilizzare un ruolo IAM per permettere a un utente (un principale affidabile) con un account diverso di accedere alle risorse nell'account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per informazioni sulle differenze tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.
- Accesso a più servizi: alcuni Servizi AWS utilizzano le funzionalità di altri Servizi AWS. Ad esempio, quando effettui una chiamata in un servizio, è normale che quel servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.
  - Sessioni di accesso inoltrato (FAS): quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, combinate con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta Forward access sessions.
  - Ruolo di servizio: un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Create a role to delegate permissions to an Servizio AWS nella Guida per l'utente IAM.
  - Ruolo collegato al servizio: un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.
- Applicazioni in esecuzione su Amazon EC2: puoi utilizzare un ruolo IAM per gestire le credenziali temporanee per le applicazioni in esecuzione su un' EC2 istanza e che AWS CLI effettuano richieste AWS API. Questa soluzione è preferibile alla memorizzazione delle chiavi di accesso all'interno dell' EC2 istanza. Per assegnare un AWS ruolo a un' EC2 istanza e renderlo disponibile per tutte le sue applicazioni, create un profilo di istanza collegato all'istanza. Un profilo di

istanza contiene il ruolo e consente ai programmi in esecuzione sull' EC2 istanza di ottenere credenziali temporanee. Per ulteriori informazioni, consulta Utilizzare un ruolo IAM per concedere le autorizzazioni alle applicazioni in esecuzione su EC2 istanze Amazon nella IAM User Guide.

Gestione dell'accesso con policy

Puoi controllare l'accesso AWS creando policy e collegandole a AWS identità o risorse. Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale (utente, utente root o sessione di ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per ulteriori informazioni sulla struttura e sui contenuti dei documenti delle policy JSON, consulta Panoramica delle policy JSON nella Guida per l'utente IAM.

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti l'autorizzazione a eseguire operazioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Le policy IAM definiscono le autorizzazioni relative a un'operazione, a prescindere dal metodo utilizzato per eseguirla. Ad esempio, supponiamo di disporre di una policy che consente l'operazione `iam:GetRole`. Un utente con tale policy può ottenere informazioni sul ruolo dall' AWS Management Console AWS CLI, dall'o dall' AWS API.

Policy basate sull'identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le operazioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente nella Guida per l'utente IAM.

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono integrate direttamente in un singolo utente, gruppo o ruolo. Le politiche gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli nel tuo Account AWS.

Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta Scelta fra policy gestite e policy inline nella Guida per l'utente IAM.

Policy basate sulle risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Esempi di policy basate sulle risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le operazioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non puoi utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Liste di controllo degli accessi () ACLs

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3 e Amazon VPC sono esempi di servizi che supportano. AWS WAF ACLs Per ulteriori informazioni ACLs, consulta la panoramica della lista di controllo degli accessi (ACL) nella Amazon Simple Storage Service Developer Guide.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi e meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- Limiti delle autorizzazioni: un limite delle autorizzazioni è una funzionalità avanzata nella quale si imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM (utente o ruolo IAM). È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal`sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta Limiti delle autorizzazioni per le entità IAM nella Guida per l'utente IAM.

- Politiche di controllo del servizio (SCPs): SCPs sono politiche JSON che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in. AWS Organizations AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più di proprietà dell' Account AWS azienda. Se abiliti tutte le funzionalità di un'organizzazione, puoi applicare le politiche di controllo del servizio (SCPs) a uno o tutti i tuoi account. L'SCP limita le autorizzazioni per le entità presenti negli account dei membri, inclusa ciascuna di esse. Utente root dell'account AWS Per ulteriori informazioni su Organizations and SCPs, consulta le politiche di controllo dei servizi nella Guida AWS Organizations per l'utente.
- Politiche di controllo delle risorse (RCPs): RCPs sono politiche JSON che puoi utilizzare per impostare le autorizzazioni massime disponibili per le risorse nei tuoi account senza aggiornare le politiche IAM allegate a ciascuna risorsa di tua proprietà. L'RCP limita le autorizzazioni per le risorse negli account dei membri e può influire sulle autorizzazioni effettive per le identità, incluse le Utente root dell'account AWS, indipendentemente dal fatto che appartengano o meno all'organizzazione. Per ulteriori informazioni su Organizations e RCPs, incluso un elenco di Servizi AWS tale supporto RCPs, vedere Resource control policies (RCPs) nella Guida per l'AWS Organizations utente.
- Policy di sessione: le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta Policy di sessione nella Guida per l'utente IAM.

Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire o meno una richiesta quando sono coinvolti più tipi di policy, consulta la logica di valutazione delle policy nella IAM User Guide.

Come AWS SimSpace Weaver funziona con IAM

Prima di utilizzare IAM per gestire l'accesso a SimSpace Weaver, scopri con quali funzionalità IAM è disponibile l'uso SimSpace Weaver.

Funzionalità IAM che puoi utilizzare con AWS SimSpace Weaver

Funzionalità IAM	SimSpace Weaver supporto
Policy basate su identità	Sì
Policy basate su risorse	No
Azioni di policy	Sì
Risorse relative alle policy	Sì
Chiavi di condizione della policy (specifica del servizio)	Sì
ACLs	No
ABAC (tag nelle policy)	Sì
Credenziali temporanee	Sì
Autorizzazioni del principale	Sì
Ruoli di servizio	Sì
Ruoli collegati al servizio	No

Per avere una panoramica di alto livello su come SimSpace Weaver e altri AWS servizi funzionano con la maggior parte delle funzionalità IAM, consulta AWS i servizi che funzionano con IAM nella IAM User Guide.

Politiche basate sull'identità per SimSpace Weaver

Supporta le policy basate su identità: sì

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le operazioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente nella Guida per l'utente IAM.

Con le policy basate su identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Non è possibile specificare l'entità principale in una policy basata sull'identità perché si applica all'utente o al ruolo a cui è associato. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta Guida di riferimento agli elementi delle policy JSON IAM nella Guida per l'utente di IAM.

Esempi di politiche basate sull'identità per SimSpace Weaver

Per visualizzare esempi di politiche basate sull' SimSpace Weaver identità, vedere. Esempi di policy basate sull'identità per AWS SimSpace Weaver

Politiche basate sulle risorse all'interno SimSpace Weaver

Supporta le policy basate su risorse: no

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Esempi di policy basate sulle risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le operazioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per consentire l'accesso multi-account, puoi specificare un intero account o entità IAM in un altro account come principale in una policy basata sulle risorse. L'aggiunta di un principale multi-account a una policy basata sulle risorse rappresenta solo una parte della relazione di trust. Quando il principale e la risorsa sono diversi Account AWS, un amministratore IAM dell'account affidabile deve inoltre concedere all'entità principale (utente o ruolo) l'autorizzazione ad accedere alla risorsa. L'autorizzazione viene concessa collegando all'entità una policy basata sull'identità. Tuttavia, se una policy basata su risorse concede l'accesso a un principale nello stesso account, non sono richieste ulteriori policy basate su identità. Per ulteriori informazioni, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

Azioni politiche per SimSpace Weaver

Supporta le operazioni di policy: si

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento `Action`di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Le azioni politiche in genere hanno lo stesso nome dell'operazione AWS API associata. Ci sono alcune eccezioni, ad esempio le operazioni di sola autorizzazione che non hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per visualizzare un elenco di SimSpace Weaver azioni, vedere Azioni definite da AWS SimSpace Weaver nel Service Authorization Reference.

Le azioni politiche in SimSpace Weaver uso utilizzano il seguente prefisso prima dell'azione:

```
simspaceweaver
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

```
"Action": [
      "simspaceweaver:action1",
      "simspaceweaver:action2"
         ]
```

Per visualizzare esempi di politiche SimSpace Weaver basate sull'identità, vedere. Esempi di policy basate sull'identità per AWS SimSpace Weaver

Risorse politiche per SimSpace Weaver

Supporta le risorse di policy: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento JSON `Resource`della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Le istruzioni devono includere un elemento `Resource`o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo nome della risorsa Amazon (ARN). È possibile eseguire questa operazione per operazioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

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

Per visualizzare un elenco dei tipi di SimSpace Weaver risorse e relativi ARNs, vedere Resources defined by AWS SimSpace Weaver nel Service Authorization Reference. Per informazioni sulle operazioni con cui è possibile specificare l'ARN di ogni risorsa, consulta la sezione Operazioni definite da AWS SimSpace Weaver.

Per visualizzare esempi di politiche SimSpace Weaver basate sull'identità, vedere. Esempi di policy basate sull'identità per AWS SimSpace Weaver

Chiavi relative alle condizioni delle politiche per SimSpace Weaver

Supporta le chiavi di condizione delle policy specifiche del servizio: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento `Condition`(o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition`è facoltativo. È possibile compilare espressioni condizionali che utilizzano operatori di condizione, ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi `Condition`in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione `AND`logica. Se si specificano più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione logica. `OR` Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

È possibile anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, è possibile autorizzare un utente IAM ad accedere a una risorsa solo se è stata taggata con il relativo nome utente IAM. Per ulteriori informazioni, consulta Elementi delle policy IAM: variabili e tag nella Guida per l'utente di IAM.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'utente IAM.

Per visualizzare un elenco di chiavi di SimSpace Weaver condizione, consulta Condition keys for AWS SimSpace Weaver nel Service Authorization Reference. Per sapere con quali azioni e risorse puoi utilizzare una chiave di condizione, vedi Azioni definite da AWS SimSpace Weaver.

Per visualizzare esempi di politiche SimSpace Weaver basate sull'identità, vedere. Esempi di policy basate sull'identità per AWS SimSpace Weaver

Liste di controllo degli accessi () in ACLs SimSpace Weaver

Supporti ACLs: no

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

Controllo degli accessi basato sugli attributi (ABAC) con SimSpace Weaver

Supporta ABAC (tag nelle policy): sì

Il controllo dell'accesso basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, questi attributi sono chiamati tag. Puoi allegare tag a entità IAM (utenti o ruoli) e a molte AWS risorse. L'assegnazione di tag alle entità e alle risorse è il primo passaggio di ABAC. In seguito, vengono progettate policy ABAC per consentire operazioni quando il tag dell'entità principale corrisponde al tag sulla risorsa a cui si sta provando ad accedere.

La strategia ABAC è utile in ambienti soggetti a una rapida crescita e aiuta in situazioni in cui la gestione delle policy diventa impegnativa.

Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Yes (Sì). Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per ulteriori informazioni su ABAC, consulta Definizione delle autorizzazioni con autorizzazione ABAC nella Guida per l'utente IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta Utilizzo del controllo degli accessi basato su attributi (ABAC) nella Guida per l'utente di IAM.

Utilizzo di credenziali temporanee con SimSpace Weaver

Supporta le credenziali temporanee: sì

Alcuni Servizi AWS non funzionano quando si accede utilizzando credenziali temporanee. Per ulteriori informazioni, incluse quelle che Servizi AWS funzionano con credenziali temporanee, consulta la sezione relativa alla Servizi AWS compatibilità con IAM nella IAM User Guide.

Stai utilizzando credenziali temporanee se accedi AWS Management Console utilizzando qualsiasi metodo tranne nome utente e password. Ad esempio, quando accedete AWS utilizzando il link Single Sign-On (SSO) della vostra azienda, tale processo crea automaticamente credenziali temporanee. Le credenziali temporanee vengono create in automatico anche quando accedi alla console come utente e poi cambi ruolo. Per ulteriori informazioni sullo scambio dei ruoli, consulta Passaggio da un ruolo utente a un ruolo IAM (console) nella Guida per l'utente IAM.

È possibile creare manualmente credenziali temporanee utilizzando l'API or. AWS CLI AWS È quindi possibile utilizzare tali credenziali temporanee per accedere. AWS AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta Credenziali di sicurezza provvisorie in IAM.

Autorizzazioni principali multiservizio per SimSpace Weaver

Supporta l'inoltro delle sessioni di accesso (FAS): sì

Quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, in combinazione con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta Forward access sessions.

Ruoli di servizio per SimSpace Weaver

Supporta i ruoli di servizio: sì

Un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Create a role to delegate permissions to an Servizio AWS nella Guida per l'utente IAM.

> **Warning**
>
> La modifica delle autorizzazioni per un ruolo di servizio potrebbe compromettere la funzionalità. SimSpace Weaver Modifica i ruoli di servizio solo quando viene SimSpace Weaver fornita una guida in tal senso.

Gli script SDK SimSpace Weaver dell'app utilizzano un AWS CloudFormation modello per creare risorse in altri AWS servizi a supporto della simulazione. Una di queste risorse è il ruolo dell'app per la simulazione. SimSpace Weaver assume il ruolo dell'app di eseguire azioni per conto dell'utente Account AWS , ad esempio scrivere i dati di registro in Logs. CloudWatch Per ulteriori informazioni sul ruolo dell'app, consulta. Autorizzazioni create per te SimSpace Weaver

Ruoli collegati ai servizi per SimSpace Weaver

Supporta i ruoli collegati ai servizi: no

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

Per ulteriori informazioni su come creare e gestire i ruoli collegati ai servizi, consulta Servizi AWS supportati da IAM. Trova un servizio nella tabella che include un `Yes` nella colonna Service-linked role (Ruolo collegato ai servizi). Scegli il collegamento Sì per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Esempi di policy basate sull'identità per AWS SimSpace Weaver

Per impostazione predefinita, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare risorse SimSpace Weaver . Inoltre, non possono eseguire attività utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI) o l'API. AWS Per concedere agli utenti l'autorizzazione a eseguire operazioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta Creazione di policy IAM (console) nella Guida per l'utente IAM.

Per informazioni dettagliate sulle azioni e sui tipi di risorse definiti da SimSpace Weaver, incluso il formato di ARNs per ogni tipo di risorsa, consulta Azioni, risorse e chiavi di condizione AWS SimSpace Weaver nel Service Authorization Reference.

Argomenti

- Best practice per le policy
- Utilizzo della console di SimSpace Weaver
- Consentire agli utenti di visualizzare le loro autorizzazioni
- Consenti agli utenti di creare ed eseguire simulazioni

Best practice per le policy

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare SimSpace Weaver risorse nel tuo account. Queste azioni possono comportare costi aggiuntivi per l' Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche gestite che concedono le autorizzazioni per molti casi d'uso comuni.AWS Sono disponibili nel tuo. Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta Policy gestite da AWSo Policy gestite da AWS per le funzioni dei processi nella Guida per l'utente IAM.
- Applica le autorizzazioni con privilegio minimo: quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. È possibile farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso: per limitare l'accesso a operazioni e risorse è possibile aggiungere una condizione alle tue policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente IAM.
- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla

sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta Convalida delle policy per il Sistema di analisi degli accessi IAM nella Guida per l'utente IAM.

- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungi le condizioni MFA alle policy. Per ulteriori informazioni, consulta Protezione dell'accesso API con MFA nella Guida per l'utente IAM.

Per maggiori informazioni sulle best practice in IAM, consulta Best practice di sicurezza in IAM nella Guida per l'utente di IAM.

Utilizzo della console di SimSpace Weaver

Per accedere alla AWS SimSpace Weaver console, devi disporre di un set minimo di autorizzazioni. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle SimSpace Weaver risorse del tuo. Account AWS Se crei una policy basata sull'identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti o ruoli) associate a tale policy.

Non è necessario consentire autorizzazioni minime per la console agli utenti che effettuano chiamate solo verso AWS CLI o l' AWS API. Al contrario, concedi l'accesso solo alle operazioni che corrispondono all'operazione API che stanno cercando di eseguire.

Per garantire che utenti e ruoli possano ancora utilizzare la SimSpace Weaver console, allega anche la policy SimSpace Weaver `ConsoleAccess` o la policy `ReadOnly` AWS gestita alle entità. Per ulteriori informazioni, consulta Aggiunta di autorizzazioni a un utente nella Guida per l'utente IAM.

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono cpllegate alla relativa identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando l'API o a livello di codice. AWS CLI AWS

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

```
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Consenti agli utenti di creare ed eseguire simulazioni

Questo esempio di policy IAM fornisce le autorizzazioni di base necessarie per creare ed eseguire simulazioni in. 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"
                }
            }
        }
    ]
}
```

Autorizzazioni create per te SimSpace Weaver

Quando crei un SimSpace Weaver progetto, il servizio creerà un ruolo AWS Identity and Access Management (IAM) con il nome `weaver-`*`project-name`*`-app-role` e una policy di fiducia IAM. La politica di fiducia SimSpace Weaver consente di assumere il ruolo in modo che possa eseguire operazioni per te.

Politica sulle autorizzazioni dei ruoli dell'app

Il ruolo dell'app di simulazione ha i seguenti criteri di autorizzazione.

