



1.16.0 版的使用者指南

AWS SimSpace Weaver



AWS SimSpace Weaver: 1.16.0 版的使用者指南

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

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能隸屬於 Amazon，或與 Amazon 有合作關係，或由 Amazon 贊助。

Table of Contents

什麼是 SimSpace Weaver ?	1
重要概念	1
SimSpace Weaver 的運作方式	2
您如何使用 SimSpace Weaver	5
模擬綱要	5
工作者與資源單位	5
模擬時鐘	6
資料分割	6
國家面料	6
實體	7
應用程式	7
範例 使用案例	9
設定	11
設定您的帳戶	11
註冊 AWS 帳戶	11
建立管理使用者	11
新增要使用的權限 SimSpace Weaver	12
設定您的本機環境	14
AL2 輸入 Docker	14
AL2 輸入 WSL	15
使用授權軟體	19
入門	20
Quick Start	20
步驟 1：建立專案	21
步驟 2：開啟記錄	24
步驟 3：執行快速啟動指令碼	25
步驟 4：獲取您的 IP 地址和端口號	28
步驟 5：檢視您的模擬	35
步驟 6：停止並清理您的模擬	41
詳教學課程	47
步驟 1：建立專案	47
步驟 2：開啟記錄	51
步驟 3：上傳您的模擬結構描述	52
步驟 4：建立您的專案	54

步驟 5：上傳應用程式	56
步驟 6：開始模擬	58
步驟 7：取得模擬詳細資料	62
步驟 8：啟動自訂應用程式	68
步驟 9：啟動時鐘	72
步驟 10：檢查記錄	74
步驟 11：檢視您的模擬	76
步驟 12：停止並清理您的模擬	83
使用 SimSpace Weaver	90
規劃您的模擬	90
模擬規劃參數	92
SDK 版本	93
模擬性質	93
工人	93
時鐘	94
分割策略	97
網域	98
持續時間上限	107
最大值	107
預設值	107
最小值	107
使用 SimSpace Weaver 應用程式 SDK 指令碼啟動模擬	108
使用控制台啟動模擬	108
達到其最長持續時間的模擬狀態	108
開發應用程式	108
空間應用	109
自定義應用	110
開發客戶端應用	111
本機開發	112
組建	113
執行	114
檢視	115
停止	116
偵錯	116
在 1.15.3 中的差異	117
SimSpace Weaver 應用程式 SDK	122

API 方法返回一個 Result	123
在頂層與應用程式 SDK 互動	124
模擬管理	124
訂閱	127
實體	128
實體事件	140
Result和錯誤處理	146
泛型和域類型	148
其他應用程式 SDK 作業	148
SimSpace Weaver演示框架	151
使用	152
取得應用程式的限制	152
獲取應用程序使用的資源量	153
重設指標	154
超過	154
記憶體	155
最佳實務	155
除錯模擬	155
使用SimSpace Weaver Local並查看控制台輸出	156
在 Amazon 日誌中查看您的 CloudWatch 日誌	156
使用 describe API 呼叫	156
Connect 用戶端	157
偵錯本機模擬	157
自訂容器	158
建立自訂容器	159
修改專案以使用自訂容器	159
常見問答集	162
故障診斷	163
使用 Python	163
創建一個項目	164
開始模擬	166
示例 Python 客戶端	167
編寫您自己的構建腳本	168
常見問答集	168
疑難排解	168
Support 其他引擎	170

Unity	170
Unreal Engine	172
使用授權軟體	172
管理資源 AWS CloudFormation	172
快照	174
快照	175
快照使用案例	175
SimSpace Weaver應用程式 SDK	176
SimSpace Weaver 主控台	180
AWS CLI	182
SimSpace Weaver API	185
常見問答集	185
簡訊	186
訊息傳送的使用案例	186
使用訊息傳遞 API	187
何時使用訊息	194
使用消息傳遞時的提示	198
訊息錯誤和疑難排解	199
最佳實務	202
設定帳單警示	202
使用 SimSpace Weaver Local	202
停止不需要的模擬	203
刪除不需要的資源	203
有備份	203
安全	204
資料保護	204
靜態加密	205
傳輸中加密	206
網際網路流量隱私權	206
身分和存取權管理	206
物件	207
使用身分驗證	207
使用政策管理存取權	210
AWS SimSpace Weaver 搭配 IAM 的運作方式	212
身分型政策範例	218
為您建SimSpace Weaver立的權限	222

預防跨服務混淆代理人	223
疑難排解	226
安全性事件記錄與監控	228
合規驗證	229
恢復能力	230
基礎設施安全	230
網路連線安全性模型	230
組態與漏洞分析	231
安全最佳實務	231
加密應用程式與其用戶端之間的通訊	232
定期備份模擬狀態	232
維護您的應用程式和 SDK	232
日誌記錄和監控	233
登入 CloudWatch	233
存取您的SimSpace Weaver記錄	233
SimSpace Weaver 日誌	234
使用監控 CloudWatch	237
SimSpace Weaver帳戶層級的指標	237
CloudTrail 日誌	237
SimSpace Weaver中的資訊 CloudTrail	238
了解 SimSpace Weaver 日誌檔案項目	238
端點和服務配額	241
服務端點	241
Service Quotas	242
郵件配額	244
時脈速率	244
的服務配額 SimSpace Weaver Local	245
疑難排解	246
AssumeRoleAccessDenied	246
InvalidBucketName	247
ServiceQuotaExceededException	249
TooManyBuckets	249
模擬啟動期間拒絕權限	250
與使用時間有關的問題 Docker	250
主控台用戶端無法連線	250
沒simspaceweaver有在 AWS CLI	252

綱要參考	254
一個完整的模式的例子	254
架構格式	256
SDK 版本	256
模擬性質	257
工作程序	258
時鐘	259
分割策略	260
網域	261
位置限制	271
API 參考	273
SimSpace Weaver 版本	274
最新版本	274
如何找到您目前的版本	274
下載最新版本	274
排解應用程式 SDK 下載	275
安裝最新版本	275
服務版本	276
1.15.1	286
將現有的 Python 項目更新為 1.15.1	286
1.15.1 版的疑難排解	287
關於版本 1.15.1 的常見問題	287
文件歷史紀錄	288
詞彙表	294
.....	CCXCVIII

什麼是 AWS SimSpace Weaver ?

AWS SimSpace Weaver 是一項服務，可用於在中建置和執行大規模空間模擬 AWS 雲端。例如，您可以建立人群模擬、大型真實世界環境，以及身歷其境的互動式體驗。

您能夠在 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體之間分發模擬工作負載。SimSpace Weaver 為您部署基礎設施，並處理執行模擬的 Amazon EC2 執行個體之間的模擬資料管理和網路通訊。

重要概念 SimSpace Weaver

模擬或遊戲受限於執行模擬或遊戲的電腦。隨著虛擬世界的規模和複雜性的增加，處理效能開始降低。計算所花費的時間較長，系統記憶體不足，而且用戶端畫面播放速率下降。對於不需要實時性能的模擬，這可能只是令人討厭。或者，這可能是業務關鍵情況，因為處理延遲的增加導致成本增加。如果您的模擬或遊戲需要即時效能，那麼效能降低絕對是個問題。

達到效能限制的模擬的常見解決方案是簡化模擬。有許多用戶的在線遊戲通常通過在不同的服務器上複製他們的虛擬世界並將用戶分散到他們之間來解決擴展問題。

SimSpace Weaver 透過在空間上劃分虛擬世界，並將片段分配到中執行的計算執行個體叢集，藉此解決擴展問題 AWS 雲端。運算執行個體共同運作，以 parallel 處理整個模擬世界。您的模擬世界顯示為一個單一的集成空間，對於其中的所有內容以及連接到該世界的所有客戶。由於硬體效能限制，您不再需要簡化模擬。您可以改為在雲端中新增更多運算容量。

主題

- [SimSpace Weaver 的運作方式](#)
- [您如何使用 SimSpace Weaver](#)
- [模擬綱要](#)
- [工作者與資源單位](#)
- [模擬時鐘](#)
- [資料分割](#)
- [國家面料](#)
- [實體](#)
- [應用程式](#)

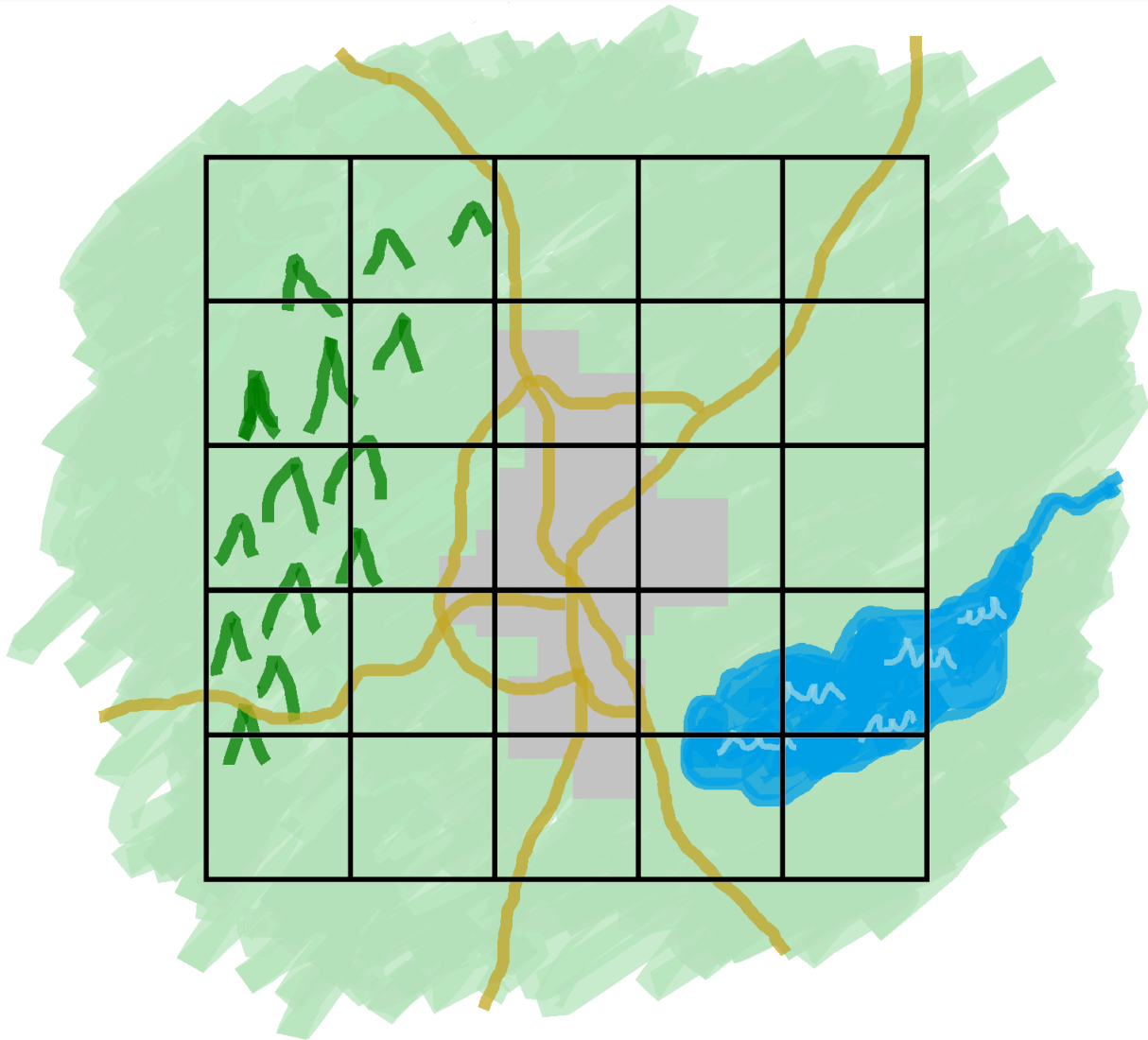
SimSpace Weaver 的運作方式

您的模擬由一個包含物件的世界組成。一些物體（例如人和車輛）移動並做事情。其他物件（例如樹木和建築物）是靜態的。在中 SimSpace Weaver，圖元是模擬世界中的物件。

您可以定義模擬世界的邊界並將其分割為網格。而不是建立在整個網格上運作的模擬邏輯，而是建立在網格的一個儲存格上運作的模擬邏輯。在中 SimSpace Weaver，空間應用程式是您撰寫的程式，可為網格的儲存格實作模擬邏輯。這包括該儲存格中所有實體的邏輯。空間應用程式的擁有權區域是空間應用程式控制的網格儲存格。

Note

在中 SimSpace Weaver，術語「app」可以指應用程式的程式碼或該程式碼的執行中執行個體。



您的模擬世界分為一個網格

您可以將模擬世界劃分為網格。每個空間應用程式都會為該網格中的單一儲存格實作模擬邏輯。

SimSpace Weaver 為網格的每個儲存格執行空間應用程式程式碼執行個體。所有空間應用程式執行個體並行執 parallel。基本上，SimSpace Weaver 將整體模擬劃分為多個較小的模擬。每個較小的模擬都可以處理整體模擬世界的一部分。SimSpace Weaver 可以將這些較小的模擬分發及執行個體上，並執行這些較小的模擬至 AWS 雲端。單一 Worker 可以執行多個空間應用程式。

圖元可以在模擬世界中移動。如果實體進入另一個空間應用程式的擁有權區域 (網格中的另一個儲存格)，則新區域的空間應用程式擁有者會接管實體的控制權。如果您的模擬在多個 Worker 上執行，則

實體可能會從一個 Worker 上的空間應用程式控制項移至不同 Worker 上的空間應用程式。當實體移至不同 Worker 時，會 SimSpace Weaver 處理基礎網路通訊。

訂閱

空間應用程式的世界觀是其自己的所有權區域。為了找出在模擬世界的另一個部分發生了什麼，空間應用程式創建了一個訂閱。訂閱區域是整體模擬世界區域的一個子集。訂閱區域可以包含多個擁有權區域的一部分，包括空間應用程式自己的擁有權區域。SimSpace Weaver 通知空間應用程式發生在訂閱區域內的所有實體事件 (例如，輸入、結束、建立、更新和刪除)。



空間應用程式的世界觀

空間應用程式的世界觀是它的所有權區域，這是世界網格中的一個單元格。



帶有添加訂閱區域的空間應用程式視圖

空間應用程式使用訂閱來了解模擬世界的另一個部分發生了什麼。訂閱區域可以包含多個網格儲存格和儲存格的一部分。

例如，模擬實體交互的應用程式可能需要了解超出其所有權區域空間邊界的實體。為此，該應用程式可以訂閱與其所有權區域邊界的區域。創建訂閱後，應用程式會收到有關這些區域中實體事件的通知，並且可以讀取實體。另一個例子是自動駕駛汽車，無論哪個應用程式擁有該區域，都需要查看前面 200 米的所有實體。(AABB)該車輛的應用程式可以創建帶有過濾器的訂閱，作為涵蓋可視區域的軸對齊邊界框。

您可以建立不負責管理模擬空間層面的模擬邏輯。自訂應用程式是在單一 Worker 上執行的可執行程式。您可以控制自訂應用程式的生命週期 (開始和停止)。模擬用戶端可以連線至自訂應用程式，以檢視模擬或與模擬互動。您也可以建立在每個 Worker 上執行的服務應用程式。SimSpace Weaver在執行模擬的每個 Worker 上啟動服務應用程式執行個體。

自訂應用程式和服務應用程式會建立訂閱，以了解實體事件和讀取實體。這些應用程序沒有所有權區域，因為它們不是空間的。使用訂閱是他們可以找出模擬世界中發生的事情的唯一方法。

您如何使用SimSpace Weaver

使用時SimSpace Weaver，以下是您遵循的主要步驟：

1. 撰寫並建置整合C++應用程式 SDK 的SimSpace Weaver應用程式。
 - a. 您的應用程式會進行 API 呼叫，以便與模擬狀態互動。
2. 撰寫可透過某些應用程式檢視您的模擬並與其互動的用戶端。
3. 在文字檔中規劃模擬。
4. 將您的應用程式套件和模擬設定上傳至服務。
5. 開始您的模擬。
6. 視需要啟動和停止您的自訂應用程式。
7. 將用戶端 Connect 至您的自訂或服務應用程式，以檢視模擬或與其互動。
8. 在亞馬遜日誌中檢查您的模擬CloudWatch日誌。
9. 停止您的模擬。
10. 清理您的模擬。

模擬綱要

模擬資料架構 (或資料架構) 是YAML格式化的文字檔，其中包含模擬的規劃資訊。SimSpace Weaver 啟動模擬時會使用您的資料架構。SimSpace Weaver應用程式 SDK 可散發套件包含範例專案的結構描述。您可以使用這個做為自有結構描述的起點。如需模擬資料架構的資訊，請參閱[SimSpace Weaver 模擬綱要參考](#)。

工作者與資源單位

工作者是執行模擬的 Amazon EC2 執行個體。您可以在模擬資料架構中指定 Worker 類型。SimSpace Weaver將您的工作者類型對應到服務使用的特定 Amazon EC2 執行個體類型。SimSpace

Weaver為您啟動和停止您的員工，並管理員工之間的網路通訊。SimSpace Weaver為每個模擬啟動一組工作者。不同的模擬使用不同的工作者。

Worker 上可用的運算 (處理器和記憶體) 容量分為邏輯單位，稱為計算資源單位 (或資源單位)。資源單位代表固定數量的處理器和記憶體容量。

Note

我們先前將計算資源單位稱為插槽。您可能仍會在我們的文件中看到上一個術語。

模擬時鐘

每個模擬都有自己的時鐘。您可以使用 API 呼叫或 SimSpace Weaver 主控台來啟動和停止時鐘。模擬只有在時鐘執行時才會更新。模擬中的所有作業均在稱為刻度的時間段內進行。時鐘向所有員工宣布每個刻度的開始時間。

時脈速率 (或刻度速率) 是時脈宣布的每秒刻度數 (赫茲或 Hz)。模擬所需的時脈速率是模擬結構描述的一部分。刻度線的所有操作必須在下一個刻度開始之前完成。因此，有效時脈速率可以低於所需的時脈速率。有效時脈速率不會高於所需的時脈速率。

資料分割

磁碟分割是 Worker 上共用記憶體的區段。每個分割區都保存模擬狀態資料的一部分。

空間應用程式的磁碟分割 (也稱為空間應用程式磁碟分割或空間分割區) 包含空間應用程式擁有權區域中的所有實體。SimSpace Weaver 根據每個實體的空間位置，將實體置於空間應用程式磁碟分割中。這表示 SimSpace Weaver 會嘗試將空間上彼此靠近的實體放在同一個 Worker 上。這樣可以最大限度地減少應用程序所需的知識量，它不擁有的實體來模擬它擁有的實體。

國家面料

狀態結構是所有 Worker 上共享內存 (所有分區的集合) 的系統。它會保留模擬的所有狀態資料。

State Fabric 使用自訂二進位格式，針對該實體的每個資料欄位，將實體描述為一組初始資料和更新記錄檔。使用此格式，您可以存取模擬時間中上一個點的圖元狀態，並將其對映回真實時間中的某個點。緩衝區具有有限的大小，並且不可能回到超出緩衝區中的內容。SimSpace Weaver 使用指向每個字段更新日誌中當前偏移量的指針，並在字段更新過程中更新指針。SimSpace Weaver 使用共享內存將這些更新日誌映射到應用程序的進程空間。

此物件格式會導致低額外負荷且無序列化成本。 SimSpace Weaver 也會使用此物件格式來剖析和識別索引欄位 (例如實體位置)。

實體

圖元是模擬中最小的資料建置區塊。實體的範例包括演員 (例如人員和車輛) 和靜態物件 (例如建築物和障礙物)。實體具有可儲存為永久性資料的屬性 (例如位置和方向) SimSpace Weaver。實體存在於磁碟分割內。

應用程式

SimSpace Weaver 應用程式是您編寫的軟件，其中包含運行每個模擬刻度的自定義邏輯。大多數應用程式的目的是在模擬執行時更新實體。您的應用程式會呼叫 SimSpace Weaver 應用程式 SDK 中的 API，以對模擬中的實體執行動作 (例如讀取和更新)。

您可以將應用程式及其所需資源 (例如程式庫) 封裝為 .zip 檔案，然後將它們上傳到 SimSpace Weaver。應用程式在背景工作上的 Docker 容器中執行的應用程式。 SimSpace Weaver 在 Worker 上為每個應用程式分配固定數量的資源單元。

SimSpace Weaver 為每個應用程式指派一個 (且只有一個) 分割區的擁有權。應用程式及其分割區位於同一個 Worker 上。每個分區只有一個應用程式所有者。應用程式可以建立、讀取、更新及刪除分割區中的實體。一個應用程式擁有其分區中的所有實體。

有三種類型的應用程式：空間應用程式，自定義應用程式和服務應用程式。它們因使用案例和生命週期而異。

Note

在中 SimSpace Weaver，術語「app」可以指應用程式的程式碼或該程式碼的執行中執行個體。

空間 App

空間應用程式會更新模擬中空間上存在的圖元的狀態。例如，您可以定義一個 Physics 應用程式，該應用程式負責根據其速度，形狀和大小為每個刻度移動和衝突實體。在此情況下， SimSpace Weaver 會 parallel 執行 Physics 應用程式的多個執行個體，以處理工作負載的大小。

SimSpace Weaver 管理空間應用程式的生命週期。您可以在模擬結構描述中指定空間應用程式分割區的排列。啟動模擬時，請為每個空間應用程式分割區 SimSpace Weaver 啟動空間應用程式。當您停止模擬時，會 SimSpace Weaver 關閉您的空間應用程式。

其他類型的應用程式可以建立實體，但只有空間應用程式可以更新實體。其他類型的應用程式必須將其建立的實體傳輸至空間網域。SimSpace Weaver 使用實體的空間位置，將實體移至空間應用程式的分割區。這會將實體的擁有權轉移至空間應用程式。

自訂 App

您可以使用自訂應用程式與模擬互動。自定義應用程序使用訂閱讀取實體數據。自訂應用程式可以建立實體。不過，應用程式必須將實體轉移至空間應用程式，才能將實體包含在模擬中並進行更新。您可以將網路端點 SimSpace Weaver 指派給自訂應用程式。模擬客戶端可以連接到網路端點以與模擬進行交互。您可以在模擬結構描述中定義自訂應用程式，但您有責任啟動和停止它們（使用 SimSpace Weaver API 呼叫）。在 Worker 上啟動自訂應用程式執行個體之後，SimSpace Weaver 不會將執行個體轉移給其他背景工作。

服務應用

當您需要在每個 Worker 上執行唯讀程序時，可以使用服務應用程式。例如，如果您有一個大型模擬，並且您需要一個檢視用戶端，該用戶端會在模擬中移動並僅向使用者顯示可見實體，則可以使用服務應用程式。在這種情況下，單個自定義應用程序實例無法處理模擬中的所有實體。您可以將服務應用程式設定為在每個 Worker 上啟動。然後，每個服務應用程式都可以篩選其指派 Worker 上的實體，並僅將相關實體傳送至其連線的用戶端。然後，您的觀看客戶端可以在模擬空間中移動時連接到不同的服務應用程序。您可在模擬結構描述中設定服務應用程式。SimSpace Weaver 為您啟動和停止服務應用程式。

App 摘要

下表摘要列出不同類型之 SimSpace Weaver 應用程式的特性。

	空間 App	自訂 App	服務應用
讀取實體	是	是	是
更新實體	是	否	否
建立實體	是	是*	是*

	空間 App	自訂 App	服務應用
生命週期	管理 (SimSpace Weaver控制它。)	未受管理 (您可以控制它。)	管理 (SimSpace Weaver控制它。)
啟動方法	SimSpace Weaver 為每個空間分區啟動一個應用程式實例，如模式中指定的那樣。	啟動每個應用程式執行個體。	SimSpace Weaver 根據結構描述中的指定，在每個 Worker 上啟動一個或多個應用程式執行個體。
用戶端可以連線	否	是	是

* 當自訂應用程式或服務應用程式建立實體時，應用程式必須將實體的擁有權轉移到空間應用程式，以便空間應用程式可以更新實體的狀態。

網域

SimSpace Weaver 網域是執行相同可執行應用程式程式碼且具有相同啟動選項和命令的應用程式執行個體集合。我們會依包含的應用程式類型來參考網域：空間網域、自訂網域和服務網域。您可以在網域中設定應用程式。

訂閱和複寫

應用程式會建立空間區域的訂閱，以了解該區域中的實體事件 (例如，輸入、結束、建立、更新和刪除)。應用程式會先處理訂閱中的實體事件，然後再讀取未擁有之分割區中實體的資料。

分區可以存在於與應用程式相同的 Worker 上 (這稱為本地分區)，但另一個應用程式可以擁有該分區。分割區也可以存在於不同的 Worker 上 (這稱為遠端磁碟分割)。如果訂閱是遠端磁碟分割，Worker 會透過稱為複寫的程序建立遠端磁碟分割的本機副本。然後 Worker 會讀取本機副本 (複製的遠端磁碟分割)。如果 Worker 上的另一個應用程式需要在同一個刻度上從該磁碟分割讀取，則 Worker 會讀取相同的本機副本。

範例使用案例 SimSpace Weaver

您可以用 SimSpace Weaver 於以代理程式為基礎的模型，以及具有空間元件的離散時間步長模擬。

建立大量人群模擬

您可以使SimSpace Weaver用模擬真實環境中的人群。 SimSpace Weaver可讓您將模擬縮放至數百萬個具有自己行為的動態物件。

建立城市規模的環境

用SimSpace Weaver於創建整個城市的數字雙胞胎。創建城市規劃模擬，設計交通路線，並規劃環境危害響應。您可以使用自己的空間資料來源做為環境的建置圖塊。

創建身臨其境的互動體驗

創建多個用戶可以參與和互動的模擬體驗。使用流行的開發工具，如虛幻引擎和統一來構建三維（3D）虛擬世界。使用您自己的內容和行為自訂3D體驗。

設定 SimSpace Weaver

若要設定第一次使 SimSpace Weaver 用，您必須設定您的 AWS 帳戶和您的本機環境。完成這些任務後，您將準備好進行 [入門教學課程](#)。

設定任務

1. [設置您的 AWS 帳戶使用 SimSpace Weaver](#).
2. [設定您的本機環境 SimSpace Weaver](#).

設置您的 AWS 帳戶使用 SimSpace Weaver

完成以下任務來設置您的 AWS 帳戶使用 SimSpace Weaver。

註冊 AWS 帳戶

如果您還沒有 AWS 帳戶，請完成以下步驟建立新帳戶。

註冊 AWS 帳戶

1. 開啟 <https://portal.aws.amazon.com/billing/signup>。
2. 請遵循線上指示進行。

部分註冊程序需接收來電，並在電話鍵盤輸入驗證碼。

註冊 AWS 帳戶時，會建立 AWS 帳戶根使用者。根使用者有權存取該帳戶中的所有 AWS 服務和資源。作為最佳安全實務，[將管理存取權指派給管理使用者](#)，並且僅使用根使用者來執行 [需要根使用者存取權的任務](#)。

註冊程序完成後，AWS 會傳送一封確認電子郵件給您。您可以隨時登錄 <https://aws.amazon.com/> 並選擇 [我的帳戶](#)，以檢視您目前的帳戶活動並管理帳戶。

建立管理使用者

在您註冊 AWS 帳戶之後，請保護您的 AWS 帳戶根使用者、啟用 AWS IAM Identity Center，以及建立管理使用者，讓您可以不使用根使用者處理日常作業。

保護您的 AWS 帳戶根使用者

1. 選擇 根使用者 並輸入您的 AWS 帳戶電子郵件地址，以帳戶擁有者身分登入 [AWS Management Console](#)。在下一頁中，輸入您的密碼。

如需使用根使用者登入的說明，請參閱 AWS 登入使用者指南中的 [以根使用者身分登入](#)。

2. 若要在您的根使用者帳戶上啟用多重要素驗證 (MFA)。

如需指示，請參閱《IAM 使用者指南》中的 [為 AWS 帳戶根使用者啟用虛擬 MFA 裝置 \(主控台\)](#)。

建立管理使用者

1. 啟用 IAM Identity Center。

如需指示，請參閱 AWS IAM Identity Center 使用者指南中的 [啟用 AWS IAM Identity Center](#)。

2. 在 IAM Identity Center 中，將管理存取權授予管理使用者。

有關如何使用 IAM Identity Center 目錄 作為身分來源的教學課程，請參閱《AWS IAM Identity Center 使用者指南》中的 [以預設 IAM Identity Center 目錄 設定使用者存取權](#)。

以管理員的身分登入

- 若要使用您的 IAM 身分中心使用者登入，請使用建立 IAM 身分中心使用者時傳送至您電子郵件地址的登入 URL。

如需有關如何使用 IAM Identity Center 使用者登入的說明，請參閱《AWS 登入 使用者指南》中的 [登入 AWS 存取入口網站](#)。

新增要使用的權限 SimSpace Weaver

若要提供存取權，請新增權限至您的使用者、群組或角色：

- AWS IAM Identity Center 中的使用者和群組：

建立權限合集。請遵循 AWS IAM Identity Center 使用者指南的 [建立權限合集](#) 中的指示。

- 透過身分提供者在 IAM 中管理的使用者：

建立聯合身分的角色。請遵循 IAM 使用者指南的 [為第三方身分提供者 \(聯合\) 建立角色](#) 中的指示。

- IAM 使用者：
 - 建立您的使用者可擔任的角色。請遵循 IAM 使用者指南的 [為 IAM 使用者建立角色](#) 中的指示。
 - (不建議) 將政策直接附加至使用者，或將使用者新增至使用者群組。請遵循 IAM 使用者指南的 [新增權限至使用者 \(主控台\)](#) 中的指示。

Example 授予使用許可的 IAM 政策 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"
      }
    }
  }
]
```

設定您的本機環境SimSpace Weaver

SimSpace Weaver 模擬在容器化 Amazon Linux 2 (AL2) 環境中執行。您必須擁有 AL2 環境，才能編譯並將應用程式與應用 SimSpace Weaver 程式 SDK 連結。標準本機開發環境是中的 AL2 容器 Docker。如果您選擇不使用 Docker，我們會提供在中執行 AL2 環境的替代指示 Windows Subsystem for Linux (WSL)。您也可以使用您自己的方法建立本機 AL2 環境。如需在本機執行 AL2 的其他方法，請參閱 [Amazon EC2 文件](#)。

Important

Docker on Microsoft Windows 是標準的開發環境。為方便起見，我們建議使用其他方式來設定您的本機開發環境，但這些方法並非標準且不受支援。

主題

- [在中設定 Amazon Linux 2 \(AL2\) Docker](#)
- [在中設定 Amazon Linux 2 \(AL2\) Windows Subsystem for Linux \(WSL\)](#)

在中設定 Amazon Linux 2 (AL2) Docker

本節提供在 Docker 中設定本機 AL2 環境的指示。如需在中設定 AL2 的指示 Windows Subsystem for Linux (WSL)，請參閱 [在中設定 Amazon Linux 2 \(AL2\) Windows Subsystem for Linux \(WSL\)](#)。

需求

- Microsoft Windows 10 或更高
- [Microsoft Visual Studio 2019](#) 或更新版本，安裝 [Desktop development with C++](#) 工作負載

- [CMake3](#)
- [Git](#)
- [Docker Desktop](#)
- [AWS CLI](#)

若要在中設定 AL2 的步驟 Docker

1. 如果您尚未設定的AWS登入資料AWS CLI，請遵循下列指示：[設定 AWS CLI](#)。如果您只是使用，SimSpace Weaver則可以將其配置AWS CLI為默認使用SimSpace Weaver憑據。
2. [下載SimSpace Weaver應用程式 SDK 可散發套件](#)。它包含以下內容：
 - 應用程式開發的二進位檔和SimSpace Weaver程式庫
 - 自動化部分開發工作流程的協助程式指令碼
 - 示範SimSpace Weaver概念的範例應用程式
3. 將文件解壓縮到您選擇`sdk-folder`的文件。
4. 前往 `sdk-folder`。
5. 輸入以下指令以建立Docker影像。

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

Note

如果在此步驟中出現錯誤，請確定該步驟Docker正在執行。

在中設定 Amazon Linux 2 (AL2) Windows Subsystem for Linux (WSL)

本節提供在Windows Subsystem for Linux (WSL)中設定本機 AL2 環境的指示。如需在中設定 AL2 的指示Docker，請參閱[在中設定 Amazon Linux 2 \(AL2\) Docker](#)。

Important

本節說明使用非 Amazon 擁有、開發或支援的 AL2 版本的解決方案。如果您選擇不使用，此解決方案僅為了您的方便而提供Docker。亞馬遜，AWS如果您選擇使用此解決方案，則不承擔任何責任。

需求

- [Hyper-V 上的 Windows 10](#)
- [Windows Subsystem for Linux \(WSL\)](#)
- [第三方開源 AL2 發行版 WSL \(下載版本 2.0.202000 更新。2 \) \(請參閱說明書 \)](#)

Important

我們的 WSL 指示使用 AL2 發行版本的 [2.0.20200722.0 更新](#)。WSL 如果您使用任何其他版本，則可能會遇到錯誤。

若要在中設定 AL2 的步驟 WSL

1. 在 WSL Windows 命令提示字元中，啟動中的 AL2 環境。