```
{
    "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": "*"
        }
    ]
}
```

Politica di attendibilità dei ruoli dell'app

SimSpace Weaver aggiunge una relazione di fiducia al ruolo dell'app di simulazione come politica di fiducia. SimSpace Weaver crea una politica di fiducia per ogni simulazione, simile all'esempio seguente.

```
{
  "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**
>
> In questo esempio, il numero di conto è `111122223333` e il nome della simulazione è. `MySimName` Questi valori sono diversi nelle vostre politiche di fiducia.

Prevenzione del problema "confused deputy" tra servizi

Il problema della confusione secondaria è un problema di sicurezza in cui un'entità che non è autorizzata a eseguire un'azione può indurre un'entità con maggiori privilegi a eseguire l'azione. Nel frattempo AWS, l'impersonificazione tra diversi servizi può portare alla confusione del vicesceriffo. La rappresentazione tra servizi può verificarsi quando un servizio (il servizio chiamante) effettua una chiamata a un altro servizio (il servizio chiamato). Il servizio chiamante può essere manipolato per utilizzare le proprie autorizzazioni e agire sulle risorse di un altro cliente, a cui normalmente non avrebbe accesso. Per evitare ciò, AWS fornisce alcuni strumenti che consentono di proteggere i dati per tutti i servizi che dispongono di principali del servizio a cui è stato consentito l'accesso alle risorse del tuo account.

Si consiglia di utilizzare `aws:SourceArn`le chiavi di contesto della condizione `aws:SourceAccount`globale nelle politiche delle risorse per limitare le autorizzazioni che AWS SimSpace Weaver forniscono un altro servizio alla risorsa. Se il valore `aws:SourceArn` non

contiene l'ID account, ad esempio il nome della risorsa Amazon (ARN) di un bucket Amazon S3, è necessario utilizzare entrambe le chiavi di contesto delle condizioni globali per limitare le autorizzazioni. Se si utilizzano entrambe le chiavi di contesto delle condizioni globali e il valore `aws:SourceArn` contiene l'ID account, il valore `aws:SourceAccount` e l'account nel valore `aws:SourceArn` deve utilizzare lo stesso ID account nella stessa dichiarazione di policy. Utilizzare `aws:SourceArn` se si desidera consentire l'associazione di una sola risorsa all'accesso tra servizi. Utilizza `aws:SourceAccount` se desideri consentire l'associazione di qualsiasi risorsa in tale account all'uso tra servizi.

Il valore di `aws:SourceArn` deve utilizzare l'ARN dell'estensione.

Il modo più efficace per proteggersi dal problema "confused deputy" è quello di usare la chiave di contesto della condizione globale `aws:SourceArn`con l'ARN completo della risorsa. Se non conosci l'ARN completo dell'estensione o specifichi più estensioni, utilizza la chiave di contesto della condizione globale `aws:SourceArn` con caratteri jolly (`*`) per le parti sconosciute dell'ARN. Ad esempio, `arn:aws:`*`simspaceweaver`*`:*:`*`111122223333`*`:*`.

L'esempio seguente mostra come utilizzare le chiavi di contesto `aws:SourceArn` e `aws:SourceAccount` global condition SimSpace Weaver per evitare il confuso problema del vice. Questa politica consentirà di SimSpace Weaver assumere il ruolo solo quando la richiesta proviene dall'account di origine specificato e viene fornita con l'ARN specificato. In questo caso, SimSpace Weaver può assumere il ruolo solo per le richieste provenienti dalle simulazioni nell'account del richiedente (*111122223333*) e solo nella regione specificata (). *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/*"
```

```
        }
      }
    }
  ]
}
```

Un modo più sicuro per scrivere questa policy consiste nell'includere il nome della simulazione in`aws:SourceArn`, come mostrato nell'esempio seguente, che limita la policy a una simulazione denominata: `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"
        }
      }
    }
  ]
}
```

Quando includi `aws:SourceArn` esplicitamente un numero di account, puoi omettere l'`Condition`elemento test per `aws:SourceAccount` (consulta la IAM User Guide per maggiori informazioni), ad esempio nella seguente policy semplificata:

```
{
  "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"
        }
      }
    }
  ]
}
```

Risoluzione dei problemi di AWS SimSpace Weaver identità e accesso

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con un SimSpace Weaver IAM.

Argomenti

- Non sono autorizzato a eseguire alcuna azione in SimSpace Weaver
- Non sono autorizzato a eseguire iam: PassRole
- Desidero visualizzare le mie chiavi di accesso
- Sono un amministratore e desidero consentire ad altri di accedere SimSpace Weaver
- Voglio consentire a persone esterne a me di accedere Account AWS alle mie risorse SimSpace Weaver

Non sono autorizzato a eseguire alcuna azione in SimSpace Weaver

Se ti AWS Management Console dice che non sei autorizzato a eseguire un'azione, devi contattare l'amministratore per ricevere assistenza. L'amministratore è la persona da cui si sono ricevuti il nome utente e la password.

L'errore di esempio seguente si verifica quando l'utente `mateojackson` IAM prova a utilizzare la console per visualizzare i dettagli relativi a una risorsa *`my-example-widget`* fittizia ma non dispone di autorizzazioni `simspaceweaver:`*`GetWidget`* fittizie.

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

In questo caso, Mateo richiede al suo amministratore di aggiornare le policy per poter accedere alla risorsa *my-example-widget* utilizzando l'azione `simspaceweaver:`*GetWidget*.

Non sono autorizzato a eseguire iam: PassRole

Se ricevi un errore che indica che non sei autorizzato a eseguire l'operazione `iam:PassRole`, le tue policy devono essere aggiornate per poter passare un ruolo a SimSpace Weaver.

Alcuni Servizi AWS consentono di passare un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

L'errore di esempio seguente si verifica quando un utente IAM denominato `marymajor` cerca di utilizzare la console per eseguire un'operazione in SimSpace Weaver. Tuttavia, l'operazione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per passare il ruolo al servizio.

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

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è colui che ti ha fornito le credenziali di accesso.

Desidero visualizzare le mie chiavi di accesso

Dopo aver creato le chiavi di accesso utente IAM, è possibile visualizzare il proprio ID chiave di accesso in qualsiasi momento. Tuttavia, non è possibile visualizzare nuovamente la chiave di accesso segreta. Se perdi la chiave segreta, dovrai creare una nuova coppia di chiavi di accesso.

Le chiavi di accesso sono composte da due parti: un ID chiave di accesso (ad esempio `AKIAIOSFODNN7EXAMPLE`) e una chiave di accesso segreta (ad esempio, `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`). Come un nome utente e una password, è necessario utilizzare sia l'ID chiave di accesso sia la chiave di accesso segreta insieme per autenticare le richieste dell'utente. Gestisci le tue chiavi di accesso in modo sicuro mentre crei il nome utente e la password.

> **Important**
>
> Non fornire le chiavi di accesso a terze parti, neppure per aiutare a trovare l'ID utente canonico. In questo modo, potresti concedere a qualcuno l'accesso permanente al tuo Account AWS.

Quando crei una coppia di chiavi di accesso, ti viene chiesto di salvare l'ID chiave di accesso e la chiave di accesso segreta in una posizione sicura. La chiave di accesso segreta è disponibile solo al momento della creazione. Se si perde la chiave di accesso segreta, è necessario aggiungere nuove chiavi di accesso all'utente IAM. È possibile avere massimo due chiavi di accesso. Se se ne hanno già due, è necessario eliminare una coppia di chiavi prima di crearne una nuova. Per visualizzare le istruzioni, consulta Gestione delle chiavi di accesso nella Guida per l'utente di IAM.

Sono un amministratore e desidero consentire ad altri di accedere SimSpace Weaver

Per consentire ad altri di accedere SimSpace Weaver, devi concedere l'autorizzazione alle persone o alle applicazioni che necessitano dell'accesso. Se si utilizza AWS IAM Identity Center per gestire persone e applicazioni, si assegnano set di autorizzazioni a utenti o gruppi per definirne il livello di accesso. I set di autorizzazioni creano e assegnano automaticamente le policy IAM ai ruoli IAM associati alla persona o all'applicazione. Per ulteriori informazioni, consulta Set di autorizzazioni nella Guida per l'AWS IAM Identity Center utente.

Se non utilizzi IAM Identity Center, devi creare entità IAM (utenti o ruoli) per le persone o le applicazioni che necessitano di accesso. Dovrai quindi collegare all'entità una policy che conceda le autorizzazioni corrette in SimSpace Weaver. Dopo aver concesso le autorizzazioni, fornisci le credenziali all'utente o allo sviluppatore dell'applicazione. Utilizzeranno tali credenziali per accedere. AWSPer ulteriori informazioni sulla creazione di utenti, gruppi, policy e autorizzazioni IAM, consulta IAM Identities and Policies and permissions in IAM nella IAM User Guide.

Voglio consentire a persone esterne a me di accedere Account AWS alle mie risorse SimSpace Weaver

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per i servizi che supportano politiche basate sulle risorse o liste di controllo degli accessi (ACLs), puoi utilizzare tali politiche per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se SimSpace Weaver supporta queste funzionalità, consulta. Come AWS SimSpace Weaver funziona con IAM
- Per scoprire come fornire l'accesso alle tue risorse attraverso Account AWS le risorse di tua proprietà, consulta Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà nella IAM User Guide.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta Fornire l'accesso a soggetti Account AWS di proprietà di terze parti nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta Fornire l'accesso a utenti autenticati esternamente (Federazione delle identità) nella Guida per l'utente IAM.
- Per informazioni sulle differenze di utilizzo tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

Registrazione e monitoraggio degli eventi di sicurezza AWS SimSpace Weaver

Il monitoraggio è un elemento importante per mantenere l'affidabilità, la disponibilità e le prestazioni delle SimSpace Weaver vostre AWS soluzioni. È necessario raccogliere i dati di monitoraggio da tutte le parti della AWS soluzione in modo da poter eseguire più facilmente il debug di un errore multipunto, se si verifica.

AWS e SimSpace Weaver forniscono diversi strumenti per monitorare le risorse di simulazione e rispondere a potenziali incidenti.

Accedi in Amazon CloudWatch

SimSpace Weaver memorizza i dati di accesso. CloudWatch È possibile utilizzare questi registri per monitorare gli eventi della simulazione (come l'avvio e l'arresto delle app) e per il debug. Per ulteriori informazioni, consulta SimSpace Weaver accessi in Amazon CloudWatch Logs.

CloudWatch Allarmi Amazon

Utilizzando Amazon CloudWatch alarms, controlli una singola metrica per un periodo di tempo specificato. Se la metrica supera una determinata soglia, viene inviata una notifica a un argomento o a una politica di Auto AWS Scaling di Amazon SNS. CloudWatch gli allarmi vengono attivati quando il loro stato cambia e vengono mantenuti per un determinato numero di periodi, non perché si trovano

in uno stato particolare. Per ulteriori informazioni, consulta Monitoraggio SimSpace Weaver con Amazon CloudWatch.

AWS CloudTrail Registri

CloudTrail fornisce un registro delle azioni intraprese da un utente, un ruolo o un AWS servizio in SimSpace Weaver. Utilizzando le informazioni raccolte da CloudTrail, è possibile determinare a quale richiesta è stata inviata SimSpace Weaver, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi. Per ulteriori informazioni, consulta Registrazione delle chiamate AWS SimSpace Weaver API utilizzando AWS CloudTrail.

Convalida della conformità per AWS SimSpace Weaver

SimSpace Weaver non rientra nell'ambito di alcun programma di AWS conformità.

I revisori esterni valutano la sicurezza e la conformità di altri AWS servizi nell'ambito di più programmi di AWS conformità. Questi includono SOC, PCI, FedRAMP, HIPAA e altri.

Per sapere se un Servizio AWS programma rientra nell'ambito di specifici programmi di conformità, consulta Servizi AWS la sezione Ambito per programma di conformitàServizi AWS di conformità e scegli il programma di conformità che ti interessa. Per informazioni generali, consulta Programmi di AWS conformità Programmi di di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact .

La vostra responsabilità di conformità durante l'utilizzo Servizi AWS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- Governance e conformità per la sicurezza: queste guide all'implementazione di soluzioni illustrano considerazioni relative all'architettura e i passaggi per implementare le funzionalità di sicurezza e conformità.
- Riferimenti sui servizi conformi ai requisiti HIPAA: elenca i servizi HIPAA idonei. Non tutti Servizi AWS sono idonei alla normativa HIPAA.
- AWS Risorse per la per la conformità: questa raccolta di cartelle di lavoro e guide potrebbe essere valida per il tuo settore e la tua località.
- AWS Guide alla conformità dei clienti: comprendi il modello di responsabilità condivisa attraverso la lente della conformità. Le guide riassumono le migliori pratiche per la protezione Servizi AWS e

mappano le linee guida per i controlli di sicurezza su più framework (tra cui il National Institute of Standards and Technology (NIST), il Payment Card Industry Security Standards Council (PCI) e l'International Organization for Standardization (ISO)).

- Valutazione delle risorse con regole nella Guida per gli AWS Config sviluppatori: il AWS Config servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida e alle normative del settore.
- AWS Security Hub— Ciò Servizio AWS fornisce una visione completa dello stato di sicurezza interno. AWS La Centrale di sicurezza utilizza i controlli di sicurezza per valutare le risorse AWS e verificare la conformità agli standard e alle best practice del settore della sicurezza. Per un elenco dei servizi e dei controlli supportati, consulta la pagina Documentazione di riferimento sui controlli della Centrale di sicurezza.
- Amazon GuardDuty: Servizio AWS rileva potenziali minacce ai tuoi carichi di lavoro Account AWS, ai contenitori e ai dati monitorando l'ambiente alla ricerca di attività sospette e dannose. GuardDuty può aiutarti a soddisfare vari requisiti di conformità, come lo standard PCI DSS, soddisfacendo i requisiti di rilevamento delle intrusioni imposti da determinati framework di conformità.
- AWS Audit Manager— Ciò Servizio AWS consente di verificare continuamente l' AWS utilizzo per semplificare la gestione del rischio e la conformità alle normative e agli standard di settore.

Resilienza in AWS SimSpace Weaver

L'infrastruttura AWS globale è costruita attorno a Regioni AWS zone di disponibilità. Regioni AWS forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità, puoi progettare e gestire applicazioni e database che eseguono automaticamente il failover tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

Per ulteriori informazioni sulle zone di disponibilità, vedere Global Regioni AWS Infrastructure.AWS

Sicurezza dell'infrastruttura in AWS SimSpace Weaver

In quanto servizio gestito, AWS SimSpace Weaver è protetto dalla sicurezza di rete AWS globale. Per informazioni sui servizi AWS di sicurezza e su come AWS protegge l'infrastruttura, consulta AWS Cloud Security. Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Utilizzate chiamate API AWS pubblicate per accedere SimSpace Weaver attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. O puoi utilizzare AWS Security Token Service (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

Modello di sicurezza della connettività di rete

Le tue simulazioni vengono eseguite su istanze di calcolo all'interno di un Amazon VPC situato in una AWS regione selezionata. Un Amazon VPC è una rete virtuale nel AWS cloud, che isola l'infrastruttura in base al carico di lavoro o all'entità organizzativa. Le comunicazioni tra le istanze di calcolo all'interno di Amazon VPC rimangono all'interno della AWS rete e non viaggiano su Internet. Alcune comunicazioni di servizio interno attraversano Internet e sono crittografate. Le simulazioni per tutti i clienti che operano nella stessa AWS regione condividono lo stesso Amazon VPC. Le simulazioni per clienti diversi utilizzano istanze di calcolo separate all'interno dello stesso Amazon VPC.

Comunicazioni tra i tuoi client di simulazione e le simulazioni eseguite durante i viaggi su Internet. SimSpace Weaver SimSpace Weaver non gestisce queste connessioni. È tua responsabilità proteggere le connessioni dei tuoi client.

Le connessioni al SimSpace Weaver servizio attraversano Internet e sono crittografate. Ciò include le connessioni che utilizzano AWS Management Console, AWS Command Line Interface (AWS CLI), i kit di sviluppo AWS software (SDK) e l'SDK dell' SimSpace Weaver app.

Analisi della configurazione e della vulnerabilità in AWS SimSpace Weaver

La configurazione e i controlli IT sono una responsabilità condivisa tra te AWS e te. Per ulteriori informazioni, consulta il modello di responsabilità AWS condivisa di . AWS gestisce le attività di sicurezza di base per l'infrastruttura sottostante, come l'applicazione di patch al sistema operativo

sulle istanze di calcolo, la configurazione del firewall e il disaster recovery AWS dell'infrastruttura. Queste procedure sono state riviste e certificate dalle terze parti appropriate. Per ulteriori dettagli, consulta Best practice per la sicurezza, l'identità e la conformità.

Sei responsabile della sicurezza del tuo software di simulazione:

- Gestisci il codice dell'app, inclusi aggiornamenti e patch di sicurezza.
- Autentica e crittografa le comunicazioni tra i tuoi client di simulazione e le app a cui si connettono.
- Aggiorna le simulazioni per utilizzare le versioni SDK più recenti, inclusi l'SDK e l'SDK dell' AWS app. SimSpace Weaver

> **Note**
>
> SimSpace Weaver non supporta gli aggiornamenti delle app in una simulazione in esecuzione. Se devi aggiornare le app, devi interrompere ed eliminare la simulazione, quindi creare una nuova simulazione con il codice dell'app aggiornato. Ti consigliamo di salvare lo stato della simulazione in un archivio dati esterno in modo da poterlo ripristinare se devi ricreare la simulazione.

Le migliori pratiche di sicurezza per SimSpace Weaver

Questa sezione descrive le migliori pratiche di sicurezza specifiche per SimSpace Weaver. Per ulteriori informazioni sulle best practice di sicurezza in AWS, consulta Best practice for Security, Identity and Compliance.

Argomenti

- Crittografa le comunicazioni tra le tue app e i relativi client
- Effettua periodicamente il backup dello stato della simulazione
- Gestisci le tue app e SDKs

Crittografa le comunicazioni tra le tue app e i relativi client

SimSpace Weaver non gestisce le comunicazioni tra le tue app e i loro client. È necessario implementare una qualche forma di autenticazione e crittografia per le sessioni client.

Effettua periodicamente il backup dello stato della simulazione

SimSpace Weaver non salva lo stato della simulazione. Le simulazioni interrotte (a seguito di una chiamata API, di un'opzione di console o di un arresto anomalo del sistema) non salvano il loro stato e non hanno un modo intrinseco per ripristinarle. Le simulazioni interrotte non possono essere riavviate. L'unico modo per eseguire l'equivalente di un riavvio è ricreare la simulazione utilizzando la stessa configurazione e gli stessi dati. È possibile utilizzare i backup dello stato della simulazione per inizializzare la nuova simulazione. AWS offre servizi di archiviazione e database su cloud altamente affidabili e disponibili che è possibile utilizzare per salvare lo stato della simulazione.

Gestisci le tue app e SDKs

Gestisci le app, le installazioni locali dei kit di sviluppo AWS software (SDKs) e l'SDK dell' SimSpace Weaver app. Puoi scaricare e installare nuove versioni di. AWS SDKs Prova le nuove versioni dell'SDK dell' SimSpace Weaver app con versioni di app non di produzione per assicurarti che le tue app continuino a funzionare come previsto. Non puoi aggiornare le app in una simulazione in esecuzione. Per aggiornare le app:

1. Aggiorna e testa il codice dell'app localmente (o in un ambiente di test).
2. Smetti di cambiare lo stato della simulazione e salvalo (se necessario).
3. Interrompi la simulazione (una volta interrotta, non può essere riavviata).
4. Elimina la simulazione (le simulazioni interrotte che non sono state eliminate vengono conteggiate ai fini dei limiti del servizio).
5. Ricrea la simulazione con la stessa configurazione e il codice dell'app aggiornato.
6. Inizializza la simulazione utilizzando i dati di stato salvati (se disponibili).
7. Inizia la tua nuova simulazione.

> **Note**
>
> Una nuova simulazione creata con la stessa configurazione è separata dalla vecchia simulazione. Avrà un nuovo ID di simulazione e invierà i log a un nuovo flusso di log in Amazon. CloudWatch

Registrazione e monitoraggio SimSpace Weaver

Il monitoraggio è un elemento importante per mantenere l'affidabilità, la disponibilità e le prestazioni delle SimSpace Weaver altre AWS soluzioni esistenti. AWS fornisce i seguenti strumenti di monitoraggio per osservare SimSpace Weaver, segnalare quando qualcosa non va e intraprendere azioni automatiche quando necessario:

- Amazon CloudWatch monitora AWS le tue risorse e le applicazioni su cui esegui AWS in tempo reale. Puoi raccogliere i parametri e tenerne traccia, creare pannelli di controllo personalizzati e impostare allarmi per inviare una notifica o intraprendere azioni quando un parametro specificato raggiunge una determinata soglia. Per ulteriori informazioni, consulta la Amazon CloudWatch User Guide.
- Amazon CloudWatch Logs ti consente di monitorare, archiviare e accedere ai dati di log provenienti dai tuoi SimSpace Weaver dipendenti e da altre fonti. CloudTrail CloudWatch I log possono monitorare le informazioni contenute nei dati di registro e avvisarti quando vengono raggiunte determinate soglie. Puoi inoltre archiviare i dati del log in storage estremamente durevole. Per ulteriori informazioni, consulta la Amazon CloudWatch Logs User Guide.
- AWS CloudTrail acquisisce le chiamate API e gli eventi correlati effettuati da o per conto del tuo Account AWS e fornisce i file di log a un bucket Simple Storage Service (Amazon S3) specificato. Puoi identificare quali utenti e account hanno richiamato AWS, l'indirizzo IP di origine da cui sono state effettuate le chiamate e quando sono avvenute. Per ulteriori informazioni, consulta la AWS CloudTrail Guida per l'utente di .

Argomenti

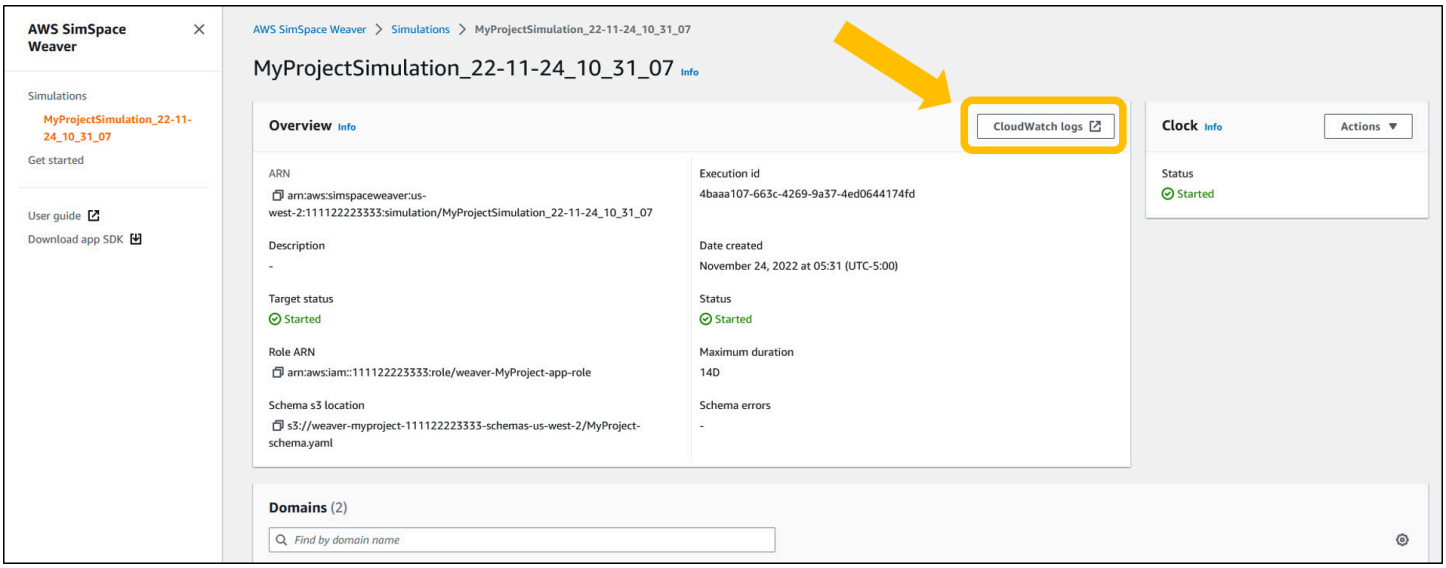
- SimSpace Weaver accessi in Amazon CloudWatch Logs
- Monitoraggio SimSpace Weaver con Amazon CloudWatch
- Registrazione delle chiamate AWS SimSpace Weaver API utilizzando AWS CloudTrail

SimSpace Weaver accessi in Amazon CloudWatch Logs

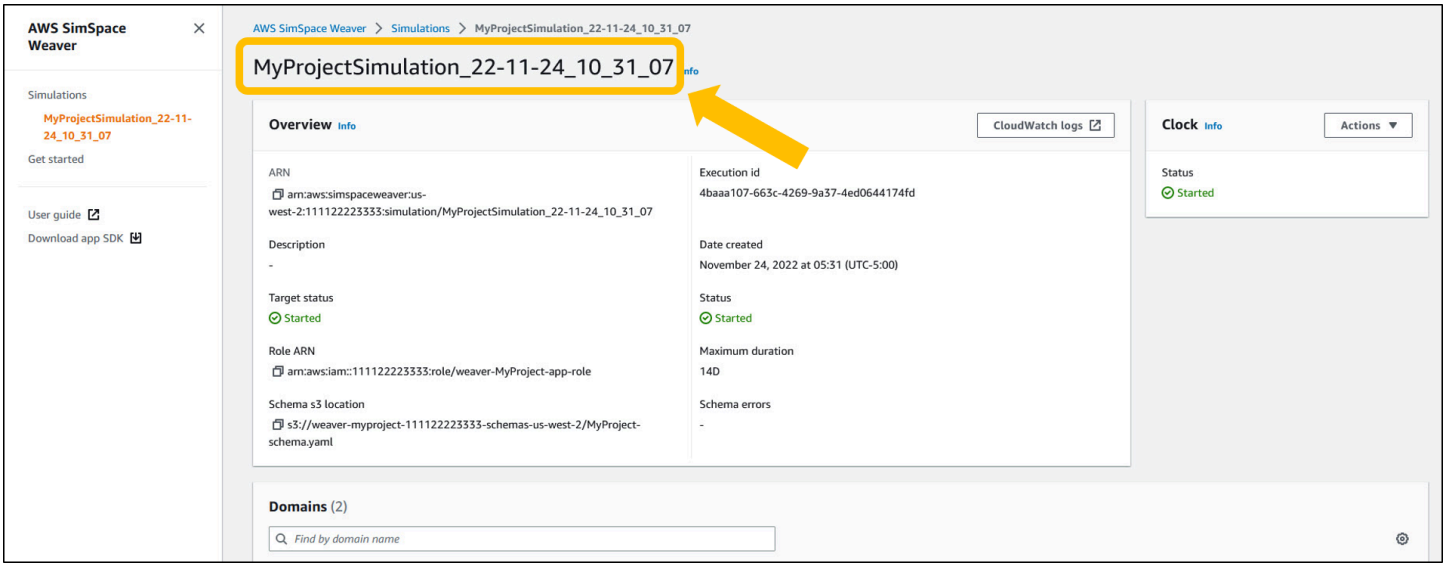
Accedere ai tuoi registri SimSpace Weaver

Tutti i log generati dalle tue SimSpace Weaver simulazioni vengono archiviati in Amazon CloudWatch Logs. Per accedere ai log, puoi utilizzare il pulsante CloudWatch Logs nel riquadro Panoramica della

simulazione nella SimSpace Weaver console, che ti porterà direttamente ai log di quella specifica simulazione.



Puoi accedere ai log anche tramite la console. CloudWatch Avrai bisogno del nome della simulazione per cercarne i log.



SimSpace Weaver registri

SimSpace Weaver scrive i messaggi di gestione della simulazione e l'output della console dalle tue app su Amazon CloudWatch Logs. Per ulteriori informazioni sull'utilizzo dei log, consulta Working with log groups and log stream nella Amazon CloudWatch Logs User Guide.

Ogni simulazione che crei ha il proprio gruppo di log in Logs. CloudWatch Il nome del gruppo di log è specificato nello schema di simulazione. Nel frammento di schema

seguente, il valore di è. `log_destination_service` `logs` Ciò significa che il valore di `log_destination_resource_name` è il nome di un gruppo di log. In questo caso, il gruppo di log è`MySimulationLogs`.

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

Puoi anche utilizzare l'DescribeSimulationAPI per trovare il nome del gruppo di log per la simulazione dopo averlo avviato.

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

L'esempio seguente mostra la parte dell'output DescribeSimulation che descrive la configurazione di registrazione. Il nome del gruppo di log viene visualizzato alla fine di`LogGroupArn`.

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

Ogni gruppo di log di simulazione contiene diversi flussi di log:

- Flusso di log di gestione: messaggi di gestione della simulazione prodotti dal servizio. SimSpace Weaver

  ```
  /sim/management
  ```

- Flusso di registro degli errori: messaggi di errore prodotti dal SimSpace Weaver servizio. Questo flusso di log esiste solo in presenza di errori. SimSpace Weaver archivia gli errori scritti dalle tue app nei rispettivi flussi di log delle app (vedi i seguenti flussi di log).

```
/sim/errors
```

- Stream di log spaziali delle app (1 per ogni app spaziale su ciascun worker): output della console prodotto dalle app spaziali. Ogni app spaziale scrive nel proprio flusso di log. Sono *spatial-app-id* tutti i caratteri dopo la barra finale alla fine di. *worker-id*

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

- Stream di log delle app personalizzati (1 per ogni istanza di app personalizzata): output della console prodotto da app personalizzate. Ogni istanza dell'app personalizzata scrive nel proprio flusso di log.

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

- Stream di log delle app di servizio (1 per ogni istanza dell'app di servizio): output della console prodotto dalle app di servizio. Ogni app di servizio scrive nel proprio flusso di log. Sono *service-app-id* tutti i caratteri dopo la barra finale alla fine di. *service-app-name*

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

Monitoraggio SimSpace Weaver con Amazon CloudWatch

Puoi monitorare SimSpace Weaver utilizzando Amazon CloudWatch, che raccoglie dati grezzi e li elabora in metriche leggibili quasi in tempo reale. Queste statistiche vengono conservate per un periodo di 15 mesi, per permettere l'accesso alle informazioni storiche e offrire una prospettiva migliore sulle prestazioni del servizio o dell'applicazione web. È anche possibile impostare allarmi che controllano determinate soglie e inviare notifiche o intraprendere azioni quando queste soglie vengono raggiunte. Per ulteriori informazioni, consulta la Amazon CloudWatch User Guide.

Il SimSpace Weaver servizio riporta le seguenti metriche nel `AWS/simspaceweaver` namespace.