```
wsl -d Amazon2
```

Important

在中執行時 WSL，請使用 Linux 指示來取代本指南中的 Windows 指示 (如果您有選擇的話)。如果沒有 Linux 指令，請使用以下替換：

- 使用 `tools/linux` 而不是 `tools\windows`
- 使用 `.sh` 腳本而不是 `.bat` 腳本

2. 在 Linux 命令介面提示字元中，更新您的 yum 套件管理員。

```
yum update -y
```

Important

如果此步驟逾時，您可能需要切換至 WSL1 並重試這些程序。結束 WSL AL2 工作階段，然後在 Windows 命令提示字元中輸入下列命令：


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

3. 安裝解壓縮工具。

```
yum install -y unzip
```

4. 刪除任何AWS CLI已yum安裝的。如果您不確定是否yum已安裝AWS CLI。

```
yum remove awscli
```

```
yum remove aws-cli
```

5. 創建一個臨時目錄並轉到它。

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

6. 下載並安裝AWS CLI：

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

7. 您可以移除暫存目錄。

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

8. 重新啟動 shell 工作階段以更新環境中的路徑。

```
exec
```

9. 在 AL2 環境AWS CLI中設定您的AWS認證。如需詳細資訊，請參閱[設定 AWS CLI](#)。如果您使用AWS IAM Identity Center，請參閱《[使用AWS Command Line Interface者AWS CLI指南](#)》AWS IAM Identity Center中的〈配置使用〉。

```
aws configure
```

10. 安裝 Git。

```
yum install -y git
```

11. 安裝 wget。

```
yum install -y wget
```

12. 建立 SimSpace Weaver 應用程式 SDK 的資料夾。

```
mkdir sdk-folder
```

13. 轉到您的 SDK 文件夾。

```
cd sdk-folder
```

14. 下載 SimSpace Weaver 應用程式 SDK 可散發檔案。它包含以下內容：

- 應用程式開發的二進位檔和 SimSpace Weaver 程式庫
- 自動化部分開發工作流程的協助程式指令碼
- 示範 SimSpace Weaver 概念的範例應用程式


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

15. 解壓縮檔案。

```
unzip *.zip
```

16. 執行其他安裝指令碼。

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

 Note

您只需要針對中 WSL 的 AL2 環境執行此操作一次。

使用授權軟體AWS SimSpace Weaver

AWS SimSpace Weaver 允許您使用自己選擇的模擬引擎和內容來構建模擬。就您的使用而言 SimSpace Weaver，您有責任取得、維護及遵守您在模擬中使用之任何軟體或內容的授權條款。驗證授權合約允許在虛擬託管環境中部署軟體和內容。

SimSpace Weaver 入門

本節提供教程，可協助您開始使用的是本節提供教SimSpace Weaver程 這些自學課程向您介紹使用建置模擬的一般工作流程SimSpace Weaver。在這兩個自學課程中，您將學習如何在中建立、部署和執行模擬SimSpace Weaver。我們建議您從快速入門自學課程開始，以在幾分鐘內執行模擬。然後，通過詳細的教程以了解有關該過程中每個步驟的更多信息。

這些教學課程使用您在安裝程序期間下載的SimSpace Weaver應用程式 SDK .zip 檔案中包含的範例應用程式 (PathfindingSample)。範例應用程式示範所有SimSpace Weaver模擬共用的概念，包括空間磁碟分割、跨分割區實體遞交、應用程式和訂閱。

在自學課程中，您將建立具有四個空間分割區的模擬。PathfindingSample空間應用程序的單獨實例管理每個單獨的分區。空間應用程式會在自己的磁碟分割中建立實體。圖元會移動到模擬世界中的特定位置，避開移動時的障礙物。您可以使用單獨的用戶端應用程式 (包含在SimSpace Weaver應用程式 SDK 中) 來檢視模擬。

主題

- [快速入門教學課程：在幾分鐘內建置並執行模擬](#)
- [詳細教學課程：在建置範例應用程式時瞭解詳細資料](#)

快速入門教學課程：在幾分鐘內建置並執行模擬

本自學課程將引導您完成SimSpace Weaver在幾分鐘內建置和執行模擬的過程。我們建議您從本教程開始，然後通過詳細的教程之後。

請求

開始之前，請務必先完成「」中的步驟[設定 SimSpace Weaver](#)。

步驟

- [步驟 1：建立專案](#)
- [步驟 2：開啟記錄功能 \(選用\)](#)
- [步驟 3：執行快速啟動指令碼](#)
- [步驟 4：獲取您的 IP 地址和端口號](#)
- [步驟 5：檢視您的模擬](#)
- [步驟 6：停止並清理您的模擬](#)

步驟 1：建立專案

可散發的 SimSpace Weaver 應用程式 SDK 包含一個腳本，該腳本從捆綁的項目創建 PathfindingSample 項目。您必須從指令碼在檔案系統中的位置執行指令碼。指令碼會使用您在命令列中提供的值在 `##project-name` 內建立路徑。

Docker

建立專案

1. 在 Windows 指令提示下，變更至您的專案資料夾。

```
cd sdk-folder
```

2. 執行 `create-project.bat` 指令碼。

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

完整的指令和參數清單如下。

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

Important

請勿使用超過 20 個字元的專案名稱。超過此限制可能會導致錯誤。

Note

如果您收到共用磁碟機的提示 Docker，請選擇共 Yes 用磁碟機。

Important

如果您為 AWS Command Line Interface (AWS CLI) 使用 AWS IAM Identity Center 或命名描述檔，您必須使用 SimSpace Weaver 應用程式 SDK 1.12.1 版或更新版本。最新的版本是 1.16.0。如需 SimSpace Weaver 版本的詳細資訊，請參閱 [SimSpace](#)

[Weaver 版本](#)。SimSpace Weaver應用程式 SDK 指令碼會使用AWS CLI。如果您使用IAM 身分中心，您可以將您的IAM 身分中心設定檔複製到您的default設定檔，或將IAM 身分中心設定檔的名稱提供給具有`--profile cli-profile-name`參數的SimSpace Weaver應用程式 SDK 指令碼。如需詳細資訊，請參閱《[使AWS CLI用AWS Command Line Interface者指南](#)》AWS IAM Identity Center中的〈[配置使用指南](#)〉和〈[組態和認證檔案設定AWS Command Line Interface](#)〉。

WSL

⚠ Important

為了您的方便，我們提供這些說明。它們適用於搭配使用Windows Subsystem for Linux (WSL)，且不受支援。如需詳細資訊，請參閱 [設定您的本機環境SimSpace Weaver](#)。

建立專案

1. 在Linuxshell 提示符下，切換到您的項目文件夾。

```
cd sdk-folder
```

2. 執行 `create-project.sh` 指令碼。

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

完整的指令和參數清單如下。

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

⚠ Important

請勿使用超過 20 個字元的專案名稱。超過此限制可能會導致錯誤。

⚠ Important

如果您為 AWS Command Line Interface (AWS CLI) 使用AWS IAM Identity Center 或命名描述檔，您必須使用SimSpace Weaver應用程式 SDK 1.12.1 版或更新版本。最新的版本是 1.16.0。如需 SimSpace Weaver 版本的詳細資訊，請參閱[SimSpace Weaver 版本](#)。SimSpace Weaver應用程式 SDK 指令碼會使用AWS CLI。如果您使用 IAM 身分中心，您可以將您的 IAM 身分中心設定檔複製到您的default設定檔，或將 IAM 身分中心設定檔的名稱提供給具有--profile *cli-profile-name* 參數的SimSpace Weaver應用程式 SDK 指令碼。如需詳細資訊，請參閱《[使AWS CLI 用AWS Command Line Interface者指南](#)》AWS IAM Identity Center中的〈[配置使用指南](#)〉和〈[組態和認證檔案設AWS Command Line Interface定](#)〉。

參數**name**

專案的名稱。

path

專案在檔案系統中的位置。專案結構有時可能具有較長的檔案路徑，這些路徑可能會超過作業系統中的路徑長度限制。我們建議您使用盡可能短的路徑名稱。

profile

指令碼應該用於驗證的AWS CLI設定檔名稱。如需詳細資訊，請參閱AWS Command Line Interface使用指南中的[組態和認證檔案設定](#)。此參數僅適用於SimSpace Weaver應用程式 SDK 1.12.1 版或更新版本。如需 SimSpace Weaver 版本的詳細資訊，請參閱 [SimSpace Weaver 版本](#)。

app-sdk-version

(可選) 您的項目使用的SimSpace Weaver應用程序 SDK 版本。您將使用此版本構建並鏈接您的應用程序。如果指令碼在可發佈的位置找不到版本，或者您沒有提供版本號碼，指令碼就會自動下載最新版本。

template

(選擇性) 指令碼用來建立專案的專案範本。如果您未提供範本，指令碼會使用PathfindingSample. 有效值：

- **PathfindingSample**— 使用單一 [Worker](#) 的範例應用程式。
- **MultiWorkerPathfindingSample**— 使用多個 [Worker](#) 的範例應用程式版本。

overwriteproject

(選擇性) 使用此選項可以使用相同的專案名稱和路徑覆寫既有的專案資料夾。

步驟 2：開啟記錄功能 (選用)

依預設，PathfindingSample專案的記錄處於關閉狀態。本教學課程假設已開啟記錄。您可以選擇開啟記錄功能，但這是選擇性的。

Important

會PathfindingSample產生大量的記錄資料。如果您選擇開啟記錄功能，則會收到記錄資料的帳單費用。只要記錄資料存在，您就會繼續收取帳單費用。如果您打開記錄功能，我們強烈建議您停止此模擬並儘快在自學課程結束時執行清理步驟。

開啟記錄

1. 在文字編輯器中開啟下列檔案：

Docker

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

Note


您的####*path**project-name*使用您在建立專案時提供的值。

WSL

Important

為了您的方便，我們提供這些說明。它們適用於搭配使用Windows Subsystem for Linux (WSL)，且不受支援。如需詳細資訊，請參閱[設定您的本機環境SimSpace Weaver](#)。

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

 Note

您的####*path/project-name*使用您在建立專案時提供的值。

2. 找到simulation_properties:文件開頭的部分：

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

3. 在該行後插入以下兩行simulation_properties:：

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

4. 確認您的simulation_properties:區段與以下內容相同：

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

5. 儲存檔案並結束您的文字編輯器。

步驟 3：執行快速啟動指令碼

範例應用程式包含快速啟動指令碼。該腳本將創建，構建，上傳和啟動您的模擬及其應用程序。

Docker

執行快速啟動指令碼

1. 如果您還不在那裡，請轉到項目和平台的工具文件夾。您*project-folder*正在*path\project-name*使用您在建立專案時提供的值。

於Windows指令提示下，輸入：

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

2. 執行此專案的快速入門指令碼。

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

Important

從版本 1.12.3 開始，quick-start 指令碼會以 1 小時的最長持續時間啟動您的模擬。您可以使用 `--maximum-duration` 參數來指定不同的最大持續時間。在版本 1.12.2 或更低版本中，您無法為腳本提供最長持續時間，並且模擬的最長持續時間為 14 天。若要取得有關模擬最長持續時間的更多資訊，請參閱 [〈模擬持續時間上限〉](#)。

Important

如果您為 AWS Command Line Interface (AWS CLI) 使用 AWS IAM Identity Center 或命名描述檔，您必須使用 SimSpace Weaver 應用程式 SDK 1.12.1 版或更新版本。最新的版本是 1.16.0。如需 SimSpace Weaver 版本的詳細資訊，請參閱 [SimSpace Weaver 版本](#)。SimSpace Weaver 應用程式 SDK 指令碼會使用 AWS CLI。如果您使用 IAM 身分中心，您可以將您的 IAM 身分中心設定檔複製到您的 default 設定檔，或將 IAM 身分中心設定檔的名稱提供給具有 `--profile cli-profile-name` 參數的 SimSpace Weaver 應用程式 SDK 指令碼。如需詳細資訊，請參閱 [《使 AWS CLI 用 AWS Command Line Interface 者指南》](#) AWS IAM Identity Center 中的 [〈配置使用指南〉](#) 和 [〈組態和認證檔案設 AWS Command Line Interface 定〉](#)。

WSL

Important

為了您的方便，我們提供這些說明。它們適用於搭配使用 Windows Subsystem for Linux (WSL)，且不受支援。如需詳細資訊，請參閱 [設定您的本機環境 SimSpace Weaver](#)。

執行快速啟動指令碼

1. 如果您尚未存在，請轉到項目和平台的 tools 文件夾。您 *project-folder* 正在 *path/project-name* 使用創建項目時提供的值。

在Linux薄殼提示下，輸入：

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

2. 執行此專案的快速入門指令碼。

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

Important

從版本 1.12.3 開始，quick-start指令碼會以 1 小時的最長持續時間啟動您的模擬。您可以使用--maximum-duration參數來指定不同的最大持續時間。在版本 1.12.2 或更低版本中，您無法為腳本提供最長持續時間，並且模擬的最長持續時間為 14 天。若要取得有關模擬最長持續時間的更多資訊，請參閱〈[模擬持續時間上限](#)〉。

Important

如果您為 AWS Command Line Interface (AWS CLI) 使用AWS IAM Identity Center 或命名描述檔，您必須使用SimSpace Weaver應用程式 SDK 1.12.1 版或更新版本。最新的版本是 1.16.0。如需 SimSpace Weaver 版本的詳細資訊，請參閱[SimSpace Weaver 版本](#)。SimSpace Weaver應用程式 SDK 指令碼會使用AWS CLI。如果您使用 IAM 身分中心，您可以將您的 IAM 身分中心設定檔複製到您的default設定檔，或將 IAM 身分中心設定檔的名稱提供給具有--profile *cli-profile-name* 參數的SimSpace Weaver應用程式 SDK 指令碼。如需詳細資訊，請參閱《[使AWS CLI 用AWS Command Line Interface者指南](#)》AWS IAM Identity Center中的〈[配置使用指南](#)〉和〈[組態和認證檔案設AWS Command Line Interface定](#)〉。

該腳本將啟動一個循環並在所有組件後自動停止STARTED。尋找類似下列內容的輸出：

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

```
"CreationTime": 1664921418.09,
```

步驟 4：獲取您的 IP 地址和端口號

您必須獲取視圖 (自定義) 應用程式的 IP 地址和端口號，以便可以連接到模擬。以下程序假設您對模擬一無所知 (例如模擬名稱)。您可以隨時使用此程序來尋找自訂應用程式或服務應用程式的 IP 位址和連接埠號碼。下列範例輸出適用於名為的專案MyProject。

Docker

取得您的 IP 位址和連接埠號碼

1. 如果您還不在那裡，請轉到項目和平台的工具文件夾。您 *project-folder* 正在 *path\project-name* 使用您在建立專案時提供的值。

於Windows指令提示下，輸入：

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

2. 使用 ListSimulationsAPI 取得模擬的名稱。

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

Important

如果您為 AWS Command Line Interface (AWS CLI) 使用AWS IAM Identity Center 或命名描述檔，您必須使用SimSpace Weaver應用程式 SDK 1.12.1 版或更新版本。最新的版本是 1.16.0。如需 SimSpace Weaver 版本的詳細資訊，請參閱[SimSpace Weaver 版本](#)。SimSpace Weaver應用程式 SDK 指令碼會使用AWS CLI。如果您使用 IAM 身分中心，您可以將您的 IAM 身分中心設定檔複製AWS CLI到您的default設定檔，或將 IAM 身分中心設定檔的名稱提供給具有 `--profile cli-profile-name` 參數的SimSpace Weaver應用程式 SDK 指令碼。如需詳細資訊，請參閱[《使AWS CLI 用AWS Command Line Interface者指南》](#) AWS IAM Identity Center中的〈配置使用指南〉和〈組態和認證檔案設AWS Command Line Interface定〉。

輸出範例：

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

3. 使用 DescribeSimulationAPI 取得模擬中的網域清單。

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

查找輸出 Domains 部 LiveSimulationState 分中的部分。

輸出範例：

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

4. 使用 ListAppsAPI 取得網域中的自訂應用程式清單。範例專案中檢視 (自訂) 應用程式的網域名稱為MyViewDomain。在輸出中尋找應用程式名稱。

```
.\weaver-project-name-cli.bat list-apps --simulation simulation-name --  
domain domain-name
```

輸出範例：

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

5. 使用 DescribeAppAPI 取得 IP 位址和連接埠號碼。對於範例專案，網域名稱為MyViewDomain，應用程式名稱為ViewApp。

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

IP 位址和連接埠號碼位於輸出的EndpointInfo區塊中。IP 位址是的值Address，連接埠號碼為的值Actual。

輸出範例：

```
{  
  "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

的值Declared是應用程式程式碼應該繫結到的連接埠號碼。的值Actual是SimSpace Weaver公開給用戶端以連線到您的應用程式的連接埠號碼。 SimSpace Weaver 將Declared連接埠對應至連Actual接埠。

WSL

Important

為了您的方便，我們提供這些說明。它們適用於搭配使用Windows Subsystem for Linux (WSL)，且不受支援。如需詳細資訊，請參閱 [設定您的本機環境SimSpace Weaver](#)。

取得您的 IP 位址和連接埠號碼

1. 如果您尚未存在，請轉到項目和平台的 tools 文件夾。您`project-folder`正在`path/project-name`使用創建項目時提供的值。

在Linux薄殼提示下，輸入：

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

2. 使用 ListSimulationsAPI 取得模擬的名稱。

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

⚠ Important

如果您為 AWS Command Line Interface (AWS CLI) 使用AWS IAM Identity Center 或命名描述檔，您必須使用SimSpace Weaver應用程式 SDK 1.12.1 版或更新版本。最新的版本是 1.16.0。如需 SimSpace Weaver 版本的詳細資訊，請參閱[SimSpace Weaver 版本](#)。SimSpace Weaver應用程式 SDK 指令碼會使用AWS CLI。如果您使用 IAM 身分中心，您可以將您的 IAM 身分中心設定檔複製到您的default設定檔，或將 IAM 身分中心設定檔的名稱提供給具有--profile *cli-profile-name* 參數的SimSpace Weaver應用程式 SDK 指令碼。如需詳細資訊，請參閱《[使AWS CLI 用AWS Command Line Interface者指南](#)》AWS IAM Identity Center中的〈[配置使用指南](#)〉和〈[組態和認證檔案設AWS Command Line Interface定](#)〉。

輸出範例：

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

3. 使用 DescribeSimulationAPI 取得模擬中的網域清單。

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

查找輸出Domains部LiveSimulationState分中的部分。

輸出範例：

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

4. 使用 ListAppsAPI 取得網域中的自訂應用程式清單。範例專案中檢視 (自訂) 應用程式的網域名稱為 MyViewDomain。在輸出中尋找應用程式名稱。

```
./weaver-project-name-cli.sh list-apps --simulation simulation-name --
domain domain-name
```

輸出範例：

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

5. 使用 DescribeAppAPI 取得 IP 位址和連接埠號碼。對於範例專案，網域名稱為 MyViewDomain，應用程式名稱為 ViewApp。

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

IP 位址和連接埠號碼位於輸出的 EndpointInfo 區塊中。IP 位址是的值 Address，連接埠號碼為的值 Actual。

輸出範例：

```
{  
  "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

的值 Declared 是應用程式程式碼應該繫結到的連接埠號碼。的值 Actual 是 SimSpace Weaver 公開給用戶端以連線到您的應用程式的連接埠號碼。 SimSpace Weaver 將 Declared 連接埠對應至連 Actual 接埠。

步驟 5：檢視您的模擬

SimSpace Weaver 應用程式 SDK 提供不同的選項來檢視範例應用程式。如果您沒有任何本地 Unreal Engine 開發支持，則可以使用示例控制台客戶端。用 Unreal Engine 客戶端的指示假設您正在使用 Windows。

主控台用戶端會在實體事件發生時顯示實體事件清單。用戶端會從取得實體事件資訊 ViewApp。如果您的主控台用戶端顯示事件清單，則它會確認與模擬中的 ViewApp 和活動的網路連線。

PathfindingSample 模擬會在 2D 平面上建立固定和移動的圖元。移動的圖元會繞著固定的圖元移動。虛幻引擎用戶端會提供實體事件的視覺效果。

Windows console client

要求

- Microsoft Windows 10 或更高
- [Microsoft Visual Studio 2019](#) 或更新版本，安裝 [Desktop development with C++](#) 工作負載
- [CMake3](#)
- [Git](#)

使用範例主控台用戶端連線至範例應用程式

1. 在命令提示符窗口中，轉到控制台客戶端的文件夾（在應用程式 SDK 文件夾中）。

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

2. 用 CMake3 於在此資料夾中建立 Visual Studio 解決方案。

```
cmake .
```

Note

確保在末尾包括空格和句點。

⚠ Important

保持命令提示符窗口打開以進行進一步的步驟。

3. 在中Visual Studio，開啟您PathfindingSampleConsoleClient.sln在上一個步驟中建立的。
4. 選擇構RelWithDebInfo建配置。
5. 選擇「Build>」Build Solution。
6. 在之前的命令提示符窗口中，轉到控制台客戶端文件夾中的構建輸出文件夾。

```
cd RelWithDebInfo
```

7. 使用您的 IP 地址和端口號運行客戶端ViewApp。

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

您的命令提示字元視窗應該會顯示實體更新、刪除和建立事件的數字，類似下列範例輸出。

📘 Note


下列範例輸出中的 IP 位址和連接埠號碼為預留位置。將您的 IP 位址和連接埠號碼提供ViewApp給主控台用戶端。如果您要連線至在中執行的連接Actual埠號碼ViewApp，請提供連接埠號碼AWS 雲端。當您連線到本機系統上執行的 IP 位址和連接埠號碼 127.0.0.1:7000 時，請提供 IP 位ViewApp址和連接埠號碼。如需詳細資訊，請參閱[本機開發](#)。

```
##PathfindingSample#ViewApp Message Reader##  
  
Added argument url:tcp://198.51.100.135:4321  
Some subscription arguments are missing, restoring defaults.  
  
*****  
Sample usage without a MoveStrategy:
```

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

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


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

 Note

如需疑難排解指引，請參閱[PathfindingSample 主控台用戶端無法連線](#)。

8. 按下CTRL+C可結束主控台用戶端。

Linux console client

 Important

為了您的方便，我們提供這些說明。它們可能無法在某些Linux環境中運作。這些程序不受支援。

此程序假設您完全在Linux環境中工作。您也可以使用內建的用戶端來檢視模擬Windows。

要求

- CMake3
- C編譯器 (已包含在中Amazon Linux 2)
- Git

使用範例主控台用戶端連線至範例應用程式

1. 在 Linuxshell 提示符下，轉到控制台客戶端的文件夾 (在應用程式 SDK 文件夾中)。

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

2. 創建一個構建文件夾。

```
mkdir build
```

3. 轉到構建文件夾。

```
cd build
```

4. 用CMake3於建置用戶端。

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

Note

確保在末尾包括空格和句點。

5. 使用您的 IP 地址和端口號運行客戶端ViewApp。

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

您的命令提示字元視窗應該會顯示實體更新、刪除和建立事件的數字，類似下列範例輸出。

Note

下列範例輸出中的 IP 位址和連接埠號碼為預留位置。將您的 IP 位址和連接埠號碼提供ViewApp給主控台用戶端。如果您要連線至在中執行的連接Actual埠號

碼ViewApp，請提供連接埠號碼AWS 雲端。當您連線到本機系統上執行的 IP 位址和連接埠號碼 127.0.0.1:7000 時，請提供 IP 位ViewApp址和連接埠號碼。如需詳細資訊，請參閱[本機開發](#)。

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

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

```
*****
```

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

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

```
Sample usage with CircleMoveStrategy:
```

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

```
*****
```

```
Starting NNG client. NNG version: 1.2.4
```

```
Creating socket ...done.
```

```
Connecting to View App ... done.
```

```
Initiating connection to tcp:// 198.51.100.135:4321 ... done.
```

```
Receiving messages ...
```

```
[2022-10-04 19:13:00.710] CreateEntity Count: 72
```

```
[2022-10-04 19:13:00.756] UpdateEntity Count: 42
```

```
[2022-10-04 19:13:00.794] DeleteEntity Count: 72
```

```
[2022-10-04 19:13:03.690] CreateEntity Count: 11
```

```
[2022-10-04 19:13:03.725] UpdateEntity Count: 2
```

```
[2022-10-04 19:13:03.757] UpdateEntity Count: 2
```

```
[2022-10-04 19:13:03.790] UpdateEntity Count: 2
```

Note

如需疑難排解指引，請參閱[PathfindingSample 主控台用戶端無法連線](#)。

6. 按下CTRL+C可結束主控台用戶端。

虛擬引擎 on Windows

要求

- Unreal Engine 5開發環境
- Microsoft .NET Framework 4.8 Developer Pack
- Windows主控台用戶端 (請參閱此頁面上的Windows主控台用戶端標籤)

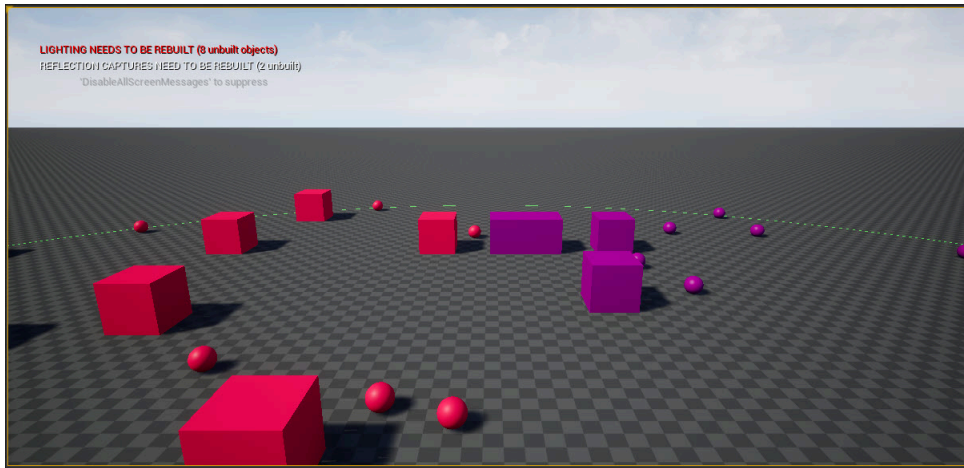
Important

其他版本的Unreal Engine和不.NET受支援，可能會造成問題。

使用範例用Unreal戶端連線至範例應用程式

1. 用Unreal Engine戶端會使用主控台用戶端的NNG程式庫。如果尚未構建控制台客戶端，Windows則必須為其構建。如需詳細資訊，請參閱此頁面上的「Windows主控台用戶端」標籤。
2. 在檔案管理員視窗中，移至`sdk-folder\packaging-tools\clients\PathfindingSampleClients\UnrealClient`。
3. 打開 `UnrealClient.uproject`。
4. 如果編輯器詢問您是否要重建UnrealClient模組，請選擇yes。
5. 在文字編輯器中，開啟`sdk-folder\packaging-tools\clients\PathfindingSampleClients\UnrealClient\view_app_url.txt`。
6. 使用視圖應用程序的 IP 地址和端口號更新 URL：`tcp://ip-address:port-number` (它應該看起來像`tcp://198.51.100.135:1234`)。
7. 在Unreal編輯器中，選擇play。

您的Unreal編輯器應該會顯示模擬的視覺效果，類似於下列螢幕擷取畫面。



Note

視您本機開發系統的功能而定，Unreal編輯器可能需要幾分鐘的時間才能顯示模擬。在此期間，系統可能似乎無回應。

使用W、AS、D鍵在用Unreal戶端中移動。按住滑鼠按鈕並拖曳滑鼠即可轉動。

您可以按下[(左方括號) 鍵來縮小訂閱區域的大小。您可以按] (右方括號) 鍵來增加訂閱區域的大小。訂閱區域的大小決定了顯示在用戶端中的實體數量。

您可以按C鍵以在模擬中建立圖元。客戶端向視圖應用程序發送CreateEntity命令。然後，檢視應用程式將建立實體並將其傳輸到空間網域。

您可以檢查ViewAppDriver::HandleEntityCreationRequests中的代碼，*project-folder*\src\PathfindingSample\ViewApp\Driver\ViewAppDriver.cpp以查看應用程序如何實現此過程。

步驟 6：停止並清理您的模擬

當您不再需要模擬時，清理模擬非常重要。SimSpace Weaver即使您的模擬已停止，模擬資源也會計入您的服務配額 (限制) 中。對於正在執行的模擬，您將繼續收取帳單費用。您也可能會在支援服務 (例如 Amazon CloudWatch Logs 和 Amazon 簡單儲存服務) 中收取資料儲存的帳單費用。如需有關服務配額的詳細資訊，請參閱[SimSpace 韋弗端點和配額](#)。

準備好清理模擬時，請遵循本節中的程序。

⚠ Important

您無法重新啟動已停止的模擬。

⚠ Important

您無法恢復已刪除的模擬。

清理中的模擬資源 SimSpace Weaver

您必須先停止模擬，然後才能刪除模擬。刪除模擬 SimSpace Weaver 僅會移除中的資源。您必須執行單獨的步驟來刪除模擬在其他服務中建立或使用的資源 (請參閱下節)。

Docker

清理模擬的步驟

1. 如果您還不在那裡，請轉到項目和平台的工具文件夾。您 *project-folder* 正在 *path\project-name* 使用您在建立專案時提供的值。

於 Windows 指令提示下，輸入：

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

2. 尋找模擬的名稱。

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

⚠ Important

如果您為 AWS Command Line Interface (AWS CLI) 使用 AWS IAM Identity Center 或命名描述檔，您必須使用 SimSpace Weaver 應用程式 SDK 1.12.1 版或更新版本。最新的版本是 1.16.0。如需 SimSpace Weaver 版本的詳細資訊，請參閱 [SimSpace Weaver 版本](#)。SimSpace Weaver 應用程式 SDK 指令碼會使用 AWS CLI。如果您使用 IAM 身分中心，您可以將您的 IAM 身分中心設定檔複製到您的 default 設定檔，或將 IAM 身分中心設定檔的名稱提供給具有 `--profile cli-profile-name` 參數的 SimSpace Weaver 應用程式 SDK 指令碼。如需詳細資訊，請參閱 [《使 AWS CLI](#)

用[AWS Command Line Interface](#)者指南》AWS IAM Identity Center中的〈[配置使用指南](#)〉和〈[組態和認證檔案設AWS Command Line Interface](#)〉。

3. 停止模擬。

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

4. 刪除已停止的模擬。

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

WSL

Important

為了您的方便，我們提供這些說明。它們適用於搭配使用Windows Subsystem for Linux (WSL)，且不受支援。如需詳細資訊，請參閱 [設定您的本機環境SimSpace Weaver](#)。

清理模擬的步驟

1. 如果您尚未存在，請轉到項目和平台的 tools 文件夾。您 *project-folder* 正在 *path/project-name* 使用創建項目時提供的值。

在Linux薄殼提示下，輸入：

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

2. 尋找模擬的名稱。

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

Important

如果您為 AWS Command Line Interface (AWS CLI) 使用AWS IAM Identity Center 或命名描述檔，您必須使用SimSpace Weaver應用程式 SDK 1.12.1 版或更新版本。最新的版本是 1.16.0。如需 SimSpace Weaver 版本的詳細資訊，請參閱[SimSpace Weaver 版本](#)。SimSpace Weaver應用程式 SDK 指令碼會使用AWS CLI。如果您使用

IAM 身分中心，您可以將您的 IAM 身分中心設定檔複製到您的 default 設定檔，或將 IAM 身分中心設定檔的名稱提供給具有 `--profile cli-profile-name` 參數的 SimSpace Weaver 應用程式 SDK 指令碼。如需詳細資訊，請參閱 [《使 AWS CLI 用 AWS Command Line Interface 者指南》](#) AWS IAM Identity Center 中的 [〈配置使用指南〉](#) 和 [〈組態和認證檔案設定〉](#)。

3. 停止模擬。

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

4. 刪除已停止的模擬。

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

AWS Management Console

清理模擬的步驟

1. 在 SimSpace Weaver 主控台開啟 [SimSpace Weaver 主控台](#)。
2. 在導覽窗格中，選擇 Simulations。
3. 從 Simulations 清單中，選取您要刪除的模擬名稱旁邊的選項。
4. 如果 Status 您選取的模擬為 STARTED：
 - a. 選擇下 Actions 拉式功能表。
 - b. 選擇 Stop。
 - c. 若要確認，請輸入您的模擬名稱。
 - d. 選擇 Stop。
 - e. 等到你 Status 的模擬是 STOPPED。
5. 選擇下 Actions 拉式功能表。
6. 選擇 Delete。
7. 若要確認，請選擇 Delete。

清理支援服務中的模擬資源

為了支援您的模擬，請在其他服務中 SimSpace Weaver 建立資源。SimSpace Weaver 刪除模擬時，不會刪除這些資源。如果您不需要這些額外資源，可以刪除這些資源。

⚠ Important

您可能會針對這些未刪除的任何資源收取帳單費用。

若要刪除專案的支援資源

1. 如果您已完成專案，請刪除其AWS CloudFormation堆疊。如需有關使用的詳細資訊AWS CloudFormation，請參閱《[使用指南](#)》中的〈[刪除AWS CloudFormation主控台上的堆疊AWS CloudFormation](#)〉。
 - 韋弗-###-堆棧

⚠ Important

您從相同專案啟動的模擬會共用資源，例如應用程式角色。當您刪除AWS CloudFormation堆疊時，您會刪除應用程式角色。如果您有其他共用相同資源的模擬，請勿刪除您的AWS CloudFormation堆疊。

📘 Note

您的AWS CloudFormation堆疊可能會報告，DELETE_FAILED因為它無法刪除非空白的Amazon S3 儲存貯體。您將在接下來的步驟中刪除 Amazon S3 儲存貯體。

2. 如果您已完成專案，請刪除其 Amazon S3 儲存貯體。如需使用 Amazon S3 儲存貯體的詳細資訊，請參閱 Amazon 簡單儲存服務使用者指南中的刪除儲存貯體。
 - `weaver-lowercase-project-name-account-number-region`

例如，在「區域」MyProject 中的帳戶111122223333中指定的專案具有下列值us-west-2區：

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

Note

您必須先刪除 Amazon S3 儲存貯體的內容，才能刪除儲存貯體。

Note

SimSpace Weaver 應用程式 SDK 版本 1.12.x 項目為應用程式 .zip 文件和模式使用單獨的存儲桶：

- *weaver-lowercase-project-name-####-#####-##*
- *weaver-lowercase-project-name-####-##-##*

3. 如果您已開啟模擬的記錄功能，請刪除記 CloudWatch 錄檔記錄群組。如需使用 CloudWatch 日誌的詳細資訊，請參閱 Amazon 日誌使用指南中的使用 CloudWatch 日誌群組和日誌[串流](#)。

模擬的記錄群組名稱在其結構描述 (規劃檔) 中指定：*project-folder\tools\project-name.yaml*

記錄群組的名稱為的值 `log_destination_resource_name`。下列結構描述程式碼片段顯示範例應用程式的記錄群組為 `MySimulationLogs`。

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

Warning

如果您啟動多個指定相同記錄群組的模擬，則所有這些模擬的記錄資料都將移至同一個記錄群組。如果刪除記錄群組，則會刪除使用該記錄群組的所有模擬的記錄資料。如果您刪除執行中模擬的記錄群組，則模擬將失敗。

⚠ Important

如果指定了模擬的結構描述，`log_destination_resource_name`但是在 CloudWatch 記錄檔中找不到記錄群組，請確保檢查 `log_destination_service: "logs"` 與執行模擬 AWS 區域相同的記錄群組。

詳細教學課程：在建置範例應用程式時瞭解詳細資料

本自學課程會引導您完成與快速入門教學課程所呈現的相同整體程序，但更詳細。快速入門教程簡化了許多步驟，並在自動化中隱藏了詳細信息。本教程公開並解釋了這些細節。

我們建議您先完成 [快速入門教學課程](#)，然後再完成此自學課程。

要求

開始之前，請務必先完成「」中的步驟 [設定 SimSpace Weaver](#)。

步驟

- [步驟 1：建立專案](#)
- [步驟 2：開啟記錄功能 \(選用\)](#)
- [步驟 3：上傳您的模擬結構描述](#)
- [步驟 4：建立您的專案](#)
- [步驟 5：上傳應用程式](#)
- [步驟 6：開始模擬](#)
- [步驟 7：取得模擬詳細資料](#)
- [步驟 8：啟動自訂應用程式](#)
- [步驟 9：啟動時鐘](#)
- [步驟 10：檢查記錄](#)
- [步驟 11：檢視您的模擬](#)
- [步驟 12：停止並清理您的模擬](#)

步驟 1：建立專案

可散發的SimSpace Weaver應用程式 SDK 包含一個腳本，該腳本從捆綁的項目創建PathfindingSample項目。您必須從指令碼在檔案系統中的位置執行指令碼。指令碼會使用您在命令列中提供的值在`##project-name`內建立路徑。

Docker

建立專案

1. 在Windows指令提示下，變更至您的專案資料夾。