SimSpace Weaver metriche a livello di account

Il SimSpace Weaver namespace include le seguenti metriche relative all'attività a livello di account. AWS

Parametro	Descrizione
`SimulationCount`	Il numero di simulazioni per l'account corrente. Unità: numero Dimensioni: nessuna Statistiche: media, minima, massima

Registrazione delle chiamate AWS SimSpace Weaver API utilizzando AWS CloudTrail

AWS SimSpace Weaver è integrato con AWS CloudTrail, un servizio che fornisce una registrazione delle azioni intraprese da un utente, un ruolo o un AWS servizio in SimSpace Weaver. CloudTrail acquisisce tutte le chiamate API SimSpace Weaver come eventi. Le chiamate acquisite includono chiamate dalla SimSpace Weaver console e chiamate di codice alle operazioni SimSpace Weaver API. Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per. SimSpace Weaver Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console in. Event history Utilizzando le informazioni raccolte da CloudTrail, è possibile determinare a quale richiesta è stata inviata SimSpace Weaver, l'indirizzo IP da cui è stata effettuata, chi ha effettuato la richiesta, quando è stata effettuata e ulteriori dettagli.

Per ulteriori informazioni CloudTrail, consulta la Guida AWS CloudTrail per l'utente.

SimSpace Weaver informazioni in CloudTrail

CloudTrail è abilitato sul tuo account al Account AWS momento della creazione dell'account. Quando si verifica un'attività in SimSpace Weaver, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi AWS di servizio in Event history. Puoi visualizzare, cercare e scaricare eventi recenti in Account AWS. Per ulteriori informazioni, consulta Visualizzazione degli eventi con la cronologia degli CloudTrail eventi.

Per una registrazione continua degli eventi del tuo Account AWS, inclusi gli eventi di SimSpace Weaver, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando si crea un percorso nella console, questo sarà valido in tutte le Regioni AWS. Il trail registra gli eventi di tutte le regioni della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consulta gli argomenti seguenti:

- Panoramica della creazione di un percorso
- CloudTrail servizi e integrazioni supportati
- Configurazione delle notifiche Amazon SNS per CloudTrail
- Ricezione di file di CloudTrail registro da più regioni e ricezione di file di CloudTrail registro da più account

Tutte SimSpace Weaver le azioni vengono registrate CloudTrail e documentate nell'AWS SimSpace Weaver API Reference. Ad esempio, le chiamate a `DescribeSimulation` e `ListSimulations` le `DeleteSimulation` azioni generano voci nei file di CloudTrail registro.

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con credenziali utente root o AWS Identity and Access Management (IAM).
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, consulta Elemento CloudTrail userIdentity.

Comprensione delle SimSpace Weaver voci dei file di registro

Un trail è una configurazione che consente la distribuzione di eventi come file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'azione richiesta, come la data e l'ora dell'azione, i parametri della richiesta e altri dettagli. CloudTrail i file di registro non sono una traccia ordinata delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail registro che illustra l'`ListSimulationsazione`.


```
{
    "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 Endpoint e quote Weaver

Le tabelle seguenti descrivono gli endpoint e le quote di servizio per Weaver. SimSpace Le service quotas, a cui si fa riferimento anche come limiti, rappresentano il numero massimo di risorse di servizio o operazioni per l' Account AWS. Per ulteriori informazioni, consulta AWS Service Quotas in Riferimenti generali di AWS .

Endpoint del servizio

Nome Regione	Regione	Endpoint	Protocollo
US East (N. Virginia)	us-east-1	simspaceweaver.us-east-1.amazonaws.com	HTTPS
Stati Uniti orientali (Ohio)	us-east-2	simspaceweaver.us-east-2.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	simspaceweaver.us-west-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	simspaceweaver.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	simspaceweaver.ap-southeast-2.amazonaws.com	HTTPS
Europa (Stoccolma)	eu-north-1	simspaceweaver.eu-north-1.amazonaws.com	HTTPS

Nome Regione	Regione	Endpoint	Protocollo
Europe (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 (Stati Uniti orientali)	us-gov-east-1	simspaceweaver.us-gov-east-1.amazonaws.com	HTTPS
AWS GovCloud (Stati Uniti occidentali)	us-gov-west-1	simspaceweaver.us-gov-west-1.amazonaws.com	HTTPS

Quote del servizio

Nome	Predefinita	Adattabile	Descrizione
Unità di risorse di calcolo per ogni app	Ogni regione supportata: 4	No	Il numero massimo di unità di risorse di calcolo che puoi allocare a ciascuna app.
Unità di risorse di calcolo per ogni worker	Ogni regione supportata: 17	No	Il numero di unità di risorse di calcolo disponibili per ogni worker.
Campi di dati per ciascuna entità	Ogni regione supportata: 7	No	Il numero massimo di campi di dati (non indice) che può avere un'entità.

Nome	Predefinita	Adattabile	Descrizione
Entità in una partizione	Ogni regione supportata: 8.192	No	Il numero massimo di entità in 1 partizione.
Dimensione del campo di dati dell'entità	Ogni regione supportata: 1.024 byte	No	La dimensione massima di un campo di dati (non indice) di un'entità.
Trasferimenti di entità tra worker	Ogni regione supportata: 25	No	Il numero massimo di trasferimenti di entità tra worker, per ogni partizione e ogni tick.
Trasferimenti di entità sullo stesso worker	Ogni regione supportata: 500	No	Il numero massimo di trasferimenti di entità sullo stesso worker, per ogni partizione e ogni tick.
Campi di indice per ogni entità	Ogni regione supportata: 1	No	Il numero massimo di campi di indice che può avere un'entità.
Durata massima (in giorni) per una simulazione	Ogni regione supportata: 14	No	Il numero massimo di giorni che è possibile specificare come durata massima per una simulazione. Tutte le simulazioni hanno una durata massima, anche se non si specifica il valore. Una simulazione si interrompe automaticamente quando raggiunge la durata massima.

Nome	Predefinita	Adattabile	Descrizione
Memoria per ogni unità di risorse di calcolo	Ogni regione supportata: 1 GB	No	La quantità di memoria ad accesso casuale (RAM) che un'app ottiene per ogni unità di risorsa di calcolo.
Sottoscrizioni remote per ogni worker	Ogni regione supportata: 24	No	Il numero massimo di sottoscrizioni remote per ogni worker.
Numero di simulazione	Ogni regione supportata: 2	Sì	Il numero massimo di simulazioni con lo stato obiettivo di STARTED nel tuo account. Puoi richiedere un aumento della quota fino a 10.
Worker per una simulazione	Ogni regione supportata: 2	Sì	Il numero massimo di lavoratori che puoi assegnare a 1 simulazione. Puoi richiedere un aumento della quota fino a 10.
v CPUs per ogni unità di risorse di calcolo	Ogni regione supportata: 2	No	Il numero di unità di elaborazione centrale virtuali (vCPUs) che un'app ottiene per ogni unità di risorsa di elaborazione.

Quote di messaggistica

Le seguenti quote si applicano alla messaggistica da app a app per SimSpace Weaver Local e in. Cloud AWS

Nome	Predefinita	Adattabile	Descrizione
Dimensione massima del messaggio	Ogni regione supportata: 256 byte	No	La dimensione massima del payload di un messaggio.
Velocità massima di invio dei messaggi	Ogni regione supportata: 128	No	Il numero massimo di messaggi che ogni app può inviare per segno di spunta.

Frequenze di clock

Lo schema di simulazione specifica la frequenza di clock (chiamata anche frequenza di ticking) per una simulazione. La tabella seguente specifica le frequenze di clock valide che è possibile utilizzare.

Nome	Valori validi	Descrizione
Frequenza di clock	Ogni regione supportata: «10", «15", «30", «illimitato»	Le frequenze di clock valide per una simulazione.
Frequenza di clock (versioni 1.13 e 1.12)	Ogni regione supportata: 10, 15, 30	Le frequenze di clock valide per una simulazione.

Quote di servizio per SimSpace Weaver Local

Le seguenti quote di servizio si applicano a SimSpace Weaver Local solo. Tutte le altre quote si applicano anche a SimSpace Weaver Local.

Nome	Predefinita	Adattabile	Descrizione
Numero massimo di partizioni	SimSpace Weaver Local: 24	No	Il numero massimo di partizioni per una simulazione.
Numero massimo di app	SimSpace Weaver Local: 24	No	Il numero totale massimo di app (di qualsiasi tipo) per una simulazione.
Numero massimo di domini	SimSpace Weaver Local: 24	No	Il numero totale massimo di domini (di qualsiasi tipo) per una simulazione.
Entità per partizione	SimSpace Weaver Local: 4.096	No	Il numero massimo di entità in ogni partizione.
Campi per entità	SimSpace Weaver Local: 8	No	Il numero massimo di campi per ogni entità.
Dimensione dei campi	SimSpace Weaver Local: 1024 byte	No	La dimensione massima di un campo di entità.

Risoluzione dei problemi in SimSpace Weaver

Argomenti

- AssumeRoleAccessDenied
- InvalidBucketName
- ServiceQuotaExceededException
- TooManyBuckets
- Autorizzazione negata durante l'avvio della simulazione
- Problemi legati al tempo di utilizzo Docker
- PathfindingSample il client della console non riesce a connettersi
- Il AWS CLI non riconosce simspaceweaver

AssumeRoleAccessDenied

Potresti ricevere il seguente errore se la simulazione non si avvia:

```
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
```

Puoi ricevere questo errore se si verifica una delle seguenti condizioni relative al ruolo AWS Identity and Access Management (IAM) per la tua simulazione:

- Il nome della risorsa Amazon (ARN) si riferisce a un ruolo IAM che non esiste.
- La politica di fiducia per il ruolo IAM che non consente al nome della nuova simulazione di assumere il ruolo.

Verifica che il ruolo esista. Se il ruolo esiste, controlla la tua politica di fiducia per il ruolo. L'`aws:SourceArn`esempio seguente, la politica di fiducia consente di assumere il ruolo solo con una simulazione (nell'account 111122223333) il cui nome inizia con`MySimulation`.

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

```
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "simspaceweaver.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:simspaceweaver:us-
west-2:111122223333:simulation/MySimulation*"
                }
            }
        }
    ]
}
```

Per consentire a un'altra simulazione il cui nome inizia con `MyOtherSimulation` di assumere il ruolo, la politica di fiducia deve essere modificata come nel seguente esempio modificato:

```
{
    "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

Potresti ricevere il seguente errore durante la creazione di un progetto:

```
An error occurred (InvalidBucketName) when calling the CreateBucket operation: The
 specified bucket is not valid.
```

Hai ricevuto questo errore perché il nome SimSpace Weaver passato ad Amazon Simple Storage Service (Amazon S3) violava le regole di denominazione dei bucket (per ulteriori informazioni, consulta le regole di denominazione dei bucket nella Guida per l'utente di Amazon Simple Storage Service).

Lo `create-project` script nell'SDK dell' SimSpace Weaver app crea i nomi dei bucket utilizzando il nome del progetto che fornisci allo script. I nomi dei bucket utilizzano i seguenti formati:

- Versione 1.13.x o successiva
  - `weaver-`*`lowercase-project-name`*`-`*`account-number`*`-`*`region`*
- Versione 1.12.x
  - `weaver-`*`lowercase-project-name`*`-`*`account-number`*`-app-zips-`*`region`*
  - `weaver-`*`lowercase-project-name`*`-`*`account-number`*`-schemas-`*`region`*

Ad esempio, date le seguenti proprietà del progetto:

- Nome del progetto: `MyProject`
- Account AWS numero: `111122223333`
- Regione AWS: `us-west-2`

Il progetto avrebbe i seguenti bucket:

- Versione 1.13.x o successiva
  - `weaver-myproject-111122223333-us-west-2`
- Versione 1.12.x
  - `weaver-myproject-111122223333-app-zips-us-west-2`
  - `weaver-myproject-111122223333-schemas-us-west-2`

Il nome del progetto non deve violare le regole di denominazione di Amazon S3. È inoltre necessario utilizzare un nome di progetto sufficientemente breve in modo che i nomi dei bucket creati dallo `create-project` script non superino il limite di lunghezza dei nomi per i bucket Amazon S3.

ServiceQuotaExceededException

Potresti ricevere il seguente errore quando avvii una simulazione:

```
An error occurred (ServiceQuotaExceededException) when calling the StartSimulation
 operation: Failed to start simulation due to: simulation quota has already been
 reached.
```

Riceverai questo errore se tenti di avviare una nuova simulazione ma al momento il tuo account ha il numero massimo di simulazioni con lo stato obiettivo di STARTED. Ciò include l'esecuzione di simulazioni, le simulazioni fallite e le simulazioni che si sono interrotte perché hanno raggiunto la durata massima. È possibile eliminare una simulazione interrotta o non riuscita per consentire l'avvio di una nuova simulazione. Se tutte le simulazioni sono in esecuzione, puoi interrompere ed eliminare una simulazione in esecuzione. Puoi anche richiedere un aumento delle quote di servizio se non hai già raggiunto il limite di richieste.

TooManyBuckets

Potresti ricevere il seguente errore durante la creazione di un progetto:

```
An error occurred (TooManyBuckets) when calling the CreateBucket operation: You have
 attempted to create more buckets than allowed.
```

Amazon Simple Storage Service (Amazon S3) ha un limite al numero di bucket che puoi avere nel AWS tuo account (per ulteriori informazioni, consulta Restrizioni e limitazioni dei bucket nella Guida per l'utente di Amazon Simple Storage Service).

Prima di continuare, devi eseguire una delle seguenti operazioni:

- Elimina 2 o più bucket Amazon S3 esistenti che non ti servono.
- Richiedi un aumento del limite di Amazon S3 (per ulteriori informazioni, consulta le restrizioni e le limitazioni dei bucket nella Guida per l'utente di Amazon Simple Storage Service).
- Usa un account diverso. AWS

> **Note**
>
> L'`DeleteSimulation`API in SimSpace Weaver non elimina le risorse Amazon S3 associate alla simulazione. Ti consigliamo di rimuovere tutte le risorse associate alle simulazioni quando non ne hai più bisogno.

Autorizzazione negata durante l'avvio della simulazione

Quando avvii una simulazione, potresti ricevere un messaggio di errore che indica che l'autorizzazione è stata negata o che si è verificato un errore durante l'accesso agli artefatti dell'app. Questo problema può verificarsi quando specifichi bucket Amazon S3 per la simulazione che SimSpace Weaver non sono stati creati per te (tramite la console o gli script SDK dell' SimSpace Weaver app).

Le seguenti situazioni sono le cause principali più probabili:

- Il servizio non dispone dell'autorizzazione per accedere a uno o più bucket Amazon S3 che hai specificato nello schema di simulazione: controlla la politica di autorizzazione dei ruoli dell'app, le policy dei bucket Amazon S3 e le autorizzazioni dei bucket Amazon S3 per assicurarti che disponga delle autorizzazioni corrette per accedere ai tuoi bucket. `simspaceweaver.amazonaws.com` Per ulteriori informazioni sulla politica di autorizzazione dei ruoli dell'app, consulta. Autorizzazioni create per te SimSpace Weaver
- Il bucket Amazon S3 potrebbe trovarsi in una situazione Regione AWS diversa dalla simulazione: i bucket Amazon S3 per gli artefatti di simulazione devono trovarsi nella stessa posizione della simulazione. Regione AWS Controlla la tua console Amazon S3 per vedere in cosa consiste Regione AWS il tuo bucket. Se il tuo bucket Amazon S3 si trova in un altro bucket Regione AWS, seleziona un bucket che si trovi nello stesso Regione AWS della simulazione.

Problemi legati al tempo di utilizzo Docker

Se stai usando Docker e ricevi errori legati al tempo durante l'esecuzione degli script dall'SDK dell' SimSpace Weaver app, la causa potrebbe essere Docker l'orologio della macchina virtuale non è corretto. Questo può succedere se il computer era in esecuzione Docker e quindi riprende dalla modalità di sospensione o ibernazione.