```
cd sdk-folder
```

2. 執行 `create-project.bat` 指令碼。

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

完整的指令和參數清單如下。

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

Important

請勿使用超過 20 個字元的專案名稱。超過此限制可能會導致錯誤。

Note

如果您收到共用磁碟機的提示 Docker，請選擇共 Yes 用磁碟機。

Important

如果您為 AWS Command Line Interface (AWS CLI) 使用 AWS IAM Identity Center 或命名描述檔，您必須使用 SimSpace Weaver 應用程式 SDK 1.12.1 版或更新版本。最新的版本是 1.16.0。如需 SimSpace Weaver 版本的詳細資訊，請參閱 [SimSpace Weaver 版本](#)。SimSpace Weaver 應用程式 SDK 指令碼會使用 AWS CLI。如果您使用 IAM 身分中心，您可以將您的 IAM 身分中心設定檔複製到您的 default 設定

檔，或將 IAM 身分中心設定檔的名稱提供給具有 `--profile cli-profile-name` 參數的 SimSpace Weaver 應用程式 SDK 指令碼。如需詳細資訊，請參閱 [《使 AWS CLI 用 AWS Command Line Interface 者指南》](#) AWS IAM Identity Center 中的 [〈配置使用指南〉](#) 和 [〈組態和認證檔案設 AWS Command Line Interface 定〉](#)。

WSL

Important

為了您的方便，我們提供這些說明。它們適用於搭配使用 Windows Subsystem for Linux (WSL)，且不受支援。如需詳細資訊，請參閱 [設定您的本機環境 SimSpace Weaver](#)。

建立專案

1. 在 Linuxshell 提示符下，切換到您的項目文件夾。

```
cd sdk-folder
```

2. 執行 `create-project.sh` 指令碼。

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

完整的指令和參數清單如下。

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

Important

請勿使用超過 20 個字元的專案名稱。超過此限制可能會導致錯誤。

Important

如果您為 AWS Command Line Interface (AWS CLI) 使用 AWS IAM Identity Center 或命名描述檔，您必須使用 SimSpace Weaver 應用程式 SDK 1.12.1 版或更新版本。

最新的版本是 1.16.0。如需 SimSpace Weaver 版本的詳細資訊，請參閱[SimSpace Weaver 版本](#)。SimSpace Weaver 應用程式 SDK 指令碼會使用 AWS CLI。如果您使用 IAM 身分中心，您可以將您的 IAM 身分中心設定檔複製到您的 default 設定檔，或將 IAM 身分中心設定檔的名稱提供給具有 `--profile cli-profile-name` 參數的 SimSpace Weaver 應用程式 SDK 指令碼。如需詳細資訊，請參閱《[使 AWS CLI 使用 AWS Command Line Interface 者指南](#)》AWS IAM Identity Center 中的〈[配置使用指南](#)〉和〈[組態和認證檔案設定](#)〉。

參數

name

專案的名稱。

path

專案在檔案系統中的位置。專案結構有時可能具有較長的檔案路徑，這些路徑可能會超過作業系統中的路徑長度限制。我們建議您使用盡可能短的路徑名稱。

profile

指令碼應該用於驗證的 AWS CLI 設定檔名稱。如需詳細資訊，請參閱 AWS Command Line Interface 使用指南中的[組態和認證檔案設定](#)。此參數僅適用於 SimSpace Weaver 應用程式 SDK 1.12.1 版或更新版本。如需 SimSpace Weaver 版本的詳細資訊，請參閱 [SimSpace Weaver 版本](#)。

app-sdk-version

(可選) 您的項目使用的 SimSpace Weaver 應用程式 SDK 版本。您將使用此版本構建並鏈接您的應用程式。如果指令碼在可發佈的位置找不到版本，或者您沒有提供版本號碼，指令碼就會自動下載最新版本。

template

(選擇性) 指令碼用來建立專案的專案範本。如果您未提供範本，指令碼會使用 `PathfindingSample`。有效值：

- **PathfindingSample**— 使用單一 [Worker](#) 的範例應用程式。
- **MultiWorkerPathfindingSample**— 使用多個 [Worker](#) 的範例應用程式版本。

overwriteproject

(選擇性) 使用此選項可以使用相同的專案名稱和路徑覆寫既有的專案資料夾。

步驟 2：開啟記錄功能 (選用)

依預設，PathfindingSample專案的記錄處於關閉狀態。本教學課程假設已開啟記錄。您可以選擇開啟記錄功能，但這是選擇性的。

Important

會PathfindingSample產生大量的記錄資料。如果您選擇開啟記錄功能，則會收到記錄資料的帳單費用。只要記錄資料存在，您就會繼續收取帳單費用。如果您打開記錄功能，我們強烈建議您停止此模擬並儘快在自學課程結束時執行清理步驟。

開啟記錄

1. 在文字編輯器中開啟下列檔案：

Docker

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

Note

您的#####*path**project-name*使用您在建立專案時提供的值。

WSL

Important

為了您的方便，我們提供這些說明。它們適用於搭配使用Windows Subsystem for Linux (WSL)，且不受支援。如需詳細資訊，請參閱[設定您的本機環境SimSpace Weaver](#)。

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

Note

您的####*path/project-name*使用您在建立專案時提供的值。

2. 找到simulation_properties:文件開頭的部分：

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

3. 在該行後插入以下兩行simulation_properties:：

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

4. 確認您的simulation_properties:區段與以下內容相同：

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

5. 儲存檔案並結束您的文字編輯器。

步驟 3：上傳您的模擬結構描述

SimSpace Weaver使用資料架構來規劃模擬。結構描述是YAML格式化的純文字檔案。如需詳細資訊，請參閱 [規劃您的模擬](#)。

Docker

範例應用程式隨附預先設定的結構描述。您可以在專案的 tools 資料夾中找到範例應用程式的結構描述檔案：

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

若要上傳結構描述

1. 如果您還不在那裡，請轉到項目和平台的工具文件夾。您*project-folder*正在*path\project-name*使用您在建立專案時提供的值。

於Windows指令提示下，輸入：

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

2. 使用輔助程式指令碼來上傳結構描述。

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

Important

如果您為 AWS Command Line Interface (AWS CLI) 使用AWS IAM Identity Center 或命名描述檔，您必須使用SimSpace Weaver應用程式 SDK 1.12.1 版或更新版本。最新的版本是 1.16.0。如需 SimSpace Weaver 版本的詳細資訊，請參閱[SimSpace Weaver 版本](#)。SimSpace Weaver應用程式 SDK 指令碼會使用AWS CLI。如果您使用 IAM 身分中心，您可以將您的 IAM 身分中心設定檔複製到您的default設定檔，或將 IAM 身分中心設定檔的名稱提供給具有 `--profile cli-profile-name` 參數的SimSpace Weaver應用程式 SDK 指令碼。如需詳細資訊，請參閱《[使AWS CLI 用AWS Command Line Interface者指南](#)》AWS IAM Identity Center中的〈[配置使用指南](#)〉和〈[組態和認證檔案設AWS Command Line Interface定](#)〉。

WSL

Important

為了您的方便，我們提供這些說明。它們適用於搭配使用Windows Subsystem for Linux (WSL)，且不受支援。如需詳細資訊，請參閱 [設定您的本機環境SimSpace Weaver](#)。

範例應用程式隨附預先設定的結構描述。您可以在專案的 `tools` 資料夾中找到範例應用程式的結構描述檔案：

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

若要上傳結構描述

1. 如果您尚未存在，請轉到項目和平台的 tools 文件夾。您 *project-folder* 正在 *path/project-name* 使用創建項目時提供的值。

在Linux薄殼提示下，輸入：

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

2. 使用輔助程式指令碼來上傳結構描述。

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

Important

如果您為 AWS Command Line Interface (AWS CLI) 使用AWS IAM Identity Center 或命名描述檔，您必須使用SimSpace Weaver應用程式 SDK 1.12.1 版或更新版本。最新的版本是 1.16.0。如需 SimSpace Weaver 版本的詳細資訊，請參閱[SimSpace Weaver 版本](#)。SimSpace Weaver應用程式 SDK 指令碼會使用AWS CLI。如果您使用 IAM 身分中心，您可以將您的 IAM 身分中心設定檔複製AWS CLI到您的default設定檔，或將 IAM 身分中心設定檔的名稱提供給具有 `--profile cli-profile-name` 參數的SimSpace Weaver應用程式 SDK 指令碼。如需詳細資訊，請參閱《[使AWS CLI 用AWS Command Line Interface](#)者指南》AWS IAM Identity Center中的〈[配置使用指南](#)〉和〈[組態和認證檔案設AWS Command Line Interface定](#)〉。

步驟 4：建立您的專案

您現在已準備好為範例專案建置空間和自訂應用程式。範例專案包含一個協助程式指令碼，可為您建置這些應用程式。

Docker

該腳本將使用您在設置本地環境時創建的 Docker 映像啟動 Docker 容器。該腳本在碼頭容器中的亞馬遜 Linux 環境中運行構建。它會將構建工件及其依賴項寫入 Windows 中的 *project-folder*\build 文件夾。

若要建立您的專案

1. 如果您還不在那裡，請轉到項目和平台的工具文件夾。您 *project-folder* 正在 *path\project-name* 使用您在建立專案時提供的值。

於Windows指令提示下，輸入：

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

2. 使用輔助腳本來構建項目。

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

WSL

Important

為方便起見，這裡也進行了總結。它們適用於搭配使用Windows Subsystem for Linux (WSL)，且不受支援。如需詳細資訊，請參閱[設定您的本機環境SimSpace Weaver](#)。

該腳本將構建工件及其依賴項寫入您的 *project-folder/build* 文件夾。

若要建立您的專案

1. 如果您尚未存在，請轉到項目和平台的 tools 文件夾。您 *project-folder* 正在 *path/project-name* 使用創建項目時提供的值。

在Linux薄殼提示下，輸入：

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

2. 使用輔助腳本來構建項目。

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

步驟 5：上傳應用程式

建置指令碼會將您的應用程式封裝為 zip 檔案。您必須將這些 zip 檔案上傳到 Amazon 簡易儲存服務中的特定儲存貯體，才能在雲端中執行 SimSpace Weaver 模擬。該 SimSpace Weaver 應用程式 SDK 提供了一個幫助腳本來處理上傳。

Docker

上傳您的應用程式

1. 如果您還不在那裡，請轉到項目和平台的工具文件夾。您 *project-folder* 正在 *path\project-name* 使用您在建立專案時提供的值。

於 Windows 指令提示下，輸入：

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

2. 使用協助程式指令碼上傳您的應用程式。

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

Important

如果您為 AWS Command Line Interface (AWS CLI) 使用 AWS IAM Identity Center 或命名描述檔，您必須使用 SimSpace Weaver 應用程式 SDK 1.12.1 版或更新版本。最新的版本是 1.16.0。如需 SimSpace Weaver 版本的詳細資訊，請參閱 [SimSpace Weaver 版本](#)。SimSpace Weaver 應用程式 SDK 指令碼會使用 AWS CLI。如果您使用 IAM 身分中心，您可以將您的 IAM 身分中心設定檔複製到您的 default 設定檔，或將 IAM 身分中心設定檔的名稱提供給具有 `--profile cli-profile-name` 參數的 SimSpace Weaver 應用程式 SDK 指令碼。如需詳細資訊，請參閱 [《使 AWS CLI 用 AWS Command Line Interface 者指南》](#) AWS IAM Identity Center 中的 [〈配置使用指南〉](#) 和 [〈組態和認證檔案設定 AWS Command Line Interface〉](#)。

WSL

⚠ Important

為了您的方便，我們提供這些說明。它們適用於搭配使用Windows Subsystem for Linux (WSL)，且不受支援。如需詳細資訊，請參閱 [設定您的本機環境SimSpace Weaver](#)。

上傳您的應用程式

1. 如果您尚未存在，請轉到項目和平台的 tools 文件夾。您 *project-folder* 正在 *path/project-name* 使用創建項目時提供的值。

在Linux薄殼提示下，輸入：

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

2. 使用協助程式指令碼上傳您的應用程式。

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

⚠ Important

如果您為 AWS Command Line Interface (AWS CLI) 使用AWS IAM Identity Center 或命名描述檔，您必須使用SimSpace Weaver應用程式 SDK 1.12.1 版或更新版本。最新的版本是 1.16.0。如需 SimSpace Weaver 版本的詳細資訊，請參閱 [SimSpace Weaver 版本](#)。SimSpace Weaver應用程式 SDK 指令碼會使用AWS CLI。如果您使用 IAM 身分中心，您可以將您的 IAM 身分中心設定檔複製AWS CLI到您的default設定檔，或將 IAM 身分中心設定檔的名稱提供給具有 `--profile cli-profile-name` 參數的SimSpace Weaver應用程式 SDK 指令碼。如需詳細資訊，請參閱 [《使AWS CLI 用AWS Command Line Interface者指南》](#) AWS IAM Identity Center中的〈配置使用指南〉和〈組態和認證檔案設AWS Command Line Interface定〉。

檢查您的 Amazon S3 資源

您可以檢查 Amazon S3 儲存貯體，以確保所有上傳都成功。如需使用 Amazon S3 的詳細資訊，請參閱 [Amazon 簡單儲存服務使用者指南中的建立、設定和使用 Amazon S3 儲存貯體](#)。

對於示例應用程序，您的模式（您在上一步中上傳的模式）和應用程序資源使用以下名稱格式：

- 架構值區：simspaceweaver-*project-name-lowercase-account-number-schemas-region*
- 結構描述檔案：*project-name*-schema.yaml
- 應用程序桶：simspaceweaver-*project-name-lowercase-account-number-app-zips-region*
- 空間應用程序：*project-name*Spatial.zip
- 視圖（自定義）應用程序：*project-name*View.zip

例如，給定以下項目屬性：

- 專案名稱：MyProject
- AWS帳號：111122223333
- AWS 區域: us-west-2

模式和應用程序資源將具有以下名稱：

- 架構值區：simspaceweaver-myproject-111122223333-schemas-us-west-2
- 結構描述檔案：MyProject-schema.yaml
- 應用程序桶：simspaceweaver-myproject-111122223333-apps-zips-us-west-2
- 空間應用程序：MyProjectSpatial.zip
- 視圖（自定義）應用程序：MyProjectView.zip

步驟 6：開始模擬

該SimSpace Weaver應用程序 SDK 提供了一個幫助腳本來啟動模擬。該腳本是圍繞 StartSimulationAPI 調用的包裝。它為 API 呼叫提供下列參數：

- 模擬結構描述的名稱（您在先前的步驟中上傳）
- 模擬名稱
- 服SimSpace Weaver務端點

Docker

開始模擬的步驟

1. 如果您還不在那裡，請轉到項目和平台的工具文件夾。您 *project-folder* 正在 *path\project-name* 使用您在建立專案時提供的值。

於Windows指令提示下，輸入：

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

2. 使用輔助程式指令碼上傳開始您的模擬。

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

Important

從版本 1.12.3 開始，start-simulation指令碼會以 1 小時的最長持續時間啟動您的模擬。您可以使用--maximum-duration參數來指定不同的最大持續時間。在版本 1.12.2 或更低版本中，您無法為腳本提供最長持續時間，並且模擬的最長持續時間為 14 天。若要取得有關模擬最長持續時間的更多資訊，請參閱〈[模擬持續時間上限](#)〉。

Important

如果您為 AWS Command Line Interface (AWS CLI) 使用AWS IAM Identity Center 或命名描述檔，您必須使用SimSpace Weaver應用程式 SDK 1.12.1 版或更新版本。最新的版本是 1.16.0。如需 SimSpace Weaver 版本的詳細資訊，請參閱[SimSpace Weaver 版本](#)。SimSpace Weaver應用程式 SDK 指令碼會使用AWS CLI。如果您使用 IAM 身分中心，您可以將您的 IAM 身分中心設定檔複製AWS CLI到您的default設定檔，或將 IAM 身分中心設定檔的名稱提供給具有--profile *cli-profile-name* 參數的SimSpace Weaver應用程式 SDK 指令碼。如需詳細資訊，請參閱《[使AWS CLI 用AWS Command Line Interface](#)者指南》AWS IAM Identity Center中的〈[配置使用指南](#)〉和〈[組態和認證檔案設AWS Command Line Interface定](#)〉。

Note

`run-project-name.bat` 是一個替代輔助腳本，它也將啟動模擬時鐘。在本教程中，您將在稍後的步驟中單獨啟動時鐘。

從版本 1.12.3 開始，`run` 指令碼會以 1 小時的最長持續時間啟動您的模擬。您可以使用 `--maximum-duration` 參數來指定不同的最大持續時間。在版本 1.12.2 或更低版本中，您無法為腳本提供最長持續時間，並且模擬的最長持續時間為 14 天。若要取得有關模擬最長持續時間的更多資訊，請參閱 [〈〉 模擬持續時間上限](#)。

WSL

Important

為了您的方便，我們提供這些說明。它們適用於搭配使用 Windows Subsystem for Linux (WSL)，且不受支援。如需詳細資訊，請參閱 [設定您的本機環境 SimSpace Weaver](#)。

開始模擬的步驟

1. 如果您尚未存在，請轉到項目和平台的 `tools` 文件夾。您 `project-folder` 正在 `path/project-name` 使用創建項目時提供的值。

在 Linux 薄殼提示下，輸入：

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

2. 使用輔助程式指令碼上傳開始您的模擬。

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

Important

從版本 1.12.3 開始，`start-simulation` 指令碼會以 1 小時的最長持續時間啟動您的模擬。您可以使用 `--maximum-duration` 參數來指定不同的最大持續時間。在版本

1.12.2 或更低版本中，您無法為腳本提供最長持續時間，並且模擬的最長持續時間為 14 天。若要取得有關模擬最長持續時間的更多資訊，請參閱 [〈 模擬持續時間上限 〉](#)。

Important

如果您為 AWS Command Line Interface (AWS CLI) 使用 AWS IAM Identity Center 或命名描述檔，您必須使用 SimSpace Weaver 應用程式 SDK 1.12.1 版或更新版本。最新的版本是 1.16.0。如需 SimSpace Weaver 版本的詳細資訊，請參閱 [SimSpace Weaver 版本](#)。SimSpace Weaver 應用程式 SDK 指令碼會使用 AWS CLI。如果您使用 IAM 身分中心，您可以將您的 IAM 身分中心設定檔複製到您的 default 設定檔，或將 IAM 身分中心設定檔的名稱提供給具有 `--profile cli-profile-name` 參數的 SimSpace Weaver 應用程式 SDK 指令碼。如需詳細資訊，請參閱 [《使 AWS CLI 用 AWS Command Line Interface 者指南》](#) AWS IAM Identity Center 中的 [〈配置使用指南〉](#) 和 [〈組態和認證檔案設 AWS Command Line Interface 定〉](#)。

Note

`run-project-name.sh` 是一個替代輔助腳本，它也將啟動模擬時鐘。在本教程中，您將在稍後的步驟中單獨啟動時鐘。

從版本 1.12.3 開始，`run` 指令碼會以 1 小時的最長持續時間啟動您的模擬。您可以使用 `--maximum-duration` 參數來指定不同的最大持續時間。在版本 1.12.2 或更低版本中，您無法為腳本提供最長持續時間，並且模擬的最長持續時間為 14 天。若要取得有關模擬最長持續時間的更多資訊，請參閱 [〈 模擬持續時間上限 〉](#)。

腳本將循環，直到模擬狀態為 STARTED 或 FAILED。模擬可能需要幾分鐘的時間才能開始。如果模擬成功啟動，您將看到類似以下內容的輸出：

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

```
"SchemaS3Location": {
  "ObjectKey": "MyProject-schema.yaml",
  "BucketName": "weaver-myproject-111122223333-us-west-2"
},
```

Note

SimSpace Weaver 應用程式 SDK 版本 1.12.x 項目為應用程式 .zip 文件和模式使用單獨的存儲桶：

- *weaver-lowercase-project-name-####-#####-##*
- *weaver-lowercase-project-name-####-##-##*

步驟 7：取得模擬詳細資料

該 SimSpace Weaver 應用程式 SDK 提供了一個幫助程序腳本，用於包裝 AWS CLI。此指令碼可透 AWS CLI 過提供 SimSpace Weaver 服務端點簡化對的呼叫。您可以使用此協助程式指令碼來呼叫 SimSpace Weaver API。DescribeSimulationAPI 提供有關模擬的詳細資訊，包括其狀態。模擬可以處於以下狀態之一：

模擬生命週期狀態

1. **STARTING**— 打電話後的初始狀態 StartSimulation
2. **STARTED**— 所有空間應用程式都已啟動並且健康
3. **STOPPING**— 打電話後的初始狀態 StopSimulation
4. **STOPPED**— 停止所有運算資源
5. **DELETING**— 打電話後的初始狀態 DeleteSimulation
6. **DELETED**— 將刪除指定給模擬的所有資源
7. **FAILED**— 模擬有一個重大的錯誤/故障和停止
8. **SNAPSHOT_IN_PROGRESS**— 正在進行[快照](#)

Docker

取得模擬詳細資料

1. 如果您還不在那裡，請轉到項目和平台的工具文件夾。您 *project-folder* 正在 *path\project-name* 使用您在建立專案時提供的值。

於Windows指令提示下，輸入：

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

2. 使用 CLI 協助程式指令碼呼叫 ListSimulations API。

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

Important

如果您為 AWS Command Line Interface (AWS CLI) 使用AWS IAM Identity Center 或命名描述檔，您必須使用SimSpace Weaver應用程式 SDK 1.12.1 版或更新版本。最新的版本是 1.16.0。如需 SimSpace Weaver 版本的詳細資訊，請參閱[SimSpace Weaver 版本](#)。SimSpace Weaver應用程式 SDK 指令碼會使用AWS CLI。如果您使用 IAM 身分中心，您可以將您的 IAM 身分中心設定檔複製到您的default設定檔，或將 IAM 身分中心設定檔的名稱提供給具有 `--profile cli-profile-name` 參數的SimSpace Weaver應用程式 SDK 指令碼。如需詳細資訊，請參閱《[使AWS CLI 用AWS Command Line Interface](#)者指南》AWS IAM Identity Center中的〈[配置使用指南](#)〉和〈[組態和認證檔案設AWS Command Line Interface](#)定〉。

該腳本應顯示有關您每個模擬的詳細信息，類似於以下內容：

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

3. 致電DescribeSimulation以獲取您的模擬詳細信息。*simulation-name*用上一步驟Name的輸出中的模擬取代。

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

該腳本應顯示有關您指定的模擬的更多詳細資料，類似於以下內容：

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

```



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

WSL

Important

為了您的方便，我們提供這些說明。它們適用於搭配使用Windows Subsystem for Linux (WSL)，且不受支援。如需詳細資訊，請參閱 [設定您的本機環境SimSpace Weaver](#)。

取得模擬詳細資料

1. 如果您尚未存在，請轉到項目和平台的 tools 文件夾。您 *project-folder* 正在 *path/project-name* 使用創建項目時提供的值。

在Linux薄殼提示下，輸入：

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

2. 使用 CLI 協助程式指令碼呼叫 ListSimulations API。

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

⚠ Important

如果您為 AWS Command Line Interface (AWS CLI) 使用AWS IAM Identity Center 或命名描述檔，您必須使用SimSpace Weaver應用程式 SDK 1.12.1 版或更新版本。最新的版本是 1.16.0。如需 SimSpace Weaver 版本的詳細資訊，請參閱[SimSpace Weaver 版本](#)。SimSpace Weaver應用程式 SDK 指令碼會使用AWS CLI。如果您使用 IAM 身分中心，您可以將您的 IAM 身分中心設定檔複製到您的default設定檔，或將 IAM 身分中心設定檔的名稱提供給具有--profile *cli-profile-name* 參數的SimSpace Weaver應用程式 SDK 指令碼。如需詳細資訊，請參閱《[使AWS CLI 用AWS Command Line Interface](#)者指南》AWS IAM Identity Center中的〈[配置使用指南](#)〉和〈[組態和認證檔案設AWS Command Line Interface定](#)〉。

該腳本應顯示有關您每個模擬的詳細信息，類似於以下內容：

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

3. 致電DescribeSimulation以獲取您的模擬詳細信息。*simulation-name*用上一步驟Name的輸出中的模擬取代。

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

該腳本應顯示有關您指定的模擬的更多詳細資料，類似於以下內容：

```
{
  "Name": "MyProjectSimulation_22-10-04_22_10_15",
  "ExecutionId": "1a2b3c4d-0ab1-1234-567a-12ab34cd5e6f",
}
```

```
"Arn": "arn:aws:simspaceweaver:us-west-2:111122223333:simulation/MyProjectSimulation_22-10-04_22_10_15",
"RoleArn": "arn:aws:iam::111122223333:role/weaver-MyProject-app-role",
"CreationTime": 1664921418.09,
"Status": "STARTED",
"TargetStatus": "STARTED",
"SchemaS3Location": {
  "ObjectKey": "MyProject-schema.yaml",
  "BucketName": "weaver-myproject-111122223333-us-west-2"
},
"SchemaError": "[]",
"LoggingConfiguration": {
  "Destinations": [
    {
      "CloudWatchLogsLogGroup": {
        "LogGroupArn": "arn:aws:logs:us-west-2:111122223333:log-group:MySimulationLogs"
      }
    }
  ]
},
"LiveSimulationState": {
  "Domains": [
    {
      "Type": "",
      "Name": "MySpatialSimulation",
      "Lifecycle": "Unknown"
    },
    {
      "Type": "",
      "Name": "MyViewDomain",
      "Lifecycle": "ByRequest"
    }
  ],
  "Clocks": [
    {
      "Status": "STARTED",
      "TargetStatus": "STARTED"
    }
  ]
},
"MaximumDuration": "1H",
"StartError": "[]"
}
```

步驟 8：啟動自訂應用程式

SimSpace Weaver 不管理自定義應用程式的生命週期。您必須啟動自訂應用程式。最佳做法是在開始模擬時鐘之前啟動自訂應用程式，但是您可以在開始時鐘後啟動自訂應用程式。

您可以使用 CLI 協助程式指令碼呼叫 StartApp API 來啟動自訂應用程式。

Important

如果您為 AWS Command Line Interface (AWS CLI) 使用 AWS IAM Identity Center 或命名描述檔，您必須使用 SimSpace Weaver 應用程式 SDK 1.12.1 版或更新版本。最新的版本是 1.16.0。如需 SimSpace Weaver 版本的詳細資訊，請參閱 [SimSpace Weaver 版本](#)。SimSpace Weaver 應用程式 SDK 指令碼使用 AWS CLI。如果您使用 IAM 身分中心，您可以將您的 IAM 身分中心設定檔複製到 AWS CLI 到您的 default 設定檔，或將 IAM 身分中心設定檔的名稱提供給具有 `--profile cli-profile-name` 參數的 SimSpace Weaver 應用程式 SDK 指令碼。如需詳細資訊，請參閱 [《使 AWS CLI 用 AWS Command Line Interface 者指南》AWS IAM Identity Center 中的〈配置使用指南〉和〈組態和認證檔案設 AWS Command Line Interface 定〉](#)。

Docker

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

WSL

Important

為了您的方便，我們提供這些說明。它們適用於搭配使用 Windows Subsystem for Linux (WSL)，且不受支援。如需詳細資訊，請參閱 [設定您的本機環境 SimSpace Weaver](#)。

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

StartAppAPI 呼叫將使用您提供的名稱建立並啟動自訂應用程式的新執行個體。如果您提供已存在的應用程式的名稱，那麼您將收到一個錯誤。如果您想要重新啟動特定應用程式 (執行個體)，您必須先停止該應用程式並將其刪除。

Note

模擬的狀態必須STARTED在啟動自訂應用程式之前。若要檢查模擬的狀態，請參閱[步驟 7：取得模擬詳細資料](#)。

範例應用程式提供用於檢視模擬的ViewApp自訂應用程式。此應用程式為您提供了一個靜態 IP 地址和端口號來連接模擬客戶端 (您將在本教程的後面步驟中執行此操作)。您可以將 a domain 視為具有相同可執行代碼和啟動選項的一類應用程式。app name識別應用程式的執行個體。如需SimSpace Weaver概念的詳細資訊，請參閱[重要概念SimSpace Weaver](#)。

啟動自訂應用程式後，您可以使用 DescribeApp API 來檢查其狀態。

Docker

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

WSL

Important

為了您的方便，我們提供這些說明。它們適用於搭配使用Windows Subsystem for Linux (WSL)，且不受支援。如需詳細資訊，請參閱[設定您的本機環境SimSpace Weaver](#)。

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

Docker

若要在本教學課程中啟動檢視應用程式

1. 如果您還不在那裡，請轉到項目和平台的工具文件夾。您*project-folder*正在*path\project-name*使用您在建立專案時提供的值。

於Windows指令提示下，輸入：

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

2. 使用 CLI 協助程式指令碼來StartApp呼叫ViewApp。

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

3. 打電話檢DescribeApp查您的自定義應用程序的狀態。

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

WSL

Important

為了您的方便，我們提供這些說明。它們適用於搭配使用Windows Subsystem for Linux (WSL)，且不受支援。如需詳細資訊，請參閱 [設定您的本機環境SimSpace Weaver](#)。

若要在本教學課程中啟動檢視應用程式

1. 如果您尚未存在，請轉到項目和平台的 tools 文件夾。您*project-folder*正在*path/project-name*使用創建項目時提供的值。