Soluzioni da provare

- Restart (Riavvia) Docker.
- Disabilita e riattiva la sincronizzazione dell'ora in Windows PowerShell:

```
Get-VMIntegrationService -VMName DockerDesktopVM -Name "Time Synchronization" |
 Disable-VMIntegrationService
Get-VMIntegrationService -VMName DockerDesktopVM -Name "Time Synchronization" |
 Enable-VMIntegrationService
```

PathfindingSample il client della console non riesce a connettersi

Potresti ricevere il seguente errore dal client della console quando ti connetti alla `PathfindingSample` simulazione descritta nei tutorial in. Guida introduttiva con SimSpace Weaver Questo errore si verifica perché il client non può aprire una connessione di rete all'indirizzo IP e `ViewApp` al numero di porta combinati che hai fornito.

```
Fatal error in function nng_dial. Error code: 268435577. Error message: no link
```

Per una simulazione in Cloud AWS

- La tua connessione di rete funziona correttamente? Verifica di poterti connettere ad altri indirizzi IP o siti Web che dovrebbero funzionare. Assicurati che il tuo browser web non stia caricando un sito web dalla sua cache.
- La simulazione è in esecuzione? Puoi utilizzare l'ListSimulationsAPI per conoscere lo stato della simulazione. Per ulteriori informazioni, consulta Ottieni l'indirizzo IP e il numero di porta di un'app personalizzata. Puoi anche usare la SimSpace Weaver console per controllare lo stato delle tue simulazioni.
- Le tue app sono in esecuzione? Puoi utilizzare l'DescribeAppAPI per conoscere lo stato delle tue app. Per ulteriori informazioni, consulta Ottieni l'indirizzo IP e il numero di porta di un'app personalizzata. Puoi anche utilizzare la SimSpace Weaver console per controllare lo stato delle tue simulazioni.
- Le tue app sono in esecuzione? Puoi utilizzare l'DescribeAppAPI per conoscere lo stato delle tue app. Per ulteriori informazioni, consulta Ottieni l'indirizzo IP e il numero di porta di un'app personalizzata. Puoi anche utilizzare la SimSpace Weaver console per controllare lo stato delle tue simulazioni.

- Hai usato l'indirizzo IP e il numero di porta corretti? Quando ti connetti tramite Internet, devi utilizzare l'indirizzo IP e il numero di `Actual` porta di`ViewApp`. Puoi trovare l'IP `Address` e il numero di `Actual` porta nel `EndpointInfo` blocco dell'output dell'DescribeAppAPI. Puoi anche utilizzare la SimSpace Weaver console per trovare l'indirizzo IP (URI) e il numero di porta (porta di ingresso) `ViewApp` nella pagina dei `MyViewDomain` dettagli.
- La tua connessione di rete attraversa un firewall? Il firewall potrebbe bloccare la connessione all'indirizzo IP o al numero di porta (o a entrambi). Controlla le impostazioni del firewall o rivolgiti all'amministratore del firewall.

Per una simulazione locale

- Puoi connetterti al tuo indirizzo di loopback (127.0.0.1)? Se disponi dello strumento da riga di `ping` comando in Windows, puoi aprire una finestra del prompt dei comandi e provare a eseguire il ping di 127.0.0.1. Premi Ctrl - C per terminare il ping.

```
ping 127.0.0.1
```

Example uscita 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 il ping indica la perdita di pacchetti, è possibile che altri software (ad esempio un firewall locale, impostazioni di sicurezza o programmi antimalware) stiano bloccando la connessione.

- Le tue app sono in esecuzione? La simulazione locale viene eseguita come finestre separate per ogni app. Assicurati che le finestre delle tue app spaziali `ViewApp` siano aperte. Per ulteriori informazioni, consulta Sviluppo locale in SimSpace Weaver.
- Hai usato l'indirizzo IP e il numero di porta corretti? È necessario utilizzare `tcp://127.0.0.1:7000` quando ci si connette a una simulazione locale. Per ulteriori informazioni, consulta Sviluppo locale in SimSpace Weaver.
- Disponi di un software di sicurezza locale che potrebbe bloccare la tua connessione? Controlla le impostazioni di sicurezza, il firewall locale o i programmi antimalware per vedere se bloccano la connessione alla porta `127.0.0.1` TCP. `7000`

Il AWS CLI non riconosce **`simspaceweaver`**

Se ti AWS CLI dà errori che suggeriscono che non sia a conoscenza SimSpace Weaver, esegui il comando seguente.

```
aws simspaceweaver help
```

Se ricevi un errore che inizia con le righe seguenti ed elenca tutte le scelte disponibili, è AWS CLI possibile che si tratti di una versione precedente.

```
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:
```

Esegui il comando seguente per verificare la versione del tuo AWS CLI.

```
aws --version
```

Se il numero di versione è precedente alla 2.9.19, devi aggiornare la tua. AWS CLI Tieni presente che la versione corrente di AWS CLI è successiva alla 2.9.19.

Per aggiornare il tuo AWS CLI, consulta Installare o aggiornare la versione più recente di AWS CLI nella Guida per l'AWS Command Line Interface utente della versione 2.

SimSpace Weaver riferimento allo schema di simulazione

SimSpace Weaver utilizza un file YAML per configurare le proprietà di una simulazione. Questo file è chiamato schema di simulazione (o semplicemente schema). La simulazione di esempio inclusa nell'SDK dell' SimSpace Weaver app include uno schema che puoi copiare e modificare per la tua simulazione.

Argomenti

- Esempio di schema completo
- Formato dello schema

Esempio di schema completo

L'esempio seguente mostra YAML-file di testo in formato che descrive una SimSpace Weaver simulazione. Questo esempio include valori fittizi per le proprietà. Il formato del file varia in base al valore `sdk_version` specificato in esso. Formato dello schemaPer una descrizione completa delle proprietà e dei relativi valori validi, vedere.

```
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 dello schema

L'esempio seguente mostra la struttura generale di uno schema. L'ordine delle proprietà a ciascun livello dello schema non ha importanza, purché le relazioni padre-figlio siano le stesse. L'ordine è importante per gli elementi di un array.

```
sdk_version: "sdk-version-number"
simulation_properties:
  simulation-properties
workers:
  worker-group-configurations
clock:
  tick_rate: tick-rate
partitioning_strategies:
  partitioning-strategy-configurations
domains:
  domain-configurations
placement_constraints:
  placement-constraints-configuration
```

Sections

- Versione SDK
- Proprietà di simulazione
- Worker
- Orologio
- Strategie di partizionamento
- Domini
- Vincoli di posizionamento

Versione SDK

La `sdk_version` sezione (obbligatoria) identifica la versione dell'SDK dell' SimSpace Weaver app che supporta questo schema. Valori validi: `1.17`, `1.16`, `1.15`, `1.14`, `1.13`, `1.12`

> **Important**
>
> Il valore di include `sdk_version` solo il numero della versione principale e il numero della prima versione secondaria. Ad esempio, il valore `1.12` specifica tutte le versioni`1.12.x`, ad esempio `1.12.0`1.12.1, e`1.12.2`.

```
sdk_version: "1.17"
```

Proprietà di simulazione

La `simulation_properties` sezione (obbligatoria) specifica varie proprietà della simulazione. Utilizzate questa sezione per configurare la registrazione e specificare un'immagine del contenitore predefinita. Questa sezione è obbligatoria anche se non configuri la registrazione o scegli di specificare un'immagine del contenitore predefinita.

```
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"
```

Proprietà

`log_destination_resource_name`

Specifica la risorsa su cui SimSpace Weaver verranno scritti i log.

Obbligatorio: No. Se questa proprietà non è inclusa, SimSpace Weaver non scriverà i log per la simulazione.

Tipo: Stringa

Valori validi:

- Il nome di un gruppo di log di CloudWatch Logs (ad esempio,) `MySimulationLogs`
- L'Amazon Resource Name (ARN) di un gruppo di log CloudWatch Logs (ad esempio,) `arn:aws:logs:us-west-2:111122223333:log-group/MySimulationLogs`

> **Note**
>
> SimSpace Weaver supporta solo una destinazione di log nello stesso account e Regione AWS nella simulazione.

`log_destination_service`

Indica il tipo di risorsa di destinazione per la registrazione quando si specifica una risorsa `logging_destination resource_name` che non è un ARN.

Obbligatorio: è necessario specificare questa proprietà se `log_destination_resource_name` è specificato e non è un ARN. Non è possibile specificare questa proprietà se `log_destination_resource_name` non è specificata o è un ARN.

Tipo: Stringa

Valori validi:

- `logs`: La risorsa di destinazione del registro è un gruppo di log.

`default_entity_index_key_type`

Speciifica il tipo di dati per il campo chiave dell'indice delle entità di simulazione.

Campo obbligatorio: sì

Tipo: Stringa

Valori validi: `Vector3<f32>`

`default_image`

Specifica l'immagine del contenitore di default per la simulazione (non supportata per la versione e). `1.13` `1.12` Se viene specificata questa proprietà, i domini che non lo specificano `image` utilizzano. `default_image`

Required: No

Tipo: Stringa

Valori validi:

- L'URI di un repository in Amazon Elastic Container Registry (Amazon ECR) (ad esempio,) `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`

Worker

La `workers` sezione (obbligatoria) specifica le configurazioni per i gruppi di lavoro (gruppi di lavoratori). SimSpace Weaver utilizza queste informazioni insieme `placement_constraints` per configurare l'infrastruttura sottostante della simulazione. Attualmente è supportato solo 1 gruppo di lavoro.

Per specificare le proprietà di un gruppo di lavoro, *`worker-group-name`* sostituiscilo con un nome a tua scelta. Il nome deve contenere da 3 a 64 caratteri e può contenere i caratteri A - Z, a - z, 0 - 9 e _ - (trattino). Specificate le proprietà del gruppo di lavoro dopo il nome.

```
workers:
  worker-group-name:
    type: "sim.c5.24xlarge"
    desired: number-of-workers
```

Proprietà

`type`

Speciifica il tipo di lavoratore.

Campo obbligatorio: sì

Tipo: Stringa

Valori validi: `sim.c5.24xlarge`

`desired`

Speciifica il numero di lavoratori desiderato per questo gruppo di lavoratori.

Campo obbligatorio: sì

Tipo: integer

Valori validi: 1 -3. La quota di servizio (limite) per il numero di lavoratori per le simulazioni determina il valore massimo di questa proprietà. Ad esempio, se la quota di servizio è2, il valore massimo per questa proprietà è2. È possibile richiedere un aumento della quota di servizio. Per ulteriori informazioni, consulta SimSpace Endpoint e quote Weaver.

Orologio

La `clock` sezione (obbligatoria) specifica le proprietà dell'orologio di simulazione.

```
clock:
  tick_rate: tick-rate
```

Proprietà

`tick_rate`

Speciifica il numero di segni di spunta al secondo che l'orologio pubblica nelle app.

Campo obbligatorio: sì

Type:

- Versione *1.14* e: stringa `1.15`
- Versione *1.13* e*1.12*: numero intero

Valori validi:

- Versione *1.14* e*1.15*: `"10"` | | `"15"` | `"30"` `"unlimited"`
  - `"unlimited"`: l'orologio invia il segno di spunta successivo non appena tutte le app terminano le operazioni di commit per il segno di spunta corrente.
- Versione *1.13* e*1.12*: | | `10` `15` `30`

Strategie di partizionamento

La `partitioning_strategies` sezione (obbligatoria) specifica l'organizzazione delle partizioni per un dominio spaziale.

> **Note**
>
> SimSpace Weaver supporta solo 1 strategia di partizionamento.

Per specificare le proprietà di una strategia di partizionamento, sostituiscila *partitioning-strategy-name* con un nome a tua scelta. Il nome deve contenere da 3 a 64 caratteri e può contenere i caratteri A - Z, a - z, 0 - 9 e _ - (trattino). Specificare le proprietà della strategia di partizionamento dopo il 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
```

Proprietà

`topology`

Specifica la topologia (schema di disposizione delle partizioni) per questa strategia di partizionamento.

Campo obbligatorio: sì

Tipo: Stringa

Valori validi: `"Grid"`

`aabb_bounds`

Speciifica i limiti del riquadro di delimitazione allineato all'asse principale (AABB) per la tua simulazione. I limiti vengono specificati come matrici ordinate a 2 elementi che descrivono i valori minimo e massimo (in quest'ordine) per ciascun asse (x e y).

Obbligatorio: condizionale. Questa proprietà è obbligatoria (e può essere specificata) solo se la topologia è impostata su. `"Grid"`

Tipo: `Float` array (per ogni asse)

Valori validi: `-3.4028235e38` - `3.4028235e38`

`grid_placement_groups`

Specificate il numero di gruppi di posizionamento lungo ciascun asse (x e y) in una topologia a griglia. Un gruppo di posizionamento è una raccolta di partizioni (nello stesso dominio) spazialmente adiacenti.

Obbligatorio: condizionale. Questa proprietà è obbligatoria (e può essere specificata) solo se la topologia è impostata su. `"Grid"` Se non specificate una configurazione dei gruppi di posizionamento, ne SimSpace Weaver calcolerà una per voi. Qualsiasi dominio che utilizza una strategia di partizionamento senza una configurazione dei gruppi di posizionamento deve specificare un `grid_partition` (vediStrategia di partizionamento del dominio spaziale).

Tipo: numero intero (per ogni asse)

Valori validi: `1` -`20`. Si consiglia che x * y sia uguale al numero di lavoratori desiderato. Altrimenti, SimSpace Weaver cercherai di bilanciare i gruppi di collocamento tra i lavoratori disponibili.

Domini

La `domains` sezione (obbligatoria) specifica le proprietà per ciascuno dei tuoi domini. Tutte le simulazioni devono avere almeno una sezione per un dominio spaziale. È possibile creare più sezioni per domini aggiuntivi. Ogni tipo di dominio ha il proprio formato di configurazione.

> **Important**
>
> Versioni `1.13` e `1.12` non supportano più domini spaziali.

> **Important**
>
> SimSpace Weaver supporta fino a 5 domini per ogni simulazione. Ciò include tutti i domini spaziali, personalizzati e di servizio.

```
domains:
  domain-name:
```

```
    domain-configuration
  domain-name:
    domain-configuration
  ...
```

Configurazione del dominio

- Configurazione del dominio spaziale
- Configurazione personalizzata del dominio
- Configurazione del dominio di servizio

Configurazione del dominio spaziale

Per specificare le proprietà di un dominio spaziale, sostituiscilo *spatial-domain-name* con un nome a tua scelta. Il nome deve contenere da 3 a 64 caratteri e può contenere i caratteri A - Z, a - z, 0 - 9 e _ - (trattino). Specificate le proprietà del dominio spaziale dopo il 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"
```

Strategia di partizionamento del dominio spaziale

La `launch_apps_by_partitioning_strategy` sezione (obbligatoria) specifica la strategia di partizionamento e le dimensioni (in numero di partizioni) dello spazio di simulazione.

```
    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
```

Proprietà

`partitioning_strategy`

Speciifica la strategia di partizionamento per questo dominio spaziale.

Campo obbligatorio: sì

Tipo: Stringa

Valori validi: il valore di questa proprietà deve corrispondere al nome di una strategia di partizionamento definita nella sezione. `partitioning_strategies` Per ulteriori informazioni, consulta Strategie di partizionamento.

`grid_partition`

Speciifica il numero di partizioni lungo ciascun asse (x e y) in una topologia a griglia. Queste dimensioni descrivono lo spazio totale di simulazione per questo dominio.

Obbligatorio: condizionale. Questa proprietà può essere specificata solo se la topologia è impostata su. `"Grid"` Questa proprietà dipende dalla `grid_placement_groups` proprietà della strategia di partizionamento specificata per questo dominio:

- Questa proprietà è obbligatoria se la strategia di partizionamento di questo dominio non specifica una configurazione. `grid_placement_groups`
- Se esiste una `grid_placement_groups` configurazione ma non viene specificata`grid_partition`, SimSpace Weaver utilizzerà le stesse dimensioni della `grid_placment_groups` configurazione.
- Se si specificano entrambe le dimensioni `grid_placement_groups` e`grid_partition`, le dimensioni di `grid_partition` devono essere multipli di quelle di `grid_placement_groups` (ad esempio, se le dimensioni sono 2x2, alcune `grid_placement_groups` dimensioni valide per `grid_partition` sono 2x2, 4x4, 6x6, 8x8, 10x10).

Tipo: numero intero (per ogni asse)

Valori validi: `1 - 20`

Configurazione spaziale dell'app

La `app_config` sezione (obbligatoria) specifica il pacchetto, la configurazione di avvio e i requisiti di risorse per le app in questo dominio.