在Linux薄殼提示下，輸入：

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

2. 使用 CLI 協助程式指令碼來StartApp呼叫ViewApp。

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

3. 打電話檢DescribeApp查您的自定義應用程序的狀態。

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

在自訂應用程式 (執行個體) 的狀態為之後STARTED，的輸出DescribeApp會包含該自訂應用程式 (執行個體) 的 IP 位址和連接埠號碼。在下列範例輸出中，IP 位址是值Address，連接埠號碼是EndpointInfo區塊Actual中的值。

```
{  
  "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

的值Declared是應用程式程式碼應該繫結到的連接埠號碼。的值Actual是SimSpace Weaver公開給用戶端以連線到您的應用程式的連接埠號碼。 SimSpace Weaver 將Declared連接埠對應至連Actual接埠。

Note

您可以使用[快速入門教學課程中的程序](#)來取得任何已啟動之自訂應用程式的 IP 位址和連接埠號碼，而不受此工作流程影響。

步驟 9：啟動時鐘

當您第一次建立模擬時，它有一個時鐘，但時鐘沒有執行。當您的時鐘未執行時，您的模擬不會更新其狀態。啟動時鐘後，它將開始向您的應用程式發送刻度。每個勾號，您的空間應用程式都會逐步執行它們擁有的實體，並將結果提交給 SimSpace Weaver

Note

啟動時鐘可能需要 30-60 秒。

Important

如果您為 AWS Command Line Interface (AWS CLI) 使用 AWS IAM Identity Center 或命名描述檔，您必須使用 SimSpace Weaver 應用程式 SDK 1.12.1 版或更新版本。最新的版本是 1.16.0。如需 SimSpace Weaver 版本的詳細資訊，請參閱[SimSpace Weaver 版本](#)。SimSpace Weaver 應用程式 SDK 指令碼會使用 AWS CLI。如果您使用 IAM 身分中心，您可以將您的 IAM 身分中心設定檔複製到您的 default 設定檔，或將 IAM 身分中心設定檔的名稱提供給具有 `--profile cli-profile-name` 參數的 SimSpace Weaver 應用程式 SDK 指令碼。如需詳細資訊，請參閱[《使 AWS CLI 用 AWS Command Line Interface 者指南》](#) AWS IAM Identity Center 中的〈配置使用指南〉和〈組態和認證檔案設 AWS Command Line Interface〉。

Docker

啟動時鐘

1. 如果您還不在那裡，請轉到項目和平台的工具文件夾。您 `project-folder` 正在 `path\project-name` 使用您在建立專案時提供的值。

於 Windows 指令提示下，輸入：


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

2. 使用 CLI 協助程式指令碼呼叫 StartClock API。

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

Note

該 StartClock API 使用您的 *simulation-name*，您可以使用 ListSimulations API 找到它：

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

WSL

Important

為了您的方便，我們提供這些說明。它們適用於搭配使用 Windows Subsystem for Linux (WSL)，且不受支援。如需詳細資訊，請參閱 [設定您的本機環境 SimSpace Weaver](#)。

啟動時鐘

1. 如果您尚未存在，請轉到項目和平台的 tools 文件夾。您 *project-folder* 正在 *path/project-name* 使用創建項目時提供的值。

在 Linux 薄殼提示下，輸入：

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

2. 使用 CLI 協助程式指令碼呼叫 StartClock API。

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

Note

該 StartClock API 使用您的 *simulation-name*，您可以使用 ListSimulations API 找到它：

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

步驟 10：檢查記錄

SimSpace Weaver 將模擬管理訊息和主控台輸出從您的應用程式寫入 Amazon CloudWatch 日誌。如需使用日誌的詳細資訊，請參閱 Amazon 日誌使用者指南中的使用日誌群組和 CloudWatch 日誌[串流](#)。

您建立的每個模擬在記錄檔中都有自己的 CloudWatch 記錄群組。記錄群組的名稱是在模擬結構描述中指定的。在下列結構描述程式碼片段中，的值 `log_destination_service` 為 `logs`。這表示的值 `log_destination_resource_name` 是記錄群組的名稱。在此情況下，記錄群組為 `MySimulationLogs`。

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

您也可以在啟動後使用 DescribeSimulation API 來尋找用於模擬的記錄群組名稱。

Important

如果您為 AWS Command Line Interface (AWS CLI) 使用 AWS IAM Identity Center 或命名描述檔，您必須使用 SimSpace Weaver 應用程式 SDK 1.12.1 版或更新版本。最新的版本是 1.16.0。如需 SimSpace Weaver 版本的詳細資訊，請參閱 [SimSpace Weaver 版本](#)。SimSpace Weaver 應用程式 SDK 指令碼會使用 AWS CLI。如果您使用 IAM 身分中心，您可以將您的 IAM 身分中心設定檔複製到您的 default 設定檔，或將 IAM 身分中心設定檔的名稱提供給具有 `--profile cli-profile-name` 參數的 SimSpace Weaver 應用程式 SDK 指令碼。如需詳細資訊，請參閱 [《使 AWS CLI 用 AWS Command Line Interface 者指](#)

南》AWS IAM Identity Center中的〈配置使用指南〉和〈組態和認證檔案設AWS Command Line Interface定〉。

Docker

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

WSL

⚠ Important

為了您的方便，我們提供這些說明。它們適用於搭配使用Windows Subsystem for Linux (WSL)，且不受支援。如需詳細資訊，請參閱 [設定您的本機環境SimSpace Weaver](#)。

```
project-folder/tools/linux/weaver-project-name-cli.sh describe-simulation --simulation simulation-name
```

下列範例會顯示描述記錄組態的輸出DescribeSimulation部分。記錄群組的名稱會顯示在的結尾LogGroupArn。

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

每個模擬記錄群組都包含數個記錄資料流：

- 管理日誌流 — SimSpace Weaver 服務產生的模擬管理消息。

```
/sim/management
```

- 錯誤記錄資料流 — SimSpace Weaver 服務產生的錯誤訊息。只有在發生錯誤時，此日誌流才存在。SimSpace Weaver將您的應用程式寫入的錯誤存儲在其自己的應用程式日誌流中（請參閱以下日誌流）。

```
/sim/errors
```

- 空間應用程式日誌流（每個工作人員上的每個空間應用程式 1 個）-空間應用程式生成的控制台輸出。每個空間應用程式都寫入自己的日誌流。*spatial-app-id*是結尾處結尾斜線之後的所有字元*worker-id*。

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

- 自定義應用程式日誌流（每個自定義應用實例 1 個）-自定義應用程式生成的控制台輸出。每個自定義應用程式實例都寫入自己的日誌流。

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

- 服務應用程式日誌流（每個服務應用程式實例 1 個）-服務應用程式生成的控制台輸出。每個服務應用程式都會寫入自己的記錄資料流。*service-app-id*是結尾處結尾斜線之後的所有字元*service-app-name*。

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

Note

範例應用程式沒有服務應用程式。

步驟 11：檢視您的模擬

SimSpace Weaver應用程式 SDK 提供不同的選項來檢視範例應用程式。如果您沒有任何本地Unreal Engine開發支持，則可以使用示例控制台客戶端。用Unreal Engine戶端的指示假設您正在使用Windows。

主控台用戶端會在實體事件發生時顯示實體事件清單。用戶端會從取得實體事件資訊ViewApp。如果您的主控台用戶端顯示事件清單，則它會確認與模擬中的ViewApp和活動的網路連線。

PathfindingSample模擬會在 2D 平面上建立固定和移動的圖元。移動的圖元會繞著固定的圖元移動。虛幻引擎用戶端會提供實體事件的視覺效果。

Windows console client

要求

- Microsoft Windows 10或更高
- [Microsoft Visual Studio 2019](#)或更新版本，安裝[Desktop development with C++](#)工作負載
- [CMake3](#)
- [Git](#)

使用範例主控台用戶端連線至範例應用程式

1. 在命令提示符窗口中，轉到控制台客戶端的文件夾（在應用程式 SDK 文件夾中）。

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

2. 用CMake3於在此資料夾中建立Visual Studio解決方案。

```
cmake .
```

Note

確保在末尾包括空格和句點。

Important

保持命令提示符窗口打開以進行進一步的步驟。

3. 在中Visual Studio，開啟您PathfindingSampleConsoleClient.sln在上一個步驟中建立的。
4. 選擇構RelWithDebInfo建配置。
5. 選擇「Build>」Build Solution。

- 在之前的命令提示符窗口中，轉到控制台客戶端文件夾中的構建輸出文件夾。

```
cd RelWithDebInfo
```

- 使用您的 IP 地址和端口號運行客戶端ViewApp。

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

您的命令提示字元視窗應顯示實體更新、刪除和建立事件的數字，類似下列範例輸出。

Note

下列範例輸出中的 IP 位址和連接埠號碼為預留位置。將您的 IP 位址和連接埠號碼提供ViewApp給主控台用戶端。如果您要連線至在中執行的連接Actual埠號碼ViewApp，請提供連接埠號碼AWS 雲端。當您連線到本機系統上執行的 IP 位址和連接埠號碼 127.0.0. 1:7000 時，請提供 IP 位ViewApp址和連接埠號碼。如需詳細資訊，請參閱[本機開發](#)。

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

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

```
*****
```

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

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

```
Sample usage with CircleMoveStrategy:
```

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

```
*****
```


```
Starting NNG client. NNG version: 1.2.4
```

```
Creating socket ...done.
```

```
Connecting to View App ... done.
```

```
Initiating connection to tcp:// 198.51.100.135:4321 ... done.
```


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

 Note

如需疑難排解指引，請參閱[PathfindingSample 主控台用戶端無法連線](#)。

8. 按下CTRL+C可結束主控台用戶端。

Linux console client

 Important

為了您的方便，我們提供這些說明。它們可能無法在某些Linux環境中運作。這些程序不受支援。

此程序假設您完全在Linux環境中工作。您也可以使用內建的用戶端來檢視模擬Windows。

要求

- CMake3
- C編譯器 (已包含在中Amazon Linux 2)
- Git

使用範例主控台用戶端連線至範例應用程式

1. 在 Linuxshell 提示符下，轉到控制台客戶端的文件夾 (在應用程式 SDK 文件夾中)。

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

2. 創建一個構建文件夾。


```
mkdir build
```

3. 轉到構建文件夾。

```
cd build
```

4. 用CMake3於建置用戶端。

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


 Note

確保在末尾包括空格和句點。

5. 使用您的 IP 地址和端口號運行客戶端ViewApp。

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

您的命令提示字元視窗應顯示實體更新、刪除和建立事件的數字，類似下列範例輸出。

 Note

下列範例輸出中的 IP 位址和連接埠號碼為預留位置。將您的 IP 位址和連接埠號碼提供ViewApp給主控台用戶端。如果您要連線至在中執行的連接Actual埠號碼ViewApp，請提供連接埠號碼AWS 雲端。當您連線到本機系統上執行的 IP 位址和連接埠號碼 127.0.0. 1:7000 時，請提供 IP 位ViewApp址和連接埠號碼。如需詳細資訊，請參閱[本機開發](#)。

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

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

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

```
*****
```



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

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

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

Note

如需疑難排解指引，請參閱[PathfindingSample 主控台用戶端無法連線](#)。

6. 按下CTRL+C可結束主控台用戶端。

虛擬引擎 on Windows

要求

- Unreal Engine 5開發環境
- Microsoft .NET Framework 4.8 Developer Pack
- Windows主控台用戶端 (請參閱此頁面上的Windows主控台用戶端標籤)

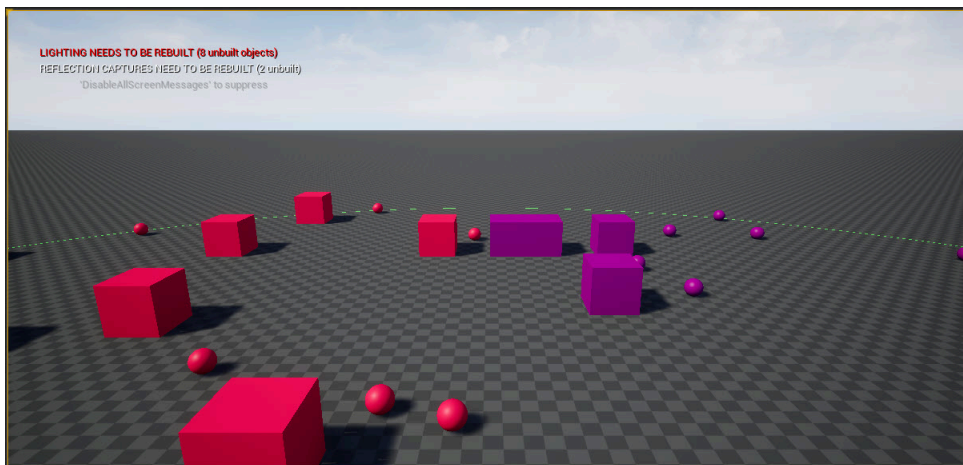
⚠ Important

其他版本的Unreal Engine和不.NET受支援，可能會造成問題。

使用範例用Unreal戶端連線至範例應用程式

1. 用Unreal Engine戶端會使用主控台用戶端的NNG程式庫。如果尚未構建控制台客戶端，Windows則必須為其構建。如需詳細資訊，請參閱此頁面上的「Windows主控台用戶端」標籤。
2. 在檔案管理員視窗中，移至`sdk-folder\packaging-tools\clients\PathfindingSampleClients\UnrealClient`。
3. 打開 `UnrealClient.uproject`。
4. 如果編輯器詢問您是否要重建UnrealClient模組，請選擇yes。
5. 在文字編輯器中，開啟`sdk-folder\packaging-tools\clients\PathfindingSampleClients\UnrealClient\view_app_url.txt`。
6. 使用視圖應用程序的 IP 地址和端口號更新 URL：`tcp://ip-address:port-number`（它應該看起來像`tcp://198.51.100.135:1234`）。
7. 在Unreal編輯器中，選擇play。

您的Unreal編輯器應該會顯示模擬的視覺效果，類似於下列螢幕擷取畫面。



Note

視您本機開發系統的功能而定，Unreal編輯器可能需要幾分鐘的時間才能顯示模擬。在此期間，系統可能似乎無回應。

使用W、AS、D鍵在用Unreal戶端中移動。按住滑鼠按鈕並拖曳滑鼠即可轉動。

您可以按下[(左方括號) 鍵來縮小訂閱區域的大小。您可以按] (右方括號) 鍵來增加訂閱區域的大小。訂閱區域的大小決定了顯示在用戶端中的實體數量。

您可以按C鍵以在模擬中建立圖元。客戶端向視圖應用程序發送CreateEntity命令。然後，檢視應用程式將建立實體並將其傳輸到空間網域。

您可以檢查ViewAppDriver::HandleEntityCreationRequests中的代碼，*project-folder*\src\PathfindingSample\ViewApp\Driver\ViewAppDriver.cpp以查看應用程序如何實現此過程。

Note

如果您不知道 View 應用程序的 IP 地址和端口號，則可以使用[快速入門教程](#)中的過程獲取該信息。

步驟 12：停止並清理您的模擬

當您不再需要模擬時，清理模擬非常重要。SimSpace Weaver即使您的模擬已停止，模擬資源也會計入您的服務配額 (限制) 中。對於正在執行的模擬，您將繼續收取帳單費用。您也可能會在支援服務 (例如 Amazon CloudWatch Logs 和 Amazon 簡單儲存服務) 中收取資料儲存的帳單費用。如需有關服務 SimSpace Weaver 配額的詳細資訊，請參閱[SimSpace 韋弗端點和配額](#)。

準備好清理模擬時，請遵循本節中的程序。

Important

您無法重新啟動已停止的模擬。

⚠ Important

您無法恢復已刪除的模擬。

清理中的模擬資源 SimSpace Weaver

您必須先停止模擬，然後才能刪除模擬。刪除模擬 SimSpace Weaver 僅會移除中的資源。您必須執行單獨的步驟來刪除模擬在其他服務中建立或使用的資源 (請參閱下節)。

Docker

清理模擬的步驟

1. 如果您還不在那裡，請轉到項目和平台的工具文件夾。您 *project-folder* 正在 *path\project-name* 使用您在建立專案時提供的值。

於 Windows 指令提示下，輸入：

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

2. 尋找模擬的名稱。

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

⚠ Important

如果您為 AWS Command Line Interface (AWS CLI) 使用 AWS IAM Identity Center 或命名描述檔，您必須使用 SimSpace Weaver 應用程式 SDK 1.12.1 版或更新版本。最新的版本是 1.16.0。如需 SimSpace Weaver 版本的詳細資訊，請參閱 [SimSpace Weaver 版本](#)。SimSpace Weaver 應用程式 SDK 指令碼會使用 AWS CLI。如果您使用 IAM 身分中心，您可以將您的 IAM 身分中心設定檔複製到您的 default 設定檔，或將 IAM 身分中心設定檔的名稱提供給具有 `--profile cli-profile-name` 參數的 SimSpace Weaver 應用程式 SDK 指令碼。如需詳細資訊，請參閱 [《使 AWS CLI 用 AWS Command Line Interface 者指南》](#) AWS IAM Identity Center 中的〈配置使用指南〉和〈組態和認證檔案設 AWS Command Line Interface 定〉。

3. 停止模擬。

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

4. 刪除已停止的模擬。

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

WSL

Important

為了您的方便，我們提供這些說明。它們適用於搭配使用Windows Subsystem for Linux (WSL)，且不受支援。如需詳細資訊，請參閱 [設定您的本機環境SimSpace Weaver](#)。

清理模擬的步驟

1. 如果您尚未存在，請轉到項目和平台的 tools 文件夾。您 *project-folder* 正在 *path/project-name* 使用創建項目時提供的值。

在Linux薄殼提示下，輸入：

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

2. 尋找模擬的名稱。

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

Important

如果您為 AWS Command Line Interface (AWS CLI) 使用AWS IAM Identity Center 或命名描述檔，您必須使用SimSpace Weaver應用程式 SDK 1.12.1 版或更新版本。最新的版本是 1.16.0。如需 SimSpace Weaver 版本的詳細資訊，請參閱 [SimSpace Weaver 版本](#)。SimSpace Weaver應用程式 SDK 指令碼會使用AWS CLI。如果您使用 IAM 身分中心，您可以將您的 IAM 身分中心設定檔複製到您的default設定檔，或將 IAM 身分中心設定檔的名稱提供給具有 `--profile cli-profile-name` 參數的SimSpace Weaver應用程式 SDK 指令碼。如需詳細資訊，請參閱 [《使AWS CLI](#)

用[AWS Command Line Interface](#)者指南》AWS IAM Identity Center中的〈[配置使用指南](#)〉和〈[組態和認證檔案設AWS Command Line Interface](#)〉。

3. 停止模擬。

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

4. 刪除已停止的模擬。

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

AWS Management Console

清理模擬的步驟

1. 在SimSpace Weaver主控台開啟[SimSpace Weaver主控台](#)。
2. 在導覽窗格中，選擇Simulations。
3. 從Simulations清單中，選取您要刪除的模擬名稱旁邊的選項。
4. 如果Status您選取的模擬為 STARTED：
 - a. 選擇下Actions拉式功能表。
 - b. 選擇 Stop。
 - c. 若要確認，請輸入您的模擬名稱。
 - d. 選擇 Stop。
 - e. 等到你Status的模擬是STOPPED。
5. 選擇下Actions拉式功能表。
6. 選擇 Delete。
7. 若要確認，請選擇Delete。

清理支援服務中的模擬資源

為了支援您的模擬，請在其他服務中SimSpace Weaver建立資源。SimSpace Weaver刪除模擬時，不會刪除這些資源。如果您不需要這些額外資源，可以刪除這些資源。

⚠ Important

您可能會針對這些未刪除的任何資源收取帳單費用。

若要刪除專案的支援資源

1. 如果您已完成專案，請刪除其AWS CloudFormation堆疊。如需有關使用的詳細資訊AWS CloudFormation，請參閱《[使用指南](#)》中的〈[刪除AWS CloudFormation主控台上的堆疊AWS CloudFormation](#)〉。
 - 韋弗-###-堆棧

⚠ Important

您從相同專案啟動的模擬會共用資源，例如應用程式角色。當您刪除AWS CloudFormation堆疊時，您會刪除應用程式角色。如果您有其他共用相同資源的模擬，請勿刪除您的AWS CloudFormation堆疊。

📘 Note

您的AWS CloudFormation堆疊可能會報告，DELETE_FAILED因為它無法刪除非空白的Amazon S3 儲存貯體。您將在接下來的步驟中刪除 Amazon S3 儲存貯體。

2. 如果您已完成專案，請刪除其 Amazon S3 儲存貯體。如需使用 Amazon S3 儲存貯體的詳細資訊，請參閱 Amazon 簡單儲存服務使用者指南中的刪除儲存貯體。
 - `weaver-lowercase-project-name-account-number-region`

例如，在「區域」MyProject 中的帳戶111122223333中指定的專案具有下列值us-west-2區：

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

Note

您必須先刪除 Amazon S3 儲存貯體的內容，才能刪除儲存貯體。

Note

SimSpace Weaver 應用程式 SDK 版本 1.12.x 項目為應用程式 .zip 文件和模式使用單獨的存儲桶：

- *weaver-lowercase-project-name-####-#####-##*
- *weaver-lowercase-project-name-####-##-##*

3. 如果您已開啟模擬的記錄功能，請刪除記 CloudWatch 錄檔記錄群組。如需使用 CloudWatch 日誌的詳細資訊，請參閱 Amazon 日誌使用指南中的使用 CloudWatch 日誌群組和日誌[串流](#)。

模擬的記錄群組名稱在其結構描述 (規劃檔) 中指定：*project-folder\tools\project-name.yaml*

記錄群組的名稱為的值 `log_destination_resource_name`。下列結構描述程式碼片段顯示範例應用程式的記錄群組為 `MySimulationLogs`。

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

Warning

如果您啟動多個指定相同記錄群組的模擬，則所有這些模擬的記錄資料都將移至同一個記錄群組。如果刪除記錄群組，則會刪除使用該記錄群組的所有模擬的記錄資料。如果您刪除執行中模擬的記錄群組，則模擬將失敗。

⚠ Important

如果指定了模擬的結構描述，`log_destination_resource_name`但是在 CloudWatch 記錄檔中找不到記錄群組，請確保檢查 `log_destination_service: "logs"` 與執行模擬 AWS 區域相同的記錄群組。

使用 SimSpace Weaver

本章提供資訊和指導，可協助您在中建置自己的應用程式 SimSpace Weaver。

主題

- [規劃您的模擬](#)
- [模擬持續時間上限](#)
- [開發應用程式](#)
- [開發客戶端應用](#)
- [本機開發](#)
- [AWS SimSpace Weaver 應用程式 SDK](#)
- [AWS SimSpace Weaver 演示框架](#)
- [使用](#)
- [除錯模擬](#)
- [自訂容器](#)
- [使用 Python](#)
- [Support 其他引擎](#)
- [使用授權軟體 AWS SimSpace Weaver](#)
- [管理您的資源 AWS CloudFormation](#)
- [快照](#)
- [簡訊](#)

規劃您的模擬

模擬資料架構 (或資料架構) 是一種 YAML 格式化的文字檔，可指定模擬的規劃。您可以使用相同的資料架構來啟動多個模擬。資料架構檔案位於模擬的專案資料夾中。您可以使用任何文字編輯器來編輯檔案。SimSpace Weaver 僅在啟動模擬時讀取您的資料架構。您對資料架構檔案所做的任何編輯都只會影響您在編輯後開始的新模擬。

Docker

若要規劃模擬，請編輯模擬資料架構檔案：

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

您可以在建立新模擬時上傳模擬資料架構。專案的快速入門輔助程式指令碼會上傳結構描述，作為建立模擬程序的一部分：

```
project-folder\tools\windows\quick-start-project-name-cli.bat
```

如果您沒有使用快速入門腳本來構建模擬，也可以為您的項目使用上傳模擬幫助程序腳本：

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

WSL

Important

為了您的方便，我們提供這些說明。它們適用於搭配使用Windows Subsystem for Linux (WSL)，且不受支援。如需詳細資訊，請參閱 [設定您的本機環境SimSpace Weaver](#)。

您可以透過編輯模擬的模擬資料架構檔來規劃模擬：

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

您可以在建立新模擬時上傳模擬資料架構。專案的快速入門輔助程式指令碼會上傳結構描述，作為建立模擬程序的一部分：

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

如果您沒有使用快速入門腳本來構建模擬，也可以為您的項目使用上傳模擬幫助程序腳本：

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

Important

如果您為 AWS Command Line Interface (AWS CLI) 使用AWS IAM Identity Center或命名描述檔，您必須使用SimSpace Weaver應用程式 SDK 1.12.1 版或更新版本。最新的

版本是 1.16.0。如需 SimSpace Weaver 版本的詳細資訊，請參閱[SimSpace Weaver 版本](#)。SimSpace Weaver 應用程式 SDK 指令碼會使用 AWS CLI。如果您使用 IAM 身分中心，您可以將您的 IAM 身分中心設定檔複製到您的 default 設定檔，或將 IAM 身分中心設定檔的名稱提供給具有 `--profile cli-profile-name` 參數的 SimSpace Weaver 應用程式 SDK 指令碼。如需詳細資訊，請參閱《[使 AWS CLI 用 AWS Command Line Interface 者指南](#)》AWS IAM Identity Center 中的〈[配置使用指南](#)〉和〈[組態和認證檔案設 AWS Command Line Interface](#)〉。

模擬規劃參數

模擬結構描述包含啟動載入資訊，包括：

- 模擬屬性 — SDK 版本和計算配置 ([工作者](#)的類型和數量)
- 時鐘 — 刻度率和容差
- 空間分割策略 — 空間拓撲 (例如網格)、邊界和放置群組 (Worker 上的空間磁碟分割群組)
- 網域及其應用程式 — 應用程式值區、路徑和啟動指令

SimSpace Weaver 使用您的結構描述配置來配置和安排空間分區，啟動應用程序，並以指定的刻度速率推進模擬。

Note

SimSpace Weaver 應用程式 SDK 中的建立專案指令碼會根據範例應用程式自動為您產生模擬結構描述。

以下主題描述模擬資料架構中的參數。若要取得模擬資料架構的完整描述，請參閱〈[SimSpace Weaver 模擬綱要參考](#)〉。

主題

- [SDK 版本](#)
- [模擬性質](#)
- [工人](#)
- [時鐘](#)
- [分割策略](#)

- [網域](#)

SDK 版本

此 `sdk_version` 欄位指定要格式化 SimSpace Weaver 結構描述的版本。有效值：1.16、1.15、1.14、1.13、1.12

Important

的值 `sdk_version` 僅包括主要版本號碼和第一個次要版本號碼。例如，值會 1.12 指定所有版本 1.12.x，例如 1.12.0、1.12.1、和 1.12.2。

模擬性質

該 `simulation_properties` 結構描述的區段會指定實體索引欄位 (通常是空間位置) 的記錄組態和資料類型。

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

的價值 `log_destination_service` 決定的價值的解釋 `log_destination_resource_name`。目前，僅支援的值為 `logs`。這意味著價值 `log_destination_resource_name` 是亞馬遜日誌組的名稱 `CloudWatch` 日誌

Note

記錄是可選的。如果您未設定記錄目的地屬性，則您的模擬將不會產生記錄檔。

需要 `default_entity_index_key_type` 屬性。唯一有效的值為 `Vector3<f32>`。

工人

該 `workers` 區段指定您要用於模擬的 `Worker` 類型和數目。SimSpace Weaver 使用對應至 Amazon EC2 執行個體類型的自己的工作類型。

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

啟用多工作者模擬

您可以建立使用 1 個以上 Worker 的模擬。依預設，模擬使用 1 個 Worker。您必須先修改模擬資料架構，然後才能開始模擬。

Note

您無法變更已啟動的模擬。如果您要為執行中的模擬啟用多工作者，則必須先停止並刪除該模擬。

若要使用多個 Worker，請設定 `desired` 大於 1 的值的運算執行個體數目。每個 Worker 都有最大數量的應用程式。如需詳細資訊，請參閱 [SimSpace 韋弗端點和配額](#)。SimSpace Weaver 當 Worker 上的應用程式數量超過此限制時，只會使用 1 個以上的 Worker。SimSpace Weaver 可以將應用程式放置在任何可用的工作人員上。不能保證在特定 Worker 上放置應用程式。

下列結構描述程式碼片段示範要求 2 個 Worker 之模擬的組態。SimSpace Weaver 如果應用程式數量超過 1 個 Worker 的最大應用程式數量，則將嘗試分配第二個 Worker。

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

時鐘

該 `clock` 區段指定模擬時鐘的性質。目前，您只能設定刻度（時鐘發送給應用程式的每秒刻度數）。價格點率是最大價格。由於價格點的所有操作（例如實體更新）必須在下一個價格點開始之前完成，因此有效價格點價格點可能會更低。價格點率也稱為時脈速率。

的有效值 `tick_rate` 取決於 `sdk_version` 在您的模式中指定。

刻度價的有效值

- 版本早於"1.14":
 - 10
 - 15
 - 30
- 版本"1.14"或更高版本：
 - "10"
 - "15"
 - "30"
 - "unlimited"

如需詳細資訊，請參閱[無限點擊率](#)。

Important

- 對於具有`sdk_version`早於"1.14"的價值`tick_rate`是一個整數，例如30。
- 對於具有`sdk_version`的"1.14"或更高版本的值`tick_rate`是一個串，例如"30"。該值必須包含雙引號。

如果您轉換版本"1.12"或者"1.13"結構描述到版本"1.14"或更高版本，您必須包含的`tick_rate`在雙引號中。

無限點擊率

您可以設定`tick_rate`至"unlimited"，讓您的模擬執行速度與程式碼可以執行的速度一樣快。無限制的價格點擊率，SimSpace Weaver在所有應用程序完成當前刻度的提交後立即發送下一個刻度。

Important

不支援無限價格點擊率SimSpace Weaver 1.14.0 之前的版本。的最小值`sdk_version`在模式中是"1.14"。

無限打勾率SimSpace Weaver Local

SimSpace Weaver Local實現"unlimited"就好像模式指定了 10 千赫 (10000) 的刻度速率一樣。該效果與中的無限價格點擊率相同AWS 雲端。您仍然指定tick_rate: "unlimited"在您的模式中。如需有關 SimSpace Weaver Local 的詳細資訊，請參閱 [本機開發](#)。

關於時鐘的常見問題

Q1. (問題 1) : 我可以更改 START 模擬以使用不同的刻度價格嗎？

您無法變更已存在於AWS 雲端在其生命週期的任何階段。您也無法更改在中運行的模擬的刻度速率 SimSpace Weaver Local。您可以設定tick_rate在資料架構中，並從該資料架構啟動新模擬。

Q2. (問題 2): 我可以在 1.14 之前的版本中以無限制的價格點執行模擬嗎？

不可以，1.14.0 之前的版本不支援無限打勾率。

排解時脈錯誤

如果模擬無法啟動，您可以檢查"StartError"在的輸出DescribeSimulationAPI。一個無效的tick_rate模式中的值會產生以下錯誤。

Note

此處顯示的錯誤輸出顯示在多行上，以提高可讀性。實際的錯誤輸出是單行。

- 該sdk_version早於"1.14"和的價值tick_rate是無效的整數。有效值：10、15、30

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

- 該sdk_version早於"1.14"和的價值tick_rate是一個字符串。有效值：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\"}"]"
```

- 該sdk_version是"1.14"或更高版本和的值tick_rate是無效的字串。有效值："10"、"15"、"30"、"unlimited"

```
"[{"errorType": "SchemaFormatInvalid", "errorMessage":
```



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

- 該`sdk_version`是"1.14"或更高版本和的值`tick_rate`是一個整數。有效值："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\"}]"
```

分割策略

該`partitioning_strategies`部分指定空間應用程式分區的配置屬性。您可以為磁碟分割策略 (本節中的一組屬性) 提供自己的名稱，並在空間應用程式設定中使用它。

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

該`topology`性質可指定模擬使用的座標系統類型。該值`Grid`指定二維 (2D) 格線。

對於一個`Grid`拓撲，模擬空間被建模為軸對齊邊界框(AABB)。您可以為 AABB 的每個軸指定坐標邊界`aabb_bounds`財產。在模擬中空間上存在的所有圖元都必須具有 AABB 內的位置。

網格放置群組

一個放置群組是您想要的空間應用程式分區的集合`SimSpace Weaver`放在同一個工人。您可以指定放置群組的數目和排列方式 (在網格中)`grid_placement_groups`財產。`SimSpace Weaver`將嘗試在放置群組之間平均分配分割區。具有相同置放群組中分割區的空間應用程式擁有權區域將與空間相鄰。

我們建議 $x * y$ 等於您想要的工作人數。如果它不相等，`SimSpace Weaver`將嘗試在可用員工之間平衡您的放置群組。

如果您未指定放置群組設定，SimSpace Weaver將為您計算一個。

網域

您可以為網域的一組組態屬性提供名稱。網域中應用程式的啟動設定會決定網域的類型：

- **launch_apps_via_start_app_call**-自定義域
- **launch_apps_by_partitioning_strategy**— 空間域
- **launch_apps_per_worker**(不包含在範例應用程式中) — 服務網域

Important

SimSpace Weaver每個模擬最多支援 5 個網域。這包括所有空間、自訂和服務網域。

```
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 應用程式 SDK 版本 1.12.x 項目為應用程式 .zip 文件和模式使用單獨的存儲桶：

- 織工 `lowercase-project-name-####-應用程式拉鍊-##`
- 織工 `lowercase-project-name-####-模式-##`

主題

- [應用配置](#)
- [配置空間領域](#)
- [網路端點](#)
- [設定服務網域](#)

應用配置

您指定應用程式的配置 (`app_config`) 做為其網域組態的一部分。所有類型的域都使用這些相同的應用程式配置屬性。

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

Note

SimSpace Weaver 應用程式 SDK 版本 1.12.x 項目為應用程式 .zip 文件和模式使用單獨的存儲桶：

- 織工 `lowercase-project-name-####-應用程式拉鍊-##`
- 織工 `lowercase-project-name-####-模式-##`

該 `package` 屬性指定 S3 存儲桶中壓縮文件的 S3 URI。zip 文件包含應用程序可執行文件 (也稱為二進制) 和它需要的任何其他資源 (例如圖書館)。應用程序可執行文件的每個實例都在 Docker 工人上的容器。

該 `launch_command` 屬性指定可執行文件的名稱和任何命令行選項來運行應用程序。的價值 `launch_command` 是一個數組。整個啟動指令字串的每個標記都是陣列中的元素。

範例

- 對於啟動命令 : `MyTestApp --option1 value1`
- 指定 : `launch_command: ["MyTestApp", "-option1", "value1"]`

該 `required_resource_units` 屬性會指定計算資源單位的數目 SimSpace Weaver 應該分配給這個應用程序。計算資源單位是固定數量的處理能力 (vCPU) 和記憶 (RAM) 在一個工人。您可以增加此值，以增加應用程式在 Worker 上執行時可用的運算能力。每個 Worker 上的運算資源單位數量有限。如需詳細資訊，請參閱 [SimSpace 韋弗端點和配額](#)。

配置空間領域

對於空間網域，您必須指定 `partitioning_strategy`。此屬性的值是您為結構描述的另一部分中定義的分割策略所指定的名稱。

```
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 應用程序 SDK 版本 1.12.x 項目為應用程序 .zip 文件和模式使用單獨的存儲桶：

- 織工 `lowercase-project-name-####-應用程式拉鍊-##`
- 織工 `lowercase-project-name-####-模式-##`

具有一個分區策略Grid拓樸 (此版本支援的唯一拓樸) 指示SimSpace Weaver在網格中安排此域的空間應用程式分區。該`grid_partition`屬性指定分割區網格的列數和欄數。

SimSpace Weaver將為分區網格中的每個單元啟動 1 個空間應用程式實例。例如，如果空間領域具有`grid_partition`值`x: 2`和`y: 2`，空間域中有 $2 * 2 = 4$ 個分區。SimSpace Weaver將啟動空間域中配置的應用程式的 4 個實例，並為每個應用程式實例分配 1 個分區。

主題

- [空間網域的資源需求](#)
- [多個空間領域](#)
- [有關空間領域的常見問題](#)
- [空間域疑難排解](#)

空間網域的資源需求

您最多可以為每個 Worker 指派 17 個運算資源單位。您可以在中指定每個空間應用程式使用的計算資源單位數`app_config`您的空間域的部分。

Example 顯示空間應用程式之計算資源單位的結構描述片段

```
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
```

若要計算網域所需的計算資源單位數，請將網格中的儲存格數目乘以 (在您的`grid_partition,x*y`) 依指派給空間應用程式的計算資源單位數量。

在前面的例子中，域`MySpatialDomain`指定：

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

的網格`MySpatialDomain`有 $2 * 2 = 4$ 個儲存格。空間網域需要 $4 * 1 = 4$ 個計算資源單位。

架構中指定之所有網域的運算資源單位總數必須小於或等於`desired`工作程式數目乘以每個 Worker 的最大計算資源單位數目 (17)。

多個空間領域

您可以將模擬規劃為使用 1 個以上的空間網域。例如，您可以使用 1 個空間網域來控制模擬中的主要演員 (例如人員和汽車) 和不同的空間網域來控制環境。

您還可以使用多個空間領域將不同的資源指定給模擬的不同部分。例如，如果您的模擬具有的實體類型實體比其他類型多 10 倍，您可以建立不同的網域來處理每個實體類型，並為具有更多實體的網域分配更多資源。

Important

SimSpace Weaver 1.14.0 之前的版本不支援多個空間網域。

Important

AWS SimSpace Weaver Local 目前不支援多個空間網域。如需有關 SimSpace Weaver Local 的詳細資訊，請參閱 [本機開發](#)。

Important

SimSpace Weaver 每個模擬最多支援 5 個網域。這包括所有空間、自訂和服務網域。

設定多個空間網域

若要配置一個以上的空間網域，請在結構描述中將其他空間網域定義新增為單獨的具名區段。每個網域都必須指定`launch_apps_by_partitioning_strategy`索引鍵。請參閱下列範例結構描述。

```
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
```

將空間領域放在一起

在某些情況下，您可能會想要將空間網域的分割區放置在 Worker 的另一個網域的磁碟分割旁邊。如果這些分割區彼此建立跨網域訂閱，這可以改善效能特性。

加入頂層索引鍵 `placement_constraints` 到您的模式以指定哪些域 SimSpace Weaver 應該放在一起。所需 `on_workerskey` 必須引用一個命名 `workers` 架構中的組態。

Example 展示放在一起的空間領域的資料架

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

Important

- 如果您使用置放群組：
 - 請確定 $x * y$ 是 Worker 數目的倍數。
 - 請確定放置群組值是您放置在一起之網域之網格維度的通用除數。
- 如果你不要使用放置群組：
 - 請確定空間網域網格的 1 個軸具有等於 Worker 數目的共同除數。

如需置入群組的詳細資訊，請參閱[分割策略](#)。

有關空間領域的常見問題

Q1. (問題 1)：如何將另一個空間領域新增至現有模擬？

- 對於執行模擬— 您無法變更執行中模擬的組態。變更結構描述中的網域組態、上傳結構描述和應用程式 zip，然後開始新的模擬。
- 對於新模擬— 將域配置添加到模式，上傳模式和應用程式 zip，然後啟動新的模擬。

空間域疑難排解

當您嘗試啟動模擬，但您的網域設定無效時，可能會收到下列錯誤。

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

可能的原因

- 結構描述為應用程式配置的運算資源單位數量超過 Worker 上可用的數量。
- SimSpace Weaver無法決定將網域放在 Worker 上的安排。當您指定多個空間網域，但網域網格之間沒有公共除數或多個，例如 2x4 網格和 3x5 網格之間) 時，就會發生這種情況。

網路端點

自訂應用程式和服務應用程式可以具有外部用戶端可連線的網路端點。您可以指定連接埠號碼清單作為下列項目的值 `ingress_ports` 在 `endpoint_config`。這些連接埠號碼都是 TCP 和 UDP。自定義或服務應用程序應綁定到您在 `endpoint_config` 中指定的端口號 `ingress_ports`。SimSpace Weaver 在運行時動態分配端口號，並將這些端口映射到動態端口。您可以呼叫 `describe-app` 您的應用程序開始找到動態 (實際) 端口號後的 API。如需詳細資訊，請參閱 [步驟 4：獲取您的 IP 地址和端口號](#) 從快速入門教程。

```
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 應用程式 SDK 版本 1.12.x 項目為應用程式 .zip 文件和模式使用單獨的存儲桶：

- 織工 `lowercase-project-name-####-應用程式拉鍊-##`
- 織工 `lowercase-project-name-####-模式-##`

Note

`endpoint_config` 是自訂應用程式和服務應用程式的選用屬性。如果您未指定 `endpoint_config` 那麼應用程式將沒有網絡端點。

設定服務網域

的存在 `launch_apps_per_worker`：在域配置中表示它是具有服務應用程式的服務域。SimSpace Weaver 為您啟動和停止服務應用程式。何時 SimSpace Weaver 啟動和停止應用程式，該應用程式被認為具有受管理生命。SimSpace Weaver 目前支持在每個工作人員上啟動 1 或 2 個服務應用程式。

Example 設定為在每個 Worker 上啟動 1 個服務應用程式的網域範例

```
domains:
  MyServiceDomain:
    launch_apps_per_worker:
      count: 1
    app_config:
      package: "s3://example-bucket/PlayerConnectionServiceApp.zip"
      launch_command: ["PlayerConnectionServiceApp"]
      required_resource_units:
        compute: 1
      endpoint_config:
        ingress_ports:
          - 9000
          - 9001
```

Example 設定為在每個 Worker 上啟動 2 個服務應用程式的網域範例

```
domains:
  MyServiceDomain:
    launch_apps_per_worker:
      count: 2
    app_config:
      package: "s3://example-bucket/PlayerConnectionServiceApp.zip"
      launch_command: ["PlayerConnectionServiceApp"]
      required_resource_units:
        compute: 1
      endpoint_config:
        ingress_ports:
          - 9000
          - 9001
```

模擬持續時間上限

中的每個模擬都AWS SimSpace Weaver具有最長持續時間設定，該設定可指定模擬可以執行的最長時間。您可以在啟動模擬時提供最長持續時間做為參數。[StartSimulation](#) [應用程序編程接口 \(API\)](#) 具有可選參數MaximumDuration。參數的值是分鐘數 (m 或 M)、小時 (h 或 H) 或天 (d 或 D)。例如，1h或1H表示 1 小時。SimSpace Weaver當模擬達到此限制時，就會停止模擬。

最大值

的最高有效值MaximumDuration是14D，或以小時 (336H) 或分鐘 (20160M) 為單位的相等值。

預設值

MaximumDuration 為選用參數。如果您未提供值，SimSpace Weaver會使用的值14D。

最小值

的最低有效值MaximumDuration是在數值上等同於的值0。例如，值0M0H0D、和都在數值上等同於0。

如果您提供最大持續時間的最小值，則模擬在到達STOPPING狀態後立即轉換為狀STARTED態。

使用SimSpace Weaver應用程式 SDK 指令碼啟動模擬

當您使用下列其中一個指令碼啟動模擬時，您可以為maximum-duration參數提供值：

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

每個指令碼都會maximum-duration將的值傳遞給StartSimulation API。

Important

如果您未提供的值maximum-duration，請SimSpace Weaver使用[預設值](#) (14D)。

使用控制台啟動模擬

當您在[SimSpace Weaver主控台](#)中啟動模擬時，您可以提供持續時間上限的值。當您選擇「開始模擬」時，在「模擬設定」表單的「最長持續時間」欄位中輸入值。

Important

如果您沒有提供最大持續時間的值，請SimSpace Weaver使用[預設值](#) (14D)。

達到其最長持續時間的模擬狀態

SimSpace Weaver自動停止達到其最長持續時間的模擬時，模擬的狀態為STOPPING (如果正在進行中) 或STOPPED。在[SimSpace Weaver主控台](#)中，模擬的目標狀態仍然是STARTED，因為這是使用者要求的最後一個狀態。

開發應用程式

SimSpace Weaver開發需要一個Amazon Linux 2 (AL2)環境來建置應用程式，因為您的模擬會Amazon Linux在AWS Cloud. 如果您正在使用Windows，則可以使用SimSpace Weaver應用程式 SDK 中的指令碼來建立並啟動Docker容器，以建置SimSpace Weaver應AL2用程式所需的相依性執行。您也可以

使用或使Windows Subsystem for Linux (WSL)用原生AL2系統來啟動AL2環境。如需詳細資訊，請參閱[設定您的本機環境SimSpace Weaver](#)。

Note

無論您如何設定本機開發環境，當Docker您將應用程式上傳到AWS 雲端。您不能直接存取主機作業系統。

應用程式的一般流SimSpace Weaver程

1. 建立 應用程式。
2. 循環：
 - a. 透過建立Transaction.
 - 如果模擬正在關閉，請結束迴路。
 - b. 處理訂閱和擁有權實體事件。
 - c. 更新模擬。
 - d. 提交Transaction以結束更新。
3. 銷毀應用程式。

空間應用

每個空間應用程序都有一個所有權區域，該區域是模擬世界的空間區域。位於空間應用程式擁有權區域中的實體會儲存在應用程式的指派分割區中。單個空間應用程序對其分配的分區內的所有實體具有完整的所有權（讀取和寫入權限）。沒有其他應用程式可以寫入這些實體。空間應用程式會先進其實體的狀態。每個空間應用程序僅擁有 1 個分區。 SimSpace Weaver使用實體的空間位置來索引並將其指派給空間應用程式磁碟分割。

SimSpace Weaver應用程式 SDK 提供範例應用程式。您可以在下列資料夾中找到範例應用程式的空間應用程式的原始程式碼：

Docker

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

WSL

Important

為方便起見，我們提供這些說明。它們適用於搭配使用Windows Subsystem for Linux (WSL)，且不受支援。如需詳細資訊，請參閱[設定您的本機環境SimSpace Weaver](#)。

```
project-folder/src/PathfindingSample/SpatialApp
```

自定義應用

您可以建立並使用自訂應用程式與模擬互動。

自定義應用可以

- 建立實體
- 訂閱其他分割區
- 遞交變更

自訂應用程式的一般流程

1. 建立 應用程式。
2. 訂閱模擬中的特定區域：
 - a. 建立Transaction以開始第一次更新。
 - b. 建立特定區域的訂閱。
 - c. 提交Transaction以結束第一次更新。
3. 循環：
 - a. 創建一個Transaction以開始更新。
 - 如果模擬正在關閉，請結束迴路。
 - b. 處理狀態變更。
 - c. 提交Transaction以結束更新。
4. 銷毀應用程式。

自訂應用程式建立實體之後，它必須將實體轉移至空間網域，以便實體在模擬中空間存在。SimSpace Weaver會使用實體的空間位置，將實體放置在適當的空間應用程式磁碟分割中。建立實體的自訂應用程式將實體傳輸至空間網域後，無法更新或刪除該實體。

SimSpace Weaver應用程式 SDK 提供範例應用程式。您可以使用範例應用程式中包含的自訂應用程式做為自訂應用程式的模型。您可以在以下文件夾中找到示例應用程序的視圖應用程序（自定義應用程序）的源代碼：

Docker

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

WSL

Important

為方便起見，我們提供這些說明。它們適用於搭配使用Windows Subsystem for Linux (WSL)，且不受支援。如需詳細資訊，請參閱 [設定您的本機環境SimSpace Weaver](#)。

```
project-folder/src/PathfindingSample/ViewApp
```

開發客戶端應用

您可能想要將用戶端連線至模擬的一些原因包括：

- 將即時交通資訊注入城市規模模擬中。
- 建立human-in-the-loop模擬，其中人工操作員可以控制模擬的某些方面。
- 讓使用者可以與模擬互動，例如用於訓練模擬。

這些範例中的自訂應用程式充當模擬狀態與外界之間的介面。客戶端連接到自定義應用程序以與模擬進行交互。

SimSpace Weaver不處理客戶端應用程序及其與您的自定義應用程序的通信。您必須負責用戶端應用程式的設計、建立、操作和安全性，以及這些應用程式與自訂應用程式的通訊。SimSpace Weaver僅公開每個自定義應用程序的 IP 地址和端口號，以便客戶端可以連接到它們。

該SimSpace Weaver應用程式 SDK 為其示例應用程式提供客戶端。您可以使用這些用戶端做為自己用戶端應用程式的模型。您可以在下列資料夾中找到範例應用程式用戶端的原始程式碼：

Docker

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

WSL

Important

為方便您使用，這裡也進行了這些說明 它們適用於搭配使用Windows Subsystem for Linux (WSL)，且不受支援。如需詳細資訊，請參閱[設定您的本機環境SimSpace Weaver](#)。

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

若要取得有關建置和使用範例應用程式用戶端[步驟 5：檢視您的模擬](#)的更多資訊，請參閱本指南中的快速入門教學課程。

本機開發

您可以在本機部署SimSpace Weaver應用程式，以便快速測試和偵錯。 SimSpace Weaver LocalMicrosoft Windows僅支援建置。

Important

如需使用和進行開發的Unity資訊Unreal Engine，請參閱[Support 其他引擎](#)。

Important

如果您在版本 1.15.3 中使用 C++、Python、或 UnityUnreal Engine，請參閱[1.15.3 版本中的本機開發差異](#)。

要求

- Microsoft Windows 10或更高
- [Microsoft Visual Studio 2019](#)或更新版本，安裝[Desktop development with C++](#)工作負載

主題

- [建置您的模擬 SimSpace Weaver Local](#)
- [使用執行模擬 SimSpace Weaver Local](#)
- [檢視您的本端模擬](#)
- [停止本端模擬](#)
- [偵錯本機模擬](#)
- [1.15.3 版本中的本機開發差異](#)

建置您的模擬 SimSpace Weaver Local

若要學習使用SimSpace Weaver Local，您可以使用在自學課程期間在雲端中執行的相同路徑尋找範例應用[SimSpace Weaver 入門](#)程式，但這次是在您的本機硬體上。

若要建置範例應用程式 SimSpace Weaver Local

1. 在命令提示字元中，移至`project-folder\tools\local`。
2. 執行 `generate_visual_studio_project.bat`。
3. `project-folder\buildlocal\PathfindingSampleLocal.sln`在視覺工作室中開啟。
4. 將您的構建配置設置為RelWithDebInfo。
5. 選擇「Build>」Build Solution。

Visual Studio會把你的構建工件放在：

- `project-folder\buildlocal\out\RelWithDebInfo`。

在該文件夾中，您應該看到以下可執行文件：

- `PathfindingSampleLocalSpatial.exe`
- `PathfindingSampleLocalView.exe`

使用執行模擬 SimSpace Weaver Local

您可以使 SimSpace Weaver Local 用在本機電腦上執行多達 24 個空間或自訂應用程式的任意組合。模擬時鐘會在結構描述中定義的所有空間應用程式開始後開始。

若要執行您的應用程式 SimSpace Weaver Local

1. 在檔案選擇器視窗中，移至。`project-folder\buildlocal\out\RelWithDebInfo`
2. SimSpace Weaver Local 應用程序需要在應用程序的工作目錄 `schema.yaml` 中命名的模式文件。從結構描述讀取所需資訊的任何失敗都會終止應用程序。

的結構描述 SimSpace Weaver Local 不一定要與相同 `project-folder\tools\project-name-schema.yaml`，但您可以使用它作為起點。

選擇下列其中一項：

- 將該結構描述複製到 `project-folder\buildlocal\out\RelWithDebInfo\schema.yaml`。
- 將環境變數設定 `WEAVERLOCAL_SCHEMA_PATH` 為具有不同路徑或檔案名稱的結構描述檔案名稱。

Example 範例

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

Note

如果您從命令列設定環境變數，則該環境變數 (具有該值) 只能從該命令提示字元工作階段 (主控台視窗) 存取。

3. 範例應用程式的結構描述定義了 2x2 網格，該網格會建立 4 個分割區。您將執行指令碼，以啟動 4 個空間應用程式執行個體，以符合結構描述中指定的空間應用程式數目。該腳本還將啟動 1 個視圖應用程序。在所有空間應用程序啟動並分配一個分區後，模擬將自動開始滴答作響。


啟動您的應用程式

- a. 在命令提示字元中，移至專案的本端工具資料夾。


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

- b. 執行指令碼以啟動應用程式。


```
launch_simulation_locally.bat
```

 Note

如果您設定WEAVERLOCAL_SCHEMA_PATH為結構定義檔案的名稱，則必須在您設定環境變數的同一工作階段 (視窗) 中的指令行啟動空間應用程式。

 Important

如果您看到 Windows 安全性快顯視窗，請選擇以Allow Access便連線至檢視應用程式以視覺化模擬。

 Note

您也可以手動啟動空間並檢視應用程式。為此，您必須手動啟動 4 個空間應用程序實例和一個視圖應用程序。

- 空間應用程序：start PathfindingSampleLocalSpatial.exe
- 查看應用程序：start PathfindingSampleLocalView.exe

檢視您的本端模擬

若要檢視您的本端模擬，您可以使用隨附的任何用戶端 SimSpaceWeaverAppSdkDistributable。若要取得有關建置和使用範例用戶端[步驟 5：檢視您的模擬](#)的更多資訊，請參閱快速入門自學課程的〈〉。

您必須更新用戶端中的 IP 位址和連接埠號碼，才能連線到 View 應用程式以進行本機模擬。始終使用以下值SimSpace Weaver Local：

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

視您選取的用戶端而定，您可以依照下列方式更新 IP 位址和連接埠號碼：

- 虛幻-更改第 1 行的網址 view_app_url.txt

- 主控台 — 以 IP 位址和連接埠號碼 URL 作為參數啟動用戶端

停止本端模擬

如果您的本機空間應用程式處於作用中狀態，則本機模擬將繼續執行。如果您關閉其中一個空間應用程式視窗，您將停止整個模擬。關閉所有其他視窗以清理模擬的其餘部分。

您可以手動關閉每個應用程式窗口，也可以使用以下腳本自動關閉所有窗口：

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

Note

即使關閉一個空間應用程式視窗會停止模擬，也請務必關閉其他應用程式視窗。如果先前模擬仍有開啟的視窗，您將無法成功啟動另一個本端模擬。

偵錯本機模擬

您可以使 SimSpace Weaver Local 用 Microsoft Visual Studio。如需如何使用偵錯的詳細資訊 Visual Studio，請參閱 [Microsoft Visual Studio documentation](#)。

除錯本端模擬的步驟

1. 確保您的位 `schema.yaml` 於工作目錄中。
2. 在中 Visual Studio，開啟您要偵錯之每個應用程式的內容功能表 (例如 `PathfindingSampleLocalSpatial` 或 `PathfindingSampleLocalView`)，並在偵錯區段中設定工作目錄。
3. 開啟您要偵錯之應用程式的右鍵功能表，然後選取 [設定為啟動專案]。
4. 選擇 F5 開始偵錯應用程式。

除錯模擬的需求與正常執行模擬的需求相同。您必須啟動架構中指定的空間應用程式數目。例如，如果您的結構描述指定了 2x2 網格，而您在偵錯模式下啟動空間應用程式，則模擬將不會執行，直到您啟動另外 3 個空間應用程式 (在偵錯模式或未處於偵錯模式)。

若要偵錯自訂應用程式，您必須先啟動空間應用程式，然後在偵錯工具中啟動自訂應用程式。

請注意，您的模擬會在鎖定步驟中執行。一旦應用程式到達斷點，所有其他應用程式將暫停。從該中斷點繼續之後，其他應用程式將繼續進行。

1.15.3 版本中的本機開發差異

本節說明從 1.15.3 版 SimSpace Weaver Local 開始的開發變更。這些變更會影響 C++、Python 和 SimSpace Weaver Local 專案的工作流程 Unreal Engine。Unity

主題

- [對檔案的變更](#)
- [將現有的 C++ 項目更新為 SimSpace Weaver Local 1.15.3](#)
- [在 SimSpace Weaver Local 1.15.3 中運行一個新的 Python 項目](#)
- [將現有的統一專案更新至 SimSpace Weaver Local 1.15.3](#)
- [將現有的虛幻引擎專案更新為 1.15.3 SimSpace Weaver Local](#)
- [關於 SimSpace Weaver Local 1.15.3 的常見問題](#)

對檔案的變更

- SimSpaceWeaverAppSdkLocal 資源庫檔案現在已命名 `weaver_app_sdk_cxx_v1_full_local`。
 - 您可以在中找到這些檔案 `sdk-folder\SimSpaceWeaverAppSdk-1.15.3\lib\weaverlocal\windows`。
- 您的專案不應再連結至或包含 `SimSpaceWeaverAppSDK-1.15.1\include\aws\weaverruntime\local_ffi`。
- Python 腳本 SimSpace Weaver Local 不再需要 `cmake`。
- 有新的或重新命名的 Python 腳本 SimSpace Weaver Local：
 - `build-local`— 建置用於啟動的 Python 應用程式。它將所有 Python 代碼和 SimSpace Weaver Local 庫文件放在 `buildlocal` 目錄中。
 - `local-config`— 定義本機指令碼的環境變數。
 - `start-python-locally`— 設置應用程式 `PythonPath` 並啟動 Python。
 - 下列指令碼相當於具有類似名稱的雲端指令碼：
 - `quick-start-local`
 - `start-simulation-local`
 - `stop-simulation-local`

將現有的 C++ 項目更新為 SimSpace Weaver Local 1.15.3

要求

- [1.15.3 版 SimSpace Weaver 應用程式開發套件](#)
- 適用於 1.15.2 或更早 SimSpace Weaver Local 版本的現有 C++ 專案

若要更新您的專案

1. 確保您的項目包含所需的標題：

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

Note

這通常包含在 `aws\weaverruntime\detail.h`

2. 確保您的項目不包含以下已棄用的標題：

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

Note

這通常包含在 `aws\weaverruntime\detail.h`

3. 在 CMake 文件或構建腳本中，用新名稱替換 `SimSpaceWeaverAppSdkLocal` 靜態庫 `weaver_app_sdk_cxx_v1_full_local` 名稱。
4. 如果您尚未 `sdk-folder\docker-create-image.bat` 從 1.15.3 版執行，請立即執行。你只需要這樣做 1 次。
5. 按照 [常規程序](#) 來構建和運行您的項目。

故障診斷

你得到鏈接器錯誤（未解析的外部符號）

您的編譯器輸出鏈接器錯誤，如下所示（為了可讀性而添加了換行符）。

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

若要修正此問題

- 確保您的項目不包括<aws/weaverruntime/local_ffi/Bridge.h>。

你得到一個類型不兼容錯誤

您可能會收到如下錯誤 (為了便於閱讀而添加了換行符) 。

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

若要修正此問題

1. 使用 1.15.3 版本更新您的 SimSpace Weaver 頭文件。
2. 確保任何使用類似 `rust::cxxbridge1::String` 於以下內容：

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

在 SimSpace Weaver Local 1.15.3 中運行一個新的 Python 項目

要求

- [1.15.3 版 SimSpace Weaver 應用程式開發套件](#)
- 一個新的 SimSpace Weaver Local 版本 1.15.3 PythonBubblesSample 項目

若要執行您的專案

1. 在命令提示字元視窗中，移至 `project-folder\tools\local\windows`。
2. 執行 `quick-start-local.bat`

4 個空間應用程式，1 個視圖應用程式和 1 個客戶端應該在本地啟動。客戶端應該顯示氣泡。

故障診斷

你得到一個錯誤，說GLIBCXX沒有找到

下列錯誤表示匯入 SimSpace Weaver Python 應用程式 SDK 失敗。一個可能的原因是你的 C++ 庫已過時。

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

若要修正此問題

1. 將libstdc++程式庫的來源 (例如gcc或msvc) 更新為包含指定版本的版本 GLIBCXX (在提供的範例中，版本為 3.4.29)。
2. 請確保將系統環境變數設定LD_LIBRARY_PATH為的正確路徑lib64。

快速啟動指令碼失敗

快速啟動指令碼會失敗，並顯示如下訊息。

```
python: can't open file 'C:\usr\project\buildlocal\bIn\bubbles_tkinter_client.py':  
[Errno 2] No such file or directory
```

若要修正此問題

1. 轉到`project-folder\tools\local\windows`並編輯`local-config.bat`。
2. 請確定下列環境變數已針對您的本機系統設定為正確的路徑：
 - TOOLS_DIR設置為您的 `project-folder\tools`
 - TOOLS_DIR_WINDOWS設置為您的 `project-folder\tools\local\windows`
 - PROJECT_ROOT設置為您的 `project-folder`
 - BUILD_DIR設置為您的 `project-folder\buildlocal`
 - APP_SDK_DIR設置為您的 `sdk-folder`

將現有的統一專案更新至 SimSpace Weaver Local 1.15.3

要求

- [1.15.3 版 SimSpace Weaver 應用程式開發套件](#)
- 1.15.2 或更早 SimSpace Weaver Local 版本的現有統一專案

若要更新您的專案

1. 在統一中，移除現有的 AWS SimSpace 韋弗套件。
 - a. 開啟您現有的 Unity 專案。
 - b. 在編輯器視窗中，選擇「視窗 > Package 管理員」。
 - c. 在「套件-統一技術」下，選取 AWS SimSpace Weaver，然後選擇「移除」。
2. 在你的版本 1.15.3 中 *sdk-folder*，運行 `download-unity-package.bat`
3. 請遵循中的指示，`Unity_SDK_for_AWS_SimSpace_Weaver.pdf` 將新下載的檔案新增 `SimSpaceWeaverUnityPackage.zip` 為統一編輯器中的 AWS SimSpace Weaver 套件。

將現有的虛幻引擎專案更新為 1.15.3 SimSpace Weaver Local

要求

- [1.15.3 版 SimSpace Weaver 應用程式開發套件](#)
- 1.15.2 或更早 SimSpace Weaver Local 版本的現有虛幻引擎專案

若要更新您的專案

1. 關閉所有虛幻專案和程式碼編輯器視窗。
2. 在命令提示符窗口中，轉到您的版本 1.15.3 *sdk-folder*。
3. 執行 `update-unreal-project.bat --path project-folder --name project-name`。

Note

這將替換您現有的插件與新的插件。任何修改都會被刪除。

4. 請遵循中的指示AWS_SimSpace_Weaver_Unreal_Guide.pdf來建置您的專案SimSpace Weaver Local。

故障診斷

無法刪除外掛程式檔案

您可能會收到類似下列內容的錯誤。

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

若要修正此問題

1. 確保 Unreal 項目編輯器窗口和代碼編輯器已關閉。
2. 執行 update-unreal-project.bat。

關於 SimSpace Weaver Local 1.15.3 的常見問題

- Q1：如何更改 Python 項目的 SimSpace Weaver Local 環境變量？
 - *project-folder*\tools\windows\local-config 為您的 Python 專案編輯。

AWS SimSpace Weaver 應用程式 SDK

應用 SimSpace Weaver 程式 SDK 提供的 API 可讓您用來控制模擬中的實體並回應 SimSpace Weaver 事件。它包括以下命名空間：

- API — API 的核心定義及其使用

鏈接到以下庫：

- `libweaver_app_sdk_cxx_v1_full.so`

Important

當您在中執行應用程式時，程式庫可用於動態連結AWS 雲端。您無需將其與應用程式一起上傳。

Note

SimSpace Weaver應用程式 SDK API 可控制模擬中的資料。這些 API 與SimSpace Weaver服務 API 分開，它們可以控制您的SimSpace Weaver服務資源（例如模擬，應用程式和時鐘）。AWS如需詳細資訊，請參閱 [SimSpace WeaverAPI 參考資料](#)。

主題

- [API 方法返回一個 Result](#)
- [在頂層與應用程式 SDK 互動](#)
- [模擬管理](#)
- [訂閱](#)
- [實體](#)
- [實體事件](#)
- [Result和錯誤處理](#)
- [泛型和域類型](#)
- [其他應用程式 SDK 作業](#)

API 方法返回一個 Result

大多數 SimSpace Weaver API 函數都具有返回類型`Aws::WeaverRuntime::Result<T>`。如果功能已成功執行，則`Result`包含`T`。否則，`Result`包含代`Aws::WeaverRuntime::ErrorCode`表來自的錯誤代碼Rust App SDK。

Example 範例

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

這種方法：

- Transaction如果成功BeginUpdate()執行，則返回。
- Aws::WeaverRuntime::ErrorCode如果BeginUpdate()失敗，則返回。

在頂層與應用程式 SDK 互動

生命週期

- SimSpace Weaver應用程式 SDK 會管理應用程式生命週期。您不需要讀取或寫入應用程式的生命週期狀態。

資料分割

- 用Result <PartitionSet> AssignedPartitions(Transaction& txn);於取得擁有的分割區。
- 用於Result <PartitionSet> AllPartitions(Transaction& txn);取得模擬中的所有分割區。

模擬管理

本節說明常見模擬管理工作的解決方案。

主題

- [開始模擬](#)
- [更新模擬](#)
- [終止模擬](#)

開始模擬

用CreateApplication()於建立應用程式。

Example 範例

```
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()));

```

更新模擬

使用下列BeginUpdate功能更新應用程式：

- Result<Transaction> BeginUpdate(Application& app)
- Result<bool> BeginUpdateWillBlock(Application& app)-告訴你是否BeginUpdate()會阻止或不阻止。

用Result<void> Commit(Transaction& txn)於提交變更。

Example 範例

```

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();
}

```

終止模擬

用 `Result<void> DestroyApplication(Application&& app)` 於終止應用程式和模擬。

其他應用程式會發現當他們接

到 `BeginUpdateWillBlock()` 或 `BeginUpdate()` 的 `ErrorCode::ShuttingDown` 來電時，模擬正在關閉。當應用程序接收時 `ErrorCode::ShuttingDown`，它可以調 `Result<void> DestroyApplication(Application&& app)` 用終止自己。

Example 範例

```

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

`Result<void> DestroyApplication(Application&& app)` 之後 `Api::Commit()` 才打電話。在更新期間銷毀應用程式可能會導致未定義的行為。

⚠ Important

您必須在程式結束 `DestroyApplication()` 前呼叫，以確保應用程式報告成功終止。程序退出 `DestroyApplication()` 時未能調用將導致狀態被視為 FATAL。

訂閱

您可以使用訂閱區域和網域識別碼建立訂閱。網域 ID 代表擁有該訂閱區域的網域。A `BoundingBox2F32` 描述了訂閱區域。使用下列函數建立訂閱：

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

Example 範例