```
app_config:
  package: "app-package-s3-uri"
  launch_command: ["app-launch-command", "parameter1", ...]
  required_resource_units:
    compute: app-resource-units
```

Proprietà

`package`

Speciifica il pacchetto (file zip) che contiene l'eseguibile/binario dell'app. Il pacchetto deve essere archiviato in un bucket Amazon S3. È supportato solo il formato di file zip.

Campo obbligatorio: sì

Tipo: Stringa

Valori validi: l'URI Amazon S3 del pacchetto in un bucket Amazon S3. Ad esempio `s3://weaver-myproject-111122223333-app-zips-us-west-2/MySpatialApp.zip`.

`launch_command`

Speciifica il nome del file eseguibile/binario e i parametri della riga di comando per avviare l'app. Ogni token di stringa della riga di comando è un elemento dell'array.

Campo obbligatorio: sì

Tipo: array di stringhe

`required_resource_units`

Speciifica il numero di unità di risorse da SimSpace Weaver allocare a ciascuna istanza di questa app. Un'unità di risorsa è una quantità fissa di unità di elaborazione centrali virtuali (vCPUs) e memoria ad accesso casuale (RAM) su un lavoratore. Per ulteriori informazioni sulle unità di risorse, vedereEndpoint e quote di servizio. La `compute` proprietà specifica un'allocazione di unità di risorse per la `compute` famiglia di lavoratori ed è attualmente l'unico tipo di allocazione valido.

Campo obbligatorio: sì

Tipo: integer

Valori validi: - 1 4

Immagine del contenitore personalizzata

La `image` proprietà (opzionale) specifica la posizione di un'immagine del contenitore che SimSpace Weaver viene utilizzata per eseguire app in questo dominio (non supportata nelle versioni `1.13` e`1.12`). Fornisci l'URI a un repository in Amazon Elastic Container Registry (Amazon ECR) che contiene l'immagine. Se questa proprietà non è specificata ma `default_image` è specificata nella `simulation_properties` sezione di primo livello, le app di questo dominio utilizzano la. `default_image` Per ulteriori informazioni, consulta Contenitori personalizzati.

```
    image: "ecr-repository-uri"
```

Proprietà

`image`

Speciifica la posizione di un'immagine del contenitore per eseguire app in questo dominio.

Required: No

Tipo: Stringa

Valori validi:

- L'URI di un repository in Amazon Elastic Container Registry (Amazon ECR) (ad esempio,) `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`

Configurazione personalizzata del dominio

Per specificare le proprietà di un dominio personalizzato, *`custom-domain-name`* sostituiscilo con un nome a tua scelta. Il nome deve contenere da 3 a 64 caratteri e può contenere i caratteri A - Z, a - z, 0 - 9 e _ - (trattino). Specificate le proprietà del dominio personalizzato dopo il nome. Ripeti questa procedura per ogni dominio personalizzato.

```
  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"
```

Proprietà

`launch_apps_via_start_app_call`

Questa proprietà è necessaria per avviare le app personalizzate utilizzando il StartApp API.

Campo obbligatorio: sì

Tipo: N/A

Valori validi: `{}`

Configurazione personalizzata dell'app

`app_config section`(obbligatorio) specifica il pacchetto, la configurazione di avvio, i requisiti di risorse e le porte di rete per le app in questo dominio personalizzato.

```
    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, ...]
```

Proprietà

`package`

Speciifica il pacchetto (file zip) che contiene l'eseguibile/binario dell'app. Il pacchetto deve essere archiviato in un bucket Amazon S3. È supportato solo il formato di file zip.

Campo obbligatorio: sì

Tipo: Stringa

Valori validi: l'URI Amazon S3 del pacchetto in un bucket Amazon S3. Ad esempio `s3://weaver-myproject-111122223333-app-zips-us-west-2/MyCustomApp.zip`.

`launch_command`

Speciifica il nome del file eseguibile/binario e i parametri della riga di comando per avviare l'app. Ogni token di stringa della riga di comando è un elemento dell'array.

Campo obbligatorio: sì

Tipo: array di stringhe

`required_resource_units`

Speciifica il numero di unità di risorse da SimSpace Weaver allocare a ciascuna istanza di questa app. Un'unità di risorsa è una quantità fissa di unità di elaborazione centrali virtuali (vCPUs) e memoria ad accesso casuale (RAM) su un lavoratore. Per ulteriori informazioni sulle unità di risorse, vedereEndpoint e quote di servizio. La `compute` proprietà specifica un'allocazione di unità di risorse per la `compute` famiglia di lavoratori ed è attualmente l'unico tipo di allocazione valido.

Campo obbligatorio: sì

Tipo: integer

Valori validi: - `1` `4`

`endpoint_config`

Speciifica gli endpoint di rete per le app in questo dominio. Il valore di `ingress_ports` specifica le porte a cui si collegano le app personalizzate per le connessioni client in entrata. SimSpace Weaver mappa le porte allocate dinamicamente alle porte di ingresso specificate. Le porte di ingresso sono sia TCP che UDP. Usa il DescribeApp API per trovare il numero di porta effettivo per connettere i tuoi client.

Obbligatorio: No. Se non specifichi la configurazione degli endpoint, le app personalizzate in questo dominio non avranno endpoint di rete.

Tipo: matrice intera

Valori validi: `1024` -`49152`. I valori devono essere univoci.

Immagine del contenitore personalizzata

La `image` proprietà (opzionale) specifica la posizione di un'immagine del contenitore che SimSpace Weaver viene utilizzata per eseguire app in questo dominio (non supportata nelle versioni `1.13`

`e1.12`). Fornisci l'URI a un repository in Amazon Elastic Container Registry (Amazon ECR) che contiene l'immagine. Se questa proprietà non è specificata ma `default_image` è specificata nella `simulation_properties` sezione di primo livello, le app di questo dominio utilizzano la. `default_image` Per ulteriori informazioni, consulta Contenitori personalizzati.

```
    image: "ecr-repository-uri"
```

Proprietà

`image`

Speciifica la posizione di un'immagine del contenitore per eseguire app in questo dominio.

Required: No

Tipo: Stringa

Valori validi:

- L'URI di un repository in Amazon Elastic Container Registry (Amazon ECR) (ad esempio,) `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`

Configurazione del dominio di servizio

Per specificare le proprietà di un dominio di servizio, *`service-domain-name`* sostituiscilo con un nome a tua scelta. Il nome deve contenere da 3 a 64 caratteri e può contenere i caratteri A - Z, a - z, 0 - 9 e _ - (trattino). Specificare le proprietà del dominio del servizio dopo il nome. Ripetere questa procedura per ogni dominio di servizio.

```
  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"
```

Avvia app per lavoratore

La `launch_apps_per_worker` sezione (obbligatoria) indica che si tratta di una configurazione del dominio di servizio e specifica il numero di app di servizio da avviare per lavoratore.

```
launch_apps_per_worker:
  count: number-of-apps-to-launch
```

Proprietà

`count`

Questa proprietà specifica il numero di app di servizio da avviare per lavoratore.

Campo obbligatorio: sì

Tipo: integer

Valori validi: `{}` | 1 |2. Un valore di `{}` specifica il valore predefinito di`1`.

Configurazione dell'app di servizio

La `app_config section` (obbligatoria) specifica il pacchetto, la configurazione di avvio, i requisiti di risorse e le porte di rete per le app in questo dominio di servizio.

```
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, ...]
```

Proprietà

`package`

Speciifica il pacchetto (file zip) che contiene l'eseguibile/binario dell'app. Il pacchetto deve essere archiviato in un bucket Amazon S3. È supportato solo il formato di file zip.

Campo obbligatorio: sì

Tipo: Stringa

Valori validi: l'URI Amazon S3 del pacchetto in un bucket Amazon S3. Ad esempio `s3://weaver-myproject-111122223333-app-zips-us-west-2/MyServiceApp.zip`.

`launch_command`

Speciifica il nome del file eseguibile/binario e i parametri della riga di comando per avviare l'app. Ogni token di stringa della riga di comando è un elemento dell'array.

Campo obbligatorio: sì

Tipo: array di stringhe

`required_resource_units`

Speciifica il numero di unità di risorse da SimSpace Weaver allocare a ciascuna istanza di questa app. Un'unità di risorsa è una quantità fissa di unità di elaborazione centrali virtuali (vCPUs) e memoria ad accesso casuale (RAM) su un lavoratore. Per ulteriori informazioni sulle unità di risorse, vedereEndpoint e quote di servizio. La `compute` proprietà specifica un'allocazione di unità di risorse per la `compute` famiglia di lavoratori ed è attualmente l'unico tipo di allocazione valido.

Campo obbligatorio: sì

Tipo: integer

Valori validi: - `1` `4`

`endpoint_config`

Speciifica gli endpoint di rete per le app in questo dominio. Il valore di `ingress_ports` specifica le porte a cui si collegano le app di servizio per le connessioni client in entrata. SimSpace Weaver mappa le porte allocate dinamicamente alle porte di ingresso specificate. Le porte di ingresso sono sia TCP che UDP. Usa il DescribeApp API per trovare il numero di porta effettivo per connettere i tuoi client.

Obbligatorio: No. Se non specifichi la configurazione degli endpoint, le app di servizio in questo dominio non avranno endpoint di rete.

Tipo: matrice intera

Valori validi: `1024` -`49152`. I valori devono essere univoci.

Immagine del contenitore personalizzata

La `image` proprietà (opzionale) specifica la posizione di un'immagine del contenitore che SimSpace Weaver viene utilizzata per eseguire app in questo dominio (non supportata nelle versioni `1.13` e`1.12`). Fornisci l'URI a un repository in Amazon Elastic Container Registry (Amazon ECR) che contiene l'immagine. Se questa proprietà non è specificata ma `default_image` è specificata nella `simulation_properties` sezione di primo livello, le app di questo dominio utilizzano la. `default_image` Per ulteriori informazioni, consulta Contenitori personalizzati.

```
    image: "ecr-repository-uri"
```

Proprietà

`image`

Speciifica la posizione di un'immagine del contenitore per eseguire app in questo dominio.

Required: No

Tipo: Stringa

Valori validi:

- L'URI di un repository in Amazon Elastic Container Registry (Amazon ECR) (ad esempio,) `111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`

Vincoli di posizionamento

La `placement_constraints` sezione (opzionale) specifica quali domini spaziali SimSpace Weaver devono essere collocati insieme sugli stessi worker. Per ulteriori informazioni, consulta Configurazione dei domini spaziali.

> **Important**
>
> Versioni `1.13` e `1.12` non supportate. `placement_constraints`

```
placement_constraints:
  - placed_together: ["spatial-domain-name", "spatial-domain-name", ...]
```