```
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();
}
```

您可以使用 `Api::SubscriptionHandle` 傳回的 `CreateSubscriptionBoundingBox2F32()` 來修改訂閱。您可以將其作為引數傳遞給以下函數：

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

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

實體

當實體進入應用程式的 `Api::Entity` 訂閱區域時 `CreateEntity()`，您可以使用 `Result<Api::Entity>` 傳回來自或從擁有權變更事件傳回的 Store 和 Load API (如需詳細資訊，請參閱 [實體事件](#))。我們建議您追蹤 `Api::Entity` 物件，以便搭配這些 API 使用它們。

主題

- [建立實體](#)
- [將實體轉移至空間網域](#)
- [寫入和讀取實體欄位資料](#)
- [儲存實體的位置](#)
- [載入圖元的位置](#)

建立實體

用 `CreateEntity()` 於建立實體。您可以定義傳遞 `Api::TypeId` 給此函數的意義。

```

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

`Api::BuiltinTypeId` 會保留的值 0-511。您的實體 `TypeId` (`k_entityTypeId` 在此範例中) 的值必須為 512 或更高。

將實體轉移至空間網域

自訂應用程式或服務應用程式建立實體之後，應用程式必須將實體轉移至空間網域，以便實體在模擬中以空間方式存在。空間網域中的實體可由其他應用程式讀取，並由空間應用程式更新。使用 `ModifyEntityDomain()` API 將實體傳輸至空間網域。

```

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

```

如果與呼叫應用程式 `Partition` 的指派不相符，則 `DomainId` 必須為 `DomainType::SpatialDomain`。 `DomainId` 轉移到新的所有權 `Domain` 會在期間發生 `Commit(Transaction&&)`。

參數

`txn`

目前的 `Transaction`。

entity

變更Entity的目標Domain。

domainId

DomainId的目Domain的地Entity。

Success如果成功變更實體網域，此 API 就會傳回。

寫入和讀取實體欄位資料

所有實體資料欄位都是 blob 類型。您最多可以將 1,024 個位元組的資料寫入實體。我們建議您將斑點盡可能小，因為較大的尺寸會降低效能。當你寫入一個 blob，你傳遞SimSpace Weaver一個指向數據及其長度的指針。當您從 blob 讀取時，會為您SimSpace Weaver提供一個指針和要讀取的長度。所有讀取必須在應用程序調用之前完成Commit()。從讀取呼叫傳回的指標會在應用程式呼Commit()叫時失效。

Important

- 在不支援之後從快取的 Commit() blob 指標讀取，並可能導致模擬失敗。
- 寫入從讀取呼叫傳回的 blob 指標不受支援，而且可能會導致模擬失敗。

主題

- [儲存實體的欄位資料](#)
- [載入實體的欄位資料](#)
- [載入已移除實體的欄位資料](#)

儲存實體的欄位資料

以下示例演示如何存儲 (寫入狀態結構) 應用程序擁有的實體的字段數據。這些範例使用下列函數：

```
AWS_WEAVERRUNTIME_API Result<void> StoreEntityField(  
    Transaction& txn,  
    const Entity& entity,  
    TypeId keyTypeId,  
    FieldIndex index,
```

```
std::int8_t* src,
std::size_t length) noexcept;
```

該 `Api::TypeId keyTypeId` 參數表示傳入的數據的數據類型。

`Api::TypeId keyTypeId` 參數應該接收對應的 `Api::TypeId` 來源 `Api::BuiltinTypeId`。如果沒有適當的轉換，則可以使用 `Api::BuiltinTypeId::Dynamic`。

對於複雜的資料類型，請使用 `Api::BuiltinTypeId::Dynamic`。

Note

的值 `FieldIndex index` 必須大於 0。值 0 會保留給索引鍵使用 (請參閱 `StoreEntityIndexKey()`)。

Example 使用原始數據類型的示例

```
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 使用 a struct 來保存數據的示例

```

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));
}

```

載入實體的欄位資料

下列範例示範如何載入 (從狀態網狀架構讀取) 實體的欄位資料。這些範例使用下列函數：

```

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

```

`Api::TypeId` `keyTypeId` 參數應該接收對應的 `Api::TypeId` 來源 `Api::BuiltinTypeId`。如果沒有適當的轉換，則可以使用 `Api::BuiltinTypeId::Dynamic`。

Note

`FieldIndex` 索引的值必須大於 0。值 0 會保留給索引鍵使用 (請參閱 `StoreEntityIndexKey()`)。

Example 使用原始數據類型的示例

```
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 使用 a struct 來保存數據的示例

```
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);
}

```

載入已移除實體的欄位資料

您無法為已從應用程式擁有權和訂閱區域移除的實體載入 (從 state Fabric 讀取) 實體欄位資料。下列範例會導致錯誤，因為它會呼叫 `Api::LoadIndexKey()` 實體的結果 `Api::ChangeListAction::Remove`。第二個示例顯示了直接在應用程序中存儲和加載實體數據的正確方法。

Example 不正確的程式碼範例

```

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 在應用程序中存儲和加載實體數據的正確方法示例

```

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;
    }
}

/* ... */
}

```

儲存實體的位置

您可以使用整數數據結構存儲 (寫入狀態結構) 實體的位置。這些範例使用下列函數：

```

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

```

Note

您必須提供 `Api::BuiltinTypeId::Vector3F32` 給 `Api::StoreEntityIndexKey()`，如下列範例所示。

Example 使用數組來表示位置的示例

```

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 使用 a struct 來表示位置的示例

```

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));
}

```

載入圖元的位置

您可以使用整數資料結構載入 (從狀態結構讀取) 實體的位置。這些範例使用下列函數：

Note

您必須提供 `Api::BuiltinTypeId::Vector3F32` 給 `Api::LoadEntityIndexKey()`，如下列範例所示。

Example 使用數組來表示位置的示例

```
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 使用 a struct 來表示位置的示例

```
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);  
    }
```

實體事件

您可以在應用SimSpace Weaver程式 SDK 中使用下列功能來取得所有權和訂閱事件：

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

如果您需要回調驅動實體事件處理，則可以使用SimSpace Weaver演示框架。如需詳細資訊，請參閱下列標頭檔案：

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

您也可以建立自己的實體事件處理。

主題

- [遍歷擁有實體的事件](#)
- [迭代已訂閱實體的事件](#)
- [迭代實體的所有權變更事件](#)

遍歷擁有實體的事件

用於`OwnershipChanges()`取得擁有實體 (應用程式擁有權區域中的實體) 的事件清單。該函數具有以下簽名：

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

然後使用迴圈重複執行實體，如下列範例所示。

Example 範例

```
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;
    }
}
```

事件類型

- None— 實體位於區域中，且未修改其位置和欄位資料。
- Remove— 圖元已從區域中移除。
- Add— 圖元已新增至區域。
- Update— 圖元位於區域中且已修改。
- Reject— 應用程式無法從區域移除實體。

Note

在發生Reject事件的情況下，應用程式會在下一個勾選時再次嘗試轉移。

迭代已訂閱實體的事件

用於AllSubscriptionEvents()取得已訂閱實體 (應用程式訂閱區域中的實體) 的事件清單。該函數具有以下簽名：

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

然後使用迴圈重複執行實體，如下列範例所示。

Example 範例

```
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;
  }
}
```

事件類型

- None— 實體位於區域中，且未修改其位置和欄位資料。

- Remove— 圖元已從區域中移除。
- Add— 圖元已新增至區域。
- Update— 圖元位於區域中且已修改。
- Reject— 應用程式無法從區域移除實體。

Note

在發生Reject事件的情況下，應用程式會在下一個勾選時再次嘗試轉移。

迭代實體的所有權變更事件

若要取得實體在擁有權區域與訂閱區域之間移動的事件，請比較目前和先前實體擁有權與訂閱事件之間的變更。

您可以閱讀以下內容來處理這些事件：

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

然後，您可以將變更與先前儲存的資料進行比較。

下列範例顯示如何處理實體擁有權變更事件。此範例假設對於在已訂閱實體與擁有實體 (任一方向) 之間進行轉換的實體，擁有權移除/新增事件會先發生，然後在下一個刻度中接著訂閱移除/新增事件。

Example 範例

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

    static EntityIdsByAction m_entityIdsByPreviousOwnershipAction;

    EntityIdSetByAction entityIdSetByAction;
```

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

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

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

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

EntityIdsByAction entityIdByAction;

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

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

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



```

switch (action)
{
case Api::ChangeListAction::Add:
case Api::ChangeListAction::Remove:
    {
        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和錯誤處理

此 `Aws::WeaverRuntime::Result<T>` 類別使用第三方案 `Outcome` 式庫。您可以使用以下模式來檢查 `Result` 並 `catch` API 調用返回的錯誤。

```

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控制語句宏

在具有返回類型的函數中`Aws::WeaverRuntime::Result<T>`，您可以使用`WEAVERRUNTIME_TRY`宏而不是先前的代碼模式。宏將執行傳遞給它的函數。如果傳遞的函數失敗，宏將使封閉函數返回一個錯誤。如果傳遞的函數成功，執行進展到下一行。下面的例子顯示了以前的`DoBeginUpdate()`功能的重寫。此版本使用`WEAVERRUNTIME_TRY`宏而不是`if-else`制結構。請注意，函數的返回類型是`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();
}

```

如果`BeginUpdate()`失敗，巨集會在失敗的情況下提前`DoBeginUpdate()`返回。您可以使用`WEAVERRUNTIME_EXPECT_ERROR`宏`Aws::WeaverRuntime::ErrorCode`從中獲取`BeginUpdate()`。下列範例顯示`Update()`函式如何呼叫`DoBeginUpdate()`並在失敗時取得錯誤代碼。

```
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);
    }
}
```

您可以將的傳回類型變更為，將錯誤碼從可BeginUpdate()用於呼叫Update()Update()的函數Aws::WeaverRuntime::Result<void>。您可以重複此過程，以繼續將錯誤代碼發送到調用堆棧下方。

泛型和域類型

該SimSpace Weaver應用程式 SDK 提供單精度數據類

型Api::Vector2F32和Api::BoundingBox2F32, 和雙精度Api::Vector2F64和

Api::BoundingBox2F64 這些數據類型是沒有方便的方法被動數據結構。請注意，API 僅使

用Api::Vector2F32和Api::BoundingBox2F32。您可以使用這些資料類型來建立和修改訂閱。

SimSpace Weaver演示框架提供了AzCore數學庫的最小版本，其中包含Vector3和Aabb。如需詳細資訊，請參閱：

- [sdk-folder/packaging-tools/samples/ext/DemoFramework/include/AzCore/Math](#)

其他應用程式 SDK 作業

主題

- [AllSubscriptionEvents並OwnershipChanges包含來自上次呼叫的事件](#)

- [處理後釋放讀取鎖定 SubscriptionChangeList](#)
- [創建一個獨立的應用實例進行測試](#)

AllSubscriptionEvents 並 OwnershipChanges 包含來自上次呼叫的事件

呼叫的傳回值，`Api::AllSubscriptionEvents()` 並 `Api::OwnershipChanges()` 包含來自上次呼叫的事件，而不是最後一次呼叫。在下列範例中，`secondSubscriptionEvents` 且 `secondOwnershipChangeList` 是空的，因為它們的函數會在第一次呼叫之後立即呼叫。

如果您等待 10 個刻度，然後調

用 `Api::AllSubscriptionEvents()` 和 `Api::OwnershipChanges()`，它們的結果將包含事件和來自最後 10 個刻度（不是最後一個刻度）的更改。

Example 範例

```
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

該函數AllSubscriptionEvents()已實現，但該功能SubscriptionEvents()未實現。

處理後釋放讀取鎖定 SubscriptionChangeList

當您開始更新時，其他磁碟分割中的已認可資料會有共用記憶體區段，供上一個刻度使用。這些共享內存段可能被讀取器鎖定。在所有讀者釋放鎖之前，應用程序無法完全提交。作為優化，應用程序應在處理Api::SubscriptionChangeList項目後調用Api::ReleaseReadLeases()以釋放鎖。這樣可以減少提交時的爭用。Api::Commit()依預設會釋出讀取租約，但最佳做法是在處理訂閱更新之後手動釋放它們。

Example 範例

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

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

    WEAVERRUNTIME_EXPECT(Api::ReleaseReadLeases(transaction));

    ...
}
```

創建一個獨立的應用實例進行測試

在實際模擬中執Api::CreateStandaloneApplication()行程式碼之前，您可以使用建立獨立應用程式來測試應用程式邏輯。

Example 範例

```
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 演示框架

AWS SimSpace Weaver 演示框架 (演示框架) 是一個可用於開發 SimSpace Weaver 應用程序的實用程序庫。

該演示框架提供

- 程式碼範例和程式設計模式供您使用和檢查
- 可簡化簡單應用程式開發的抽象和公用程式功能
- 測試 SimSpace Weaver 應用程序 SDK 實驗功能的更簡單方法

我們設計了具有 SimSpace Weaver API 低級訪問權限的 SimSpace Weaver 應用程序 SDK，以提供更高的性能。相較之下，我們設計了示範架構，以提供更高層級的抽象和 API 的存取權，讓您 SimSpace Weaver 更容易使用。與直接使用應用 SimSpace Weaver 程序 SDK 相比，易用性的成本是較低的性能水平。可以容忍較低效能的模擬 (例如那些沒有即時效能需求的模擬) 可能是使用示範架構的最佳選擇。我們建議您針對複雜的應用 SimSpace Weaver 程式使用應用程式 SDK 中的原生功能，因為示範架構不是完整的工具組。

演示框架包括

- 支持和演示的工作代碼示例：
 - 應用流程管理
 - 回呼驅動實體事件處理
- 一組協力廠商公用程式庫：
 - spdlog (一個日誌庫)
 - (數學庫 AZCore) 的最小版本，僅包含：
 - Vector3
 - Aabb

- `cxxopts` (一個命令行選項解析器庫)
- 特定的實用程序功能SimSpace Weaver

演示框架由一個庫，源文件，和CMakeLists. 這些檔案包含在SimSpace Weaver應用程式 SDK 可散發套件中。

使用

本節說明如何使用的SimSpace Weaver。配額也稱為限制。如需服務配額，請參閱[SimSpace 韋弗端點和配額](#)。本節中的 API 來自一組應用程式 API。應用程式 API 與服務 API 不同。應用程式 API 是應用SimSpace Weaver程式 SDK 的一部分。您可以在本機系統的 app SDK 資料夾中找到應用程式 API 的說明文件：

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

主題

- [取得應用程式的限制](#)
- [獲取應用程序使用的資源量](#)
- [重設指標](#)
- [超過](#)
- [記憶體](#)
- [最佳實務](#)

取得應用程式的限制

您可以使用RuntimeLimits應用程式 API 來查詢應用程式的限制。

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

參數

Application應用程式

對應用程式。

LimitType 類型

具有以下限制類型的枚舉：

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

下列範例會查詢實體計數限制。

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

獲取應用程序使用的資源量

您可以調用RuntimeMetrics應用程序 API 以獲取應用程序使用的資源量：

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

參數

Application應用程式

對應用程式。

API 會傳回包含指標的參照。struct計數器量度保持一個運行總值，並且只增加。量規度量保持可以增加或減少的值。每當事件增加值時，應用程式執行階段就會更新計數器。執行階段只會在您呼叫 API 時更新量測計。SimSpace Weaver保證引用在應用程式的生命週期內有效。對 API 的重複調用不會更改引用。

```
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  
}
```

重設指標

ResetRuntimeMetrics 應用程式 API 會重設中的值 AppRuntimeMetrics struct。

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

以下範例示範如何 ResetRuntimeMetrics 在應用程式。

```
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));  
}
```

超過

超過限制的應用程式 API 呼叫將傳回 ErrorCode::CapacityExceeded，實體轉移除外。

SimSpace Weaver 以非同步方式處理實體傳輸，做為 Commit 和 BeginUpdate app API 作業的一部分，因此如果傳輸因實體傳輸限制而失敗，則不會傳回錯誤的特定作業。若要偵測傳輸失敗，您可以將 rejected_incoming_transfer_counter 和 rejected_outgoing_transfer_counter (在

中AppRuntimeMetricsstruct) 的目前值與之前的值進行比較。被拒絕的實體不會在分區中，但應用程序仍然可以模擬它們。

記憶體

SimSpace Weaver使用垃圾收集器進程來清理和釋放釋放的內存。寫入資料的速度可能比記憶體回收器釋放記憶體的速度更快。如果發生這種情況，寫入操作可能會超過應用程序的保留內存限制。SimSpace Weaver將返回一個內部錯誤，其中包含OutOfMemory (和其他詳細信息) 的消息。如需詳細資訊，請參閱[跨時間分散寫入](#)。

最佳實務

以下最佳實務為設計應用程式以避免超過限制。它們可能不適用於您的特定應用程序設計。

頻繁監控並減慢速度

您應該經常監視指標，並減慢接近達到限制的操作速度。

避免超過訂閱限制和轉移限額

如果可能，請設計模擬以減少遠端訂閱和實體傳輸的數量。您可以使用置放群組在同一個 Worker 上放置多個分割區，並減少 Worker 之間進行遠端實體傳輸的需求。

跨時間分散寫入

刻度中的更新數量和大小可能會對確認交易所需的時間和記憶體產生重大影響。記憶體需求較大可能會導致應用程式執行階段記憶體不足。您可以跨時間分散寫入，以降低每個刻度更新的平均總大小。這可以幫助提高性能並避免超出限制。我們建議您在每個刻度上寫入的平均值不要超過 12 MB，或每個實體寫入的平均值不要超過 1.5 KB。

除錯模擬

您可以使用以下方法取得有關模擬的資訊。

主題

- [使用SimSpace Weaver Local並查看控制台輸出](#)
- [在 Amazon 日誌中查看您的 CloudWatch 日誌](#)
- [使用 describe API 呼叫](#)

- [Connect 用戶端](#)

使用SimSpace Weaver Local並查看控制台輸出

我們建議您先在本端開發模擬，然後在中執行它們。AWS 雲端當您使用執行時，您可以直接檢視主控台輸出SimSpace Weaver Local。如需詳細資訊，請參閱 [本機開發](#)。

在 Amazon 日誌中查看您的 CloudWatch 日誌

當您在應用程式的主控制台輸出中執行模擬時，會傳送到 Amazon CloudWatch Logs 中的日誌串流。AWS 雲端您的模擬也會寫入其他記錄資料。如果您希望模擬寫入記錄資料，則必須在模擬資料架構中啟用記錄。如需詳細資訊，請參閱 [SimSpace WeaverAmazon 日誌中的 CloudWatch 日誌](#)。

Warning

您的模擬可以產生大量的記錄資料。日誌數據可以非常快速地增長。您應該密切關注您的日誌，並在不再需要它們運行時停止模擬。記錄可能會產生很大的成本。

使用 describe API 呼叫

您可以使用下列服務 API 取得中模擬的相關資訊。AWS 雲端

Important

如果您為 AWS Command Line Interface (AWS CLI) 使用AWS IAM Identity Center或命名描述檔，您必須使用SimSpace Weaver應用程式 SDK 1.12.1 版或更新版本。最新的版本是 1.16.0。如需 SimSpace Weaver 版本的詳細資訊，請參閱[SimSpace Weaver 版本](#)。SimSpace Weaver應用程式 SDK 指令碼會使用AWS CLI。如果您使用 IAM 身分中心，您可以將您的 IAM 身分中心設定檔複製AWS CLI到您的default設定檔，或將 IAM 身分中心設定檔的名稱提供給具有--profile *cli-profile-name*參數的SimSpace Weaver應用程式 SDK 指令碼。如需詳細資訊，請參閱《[使AWS CLI用AWS Command Line Interface者指南](#)》AWS IAM Identity Center中的〈[配置使用指南](#)〉和〈[組態和認證檔案設AWS Command Line Interface定](#)〉。

- ListSimulations— 獲取您的所有模擬的列表。AWS 雲端

Example 範例

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

- DescribeSimulation— 取得有關模擬的詳細資訊。

Example 範例

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

- DescribeApp— 獲取有關應用程式的詳細信息。

Example 範例

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

如需 SimSpace Weaver API 的詳細資訊，請參閱[SimSpace Weaver API 參考資料](#)。

Connect 用戶端

您可以將用戶端連接至您在模擬結構描述中使用定義的執行 `endpoint_config` 中自訂或服務應用程式。SimSpace Weaver 應用程式 SDK 包含可用來檢視範例應用程式的範例用戶端。您可以查看這些示例客戶端的源代碼和示例應用程序，以了解如何創建自己的客戶端。如需如何建置和執行範例用戶端的詳細資訊，請參閱[步驟 5：檢視您的模擬](#)。

您可以在下列資料夾中找到範例用戶端的原始程式碼：

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

偵錯本機模擬

您可以使 SimSpace Weaver Local 用 Microsoft Visual Studio。如需如何使用偵錯的詳細資訊 Visual Studio，請參閱 [Microsoft Visual Studio documentation](#)。

除錯本端模擬的步驟

1. 確保您的位 `schema.yaml` 於工作目錄中。

2. 在中 Visual Studio，開啟您要偵錯之每個應用程式的內容功能表 (例如 PathfindingSampleLocalSpatial 或 PathfindingSampleLocalView)，並在偵錯區段中設定工作目錄。
3. 開啟您要偵錯之應用程式的右鍵功能表，然後選取 [設定為啟動專案]。
4. 選擇 F5 開始偵錯應用程式。

除錯模擬的需求與正常執行模擬的需求相同。您必須啟動架構中指定的空間應用程式數目。例如，如果您的結構描述指定了 2x2 網格，而您在偵錯模式下啟動空間應用程式，則模擬將不會執行，直到您啟動另外 3 個空間應用程式 (在偵錯模式或未處於偵錯模式)。

若要偵錯自訂應用程式，您必須先啟動空間應用程式，然後在偵錯工具中啟動自訂應用程式。

請注意，您的模擬會在鎖定步驟中執行。一旦應用程序到達斷點，所有其他應用程序將暫停。從該中斷點繼續之後，其他應用程式將繼續進行。

自訂容器

AWS SimSpace Weaver 應用程式在容器化 Amazon Linux 2 (AL2) 環境中執行。在中 AWS 雲端，SimSpace Weaver 在從 Amazon Elastic Container Registry (Amazon ECR) 提供的 `amazonlinux:2` 映像構建的 Docker 容器中運行模擬。您可以建立自訂 Docker 映像，將其存放在 Amazon ECR 中，然後將該映像用於模擬，而不是我們提供的預設 Docker 映像。

您可以使用自訂容器來管理軟體相依性，並包含不在標準 Docker 映像檔中的其他軟體元件。例如，您可以將應用程序使用的公開可用軟件庫添加到容器中，並且只將自定義代碼放入應用程序 zip 文件中。

Important

我們僅支援在 Amazon ECR 公共圖庫或您的私人 Amazon ECR 註冊表託管在 Amazon ECR 存儲庫中託管的 AL2 Docker 映像。我們不支援在 Amazon ECR 以外託管的 Docker 圖片。如需 Amazon ECR 的詳細資訊，請參閱 [Amazon 彈性容器登錄文件](#)。

主題

- [建立自訂容器](#)
- [修改專案以使用自訂容器](#)
- [關於自訂容器的常見問題集](#)
- [排解自訂容器](#)

建立自訂容器

這些說明假設您知道如何使用碼頭和 Amazon Elastic Container Registry (Amazon ECR)。如需 Amazon ECR 的詳細資訊，請參閱 [《Amazon ECR 使用者指南》](#)。

必要條件

- 您用來執行這些動作的 IAM 身分 (使用或角色) 具有使用 Amazon ECR 的正確許可
- 碼頭工具已安裝在您的本地系統上

建立自訂容器

1. 創建您的Dockerfile.

A 運Dockerfile行AWS SimSpace Weaver應用程序從 Amazon ECR 中的Amazon Linux 2圖像開始。

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

2. 建立您的Dockerfile.

3. 將您的容器映像檔上傳到 Amazon ECR。

- [使用AWS Management Console.](#)
- [使用AWS Command Line Interface.](#)

Note

如果您在嘗試將容器映像上傳到 Amazon ECR 時收到AccessDeniedException錯誤訊息，您的 IAM 身分 (使用者或角色) 可能沒有使用 Amazon ECR 的必要許可。您可以將受AmazonEC2ContainerRegistryPowerUserAWS管政策附加到 IAM 身分，然後再試一次。如需有關如何附加政策的詳細資訊，請參閱AWS Identity and Access Management使用指南中的[新增和移除 IAM 身分許可](#)。

修改專案以使用自訂容器

這些指示假設您已經知道如何使用AWS SimSpace Weaver並希望讓您的應用程式儲存和開發工作流程AWS 雲端更有效率。

必要條件

- 您正在修改由create-project.bat指令碼建立的現有SimSpace Weaver專案。
- 您在 Amazon Elastic Container Registry (Amazon ECR) 中有一個自定義容器。如需建立自訂容器的詳細資訊，請參閱[建立自訂容器](#)。

若要修改專案以使用自訂容器

1. 為專案的模擬應用程式角色新增許可以使用 Amazon ECR。
 - a. 如果您還沒有具有以下許可的 IAM 政策，請建立該政策。我們建議使用策略名稱simspaceweaver-ecr。如需如何建立 IAM 政策的詳細資訊，請參閱AWS Identity and Access Management使用者指南中的[建立 IAM 政策](#)。

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

- b. 尋找專案模擬應用程式角色的名稱：
 - i. 在文字編輯器中，開啟專案的AWS CloudFormation樣板：

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

- ii. 尋找下方的RoleName屬性WeaverAppRole。此值是專案模擬應用程式角色的名稱。

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. 將simspaceweaver-ecr原則附加至專案的模擬應用程式角色。如需有關如何附加政策的詳細資訊，請參閱AWS Identity and Access Management使用指南中的[新增和移除 IAM 身分許可](#)。

2. 在專案的模擬結構描述中指定容器映像檔。

- 您可以在下新增選擇default_image性屬性，simulation_properties以指定所有網域的預設自訂容器映像檔。
- 在您要使用app_config自訂容器映像檔的網域中新增image屬性。指定 Amazon ECR 儲存庫 URI 做為值。您可以為每個網域指定不同的映像檔。
- 如果image未為網域指定且已指定，default_image則該網域中的應用程式會使用預設映像檔。
- 如果image未為網域指定且default_image未指定，則該網域中的應用程式會在標準 SimSpace Weaver 容器中執行。

Example 包含自訂容器設定的結構描述片段

```
sdk_version: "1.16.0"
simulation_properties:
  log_destination_service: "logs"
  log_destination_resource_name: "MySimulationLogs"
  default_entity_index_key_type: "Vector3<f32>"
  default_image: "111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest" # image to use if no image specified for a domain
domains:
  MyCustomDomain:
    launch_apps_via_start_app_call: {}
    app_config:
```

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

3. 像往常一樣構建和上傳您的項目。

關於自訂容器的常見問題集

Q1. (問題 1)：如果我想更改容器的內容，我該怎麼辦？

- 對於執行中模擬 — 您無法變更執行中模擬的容器。您必須建置新容器並啟動使用該容器的新模擬。
- 對於新模擬 — 建立新容器，將其上傳到 Amazon Elastic Container Registry (Amazon ECR)，然後開始使用該容器的新模擬。

Q2. (問題 2)：如何變更模擬的容器映像？

- 對於執行中模擬 — 您無法變更執行中模擬的容器。您必須啟動使用新容器的新模擬。
- 對於新模擬 — 在專案的模擬結構描述中指定新容器映像檔。如需詳細資訊，請參閱 [修改專案以使用自訂容器](#)。

排解自訂容器

主題

- [AccessDeniedException 將圖像上傳到 Amazon Elastic Container Registry \(Amazon ECR\) 時](#)
- [使用自訂容器的模擬無法啟動](#)

AccessDeniedException 將圖像上傳到 Amazon Elastic Container Registry (Amazon ECR) 時

如果您在嘗試將容器映像上傳到 Amazon ECR 時收到 AccessDeniedException 錯誤訊息，您的 IAM 身分 (使用者或角色) 可能沒有使用 Amazon ECR 的必要許可。您可以將受 AmazonEC2ContainerRegistryPowerUserAWS 管政策附加到 IAM 身分，然後再試一次。如需有關如何附加政策的詳細資訊，請參閱 AWS Identity and Access Management 使用指南中的 [新增和移除 IAM 身分許可](#)。

使用自訂容器的模擬無法啟動

對秘訣進行故障診斷

- 如果您的模擬已啟用記錄，請檢查錯誤記錄。如需詳細資訊，請參閱 [詳細的教學課程](#)。
- 在不使用自訂容器的情況下測試模擬。
- 在本端測試模擬。如需更多詳細資訊，請參閱 [本機開發](#)。

使用 Python

您可以使用 Python 為您的 SimSpace Weaver 應用程式和用戶端。Python 軟體開發工具包 (Python SDK) 作為標準的一部分 SimSpace Weaver 應用程式 SDK 可散發套件。使用 Python 進行開發的工作方式與其他支持的語言進行開發的方式相似。

Important

SimSpace Weaver 只支持 3.9 版本。

Important

SimSpace Weaver對 Python 的支持需要SimSpace Weaver版本 1.15.0 或更高版本。

主題

- [創建一個項目](#)
- [開始模擬](#)
- [示例 Python 客戶端](#)
- [編寫您自己的構建腳本](#)
- [關於使用 Python 的常見問題](#)
- [疑難排解與 Python 相關的問題](#)

創建一個項目

您使用create-project.bat腳本來創建一個 Python 項目，與創建一個非 Python 項目相同的方式。您可以使用PythonBubblesSample模板作為您的 Python 項目的起點。請參閱[創建一個蟒蛇項目](#)，下面。

蟒蛇自定義容器

運行基於 Python 的步驟SimSpace Weaver模擬中的AWS 雲端，您可以建立包含必要相依性的自訂容器。如需詳細資訊，請參閱[自訂容器](#)。

Python 自定義容器必須包含以下內容：

- 海合作
- 開放式開放
- bzip2-開放
- 利比菲-德维尔
- wget
- tar
- gzip
- make
- 蟒蛇 (版本 3.9)

如果您使用PythonBubblesSample模板來創建您的項目，您可以運行create-custom-container.bat腳本 (位於tools您的項目文件夾) 創建具有必要依賴關係的 Docker 映像。該腳本將圖像上傳到亞馬遜彈性容器註冊表 (亞馬遜 ECR)。

該create-custom-container.bat腳本使用以下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
```

您可以將自己的依賴項添加到Dockerfile:

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

該requirements.txt檔案包含所需的 Python 套件清單PythonBubblesSample樣本模擬：

```
Flask==2.1.1
```

您可以將自己的 Python 套件相依性新增至requirements.txt:

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

該Dockerfile和requirements.txt在tools您的項目的文件夾。

Important

你必須跑create-custom-container.bat在任何變更之後Dockerfile或者requirements.txt。

⚠ Important

從技術上講，您不必在 Python 模擬中使用自定義容器，但我們強烈建議您使用自定義容器。我們提供的標準亞馬遜 Linux 2 (AL2) 容器沒有 Python。因此，如果您不使用具有 Python 的自定義容器 (例如由 `create-custom-container.bat` 腳本)，您必須在您上傳到的每個應用程序 zip 文件中包含 Python 和所需的依賴關係 SimSpace Weaver。

創建一個蟒蛇項目

下列程序適用於微軟視窗。如果您正在使用適用於 Linux 的視窗子系統 (WSL)，請使用 `.sh` 的版本 `.bat` 腳本代替。您必須完成亞馬遜彈性容器登錄 (Amazon ECR) 的設定，才能使用此程序。如需詳細資訊，請參閱 [使用亞馬遜 ECR 進行設置](#) 在亞馬遜 ECR 用戶指南。

若要建立專案

1. 在一個命令提示符窗口，轉到您的 SimSpace Weaver SDK 資料夾。

```
cd sdk-folder
```

2. 執行 `create-project.bat` 與 `PythonBubblesSample` 範本。

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

3. 前往 `tools` 專案資料夾中的資料夾。您的 `#####` 是 `project-folder-parent-path\project-name`。

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

4. 建立自訂容器。

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

開始模擬

您可以使用與普通模擬相同的方式開始基於 Python 的模擬 SimSpace Weaver 模擬，兩者都在 SimSpace Weaver Local 並在 SimSpace Weaver 在 AWS 雲端。如需詳細資訊，請參閱下列內容：

SimSpace Weaver Local

- [本機開發](#)

AWS 雲端

- [步驟 3：執行快速啟動指令碼](#)在快速入門教學課程中
- [詳細教學課程：在建置範例應用程式時瞭解詳細資料](#)

該PythonBubblesSample包括它自己的 Python 示例客戶端。如需詳細資訊，請參閱[示例 Python 客戶端](#)。

示例 Python 客戶端

如果您使用PythonBubblesSample模板來創建一個項目，然後你的項目包含一個 Python 示例客戶端。您可以使用範例用戶端來檢視PythonBubblesSample模擬。您也可以使用範例用戶端做為起點來建立自己的 Python 用戶端。

下列程序假設您已建立PythonBubblesSample投影並開始其模擬。

若要啟動用戶端

1. 在一個命令提示符窗口，轉到src\PythonBubblesSample\bin您的項目的文件夾。

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

2. 運行客戶端。

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

參數

host

模擬的 IP 位址。對於在中開始的模擬AWS 雲端，您可以在[SimSpace Weaver安慰](#)或使用程序[步驟 4：獲取您的 IP 地址和端口號](#)的快速入門教程。對於本端模擬，請使用127.0.0.1作為 IP 地址。

port

模擬的連接埠號碼。對於在中開始的模擬AWS 雲端，這就是Actual連接埠號碼。您可以在[SimSpace Weaver](#) [安慰](#) 或使用程序 [步驟 4：獲取您的 IP 地址和端口號](#) 的快速入門教程。對於本端模擬，請使用7000作為連接埠號碼。

simszize

要在用戶端中顯示的實體數目上限。

編寫您自己的構建腳本

您可以為 Python 模擬編寫自己的構建腳本。若要成功建置，必須執行下列步驟：

1. 複製的內容src/PythonBubblesSample/進入Build/out目錄。
2. 複製的內容\${WEAVER_SDK_DIRECTORY}/lib/weaver/weaver_python_app_sdk_v1進入Build/out/lib/weaver_app_sdk_v1目錄。
3. 複製\${WEAVER_SDK_DIRECTORY}/lib/weaver/libweaver_app_sdk_python_v1_39.so進入Build/out/lib/weaver_app_sdk_v1目錄。
4. 重新命名 Build/out/lib/weaver_app_sdk_v1/libweaver_app_sdk_python_v1_39.so 為 libweaver_app_sdk_python_v1.so。
5. 壓縮的內容Build/out/目錄。
6. 針對在模擬結構描述中指定的每個應用程式 zip 重複 zip 程序。對於PythonBubblesSample，該模式需要`project-nameSpatial.zip`和一個`project-nameView.zip`。

完成這些步驟後，zip 檔案就可以上傳到專案的 Amazon S3 儲存貯體。

關於使用 Python 的常見問題

Q1. (問題 1)：支援哪些版本的 Python？

SimSpace Weaver只支持 3.9 版本。

疑難排解與 Python 相關的問題

主題

- [自訂容器建立期間失敗](#)

- [您的 Python 模擬無法啟動](#)
- [一個 Python 模擬或視圖客戶端拋出一個ModuleNotFound錯誤](#)

自訂容器建立期間失敗

如果您收到錯誤no basic auth credentials在你跑步之後create-custom-container.bat那麼您的亞馬遜 ECR 臨時登入資料可能存在問題。運行以下命令AWS 區域身份證和AWS帳號：

```
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

請確定AWS 區域您指定的與您用於模擬的指定相同。使用其中一個AWS 區域那個SimSpace Weaver支持。如需詳細資訊，請參閱[SimSpace 韋弗端點和配額](#)。

在您執行之後aws ecr命令，執行create-custom-container.bat再次。

其他需要檢查的疑難排解

- [排解自訂容器](#)
- [亞馬遜 ECR 故障排除](#)在亞馬遜 ECR 用戶指南
- [使用亞馬遜 ECR 進行設置](#)在亞馬遜 ECR 用戶指南

您的 Python 模擬無法啟動

您可能會看到Unable to start app模擬管理日誌中出現錯誤。如果您的自訂容器建立失敗，就會發生這種情況 如需詳細資訊，請參閱[自訂容器建立期間失敗](#)。如需日誌的詳細資訊，請參閱[SimSpace Weaver Amazon 日誌中的 CloudWatch 日誌](#)。

如果您確定容器沒有任何問題，請檢查應用程序的 Python 源代碼。您可以使用 SimSpace Weaver Local 測試您的應用程序。如需詳細資訊，請參閱 [本機開發](#)。

一個 Python 模擬或視圖客戶端拋出一個 ModuleNotFound 錯誤

蟒蛇拋出一個 ModuleNotFound 當它找不到所需的 Python 包時出錯。

如果您的模擬位於 AWS 雲端，確保您的自定義容器中列出了所有必需的依賴項 requirements.txt。記得要跑 create-custom-container.bat 再次如果你編輯 requirements.txt。

如果您收到錯誤 PythonBubblesSample 用戶端，使用 pip 安裝指定的軟件包：

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

Support 其他引擎

您可以搭配使用您自己的自訂 C++ 引擎 SimSpace Weaver。我們目前正在開發以下引擎的支持。每個引擎都有單獨的文檔。

Important

與此處列出的引擎的集成是實驗性的。它們可用於預覽。

发动机

- [Unity](#) (最低限度版本)
- [Unreal Engine](#) (最低版本 5.0)

Unity

您必須先安裝開 Unity 發環境，然後才能使用 Unity 建置 SimSpace Weaver 模擬。下載單獨 Unity SDK 的 of AWS SimSpace Weaver (Unity SDK) 並按照該軟件包中的說明進行操作。

⚠ Important

您必須使用最新版本的SimSpace Weaver應用程式 SDK。最新的版本是 1.16.0。如需詳細資訊，請參閱 [AWS SimSpace Weaver 版本](#)。

若要下載並使用 Unity SDK

1. 在 Windows 命令提示符下，轉到您的 *SDK* 文件夾。
2. 執行下載指令碼。將##取代為AWS 區域您將開始模擬的位置 (例如，us-west-2)。

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

指令碼會在您目前的資料夾SimSpaceWeaverUnityPackage.zip中下載並解壓縮。

3. 閱讀SimSpaceWeaverUnityPackage\Release\Documentation\Unity_SDK_for_AWS_SimSpace_Weaver.pdf。

⚠ Important

如果您在 Unity 中收到關於遺失JsonProperty或命名空間的錯誤JsonAttribute，請依照下列步驟新增 NewtonsoftJson 套件：

1. 在 Unity 編輯器中，從功能表列選擇「視窗 > Package 管理員」。
2. 在「Package 管理器」窗口中，選擇窗口頂部的 + (加號) 按鈕。
3. 選擇從 git 網址添加軟件包。
4. 輸入下列資料：

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

5. 選擇新增。

⚠ Important

Unity SDK不支援 AWS Command Line Interface (AWS CLI) 的已命名設定檔。如果您使用AWS IAM Identity Center或AWS CLI紀要，則必須先將具名的設定檔複製或重新命名為default紀要，然後才能使用Unity SDK。如需詳細資訊，請參閱 [《使AWS CLI用AWS](#)

[Command Line Interface](#)者指南》AWS IAM Identity Center中的〈[配置使用指南](#)〉和〈[組態和認證檔案設定AWS Command Line Interface](#)〉。

Unreal Engine

您必須從源代碼構建Unreal Engine專用服務器。 SimSpaceWeaverAppSdkDistributable 包含的版 PathfindingSample 本Unreal Engine。有關更多信息，請參閱單獨的說明：

```
sdk-folder\AWS_SimSpace_Weaver_Unreal_Guide.pdf
```

使用授權軟體AWS SimSpace Weaver

AWS SimSpace Weaver允許您使用自己選擇的模擬引擎和內容來構建模擬。就您的使用而言 SimSpace Weaver，您有責任取得、維護及遵守您在模擬中使用之任何軟體或內容的授權條款。驗證您的授權合約允許在虛擬託管環境中部署軟體和內容。

管理您的資源 AWS CloudFormation

您可以使用AWS CloudFormation來管理您的AWS SimSpace Weaver資源。 AWS CloudFormation 是一項獨立的AWS服務，可協助您以程式碼形式指定、佈建和管理AWS基礎結構。隨著AWS CloudFormation您創建一個 JSON 或 YAML 文件，稱為[模板](#)。您的範本會指定基礎結構的詳細資料。 AWS CloudFormation使用您的範本將基礎結構佈建為單一單位 (稱為[堆疊](#))。當您刪除堆疊時，您可以同時AWS CloudFormation刪除堆疊中的所有內容。您可以使用標準的原始程式碼管理程序來管理範本 (例如，在 [Git](#) 等版本控制系統中追蹤範本)。如需有關 AWS CloudFormation 的詳細資訊，請參閱 [《AWS CloudFormation 使用者指南》](#)。

您的模擬資源

在 AWS 中，資源是指您可以使用的實體。範例包括 Amazon EC2 執行個體、Amazon S3 儲存貯體或 IAM 角色。您的SimSpace Weaver模擬是一種資源。在配置中，您通常在表單中指定AWS資源AWS::*service*::resource。對於SimSpace Weaver，您可以將模擬資源指定為AWS::SimSpaceWeaver::Simulation。若要取得有關中模擬資源的更多資訊AWS CloudFormation，請參閱《AWS CloudFormation使用指南》中的〈〉—[SimSpace Weaver](#)節。

如何使AWS CloudFormation用SimSpace Weaver？

您可以建立AWS CloudFormation範本，以指定想要佈建的AWS資源。您的範本可以指定整個架構、架構的一部分或小型解決方案。例如，您可以為SimSpace Weaver解決方案指定一個架構，其中包括Amazon S3 儲存貯體、IAM 許可、Amazon 關聯式資料庫服務或 Amazon DynamoDB 中的支援資料庫以及您Simulation的資源。然後，您可AWS CloudFormation以使用將所有這些資源作為一個單元佈建，並在同一時間。

Example 建立 IAM 資源並啟動模擬的範本

下列範例範本會建立 IAM 角色和許可，以SimSpace Weaver便在您的帳戶中執行動作。當您建立專案AWS 區域時，SimSpace Weaver應用程式 SDK 指令碼會在特定專案中建立角色和權限，但您可以使用AWS CloudFormation範本將模擬部署到另一個專案，AWS 區域而無需再次執行指令碼。例如，您可以執行此操作來設置備份模擬以進行災難恢復。

在此範例中，原始模擬名稱為MySimulation。該模式的儲存桶已經存在於AWS CloudFormation將構建堆棧的AWS 區域位置中。值區包含已正確設定為在其中執行模擬的結構描述版本AWS 區域。回想一下，結構描述會指定應用程式 zip 檔案的位置，該檔案是與模擬AWS 區域相同的 Amazon S3 儲存貯體。應用程式 zip 儲存桶和文件必須已經存在於AWS CloudFormation構建堆棧AWS 區域時，否則您的模擬將無法啟動。請注意，此範例中的值區名稱包含AWS 區域，但不會決定值區實際位置的位置。您必須確定儲存貯體實際上在該儲存貯體中 AWS 區域 (您可以在 Amazon S3 主控台中檢查儲存貯體屬性、使用 Amazon S3 API 或使用中的 Amazon S3 命令AWS CLI)。

此範例使用中的一些內建函數和參數AWS CloudFormation來執行變數替換。[如需詳細資訊，請參閱《AWS CloudFormation使用指南》中的內建函數參考和虛擬參數參考。](#)

```
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'

搭配使用快照 AWS CloudFormation

快照是模擬的備份。下列範例會從快照而非結構描述啟動新模擬。此範例中的快照是從SimSpace Weaver應用程式 SDK 專案模擬建立的。AWS CloudFormation建立新模擬資源，並使用快照中的資料對其進行初始化。新模擬可以具有與原始模擬MaximumDuration不同的模擬。

我們建議您製作並使用原始模擬應用程式角色的副本。如果您刪除該模擬的AWS CloudFormation堆疊，則可能會刪除原始模擬的應用程式角色。

```
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"

快照

您可以建立快照以在任何時間點備份模擬圖元資料。 SimSpace Weaver在 Amazon S3 儲存貯體中建立 .zip 檔案。您可以快照建立新模擬應用程式。 SimSpace Weaver使用儲存在快照中的實體資料初始化新模擬的 State Fabric、啟動建立快照時執行的空間和服務應用程式，並將時鐘設定為適當的刻度。 SimSpace Weaver從快照取得模擬的組態，而不是從結構描述檔案取得。您的應用程式 .zip 檔案在 Amazon S3 中必須與原始模擬中的位置相同。您必須個別啟動任何自訂應用程式。

主題

- [快照使用案例](#)
- [使用應用SimSpace Weaver程式 SDK 來使用快照](#)
- [使用主SimSpace Weaver控制台使用快照](#)
- [使用AWS CLI來處理快照](#)
- [使用 SimSpace Weaver API 來處理快照](#)
- [搭配使用快照 AWS CloudFormation](#)
- [快照相關指南](#)

快照使用案例

返回先前的狀態並探索分支案例

您可以建立模擬應用程式的快照以便將其儲存為特定狀態。然後，您可以從該快照建立多個新模擬，並探索可從該狀態分支的不同案例。

災難復原和安全最佳實務

我們建議您定期備份模擬，特別是對於執行超過 1 小時或使用多個 Worker 的模擬。備份可協助您從災難和安全事件中復原。快照可讓您備份模擬。快照要求您的應用程式 .zip 檔案和以前一樣存在於 Amazon S3 中的相同位置。如果您需要能夠將應用程式 .zip 文件移動到其他位置，則必須使用自定義備份解決方案。

如需其他最佳實務的詳細資訊，請參閱[使用時的最佳做法 SimSpace Weaver](#)和[SimSpace Weaver 的安全最佳實務](#)。

延長模擬的持續時間

您的模擬資源是中模擬的表現法 SimSpace Weaver。所有模擬資源都有一個 MaximumDuration 設定。模擬資源到達時會自動停止 MaximumDuration。的最大值 MaximumDuration 為 14D (14 天)。

如果您需要模擬的持續時間超過其模擬資源 MaximumDuration 的時間，您可以在模擬資源到達快照之前建立快照 MaximumDuration。您可以使用快照開始新模擬 (建立新模擬資源)。 SimSpace Weaver 從快照初始化實體資料、啟動之前執行的相同空間和服務應用程式，然後還原時鐘。您可以啟動自定義應用程式並執行任何其他自定義初始化。您可以在啟動新模擬資源時將其設定為不同 MaximumDuration 的值。

使用應用 SimSpace Weaver 程式 SDK 來使用快照

您可以使用應用 SimSpace Weaver 程式 SDK 中提供的指令碼 (最低版本 1.13) 來建立和使用快照。

該 SimSpace Weaver 應用程式 SDK 按項目組織您的模擬。您可以從單一專案開始多個模擬。每個模擬都使用相同的結構描述和應用程式 .zip 檔案。 SimSpace Weaver 應用程式 SDK 指令碼會根據專案名稱、AWS 帳戶編號和，將用於模擬的資產放置在特定的 Amazon S3 儲存貯體中 AWS 區域。這些指令碼會處理位於該值區根目錄 snapshot 資料夾中的快照檔案。該 snapshot 資料夾的 Amazon S3 URI 具有以下形式：

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

範例

- 專案名稱 : MyProject
- AWS 帳戶招聘號碼:
- AWS 區域: us-west-2

- 快照資料夾 Amazon S3 URI : `s3://weaver-myproject-111122223333-artifacts-us-west-2/snapshot`

如果您想要使用其他 Amazon S3 儲存貯體，請參閱以下使用快照的替代方式。

其他使用快照的方式

- [SimSpace Weaver 主控台](#)
- [AWS CLI](#)
- [SimSpace Weaver API](#)

主題

- [使用 SimSpace Weaver 應用程式 SDK 建立快照](#)
- [使用 SimSpace Weaver 應用程式 SDK 從快照開始模擬](#)
- [使用應用 SimSpace Weaver 程式 SDK 從快照快照快速啟動模擬](#)
- [使用 SimSpace Weaver 應用程式 SDK 列出專案的快照](#)

使用 SimSpace Weaver 應用程式 SDK 建立快照

若要建立快照，您的模擬必須 STARTED 處於狀態。快照建立會在目前的刻度完成後開始。SimSpace Weaver 停止向應用程序發送刻度，但時鐘狀態仍然顯示 STARTED。模擬狀態會變更為 SNAPSHOT_IN_PROGRESS。快照完成後，模擬狀態會變回，應用程 STARTED 式會再次收到刻度。

若要建立快照

1. 在 Windows 命令提示字元中，移至專案的工具資料夾。

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

2. 如果您不知道模擬的名稱，請呼叫 list-simulations API 以查看您的模擬資源清單。確定模擬狀態為 STARTED。

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

3. 執行專案的 create-snapshot 指令碼。

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

範例

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

SimSpace Weaver在專案的成品值區中建立快照檔案。

範例

- 專案名稱 : MyProject
- AWS 帳戶招聘號碼:
- AWS 區域: us-west-2
- 快照資料夾 Amazon S3 URI : s3://weaver-myproject-111122223333-artifacts-us-west-2/snapshot
- 模擬名稱 : MyProjectSimulation_23-04-29_12_00_00
- 快照時間:二零二三年四月二十九日, 下午 3:30:27 世界標準時間
- 快照檔案名稱 : MyProjectSimulation_23-04-29_12_00_00-230429-1530-27.zip
- 快照檔案 Amazon S3 URI : s3://weaver-myproject-111122223333-artifacts-us-west-2/snapshot/MyProjectSimulation_23-04-29_12_00_00-230429-1530-27.zip

使用SimSpace Weaver應用程式 SDK 從快照開始模擬

當您使用應用程式 SDK 指令碼從快照啟動模擬時，指令碼會建立新的模擬名稱，與在不使用快照的情況下啟動模擬的方式相同。

您的快照檔案必須存在於具有下列 Amazon S3 URI 的 Amazon S3 中的快照位置：

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

應用程式 .zip 檔案必須位於建立快照時的相同位置。

SimSpace Weaver建立新的模擬資源、使用儲存在快照中的實體資料初始化 State Fabric、啟動建立快照時執行的相同空間和服務應用程式的新執行個體，並將時鐘設定為適當的刻度。您必須通過正常過程單獨啟動自定義應用程序。

此指start-from-snapshot令碼是指令碼的快照版start-simulation本。就像start-simulation腳本一樣，start-from-snapshot腳本不會為您啟動時鐘。您必須分別啟動時鐘。

從快照開始模擬的步驟

1. 在 Windows 命令提示字元中，移至專案的工具資料夾。

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

2. 執行 `start-from-snapshot` 指令碼。

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

範例

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

使用應用 SimSpace Weaver 程式 SDK 從快照快照快速啟動模擬

您可以從快照快照快速啟動模擬。這類似於沒有快照的快速啟動。

您的快照檔案必須存在於具有下列 Amazon S3 URI 的 Amazon S3 中的快照位置：

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

應用程式 .zip 檔案必須位於建立快照時的相同位置。

SimSpace Weaver 建立新的模擬資源、使用儲存在快照中的實體資料初始化 State Fabric、啟動建立快照時執行的相同空間和服務應用程式的新執行個體，並將時鐘設定為適當的刻度。您必須通過正常過程單獨啟動自定義應用程式。

此指 `quick-start-from-snapshot` 指令碼是指令碼的快照版 `quick-start` 本。就像 `quick-start` 腳本一樣，`quick-start-from-snapshot` 腳本會為您啟動時鐘。它還啟動了路徑查找示例項目的視圖應用程式。

從快照快照快速啟動模擬的步驟

1. 在 Windows 命令提示字元中，移至專案的工具資料夾。

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

2. 執行 `quick-start-from-snapshot` 指令碼。

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

範例

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

使用 SimSpace Weaver 應用程式 SDK 列出專案的快照

您可以使用 `list-snapshots` 指令碼列出專案的快照。此指令碼會列出專案資料夾中的檔案。項目是 SimSpace Weaver 應用程式 SDK 唯一的，因此您只能使用應用程式 SDK 腳本來完成此操作，並且僅適用於項目。該指令碼假設 Amazon S3 上 `snapshot` 資料夾中的所有檔案都是快照檔案。如果您移動或刪除文件夾中的文件，那些文件將不會出現在列表中。

欲列出專案的快照

1. 在 Windows 命令提示字元中，移至專案的工具資料夾。

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

2. 執行 `list-snapshots` 指令碼。

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

範例

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

使用主 SimSpace Weaver 控制台使用快照

您可以使用主 SimSpace Weaver 控制台建立模擬應用程式的快照。

其他使用快照的方式

- [SimSpace Weaver 應用程式 SDK 腳本](#)
- [AWS CLI](#)

- [SimSpace Weaver API](#)

主題

- [使用主控台建立快照](#)
- [使用控制台從快照開始模擬](#)

使用主控台建立快照

若要建立快照

1. 登入AWS Management Console並連線到[SimSpace Weaver主控台](#)。
2. 從導覽窗格中選擇模擬應用程式。
3. 選取模擬應用程式旁的選項按鈕。模擬的「狀態」必須為「已開始」。
4. 在頁面頂端，選擇建立快照相關指示。
5. 在快照設定下，針對快照目的地，輸入儲存貯體的 Amazon S3 URI，或是要SimSpace Weaver在其中建立快照的儲存貯體和資料夾。如果您偏好瀏覽可用儲存貯體並選取位置，可以選擇瀏覽 S3。

Important

Amazon S3 儲存貯體必須位於與模擬相AWS 區域同的儲存貯體。

Note

SimSpace Weaver在您選取的快照目標內建立snapshot資料夾。 SimSpace Weaver會在該snapshot資料夾中建立快照 .zip 檔案。

6. 選擇 Create snapshot (建立快照)。

使用控制台從快照開始模擬

若要從快照開始模擬，您的快照 .zip 檔案必須存在於您的模擬可存取的 Amazon S3 儲存貯體中。您的模擬會使用您在啟動模擬時選取的應用程式角色中定義的權限。原始模擬中的所有應用程式 .zip 檔案都必須存在於與建立快照時相同的位置。

從快照開始模擬的步驟

1. 登入AWS Management Console並連線到[SimSpace Weaver主控台](#)。
2. 從導覽窗格中選擇模擬應用程式。
3. 在頁面頂端，選擇 [開始模擬應用程式]。
4. 在「模擬設定」下，輸入模擬的名稱和可選描述。您的模擬名稱在您的中必須是唯一的AWS 帳戶。
5. 對於模擬啟動方法，請選擇在 Amazon S3 中使用快照。
6. 對於用於快照的 Amazon S3 URI，請輸入快照檔案的 Amazon S3 URI，或選擇瀏覽 S3 瀏覽並選取檔案。

Important

Amazon S3 儲存貯體必須位於與模擬相AWS 區域同的儲存貯體。

7. 對於 IAM 角色，請選取模擬將使用的應用程式角色。
8. 對於持續時間上限，請輸入模擬資源應執行的時間上限。最大值為 14D. [如需持續時間最長的詳細資訊，請參閱《](#)
9. 如果您要新增標籤，請在 [標籤]-[選用] 底下選擇 [新增標籤]。
10. 選擇 [開始模擬]。

使用AWS CLI來處理快照

您可以使AWS CLI用從命令提示字元呼叫 SimSpace Weaver API。您必須正確AWS CLI安裝和配置。如需詳細資訊，請參閱《[第 2 版](#)》中的[AWS安裝或更新](#) AWS Command Line Interface的最新版本。

其他使用快照的方式

- [SimSpace Weaver應用程式 SDK 腳本](#)
- [SimSpace Weaver 主控台](#)
- [SimSpace Weaver API](#)

主題

- [使用AWS CLI建立快照](#)
- [使用AWS CLI從快照啟動模擬](#)

使用AWS CLI建立快照

若要建立快照

- 在命令提示字元中，呼叫 CreateSnapshot API。

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

參數

模擬

已開始模擬的名稱。您可以使aws simspaceweaver list-simulations用查看模擬的名稱和狀態。

目的地

字串，指定快照檔案的目的地 Amazon S3 儲存貯體和選用的物件 key prefix。您的物件 key prefix 通常是值區中的資料夾。 SimSpace Weaver在此目的地的snapshot資料夾內建立快照。

Important

Amazon S3 儲存貯體必須位於與模擬相AWS 區域同的儲存貯體。

範例

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

如需 CreateSnapshot API 的詳細資訊，請參閱 AWS SimSpace WeaverAPI 參考資料[CreateSnapshot](#)中的。

使用AWS CLI從快照啟動模擬

從快照開始模擬的步驟

- 在命令提示字元中，呼叫 StartSimulation API。

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

參數

name

新模擬應用程式的名稱。該模擬名稱在您的中必須是唯一的AWS 帳戶。您可以使aws simspaceweaver list-simulations用查看既有模擬的名稱。

角色 arn

您的模擬應用程式角色的 Amazon Resource Name (ARN)。

快照-S3-位置

字串；指定快照相關指定 Amazon S3 儲存貯體和物件金鑰的物件金鑰。

Important

Amazon S3 儲存貯體必須位於與模擬相AWS 區域同的儲存貯體。

範例

```
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
```

如需 StartSimulation API 的詳細資訊，請參閱 AWS SimSpace WeaverAPI 參考資料[StartSimulation](#)中的。

使用 SimSpace Weaver API 來處理快照

您可以直接呼叫 SimSpace Weaver API 來處理快照。如需 API 的相關資訊，請參閱 [AWS SimSpace Weaver API 參考資料](#)。

其他使用快照的方式

- [SimSpace Weaver 應用程式 SDK 腳本](#)
- [SimSpace Weaver 主控台](#)
- [AWS CLI](#)

建立快照

您可以呼叫 CreateSnapshot API 來建立模擬的快照。模擬狀態必須是 STARTED。SimSpace Weaver 在您指定的 Amazon S3 儲存貯體和物件前置詞的 snapshot 資料夾中建立快照檔案。如需詳細資訊，請參閱《AWS SimSpace Weaver API 參考》中的 [CreateSnapshot](#)。

從快照開始模擬應用

您可以在呼叫 StartSimulation API 以開始新模擬時提供快照。您提供 JSON 字串做為參SnapshotS3Location 數的引數。該字串指定快照相關指定 Amazon S3 儲存貯體名稱和物件金鑰。如果您提供 SnapshotS3Location，則無法提供 SchemaS3Location。如需詳細資訊，請參閱《AWS SimSpace Weaver API 參考》中的 [StartSimulation](#)。

快照相關指南

我的模擬是否會在快照期間繼續執行？

您的模擬資源會在快照期間繼續執行，而您仍會繼續收取該期間的帳單費用。時間會計入模擬的最長持續時間。快照正在進行時，您的應用程式不會收到刻度。如果您的 STARTED 時鐘狀態是快照建立開始的時間，您的時鐘仍會顯示 STARTED 狀態。快照完成後，您的應用程式再次收到刻度。如果您的時鐘狀態為 STOPPED 則您的時鐘狀態將保留 STOPPED。請注意，即使時鐘 STARTED 狀態為，具有狀態的模擬仍在執行 STOPPED。

如果快照正在進行中，且我的模擬達到其最長持續時間，會發生什麼情況？

您的模擬將完成快照，然後在快照過程結束後立即停止 (無論是成功還是失敗)。我們建議您事先測試快照程序，以瞭解快照檔需要多長時間、您可以預期的快照檔大小，以及快照檔案是否應該成功完成。

如果我停止正在進行快照的模擬，會發生什麼情況？

當您停止模擬時，進行中的快照會立即停止。它不會創建快照文件。

如何停止進行中的快照？

停止進行中的快照的唯一方法是停止模擬。停止模擬應用程式後就無法重新開始模擬應用程式。

完成快照需要多長時間？

建立快照所需的時間取決於您的模擬。我們建議您事先測試快照程序，以瞭解模擬需要多長時間。

我的快照檔案會有多大？

快照檔案的大小取決於您的模擬。我們建議您事先測試快照程序，以瞭解檔案可用於模擬的大小。

簡訊

訊息傳遞 API 可簡化模擬中應用程式與應用程式間的通訊。用於發送和接收消息的 API 是 SimSpace Weaver 應用程式 SDK 的一部分。訊息傳送目前使用最佳的方式來傳送和接收訊息。 SimSpace Weaver 嘗試在下一個模擬刻度時傳送/接收訊息，但沒有傳送、訂購或到達時間的保證。

主題

- [訊息傳送的使用案例](#)
- [使用訊息傳遞 API](#)
- [何時使用訊息](#)
- [使用消息傳遞時的提示](#)
- [訊息錯誤和疑難排解](#)

訊息傳送的使用案例

在模擬應用程式之間

使用訊息 API 在模擬中的應用程式之間進行通訊。使用它可以變更遠距離的圖元狀態、變更實體行為，或將資訊廣播至整個模擬。

確認收到訊息

已傳送的郵件會在郵件標頭中包含寄件者的相關資訊。使用此資訊可在收到訊息時傳回確認回覆。

將自定義應用程式接收的數據轉發到模擬中的其他應用程式

消息傳遞不能替代客戶端如何連接到中運行的自定義應用程式 SimSpace Weaver。但是，消息傳遞確實允許用戶將數據從接收客戶端數據的自定義應用程式轉發到沒有外部連接的其他應用程式的方法。消息流也可以反向運行，允許沒有外部連接的應用程式將數據轉發到自定義應用程式，然後轉發到客戶端。

使用訊息傳遞 API

消息傳遞 API 包含在 SimSpace Weaver 應用程式 SDK 中（最低版本 1.16.0）。消息傳遞支持 C++，Python，並且我們與虛幻引擎 5 和統一的集成。

有兩個函數可以處理訊息交易：SendMessage 和 ReceiveMessages。所有發送的消息都包含目的地和有效負載。該 ReceiveMessages API 返回當前在應用程式的入站消息隊列中的消息列表。

C++

發送消息

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

接收訊息

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

Python

發送消息

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

接收訊息

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

主題

- [傳送訊息](#)
- [接收訊息](#)
- [回覆寄件者](#)

傳送訊息

訊息是由交易 (類似於其他 Weaver API 呼叫)、承載和目的地所組成。

訊息承載

訊息承載是高達 256 個位元組的彈性資料結構。我們建議您採用下列作為建立訊息承載的最佳作法。

若要建立訊息承載

1. 建立定義訊息內容的資料結構 (例如 C++)。struct
2. 建立包含要在郵件中傳送之值的訊息承載。
3. 建立MessagePayload物件。

訊息目的地

訊息的目的地由MessageEndpoint物件定義。這包括端點類型和端點 ID。目前支援的唯一端點類型為Partition，可讓您將訊息傳送至模擬中的其他分割區。端點 ID 是目標目標的分割區 ID。

您只能在一封郵件中提供 1 個目的地位址。如果您想同時將消息發送到 1 個以上的分區，請創建並發送多個消息。

如需如何從位置解析訊息端點的指引，請參閱[使用消息傳遞時的提示](#)。

傳送訊息

您可以在建立目的地和承載物件之後使用 SendMessage API。

C++

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

Python

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

發送消息的完整示例

下面的例子演示了如何構建和發送一般消息。此範例會傳送 16 則個別訊息。每個訊息都包含一個值為 0 和 15 的有效負載，以及目前的模擬刻度。

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
```

接收訊息

SimSpace Weaver將訊息傳遞至分割區的輸入訊息佇列。使用 `ReceiveMessages` API 取得包含佇列訊息的 `MessageList` 物件。使用 `ExtractMessage` API 處理每條消息以獲取消息數據。

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
```


回覆寄件者

每個收到的郵件都包含一個郵件標頭，其中包含郵件原始寄件者的相關資訊。您可以使用消息。頭。源端點發送回復。

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
```

何時使用訊息

中的訊息SimSpace Weaver提供了另一種模式，可在模擬應用程式之間交換資訊 訂閱提供提取機制，可從特定應用程式或模擬區域讀取資料；訊息提供推送機制，可將資料傳送至特定應用程式或模擬區域。

以下是兩個使用案例，其中使用消息傳遞推送數據，而不是通過訂閱提取或讀取數據更有幫助。

Example 1：將命令發送到另一個應用程式以更改實體位置

```
// 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);
```

在接收端，應用程式更新實體的位置，並將其寫入狀態結構。

```
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：將創建實體消息發送到空間應用程式

```

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

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);

```

在接收端，應用程式會在 State Fabric 中建立新的實體，並更新其位置。

```

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)));
}
```

使用消息傳遞時的提示

從位置或應用程式名稱解析端點

您可以使用此AllPartitions函數來取得決定郵件分割區 ID 和郵件目的地所需的空間界限和網域識別碼。但是，如果您知道要發送消息的位置，但不知道其分區 ID，則可以使用該MessageEndpointResolver 功能。

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

序列化和反序列化消息有效負載

您可以使用下列函數來建立和讀取訊息承載。如需詳細資訊，請參閱本機系統上應用程式 SDK 程式庫中的 MessagingUtils .h。

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

訊息錯誤和疑難排解

使用訊息傳遞 API 時，您可能會遇到下列錯誤。

端點解析錯誤

這些錯誤可能會在應用程式傳送訊息之前發生。

域名檢查

傳送訊息至無效的端點會導致下列錯誤：

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

當您嘗試將消息發送到自定義應用程式，並且該自定義應用程式尚未加入模擬時，可能會發生這種情況。使用 DescribeSimulation API 確保您的自定義應用程式已啟動，然後再向其發送消息。這個行為在 SimSpace Weaver Local 和中是相同的 AWS 雲端。

位置檢查

嘗試解析具有有效網域名稱但位置無效的端點會導致下列錯誤。

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

我們建議您使用MessageEndpointResolver應用SimSpace Weaver程式 SDK 中包含的程式MessageUtils庫中的。

訊息傳送錯誤

應用程式傳送訊息時，可能會發生下列錯誤。

超過每個應用程式，每個刻度的消息發送限制

每個模擬刻度的每個應用程式可以發送的消息數量的電流限制為 128。同一刻度的後續呼叫將失敗，並出現以下錯誤：

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

SimSpace Weaver嘗試在下一個刻度發送未發送的消息。降低傳送頻率以解決此問題。合併小於 256 位元組限制的郵件承載，以減少輸出訊息的數目。

在SimSpace Weaver Local和中，此行為是相同的AWS 雲端。

超過郵件承載大小限制

郵件承載大小的目前限制為 256 個位元組 (以SimSpace Weaver Local及) AWS 雲端。傳送裝載大於 256 位元組的訊息會導致下列錯誤：

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

SimSpace Weaver檢查每封郵件，並僅拒絕超出限制的郵件。例如，如果您的應用程式嘗試發送 10 條消息，而 1 則檢查失敗，則僅拒絕該 1 條消息。 SimSpace Weaver發送其他 9 條消息。