```
    on_workers: ["worker-group-name"]
```

Proprietà

`placed_together`

Speciifica i domini spaziali da unire SimSpace Weaver .

Campo obbligatorio: sì

Tipo: matrice di stringhe

Valori validi: nomi dei domini spaziali specificati nello schema

`on_workers`

Speciifica il gruppo di lavoro su cui SimSpace Weaver collocare i domini.

Campo obbligatorio: sì

Tipo: matrice di stringhe a 1 elemento

Valori validi: nome di un gruppo di lavoro specificato nello schema

SimSpace Weaver Riferimenti API

SimSpace Weaver dispone di 2 diversi set di interfacce di programmazione delle applicazioni (APIs):

- Servizio APIs: APIs controllano il servizio e le risorse del servizio, come le simulazioni, gli orologi e le app. Fanno parte del kit di sviluppo AWS software principale (SDK) e puoi utilizzare l'interfaccia a riga di AWS comando (CLI) per chiamarli. Per ulteriori informazioni sul servizio APIs, consulta il riferimento all'SimSpace Weaver API.
- App SDK APIs: APIs controllano i dati all'interno della simulazione. Li usi nel codice dell'app per eseguire operazioni come leggere e scrivere i dati dei campi delle entità, utilizzare gli abbonamenti e monitorare gli eventi nella simulazione. Per ulteriori informazioni, consulta la documentazione dell'SDK SimSpace Weaver dell'app nella cartella SDK dell'app: `sdk-folder\SimSpaceWeaverAppSdk\documentation`

> **Note**
>
> `sdk-folder`è la cartella in cui hai decompresso il pacchetto. SimSpaceWeaverAppSdkDistributable

AWS SimSpace Weaver versioni

Miglioriamo continuamente AWS SimSpace Weaver. Devi scaricare l'SDK più recente SimSpace Weaver dell'app quando rilasciamo una nuova versione se desideri sfruttare le nuove funzionalità e gli aggiornamenti delle funzionalità. Per eseguire una simulazione esistente con una versione più recente, potrebbe essere necessario aggiornarne lo schema e il codice, quindi avviare una nuova istanza della simulazione. Non è necessario eseguire l'aggiornamento e puoi continuare a eseguire simulazioni esistenti con le versioni precedenti. Puoi controllare questa pagina per vedere quali sono le differenze tra le versioni. Tutte le versioni sono attualmente supportate.

> **Important**
>
> L'ultima versione della Guida AWS SimSpace Weaver per l'utente copre solo la versione più recente del servizio. È possibile trovare la documentazione per le versioni precedenti nel catalogo delle AWS SimSpace Weaver guide, disponibile nella pagina principale dedicata alla documentazione.

Versione più recente

L'ultima versione è: 1.17.0

Come trovare la tua versione attuale

Se hai creato una simulazione con l'SDK dell' SimSpace Weaver app, lo `create-project` script ha scaricato una versione delle librerie SDK in una sottodirectory del tuo. *`sdk-folder`* La sottodirectory che contiene le librerie SDK ha un nome che include il numero di versione dell'SDK:. `SimSpaceWeaverAppSdk-`*`sdk-version`* Ad esempio, le librerie per la versione 1.16.0 sono disponibili. `SimSpaceWeaverAppSdk-1.16.0`

Puoi anche trovare la versione del pacchetto distribuibile dell' SimSpace Weaver app SDK nel file di testo del tuo. `app_sdk_distributable_version.txt` *`sdk-folder`*

Scarica la versione più recente

Utilizza uno dei seguenti link per scaricare la versione più recente.

- Pacchetto completo distribuibile dell'app SDK
- Solo le librerie SDK dell'app

Puoi anche scaricare il pacchetto completo distribuibile SimSpace Weaver dell'app SDK dalla SimSpace Weaver console in. AWS Management Console Scegli Scarica app SDK dal pannello di navigazione.

> **Warning**
>
> Non utilizzarlo AWS CLI per scaricare nulla che sembri essere il pacchetto distribuibile dell' SimSpace Weaver app SDK. Utilizza solo i link per il download in questa pagina o il link per il download nella console. Qualsiasi altro metodo o percorso di download non è supportato e potrebbe contenere codice obsoleto, errato o dannoso.

Risoluzione dei problemi relativi ai download dell'SDK delle app

Utilizziamo Amazon CloudFront (CloudFront) per distribuire i file.zip dell'SDK dell'app. Potrebbero verificarsi alcune delle seguenti situazioni.

- Il pacchetto scaricato non è la versione più recente
  - Se il file.zip scaricato non contiene la versione più recente, è possibile che la cache dell' CloudFront edge location non sia ancora aggiornata. Effettua nuovamente il download dopo 24 ore.
- Viene visualizzato un errore HTTP 4xx o 5xx utilizzando un link per il download
  - Riprova dopo 24 ore. Se ricevi lo stesso errore, utilizza il link Feedback nella parte inferiore della SimSpace Weaver console per segnalare il problema. Seleziona Segnala un problema come Tipo di feedback.
- Il tuo browser segnala che non è in grado di caricare la pagina
  - Potresti avere un problema di configurazione della rete locale o del browser. Verifica di poter caricare altre pagine. Svuota la cache del browser e riprova. Assicurati di non avere regole firewall che potrebbero bloccare l'URL di download.
- Viene visualizzato un errore quando si tenta di salvare il file
  - Controlla le autorizzazioni del file system locale per assicurarti di disporre delle autorizzazioni corrette per salvare il file.

- Il tuo browser visualizza AccessDenied
  - Se hai inserito manualmente l'URL nel browser, verifica che sia corretto. Se hai usato un link per il download, assicurati che qualcosa non interferisca con l'URL del browser; usa di nuovo il link.

Installa la versione più recente

Per installare la versione più recente

1. Scarica la versione più recente.
2. Decomprimi il SimSpaceWeaverAppSdkDistributable file.zip in una cartella.
3. Esegui `python setup.py` dalla cartella SDK dell'app decompressa dell'ultima versione SimSpace Weaver .
4. Utilizza la cartella SDK dell' SimSpace Weaver app dell'ultima versione decompressa anziché la versione precedente.

Versioni del servizio

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
1.17.0	Principali modifiche al pacchetto distribuibile dell' SimSpace Weaver app SDK • Abbiamo sostituito gli script batch di Windows e Linux Bash con script basati su Python. Python 3.9 è quindi ora necessario per utilizzar	17 aprile 2024	Questa guida	• Pacchetto completo • Solo biblioteche Vedi Risoluzione dei problemi.

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
	e gli script e gli esempi, anche se non si utilizza (o non si intende utilizzare) l'SDK Python. • Questa versione aumenta il supporto per Amazon Linux 2. • Abbiamo corretto diversi bug in SimSpace Weaver Local. Per ulteriori informazioni, consulta le note di rilascio. Correzioni di bug • Abbiamo risolto un bug che faceva sì che le entità diventassero prive di proprietà se non completavano il trasferimento tra lavoratori remoti.			

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
1.16.0	Nuova funzionalità: • Ora puoi utilizzare la messaggistica APIs nell'SDK dell' SimSpace Weaver app per inviare e ricevere messaggi tra le tue app. Questa funzionalità è disponibile per le integrazioni C++, Python e Unity e Unreal Engine 5.	12 febbraio 2024	Consulta il catalogo delle guide.AWS SimSpace Weaver	• Pacchetto completo • Solo biblioteche Vedi Risoluzione dei problemi.

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
1.15.3	SimSpace Weaver Local aggiornamento: • Abbiamo cambiato SimSpace Weaver Local per allinearlo più strettamente allo sviluppo di Cloud AWS. Queste modifiche influiscono sui progetti e sui flussi di lavoro C ++, Python, Unity e Unreal Engine per SimSpace Weaver Local.	4 dicembre 2023	Consulta il catalogo delle guide.AWS SimSpace Weaver	Non disponibile per il download

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
1.15.2	Aggiornamento del pacchetto distribuibile dell'App SDK: • L'abbiamo aggiornato `Dockerfile` per utilizzare la versione specifica richiesta di cmake. Le compilazioni dei container Docker potrebbero fallire senza questa modifica.	2 novembre 2023	Consulta il AWS SimSpace Weaver catalogo delle guide.	Non disponibile per il download
1.15.1	Aggiornamento delle funzionalità: • Python SDK: questa versione corregge un problema che causava il fallimento delle simulazioni basate su Python in. Cloud AWS	22 settembre 2023	AWS SimSpace Weaver Consulta il catalogo delle guide.	Non disponibile per il download

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
1.15.0	Nuova funzionalità: • Python SDK: ora puoi sviluppare le tue simulazioni in Python. Il pacchetto distribuibile SimSpace Weaver dell'app SDK include un modello per un progetto Python di esempio e il relativo client di visualizzazione Python.	31 agosto 2023	Consulta il catalogo delle guide.AWS SimSpace Weaver	Non disponibile per il download

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
1.14.0	Nuove caratteristiche: • Contenitori personalizzati: crea la tua immagine di contenitore basata su Amazon Linux 2 (AL2), memorizzala in Amazon Elastic Container Registry (Amazon ECR) e usala per eseguire le SimSpace Weaver tue app in. Cloud AWS • Domini spaziali multipli: crea più di un dominio spaziale in una simulazione. Separa la logica di simulazione invece di combinarla tutta in un'unica app spaziale. Assegna risorse diverse ai domini	26 luglio 2023	Consulta il AWS SimSpace Weaver catalogo delle guide.	Non disponibile per il download

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
	spaziali in base alle loro esigenze. • Tick rate illimitato: consenti alla simulazione di funzionare alla stessa velocità di esecuzione del codice. Imposta l'orologio della simulazione in modo che invii il segno di spunta successivo non appena tutte le app terminano le operazioni di commit per il segno di spunta corrente. SimSpace Weaver SDK per app: • Il valore di `tick_rate` è ora una stringa. Il valore deve includere virgolette doppie (''). Il tick rate per le versioni precedenti			

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
	è ancora un numero intero.			
1.13.1	SimSpace Weaver SDK dell'app: • Aggiornamento delle funzionalità: la creazione del progetto ora funziona correttamente con il `PathfindingSampleUnreal` modello.	7 giugno 2023	Consulta il catalogo della AWS SimSpace Weaver guida.	Non disponibile per il download

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
1.13.0	SimSpace Weaver servizio APIs: • • Nuova `CreateSnapshot` azione • Modifiche all'`StartSimulation` azione: • Aggiunto un `SnapshotS3Location` parametro per iniziare da un'istantanea. • Il `SchemaS3Location` parametro è ora facoltativo. • Modifiche all'`DescribeSimulation` output: • `SchemaError` è obsoleta. • È stato aggiunto un	29 aprile 2023	Consulta il catalogo della guida.AWS SimSpace Weaver	Non disponibile per il download

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
	`StartError` campo. • È stato aggiunto un `SnapshotS3Location` campo. • È stato aggiunto uno stato di `SNAPSHOT_IN_PROGRESS` simulazione. • Nuovo tipo di `S3Destination` dati SimSpace Weaver console: • Nuova funzionalità per creare istantanee. • Nuova funzionalità per avviare una simulazione da un'istantanea.			

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app

	SimSpace Weaver SDK per app:
	• Nuovi script per supportare le istantanee
	  • `create-snapshot-` *proje*
	  name `.bat`
	  • `start-from-snapshot-` *project-*
	  name `.bat`
	  • `quick-start-from-snapshot-` *pro*
	  name `-cli.bat`
	  • `list-snapshots-` *proje*
	  name `.bat`
	• I progetti ora utilizzano un singolo bucket Amazon S3 per progetto: weaver- - - *lowercase -project- name account-number region*			

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
1.12.3	SimSpace Weaver SDK dell'app: • I seguenti script ora supportano il `--maximum-duration` parametro: • `quick-start-`*`project-name`*`-cli.bat` • `quick-start-`*`project-name`*`-cli.sh` • `start-simulation-`*`project-name`*`.bat` • `start-simulation-`*`project-name`*`.sh` • `run-`*`project-name`*`.bat` • `run-`*`project-name`*`.sh`	27 marzo 2023	Consultate il catalogo delle AWS SimSpace Weaver guide.	Non disponibile per il download

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
1.12.2	SimSpace Weaver SDK dell'app: • Correzione di bug: `docker-create-image.bat` ora funziona correttamente.	1 marzo 2023	Consulta il catalogo delle AWS SimSpace Weaver guide.	Non disponibile per il download

Versione	Note	Data di rilascio	Documentazione	Scarica l'SDK dell'app
1.12.1	SimSpace Weaver SDK dell'app: • Gli script ora accettano un AWS CLI profilo da utilizzare per AWS l'autenti cazione. • Gli script ora supportano AWS IAM Identity Center l'autenti cazione. AWS SimSpace Weaver Local: • Correzione di bug: `Api::BeginUpdateWillBlock` ora restituisce correttamente `true` se tutte le app spaziali non hanno aderito alla simulazione.	28 febbraio 2023	Consulta il AWS SimSpace Weaver catalogo delle guide.	Non disponibile per il download
1.12.0	Versione per disponibilità generale (GA)	29 novembre 2022	Consulta il catalogo delle AWS SimSpace Weaver guide.	Non disponibile per il download

AWS SimSpace Weaver versione 1.17.0

Questa versione è una revisione del pacchetto distribuibile dell' SimSpace Weaver app SDK. Abbiamo sostituito gli script batch di Windows e Linux Bash obsoleti con script basati su Python.

> **Important**
>
> Python 3.9 è ora un requisito per utilizzare gli script e gli esempi, non solo per l'SDK Python.

Indice

- Principali modifiche per 1.17.0
- Aggiorna un progetto alla versione 1.17.0
- Domande frequenti sulla versione 1.17.0

Principali modifiche per 1.17.0

- Creazione semplificata del progetto
  - Dopo l'esecuzione`setup.py`, puoi creare il tuo progetto semplicemente copiando e incollando un esempio.
- Esempi con 1 clic
  - Il file zip di distribuzione ora contiene ready-to-use esempi che funzionano dopo aver impostato la distribuzione.
- Ogni SDK ora esiste nella propria directory:`cpp`, `pythonunreal`, `eunity`. Potrebbe essere necessario aggiornare i percorsi a seconda dell'SDK utilizzato.
- Miglioramenti agli script di supporto.
  - Gli script ora contengono più AWS CLI opzioni per massimizzarne la flessibilità.
  - Avvio e connessione del client di console integrati come parte del processo di avvio rapido.
  - Output da console migliorato.
  - La creazione di esempi di Unreal e Unity ora funziona con`quick-start`, non sono più necessari passaggi manuali.
  - SimSpace Weaver Local ora funziona semplicemente chiamando`quick-start`, non è più necessario creare e avviare manualmente.

  - SimSpace Weaver Local `quick-startha` un supporto integrato per la registrazione dell'output dell'app.
  - SimSpace Weaver Local ora può essere avviato in un ambiente non GUI, ad esempio in una sessione ssh.
  - La funzionalità «contenitore personalizzato» è ora integrata nello script. `quick-start`
  - Maggiore supporto per Amazon Linux 2 (AL2): flussi di lavoro di script per Windows e ora AL2 sono comparabili. In precedenza, AL2 i progetti richiedevano più passaggi manuali e SimSpace Weaver Local non era supportato per AL2.
- I plugin Unreal Engine e Unity sono ora inclusi come parte del pacchetto distribuibile SDK dell' SimSpace Weaver app.
- Correzioni di bug per SimSpace Weaver Local
  - Risolto un bug per cui alle entità poteva essere assegnato lo stesso ID di entità.
  - Risolto un bug per cui a due partizioni poteva essere assegnato lo stesso ID di partizione.
  - È stato corretto un bug relativo al tentativo delle app di scrivere su entità di cui non erano proprietarie.
  - Risolto un problema di perdita di memoria.

Aggiorna un progetto alla versione 1.17.0

1. Configura la distribuzione 1.17.0: Ripeti le procedure di configurazione perché le abbiamo modificate per 1.17.0. Per ulteriori informazioni, consulta Configurazione per SimSpace Weaver.
2. Ogni Weaver App SDK ora esiste nella propria directory. Aggiorna i tuoi percorsi di costruzione per rispecchiare questo aspetto.

   a. Directory C++: `SimSpaceWeaverAppSdk/cpp`

      - L'SDK dell'app SimSpace Weaver C++ ora utilizza un file. `FindSimSpaceWeaverAppSdk.cmake` Questo file imposta una `weaver` destinazione a cui viene collegato e include importanti correzioni di bug durante la creazione di Weaver in. Cloud AWS Dovresti usarlo invece di collegarti direttamente ai binari.

   b. Cartella Python: `SimSpaceWeaverAppSdk/python`

   c. Plugin Unity: `SimSpaceWeaverAppSdk/unity`

   d. Plugin Unreal Engine: `SimSpaceWeaverAppSdk/unreal`

3. Gli `tools` script precedenti non funzioneranno con la nuova SimSpace Weaver distribuzione. Per utilizzare i nuovi `tools` script con il tuo progetto:

   a. Elimina le tue vecchie `tools/windows` `tools/linux` `tools/local` cartelle e.

   b. Copia la `tools` directory di un progetto di esempio che utilizza lo stesso SDK SimSpace Weaver dell'app del progetto. Assicurati di averlo eseguito `setup.py` prima di copiare questa directory.

> **Important**
>
> È garantito che gli script degli strumenti funzionino solo con i progetti di esempio. Potrebbe essere necessario modificare questi script, in particolare lo `build.py` script, per lavorare con il progetto. Qualsiasi modifica sarà specifica per il tuo progetto, quindi non possiamo fornire alcuna guida.

Domande frequenti sulla versione 1.17.0

Devo effettuare l'aggiornamento alla versione 1.17.0?

Questo non è un aggiornamento richiesto perché non ci sono modifiche all' SimSpace Weaver API o all'SDK dell' SimSpace Weaver app. È necessario eseguire l'aggiornamento alla versione 1.17.0 se si desidera utilizzare la versione 1.17.0 SimSpace Weaver Local, che contiene diverse correzioni di bug.

Qual è la versione minima di Python richiesta?

Python 3.9 è la versione minima.

Qual è la CMake versione minima richiesta?

CMake la versione 3.13 è la versione minima.

Qual è la versione minima di Unreal Engine richiesta?

Unreal Engine 5.0 è il minimo.

Qual è la versione minima di Unity richiesta?

La versione minima di Unity 2022.3.19.F1 è la versione minima.

AWS SimSpace Weaver versione 1.15.1

Questa versione è un aggiornamento richiesto per Python SDK originariamente rilasciato nella SimSpace Weaver versione 1.15.0. Risolve un problema di mancata corrispondenza delle versioni che causava il fallimento delle simulazioni basate su Python in. Cloud AWS Usa questa versione invece della 1.15.0.

Aggiorna un progetto Python esistente a 1.15.1

Se disponi di un progetto Python esistente creato con la versione 1.15.0 Python SDK, devi eseguire i seguenti passaggi per aggiornarlo alla 1.15.1 in modo che possa essere eseguito in. Cloud AWS

Invece di seguire questa procedura, puoi anche creare un nuovo progetto Python con l'SDK Python 1.15.1 e spostare il codice personalizzato nel nuovo progetto.

Per aggiornare un progetto Python 1.15.0 a 1.15.1

1. Vai alla cartella del tuo progetto Python.
2. In `src/PythonBubblesSample/bin/run-python` modifica la seguente riga:

   ```
   export PYTHONPATH=$PYTHONPATH:/roapp/lib
   ```

   alla seguente:

   ```
   export PYTHONPATH=$PYTHONPATH:$LD_LIBRARY_PATH:/roapp/lib
   ```

3. In `CMakeLists.txt` eliminare le seguenti righe:

   - ```
     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)
     ```

Risoluzione dei problemi per la versione 1.15.1

Dopo aver aggiornato una simulazione Python 1.15.0, non riesce ad avviarsi in Cloud AWS

Sintomi: dopo circa 5-10 minuti dall'avvio della simulazione, il registro di gestione della simulazione riporta un messaggio e lo stato della simulazione è. `internal error` `FAILED`

Ciò può accadere se un file di libreria dell'SDK Python 1.15.0 è incluso in un file zip dell'app. Assicurati di aver completato i passaggi per aggiornare il progetto e assicurati che non sia presente nei file zip o che `libweaver_app_sdk_python_v1.so` non vi sia alcun riferimento.

Domande frequenti sulla versione 1.15.1

Questa versione influisce su qualcosa di diverso dall'SDK Python?

No.

Devo aggiornare alla versione 1.15.1?

Non è necessario eseguire l'aggiornamento alla versione 1.15.1 se non si intende utilizzare Python per le app spaziali. Se hai eseguito l'aggiornamento alla 1.15.0, le simulazioni basate su Python non verranno eseguite in. Cloud AWS Ti consigliamo di eseguire l'aggiornamento alla versione 1.15.1 se usi 1.15.0.

Cos'è **`$LD_LIBRARY_PATH`**?

È la posizione dell'SDK Python quando la simulazione viene eseguita in. Cloud AWS Questa è una novità per 1.15.1. Abbiamo apportato questa modifica per evitare problemi futuri con la versione di Python. Il collegamento a quella directory è funzionalmente lo stesso del collegamento alla versione 1.15.0. `libweaver_app_sdk_python_v1.so`

Cronologia dei documenti per AWS SimSpace Weaver

La tabella seguente descrive importanti modifiche alla SimSpace Weaver documentazione.

Data	Modifica	Aggiornamenti della documentazione	Versioni API aggiornate
20 maggio 2025	Avviso di fine del supporto	Avviso di fine del supporto: il 20 maggio 2026, AWS terminerà il supporto per AWS SimSpace Weaver. Dopo il 20 maggio 2026, non potrai più accedere alla SimSpace Weaver console o SimSpace Weaver alle risorse. Per ulteriori informazi oni, consulta AWS SimSpace Weaver Fine del supporto.	N/D
17 aprile 2024	Contenuti aggiornati	Aggiornamento integrale della guida per l'utente per la versione 1.17.0. Principali modifiche al Configurazione capitolo e Nozioni di base ai tutorial. Per ulteriori informazi oni, consulta le note di rilascio.	N/D
12 febbraio 2024	Contenuti aggiornati	Aggiornato il AWS SimSpace Weaver versioni capitolo per la versione 1.16.0.	N/D

Data	Modifica	Aggiornamenti della documentazione	Versioni API aggiornate
12 febbraio 2024	Nuovo contenuto	È stata aggiunta la Messaggistica sezione come parte della versione 1.16.0. Questa sezione descrive la messaggistica APIs aggiunta all'SDK dell' SimSpace Weaver app. Puoi usarli APIs per inviare e ricevere messaggi tra le tue app.	N/D
12 febbraio 2024	Contenuti aggiornati	Aggiornato il SimSpace Weaver riferimento allo schema di simulazione capitolo per la versione 1.16.0.	N/D
12 febbraio 2024	Contenuti aggiornati	Sono state aggiunte le quote di servizio per la messaggistica al capitolo. SimSpace Endpoint e quote Weaver	N/D
12 febbraio 2024	Nuove guide	Suddividi il contenuto delle versioni precedent i alla 1.16.0 in una guida separata. È stato aggiunto il catalogo delle AWS SimSpace Weaver guide (disponibile nella pagina principale della documentazione) per accedere alle guide delle versioni precedenti.	N/D

Data	Modifica	Aggiornamenti della documentazione	Versioni API aggiornate
4 dicembre 2023	Contenuti aggiornati	È stato aggiornato il AWS SimSpace Weaver versioni capitolo per la versione 1.15.3.	N/D
4 dicembre 2023	Contenuti aggiornati	È stato aggiornato il AWS SimSpace Weaver versioni capitolo per includere le istruzion i di installazione per la versione più recente.	N/D
4 dicembre 2023	Contenuti aggiornati	Aggiornato ilQuote di servizio per SimSpace Weaver Local.	N/D
4 dicembre 2023	Contenuti nuovi e aggiornati	Ha ristrutturato la Sviluppo locale in SimSpace Weaver sezione e ha aggiunto una nuova pagina che descrive le differenze SimSpace Weaver Local introdotte nella versione 1.15.3.	N/D

Data	Modifica	Aggiornamenti della documentazione	Versioni API aggiornate
7 novembre 2023	Contenuti aggiornati	Sono state aggiornate le istruzioni per configura re Docker e WSL per utilizzare il link/URL di download diretto per l'SDK dell'app. Per ulteriori informazioni, consulta Configura il tuo ambiente locale per SimSpace Weaver.	N/D
2 novembre 2023	Contenuti aggiornati	È stata aggiornata la pagina delle versioni del servizio per la versione 1.15.2. Per ulteriori informazioni, consulta Versioni del servizio.	N/D
23 ottobre 2023	Contenuti aggiornati	È stata aggiornata la pagina delle versioni del servizio con nuove istruzioni per scaricare il pacchetto distribuibile SDK dell'app. I clienti devono ora utilizzare solo uno dei nostri link per il download diretto approvati e non utilizzar lo AWS CLI per scaricare il pacchetto distribuibile SDK dell'app. Per ulteriori informazioni, consulta Scarica la versione più recente.	N/D

Data	Modifica	Aggiornamenti della documentazione	Versioni API aggiornate
22 settembre 2023	Nuovo contenuto	È stata aggiunta una pagina di note sulla versione con le istruzioni di aggiornamento per la versione 1.15.1. Per ulteriori informazioni, consulta AWS SimSpace Weaver versione 1.15.1.	N/D
10 settembre 2023	Nuovo contenuto	È stata aggiunta una sezione di risoluzione dei problemi per le situazioni in cui AWS CLI non riconosce SimSpace Weaver. Per ulteriori informazioni, consulta Il AWS CLI non riconosce `simspaceweaver` .	N/D
10 settembre 2023	Contenuti aggiornati	Istruzioni di installazione aggiornate per il AWS CLI WSL. Per ulteriori informazioni, consulta Configurare il pacchetto SimSpace Weaver di distribuzione per Amazon Linux 2 (AL2) in Windows Subsystem for Linux (WSL).	N/D

Data	Modifica	Aggiornamenti della documentazione	Versioni API aggiornate
7 settembre 2023	Aggiornamento dell'API	BucketName e ora ObjectKey sono necessari per il tipo di dati S3Location. BucketNam e è ora necessario per il tipo di dati S3Destination.	AWS SDK: 2023-09-07
31 agosto 2023	Nuovo contenuto	Aggiunta una nuova sezione per la versione 1.15.0:. Uso di Python	N/D
15 agosto 2023	Contenuti aggiornati	Le istruzioni di download sono state aggiornate AWS SimSpace Weaver versioni per elencare solo i SimSpace Weaver bucket Amazon S3 ufficiali. Altre posizioni di download non sono controllate da AWS e potrebbero contenere codice dannoso.	N/D
26 luglio 2023	Contenuti aggiornati	Aggiornato Orologio.	N/D
26 luglio 2023	Contenuti aggiornati	Aggiornato Configura zione dei domini spaziali.	N/D
26 luglio 2023	Nuovo contenuto	Aggiunta di una nuova sezione, Contenitori personalizzati,	N/D

Data	Modifica	Aggiornamenti della documentazione	Versioni API aggiornate
26 luglio 2023	Contenuti aggiornati	Aggiornato AWS SimSpace Weaver versioni per la versione 1.14.0.	N/D
6 luglio 2023	Nuovo contenuto	Aggiunta di una nuova sezione, Pathfindi ngSample il client della console non riesce a connettersi,	N/D
7 giugno 2023	Contenuti aggiornati	Aggiornato AWS SimSpace Weaver versioni per la versione 1.13.1.	N/D
15 maggio 2023	Nuovo contenuto	Aggiunta di una nuova sezione, Utilizzo di istantanee con AWS CloudFormation,	N/D
29 aprile 2023	Nuovo contenuto	Aggiunto contenuto per la versione 1.13.0. Per ulteriori informazioni, consulta AWS SimSpace Weaver versioni.	AWS SDK: 2023-04-28

Data	Modifica	Aggiornamenti della documentazione	Versioni API aggiornate
27 marzo 2023	Nuovo contenuto	Aggiunta una sezione che descrive la durata massima delle simulazioni. Sono state aggiunte delle note nei tutorial per la versione 1.12.3, che hanno aggiunto il supporto per il `--maximum-duration` parametro agli script SDK delle app. SimSpace Weaver	N/D
9 marzo 2023	Contenuti modificati	Abbiamo chiarito che forniamo solo istruzioni per l'attivazione Windows e per Docker Windows Subsystem for Linux (WSL) e che WSL (e qualsiasi altro ambiente Linux) non è supportato.	N/D
28 febbraio 2023	Nuovo contenuto	È stato aggiunto un capitolo che descrive le versioni. SimSpace Weaver	N/D
28 febbraio 2023	Contenuti modificati	È stato modificato il contenuto sull'autenticazione per includere l'uso di AWS IAM Identity Center profili denominati per AWS Command Line Interface (AWS CLI).	N/D

Data	Modifica	Aggiornamenti della documentazione	Versioni API aggiornate
17 febbraio 2023	Nuovo contenuto	È stata aggiunta una sezione sulla gestione delle risorse con AWS CloudFormation.	N/D
23 gennaio 2023	Nuovo contenuto	Sono state aggiunte istruzioni per il debug delle simulazioni locali.	N/D
29 novembre 2022	Avvio del servizio	Rilasciati la Guida per l'utente e il riferimen to API per. SimSpace Weaver	AWS SDK: 2022-11-29

Glossario

Questo glossario definisce i termini specifici di AWS SimSpace Weaver.

Per la AWS terminologia più recente, consulta il AWS glossario nella AWS Guida generale.

A

app — Codice eseguibile (chiamato anche binari) creato dall'utente. Il termine app può fare riferimento al codice o a un'istanza in esecuzione di tale codice. Un'app incapsula il comportamento di simulazione. Le app creano, eliminano, leggono e aggiornano le entità.

SDK per app — Un kit di sviluppo software (SDK) con cui integrare un'app. SimSpace Weaver L'SDK consente la lettura e APIs la scrittura dei dati delle entità e il monitoraggio del tempo di simulazione. Per ulteriori informazioni, consulta SimSpace Weaver app SDK.

C

cliente — Processi (o relative definizioni) che esistono al di fuori della simulazione SimSpace Weaver e che interagiscono con essa tramite un'app personalizzata o un'app di servizio. È possibile utilizzare un client per visualizzare o modificare lo stato della simulazione.

orologio — Un'astrazione dei processi SimSpace Weaver di pianificazione interni. L'orologio pubblica segni di spunta nelle app per mantenere la sincronizzazione dell'ora. Ogni simulazione ha il suo orologio.

frequenza — Il numero di tick al secondo che l'orologio pubblica nelle app. Per ulteriori informazioni sulle frequenze di clock supportate, consulta. SimSpace Endpoint e quote Weaver

frequenza di ticchettamento — Vedi la frequenza dell'orologio.

unità di risorse di calcolo — Un'unità di risorse di calcolo (processore e memoria) su un lavoratore. A una singola istanza di un'app viene normalmente assegnata 1 unità di

	risorse di calcolo. Puoi allocare più di 1 unità di risorse di calcolo per ogni app.
app personalizzata	Un tipo di app che usi per leggere e interagire con lo stato della simulazione. Le app personalizzate possono creare entità nella simulazione ma non le possiedono. Quando un'app personalizzata crea un'entità, deve trasferire l'entità nel dominio spaziale. Puoi controllare il ciclo di vita di un'app personalizzata utilizzando l'app. APIs Per ulteriori informazioni su SimSpace Weaver APIs, vedere. SimSpace Weaver Riferimenti API
dominio personalizzato	Un dominio che contiene app personalizzate.
partizione personalizzata	La partizione di un'app personalizzata.

D

scadenza	Un momento effettivo entro il quale un'operazione (ad esempio l'elaborazione di un segno di spunta) deve essere completata.
dominio	Un gruppo di istanze di app che eseguono lo stesso codice eseguibile (app binario) e hanno le stesse opzioni di avvio.

E

endpoint (servizio)	Un nome di dominio completo (FQDN) utilizzato da programmi (come il AWS Command Line Interface) per connettersi al servizio. SimSpace Weaver
endpoint (simulazione)	Un indirizzo IP e un numero di porta utilizzati dai client per connettersi a una simulazione. Puoi configurare gli endpoint su app personalizzate e app di servizio.
entità	oggetti di dati del cliente (o relative definizioni). Le entità possono essere statiche (rimanere in una posizione) o dinamiche (spostarsi nello spazio di simulazione). Ad esempio, persone ed edifici in una simulazione.

I

indice (simulazione)	Descrizione delle proprietà spaziali di una simulazione, compresi i confini spaziali e il sistema di coordinate.

L

ciclo di vita (di un'app)	Una descrizione dei passaggi logici previsti che un'app deve seguire durante una simulazione. I cicli di vita sono gestiti (SimSpace Weaver avvia e arresta l'app) o non gestiti (l'utente avvia e arresta l'app).
load (dati del campo dell'entità)	Leggi i dati del campo di entità da State Fabric.

P

partizione	Un segmento di memoria condivisa su un lavoratore. Ogni partizione contiene un sottoinsieme discreto di entità all'interno di un dominio. A ogni app è assegnata una partizione. Un'app possiede tutte le entità nella sua partizione. Quando un'app crea un'entità, la crea nella sua partizione. Quando le entità si spostano da una partizione all'altra, la proprietà viene trasferita dall'app della partizione di origine all'app della partizione di destinazione.

R

unità di risorse	Consultare ???.

S

schema	Un documento YAML o JSON che descrive la configurazione di una simulazione. SimSpace Weaver utilizza uno schema per creare una risorsa di simulazione.
app di servizio	Un tipo di app che si utilizza per leggere e interagire con lo stato della simulazione. Le app di servizio possono creare entità nella simulazione ma devono trasferirle nel dominio spaziale. SimSpace Weaver gestisce il

	ciclo di vita di un'app di servizio e ne avvia 1 (o più, come specificato nello schema di simulazione) su ogni lavoratore della simulazione.
dominio di servizio	Un dominio che contiene app di servizio.
partizione di servizio	La partizione di un'app di servizio.
simulazione (risorsa)	Un'astrazione di un cluster di calcolo che gestisce uno spazio virtuale simulato. Puoi avere più simulazioni. Si configura una simulazione utilizzando uno schema.
app spaziale	Un tipo di app che incapsula la logica di simulazione di base. Ogni app spaziale possiede 1 (e solo 1) partizione.
dominio spaziale	Un dominio che contiene app spaziali.
partizione spaziale	La partizione di un'app spaziale.
State Fabric	SimSpace Weaverè un database in memoria. State FabricMemorizza lo stato delle simulazioni, incluse entità e dati interni SimSpace Weaver .
store (dati del campo dell'entità)	Scrivi i dati del campo dell'entità in State Fabric.
abbonamento	Una richiesta a lungo termine per una specifica istanza di app per ricevere dati da un'area di abbonamento. L'app per gli abbonamenti utilizza un abbonamento per scoprire le modifiche alle entità all'interno dell'area di abbonamento.
area di sottoscrizione	Una regione bidimensionale dello spazio di simulazione. Un abbonamento si riferisce a un'area di abbonamento. Un'area di sottoscrizione può estendersi su più di una partizione e includere anche parti di partizioni. Un'area di sottoscrizione è continua entro i limiti definiti.

T

segno di spunta	Un valore temporale discreto (orario dell'orologio da parete o tempo di simulazione). Le app possono iterare più velocemente della durata del segno di spunta, ma ci si aspetta che scrivano tick specifici entro scadenze specifiche. Tutte le operazioni per tutte le app relative a un

	determinato segno di spunta devono essere completate prima che possa iniziare il segno di spunta successivo.
tasso di selezione	Vedi la frequenza.
ora (effettiva)	L'ora attuale dal punto di vista della realtà. SimSpace Weaver utilizza un timestamp POSIX a 64 bit che è il numero di nanosecondi dall'epoca Unix (January 1, 1970, 00:00:00 UTC).
tempo (simulazione)	L'ora corrente dal punto di vista della simulazione. SimSpace Weaver utilizza un contatore di tacche logico intero a 64 bit, che potrebbe non corrispondere direttamente all'ora effettiva.

W

lavoratore	Un'istanza Amazon Elastic Compute Cloud (Amazon EC2) che esegue codice di simulazione.