這個行為在SimSpace Weaver Local和中是相同的AWS 雲端。

目的地與來源相同

應用程式無法將消息發送到他們擁有的分區。如果應用程式將訊息傳送至其擁有的分割區，您會收到下列錯誤訊息。

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

這個行為在SimSpace Weaver Local和中是相同的AWS 雲端。

最好的消息傳遞

SimSpace Weaver 不保證郵件傳遞。該服務將嘗試在後續的模擬刻度上完成消息的傳遞，但消息可能會丟失或延遲。

使用時的最佳做法 SimSpace Weaver

我們建議您在使用時採用下列最佳作法SimSpace Weaver。

主題

- [設定帳單警示](#)
- [使用 SimSpace Weaver Local](#)
- [停止不需要的模擬](#)
- [刪除不需要的資源](#)
- [有備份](#)

設定帳單警示

即使在不再需要資源的情況下，您也可以輕鬆佈建資源AWS並始終保持運行。這可能會導致失控的成本，當你得到你的賬單可能是一個驚喜。您可以在 Amazon 中設定警示 CloudWatch ，當成本超過您設定的閾值時，就會觸發並通知您。您可以使用成本管理工具來檢查成本。如需詳細資訊，請參閱：

- [建立帳單警示以監控您的估計AWS費用](#)
- [什麼是 AWS Cost Management](#)

使用 SimSpace Weaver Local

我們建議您先使用SimSpace Weaver Local開發和測試您的模擬，然後再將它們上傳到中的SimSpace WeaverAWS 雲端服務。與開發的好處SimSpace Weaver Local包括：

- 無需等待大量上傳
- 您可以建立的本端模擬數量沒有限制
- 您不需支付本機電腦上的運算時間費用
- 直接從您的應用程式存取主控台輸出
- 修改、重新建置和重新啟動本端模擬，而不必在 AWS 雲端

停止不需要的模擬

您會在模擬執行時收到帳單費用。您必須停止模擬才能停止收取費用。執行模擬也會計入您的模擬數量上限的配額中。已設定記錄的執行中模擬也可以產生大量的記錄檔，您也會收取帳單費用。您應停止任何不需要的模擬，以停止收取額外費用。

Important

停止模擬時鐘不會停止模擬，時鐘只是停止將刻度發佈到您的應用程式。您無法在停止模擬後重新啟動模擬。

刪除不需要的資源

您在中建立的每個模擬SimSpace Weaver也會在其他AWS服務中建立資源。您可以針對這些其他服務中的資源和資料收取帳單費用。執行中和失敗的模擬會計入您的模擬數量上限的配額中。您應該刪除不需要的失敗模擬，以便可以開始新模擬。刪除模擬時，可能不會刪除存在於其他AWS服務中的模擬資源。例如，Amazon CloudWatch 日誌中的任何模擬日誌資料都會保留在該資料，直到您將其刪除為止。您將收到該日誌數據的帳單費用。如果您不再需要模擬的所有相關資源，您應該清理模擬的所有相關資源。如需詳細資訊，請參閱快速[步驟 6：停止並清理您的模擬](#)入門自學課程中的〈〉。

有備份

對所有內容進行備份和備份計劃是個好主意。您不應該假設，僅僅因為您AWS的數據存在於您不必備份它。如果需要備份模擬狀態，則必須建立自己的系統。考慮使用多個工作負載AWS 區域並制定適當的計劃，以便在需要時快速將生產工作負載切換到另一AWS 區域個工作負載。若要取得有關AWS 區域該支援的更多資訊SimSpace Weaver，請參閱[SimSpace 韋弗端點和配額](#)。

AWS SimSpace Weaver 中的安全性

雲端安全是 AWS 最重視的一環。身為 AWS 的客戶，您將能從資料中心和網路架構中獲益，這些都是專為最重視安全的組織而設計的。

安全性是 AWS 與您共同肩負的責任。[共同的責任模式](#)將其稱為雲端的安全性和雲端中的安全性：

- 雲端本身的安全 – AWS 負責保護執行 AWS 雲端內 AWS 服務的基礎設施。AWS 提供的服務，也可讓您安全使用。第三方稽核人員會定期測試和驗證我們安全性的有效性，作為 [AWS 合規計畫](#) 的一部分。若要了解適用於 AWS SimSpace Weaver 的合規計畫，請參閱 [合規計畫的 AWS 服務範圍](#)。
- 雲端內部的安全 – 您的責任取決於所使用的 AWS 服務。您也必須對其他因素負責，包括資料的機密性、您的請求和適用法律和法規。

本文件有助於您了解如何在使用 SimSpace Weaver 時套用共同責任模型。下列主題說明如何將 SimSpace Weaver 設定為達到您的安全及法規遵循目標。您也會了解如何使用其他 AWS 服務來協助監控並保護 SimSpace Weaver 資源。

主題

- [AWS SimSpace Weaver 中的資料保護](#)
- [AWS SimSpace Weaver 的身分和存取權管理](#)
- [安全事件記錄和監控 AWS SimSpace Weaver](#)
- [AWS SimSpace Weaver 的合規驗證](#)
- [AWS SimSpace Weaver 中的恢復能力](#)
- [AWS SimSpace Weaver 中的基礎設施安全](#)
- [AWS SimSpace Weaver 中的組態與漏洞分析](#)
- [SimSpace Weaver 的安全最佳實務](#)

AWS SimSpace Weaver 中的資料保護

AWS [共同的責任模型](#)適用於 AWS SimSpace Weaver 中的資料保護。如此模型所述，AWS 負責保護執行所有 AWS 雲端的全球基礎設施。您負責維護在此基礎設施上託管內容的控制權。您也必須負責您所使用 AWS 服務的安全組態和管理任務。如需資料隱私權的相關資訊，請參閱 [資料隱私權常見問答集](#)。如需有關歐洲資料保護的相關資訊，請參閱 AWS 安全性部落格上的 [AWS 共同的責任模型和 GDPR](#) 部落格文章。

基於資料保護目的，建議您使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 保護 AWS 帳戶憑證，並設定個人使用者。如此一來，每個使用者都只會獲得授與完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 使用 AWS CloudTrail 設定 API 和使用者活動日誌記錄。
- 使用 AWS 加密解決方案，以及 AWS 服務內的所有預設安全控制項。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在透過命令列介面或 API 存取 AWS 時，需要 FIPS 140-2 驗證的加密模組，請使用 FIPS 端點。如需 FIPS 和 FIPS 端點的相關資訊，請參閱[聯邦資訊處理標準 \(FIPS\) 140-2 概觀](#)。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如 Name(名稱) 欄位。這包括當您使用 SimSpace Weaver 或使用主控台、API、AWS CLI 或 AWS 開發套件的其他 AWS 服務。您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

靜態加密

當資料位於非揮發性 (持續性) 資料儲存體 (例如磁碟) 中時，就會將資料視為靜止狀態。位於易失性數據存儲中的數據，例如存儲器和寄存器，不被認為是靜態的。

使用時 SimSpace Weaver，唯一的靜態數據是：

- 您上傳到亞馬遜簡單儲存服務 (Amazon S3) 的應用程式和結構描述
- 存放在 Amazon 的模擬日誌資料 CloudWatch

停止模擬後，內部 SimSpace Weaver 使用的其他資料不會持續存在。

若要瞭解如何加密靜態資料，請參閱：

- [在 Amazon S3 中加密您的資料](#)
- [加密您的日誌數據](#)

傳輸中加密

您透過 AWS Command Line Interface (AWS CLI)、AWS SDK 和 SimSpace Weaver 應用程式 SDK 與 SimSpace Weaver API 的連線，請使用 TLS 加密搭配 [簽章版本 4 簽署程序](#)。AWS 針對您用來連線的安全登入資料，使用 IAM 定義的存取原則來管理驗證。

在內部，SimSpace Weaver 使用 TLS 連線至其使用的其他 AWS 服務。

Important

您的應用程式與其用戶端之間的通訊不涉及 SimSpace Weaver。如有需要，您有責任加密與模擬用戶端的通訊。我們建議您建立解決方案，以加密所有用戶端連線傳輸中的資料。

若要深入瞭解可支援您加密解決方案的 AWS 服務，請參閱 [AWS 安全性部落格](#)。

網際網路流量隱私權

SimSpace Weaver 運算資源位於所有 SimSpace Weaver 客戶共用的 1 個 Amazon VPC 內。所有內部 SimSpace Weaver 服務流量都保持在 AWS 網絡內，並且不會通過互聯網傳輸。模擬客戶端和您的應用程式之間的通信在互聯網上傳播。

AWS SimSpace Weaver 的身分和存取權管理

AWS Identity and Access Management (IAM) 是一種 AWS 服務，讓管理員能夠安全地控制對 AWS 資源的存取權限。IAM 管理員可以控制身分身分驗證 (已登入) 和授權 (具有許可) 以使用 SimSpace Weaver 資源。IAM 是一種您可以免費使用的 AWS 服務。

主題

- [物件](#)
- [使用身分驗證](#)
- [使用政策管理存取權](#)
- [AWS SimSpace Weaver 搭配 IAM 的運作方式](#)
- [AWS SimSpace Weaver 的身分型政策範例](#)
- [為您建 SimSpace Weaver 立的權限](#)

- [預防跨服務混淆代理人](#)
- [對 AWS SimSpace Weaver 身分與存取進行疑難排解](#)

物件

AWS Identity and Access Management (IAM) 的使用方式會不同，需視您在 SimSpace Weaver 中所執行的工作而定。

服務使用者：如果使用 SimSpace Weaver 執行任務，管理員會為您提供所需的憑證和許可。隨著您為了執行作業而使用的 SimSpace Weaver 功能數量變多，您可能會需要額外的許可。了解存取的管理方式可協助您向管理員請求正確的許可。若您無法存取 SimSpace Weaver 中的某項功能，請參閱 [對 AWS SimSpace Weaver 身分與存取進行疑難排解](#)。

服務管理員：如果您負責公司內的 SimSpace Weaver 資源，您可能具備 SimSpace Weaver 的完整存取權限。您的任務是判斷服務使用者應存取的 SimSpace Weaver 功能及資源。接著，您必須將請求提交給您的 IAM 管理員，來變更您服務使用者的許可。檢閱此頁面上的資訊，了解 IAM 的基本概念。若要進一步了解貴公司可搭配 SimSpace Weaver 使用 IAM 的方式，請參閱 [AWS SimSpace Weaver 搭配 IAM 的運作方式](#)。

IAM 管理員：如果您是 IAM 管理員，建議您掌握如何撰寫政策以管理 SimSpace Weaver 存取權的詳細資訊。若要檢視您可以在 IAM 中使用的範例 SimSpace Weaver 身分型政策，請參閱 [AWS SimSpace Weaver 的身分型政策範例](#)。

使用身分驗證

身分驗證是使用身分憑證登入 AWS 的方式。您必須以 AWS 帳戶根使用者、IAM 使用者身分，或擔任 IAM 角色進行驗證 (登入至 AWS)。

您可以使用透過身分來源 AWS IAM Identity Center 提供的憑證，以聯合身分登入 AWS。(IAM Identity Center) 使用者、貴公司的單一登入身分驗證和您的 Google 或 Facebook 憑證都是聯合身分的範例。您以聯合身分登入時，您的管理員先前已設定使用 IAM 角色的聯合身分。您 AWS 藉由使用聯合進行存取時，您會間接擔任角色。

根據您的使用者類型，您可以登入 AWS Management Console 或 AWS 存取入口網站。如需登入至 AWS 的相關資訊，請參閱《AWS 登入 使用者指南》中的 [如何登入您的 AWS 帳戶](#)。

如果您是以程式設計的方式存取 AWS，AWS 提供軟體開發套件 (SDK) 和命令列介面 (CLI)，以便使用您的憑證透過密碼編譯方式簽署您的請求。如果您不使用 AWS 工具，您必須自行簽署請求。如需使用建議的方法自行簽署請求的相關資訊，請參閱《IAM 使用者指南》中的 [簽署 AWS API 請求](#)。

無論您使用何種身分驗證方法，您可能都需要提供額外的安全性資訊。例如，AWS 建議您使用多重要素驗證 (MFA) 以提高帳戶的安全。如需更多資訊，請參閱《AWS IAM Identity Center 使用者指南》中的[多重要素驗證](#)和《IAM 使用者指南》中的[在 AWS 中使用多重要素驗證 \(MFA\)](#)。

AWS 帳戶 根使用者

如果是建立 AWS 帳戶，您會先有一個登入身分，可以完整存取帳戶中所有 AWS 服務與資源。此身分稱為 AWS 帳戶 根使用者，使用建立帳戶時所使用的電子郵件地址和密碼即可登入並存取。強烈建議您不要以根使用者處理日常作業。保護您的根使用者憑證，並將其用來執行只能由根使用者執行的任務。如需這些任務的完整清單，了解需以根使用者登入的任務，請參閱《IAM 使用者指南》中的[需要根使用者憑證的任務](#)。

聯合身分

最佳實務是要求人類使用者 (包括需要管理員存取權的使用者) 搭配身分提供者使用聯合功能，使用暫時憑證來存取 AWS 服務。

聯合身分是來自您企業使用者目錄的使用者、Web 身分供應商、AWS Directory Service、Identity Center 目錄或透過身分來源提供的憑證來存取 AWS 服務的任何使用者。聯合身分存取 AWS 帳戶時，會擔任角色，並由角色提供暫時憑證。

對於集中式存取權管理，我們建議您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中建立使用者和群組，也可以連線並同步到自己身分來源中的一組使用者和群組，以便在您的所有 AWS 帳戶和應用程式中使用。如需 IAM Identity Center 的相關資訊，請參閱《AWS IAM Identity Center 使用者指南》中的[什麼是 IAM Identity Center ?](#)。

IAM 使用者和群組

[IAM 使用者](#)是您 AWS 帳戶中的一種身分，具備單一人員或應用程式的特定許可。建議您盡可能依賴暫時憑證，而不是擁有建立長期憑證 (例如密碼和存取金鑰) 的 IAM 使用者。但是如果特定使用案例需要擁有長期憑證的 IAM 使用者，建議您輪換存取金鑰。如需詳細資訊，請參閱《[IAM 使用者指南](#)》中的為需要長期憑證的使用案例定期輪換存取金鑰。

[IAM 群組](#)是一種指定 IAM 使用者集合的身分。您無法以群組身分登入。您可以使用群組來一次為多名使用者指定許可。群組可讓管理大量使用者許可的過程變得更為容易。例如，您可以擁有一個名為 IAMAdmins 的群組，並給予該群組管理 IAM 資源的許可。

使用者與角色不同。使用者只會與單一人員或應用程式建立關聯，但角色的目的是在由任何需要它的人員取得。使用者擁有永久的長期憑證，但角色僅提供暫時憑證。若要進一步了解，請參閱《IAM 使用者指南》中的[建立 IAM 使用者 \(而非角色\) 的時機](#)。

IAM 角色

IAM 角色是您 AWS 帳戶中的一種身分，具備特定許可。它類似 IAM 使用者，但不與特定的人員相關聯。您可以在 AWS Management Console 中透過[切換角色](#)來暫時取得 IAM 角色。您可以透過呼叫 AWS CLI 或 AWS API 操作，或是使用自訂 URL 來取得角色。如需使用角色的方法的相關資訊，請參閱《IAM 使用者指南》中的[使用 IAM 角色](#)。

使用暫時憑證的 IAM 角色在下列情況中非常有用：

- 聯合身分使用者存取 — 如需向聯合身分指派許可，請建立角色，並為角色定義許可。當聯合身分進行身分驗證時，該身分會與角色建立關聯，並取得由角色定義的許可。如需有關聯合角色的詳細資訊，請參閱《IAM 使用者指南》https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_create_for-idp.html中的為第三方身分提供者建立角色。如果您使用 IAM Identity Center，則需要設定許可集。為控制身分驗證後可以存取的內容，IAM Identity Center 將許可集與 IAM 中的角色相關聯。如需有關許可集的資訊，請參閱《AWS IAM Identity Center 使用者指南》中的[許可集](#)。
- 暫時 IAM 使用者許可 – IAM 使用者或角色可以擔任 IAM 角色來暫時針對特定任務採用不同的許可。
- 跨帳戶存取權 – 您可以使用 IAM 角色，允許不同帳戶中的某人 (信任的委託人)存取您帳戶中的資源。角色是授予跨帳戶存取權的主要方式。但是，針對某些 AWS 服務，您可以將政策直接連接到資源 (而非使用角色作為代理)。若要了解跨帳戶存取角色和資源型政策間的差異，請參閱《IAM 使用者指南》中的[IAM 角色與資源類型政策的差異](#)。
- 跨服務存取 – 有些 AWS 服務會使用其他 AWS 服務中的功能。例如，當您在服務中進行呼叫時，該服務通常會在 Amazon EC2 中執行應用程式或將物件儲存在 Amazon Simple Storage Service (Amazon S3) 中。服務可能會使用呼叫主體的許可、使用服務角色或使用服務連結角色來執行此作業。
- 轉發存取工作階段 (FAS)：當您使用 IAM 使用者或角色在 AWS 中執行動作時，系統會將您視為主體。當您使用某些服務時，您可能會執行一個動作，而該動作之後會在不同的服務中啟動另一個動作。FAS 使用主體的許可呼叫 AWS 服務，搭配請求 AWS 服務以向下游服務發出請求。只有在服務收到需要與其他 AWS 服務或資源互動才能完成的請求之後，才會提出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱《[轉發存取工作階段](#)》。
- 服務角色：服務角色是服務擔任的 [IAM 角色](#)，可代表您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱《IAM 使用者指南》中的[建立角色以委派許可給 AWS 服務](#)。

- 服務連結角色 – 服務連結角色是一種連結到 AWS 服務的服務角色類型。服務可以擔任代表您執行動作角色。服務連結角色會顯示在您的 AWS 帳戶中，並由該服務所擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。
- 在 Amazon EC2 上執行的應用程式 – 針對在 EC2 執行個體上執行並提出 AWS CLI 和 AWS API 請求的應用程式，您可以使用 IAM 角色來管理暫時憑證。這是在 EC2 執行個體內儲存存取金鑰的較好方式。如需指派 AWS 角色給 EC2 執行個體並提供其所有應用程式使用，您可以建立連接到執行個體的執行個體設定檔。執行個體設定檔包含該角色，並且可讓 EC2 執行個體上執行的程式取得暫時憑證。如需詳細資訊，請參閱《IAM 使用者指南》中的[利用 IAM 角色來授予許可給 Amazon EC2 執行個體上執行的應用程式](#)。

如需了解是否要使用 IAM 角色或 IAM 使用者，請參閱《IAM 使用者指南》中的[建立 IAM 角色 \(而非使用者\) 的時機](#)。

使用政策管理存取權

您可以透過建立政策並將其附加到 AWS 身分或資源，在 AWS 中控制存取。政策是 AWS 中的一個物件，當其和身分或資源建立關聯時，便可定義其許可。AWS 會在主體 (使用者、根使用者或角色工作階段) 發出請求時評估這些政策。政策中的許可，決定是否允許或拒絕請求。大部分政策以 JSON 文件形式儲存在 AWS 中。如需 JSON 政策文件結構和內容的詳細資訊，請參閱《IAM 使用者指南》中的[JSON 政策概觀](#)。

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

預設情況下，使用者和角色沒有許可。若要授與使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

IAM 政策定義該動作的許可，無論您使用何種方法來執行操作。例如，假設您有一個允許 `iam:GetRole` 動作的政策。具備該政策的使用者便可以從 AWS Management Console、AWS CLI 或 AWS API 取得角色資訊。

身分型政策

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分類型政策，請參閱《IAM 使用者指南》中的[建立 IAM 政策](#)。

身分型政策可進一步分類成內嵌政策或受管政策。內嵌政策會直接內嵌到單一使用者、群組或角色。受管政策則是獨立的政策，您可以將這些政策附加到 AWS 帳戶中的多個使用者、群組和角色。受管政

策包含 AWS 管理政策和客戶管理政策。如需瞭解如何在受管政策及內嵌政策間選擇，請參閱 IAM 使用者指南中的[在受管政策和內嵌政策間選擇](#)。

資源型政策

資源型政策是連接到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中[指定主體](#)。主體可以包括帳戶、使用者、角色、聯合身分使用者或 AWS 服務。

資源型政策是位於該服務中的內嵌政策。您無法在資源型政策中使用來自 IAM 的 AWS 受管政策。

存取控制清單 (ACL)

存取控制清單 (ACL) 可控制哪些委託人 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

Amazon Simple Storage Service (Amazon S3)、AWS WAF 和 Amazon VPC 是支援 ACL 的服務範例。若要進一步了解 ACL，請參閱《Amazon Simple Storage Service 開發人員指南》中的[存取控制清單 \(ACL\) 概觀](#)。

其他政策類型

AWS 支援其他較少見的政策類型。這些政策類型可設定較常見政策類型授與您的最大許可。

- 許可界限 – 許可範圍是一種進階功能，可供您設定身分型政策能授予 IAM 實體 (IAM 使用者或角色) 的最大許可。您可以為實體設定許可界限。所產生的許可會是實體的身分型政策和其許可界限的交集。會在 Principal 欄位中指定使用者或角色的資源型政策則不會受到許可界限限制。所有這類政策中的明確拒絕都會覆寫該允許。如需許可範圍的更多相關資訊，請參閱《IAM 使用者指南》中的[IAM 實體許可範圍](#)。
- 服務控制政策 (SCP) – SCP 是 JSON 政策，可指定 AWS Organizations 中組織或組織單位 (OU) 的最大許可。AWS Organizations 服務可用來分組和集中管理您企業所擁有的多個 AWS 帳戶。若您啟用組織中的所有功能，您可以將服務控制政策 (SCP) 套用到任何或所有帳戶。SCP 會限制成員帳戶中實體的許可，包括每個 AWS 帳戶根使用者。如需組織和 SCP 的更多相關資訊，請參閱《AWS Organizations 使用者指南》中的[SCP 運作方式](#)。
- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過編寫程式的方式建立角色或聯合使用者的暫時工作階段時，作為參數傳遞。所產生工作階段的許可會是使用者或角色的身分型政策和工作

階段政策的交集。許可也可以來自資源型政策。所有這類政策中的明確拒絕都會覆寫該允許。如需更多資訊，請參閱《IAM 使用者指南》中的[工作階段政策](#)。

多種政策類型

將多種政策類型套用到請求時，其結果形成的許可會更為複雜、更加難以理解。若要了解 AWS 在涉及多種政策類型時如何判斷是否允許一項請求，請參閱《IAM 使用者指南》中的[政策評估邏輯](#)。

AWS SimSpace Weaver 搭配 IAM 的運作方式

在您使用 IAM 管理 SimSpace Weaver 的存取權之前，請了解搭配 SimSpace Weaver 使用的 IAM 功能有哪些。

您可搭配 AWS SimSpace Weaver 使用的 IAM 功能

IAM 功能	SimSpace Weaver 支援
身分型政策	是
資源型政策	否
政策動作	是
政策資源	是
政策條件索引鍵 (服務特定)	是
ACL	否
ABAC (政策中的標籤)	是
臨時憑證	是
主體許可	是
服務角色	是
服務連結角色	否

若要取得 SimSpace Weaver 和其他 AWS 服務如何搭配大部分 IAM 功能使用的概觀資訊，請參閱《IAM 使用者指南》中的[可搭配 IAM 使用的 AWS 服務](#)。

適用於 SimSpace Weaver 的身分型政策

支援身分型政策	是
---------	---

身分型政策是可以連接到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要瞭解如何建立身分類型政策，請參閱《IAM 使用者指南》中的[建立 IAM 政策](#)。

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及在何種條件下允許或拒絕動作。您無法在身分型政策中指定主體，因為這會套用至附加的使用者或角色。如要了解您在 JSON 政策中使用的所有元素，請參閱《IAM 使用者指南》中的[IAM JSON 政策元素參考](#)。

SimSpace Weaver 的身分型政策範例

若要檢視 SimSpace Weaver 身分型政策範例，請參閱[AWS SimSpace Weaver 的身分型政策範例](#)。

SimSpace Weaver 內的資源型政策

支援以資源基礎的政策	否
------------	---

資源型政策是附加到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中[指定主體](#)。主體可以包括帳戶、使用者、角色、聯合身分使用者或 AWS 服務。

若要啟用跨帳戶存取權，您可以指定在其他帳戶內的所有帳戶或 IAM 實體，作為資源型政策的主體。新增跨帳戶主體至資源型政策，只是建立信任關係的一半。當主體和資源在不同的 AWS 帳戶中時，受信任帳戶中的 IAM 管理員也必須授與主體實體 (使用者或角色) 存取資源的許可。其透過將身分型政策附加到實體來授予許可。不過，如果資源型政策會為相同帳戶中的主體授與存取，這時就不需要額外的身分型政策。如需詳細資訊，請參閱 IAM 使用者指南中的[IAM 角色與資源型政策有何差異](#)。

適用於 SimSpace Weaver 的政策動作

支援政策動作 是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

JSON 政策的 Action 元素描述您可以用來允許或拒絕政策中存取的動作。政策動作的名稱通常會和相關聯的 AWS API 操作相同。有一些例外狀況，例如沒有相符的 API 操作的僅限許可動作。也有一些操作需要政策中的多個動作。這些額外的動作稱為相依動作。

政策會使用動作來授與執行相關聯操作的許可。

如要查看 SimSpace Weaver 動作的清單，請參閱《服務授權參考》中的《[AWS SimSpace Weaver 定義的動作](#)》。

SimSpace Weaver 中的政策動作會在動作之前使用以下字首：

```
simspaceweaver
```

如需在單一陳述式中指定多個動作，請用逗號分隔。

```
"Action": [  
  "simspaceweaver:action1",  
  "simspaceweaver:action2"  
]
```

若要檢視 SimSpace Weaver 身分型政策範例，請參閱 [AWS SimSpace Weaver 的身分型政策範例](#)。

SimSpace Weaver 的政策資源

支援政策資源 是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Resource JSON 政策元素可指定要套用動作的物件。陳述式必須包含 Resource 或 NotResource 元素。最佳實務是使用其 [Amazon Resource Name \(ARN\)](#) 來指定資源。您可以針對支援特定資源類型的動作 (稱為資源層級許可) 來這麼做。

對於不支援資源層級許可的動作 (例如列出作業)，請使用萬用字元 (*) 來表示陳述式適用於所有資源。

```
"Resource": "*"
```

如要查看 SimSpace Weaver 資源類型及其 ARN 的清單，請參閱《服務授權參考》中的 [AWS SimSpace Weaver 定義的資源](#)。若要了解您可以使用哪些動作指定每個資源的 ARN，請參閱 [AWS SimSpace Weaver 定義的動作](#)。

若要檢視 SimSpace Weaver 身分型政策範例，請參閱 [AWS SimSpace Weaver 的身分型政策範例](#)。

SimSpace Weaver 的政策條件索引鍵

支援服務特定政策條件索引鍵 是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Condition 元素 (或 Condition 區塊)可讓您指定使陳述式生效的條件。Condition 元素是選用項目。您可以建立使用[條件運算子](#)的條件運算式 (例如等於或小於)，來比對政策中的條件和請求中的值。

若您在陳述式中指定多個 Condition 元素，或是在單一 Condition 元素中指定多個索引鍵，AWS 會使用邏輯 AND 操作評估他們。若您為單一條件索引鍵指定多個值，AWS 會使用邏輯 OR 操作評估條件。必須符合所有條件，才會授與陳述式的許可。

您也可以在指定條件時使用預留位置變數。例如，您可以只在使用者使用其 IAM 使用者名稱標記時，將存取資源的許可授與該 IAM 使用者。如需更多資訊，請參閱《IAM 使用者指南》中的 [IAM 政策元素：變數和標籤](#)。

AWS 支援全域條件索引鍵和服務特定的條件索引鍵。若要查看 AWS 全域條件索引鍵，請參閱《IAM 使用者指南》中的 [AWS 全域條件內容金鑰](#)。

如要查看 SimSpace Weaver 條件索引鍵的清單，請參閱《服務授權參考》中的 [AWS SimSpace Weaver 的條件索引鍵](#)。若要了解您可以針對何種動作及資源使用條件索引鍵，請參閱 [AWS SimSpace Weaver 定義的動作](#)。

若要檢視 SimSpace Weaver 身分型政策範例，請參閱 [AWS SimSpace Weaver 的身分型政策範例](#)。

SimSpace Weaver 中的存取控制清單 (ACL)

支援 ACL	否
--------	---

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

SimSpace Weaver 屬性型存取控制 (ABAC)

支援 ABAC (政策中的標籤)	是
------------------	---

屬性型存取控制 (ABAC) 是一種授權策略，可根據屬性來定義許可。在 AWS 中，這些屬性稱為標籤。您可以將標籤附加到 IAM 實體 (使用者或角色)，以及許多 AWS 資源。為實體和資源加上標籤是 ABAC 的第一步。您接著要設計 ABAC 政策，允許在主體的標籤與其嘗試存取的資源標籤相符時操作。

ABAC 在成長快速的環境中相當有幫助，並能在政策管理變得繁瑣時提供協助。

若要根據標籤控制存取，請使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 條件索引鍵，在政策的 [條件元素](#) 中，提供標籤資訊。

如果服務支援每個資源類型的全部三個條件索引鍵，則對該服務而言，值為 Yes。如果服務僅支援某些資源類型的全部三個條件索引鍵，則值為 Partial。

如需 ABAC 的詳細資訊，請參閱《IAM 使用者指南》中的 [什麼是 ABAC?](#)。如要查看含有設定 ABAC 步驟的教學課程，請參閱《IAM 使用者指南》中的 [使用屬性型存取控制 \(ABAC\)](#)。

將臨時憑證與 SimSpace Weaver 搭配使用

支援臨時憑證	是
--------	---

您使用臨時憑證進行登入時，某些 AWS 服務 無法運作。如需詳細資訊，包括那些 AWS 服務 搭配臨時憑證運作，請參閱 [《IAM 使用者指南》](#) 中的可搭配 IAM 運作的 AWS 服務。

如果您使用使用者名稱和密碼之外的任何方法登入 AWS Management Console，則您正在使用臨時憑證。例如，當您使用公司的單一登入(SSO)連結存取 AWS 時，該程序會自動建立臨時憑證。當您以使用者身分登入主控台，然後切換角色時，也會自動建立臨時憑證。如需切換角色的詳細資訊，請參閱 IAM 使用者指南中的[切換至角色 \(主控台\)](#)。

您可使用 AWS CLI 或 AWS API，手動建立臨時憑證。接著，您可以使用這些臨時憑證來存取 AWS。AWS 建議您動態產生臨時憑證，而非使用長期存取金鑰。如需詳細資訊，請參閱 [IAM 中的暫時性安全憑證](#)。

SimSpace Weaver 的跨服務主體權限

支援轉寄存取工作階段 (FAS) 是

當您使用 IAM 使用者或角色在 AWS 中執行動作時，您會被視為主體。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 使用主體的許可呼叫 AWS 服務，搭配請求 AWS 服務以向下游服務發出請求。只有在服務收到需要與其他 AWS 服務或資源互動才能完成的請求之後，才會提出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的策略詳細資訊，請參閱 [《轉發存取工作階段》](#)。

SimSpace Weaver 的服務角色

支援服務角色 是

服務角色是服務擔任的 [IAM 角色](#)，可代您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱《IAM 使用者指南》中的[建立角色以委派許可給 AWS 服務 服務](#)。

Warning

變更服務角色的許可有可能會讓 SimSpace Weaver 功能出現故障。只有 SimSpace Weaver 提供指引時，才能編輯服務角色。

SimSpace Weaver 應用程式 SDK 指令碼使用 AWS CloudFormation 範本在其他 AWS 服務中建立資源，以支援您的模擬。這些資源之一就是模擬的應用程式角色。SimSpace Weaver 假定應用程式角色代表您 AWS 帳戶執行操作，例如將日誌數據寫入 CloudWatch 日誌。如需有關應用程式角色的詳細資訊，請參閱 [為您建 SimSpace Weaver 立的權限](#)。

SimSpace Weaver 的服務連結角色

支援服務連結角色。

否

服務連結角色是一種連結到 AWS 服務的服務角色類型。服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的 AWS 帳戶中，並由該服務所擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

如需建立或管理服務連結角色的詳細資訊，請參閱[可搭配 IAM 運作的 AWS 服務](#)。在表格中尋找服務，其中包含服務連結角色欄中的 Yes。選擇 Yes (是) 連結，以檢視該服務的服務連結角色文件。

AWS SimSpace Weaver 的身分型政策範例

根據預設，使用者和角色不具備建立或修改 SimSpace Weaver 資源的權限。他們也無法使用 AWS Management Console、AWS Command Line Interface (AWS CLI) 或 AWS API 執行任務。若要授與使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

若要了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱《IAM 使用者指南》中的[建立 IAM 政策](#)。

如需 SimSpace Weaver 所定義之動作和資源類型的詳細資訊，包括各個資源類型的 ARN 格式，請參閱《服務授權參考》中的《[適用於 AWS SimSpace Weaver 的動作、資源和條件索引鍵](#)》。

主題

- [政策最佳實務](#)
- [使用 SimSpace Weaver 主控台](#)
- [允許使用者檢視他們自己的許可](#)
- [允許使用者建立和執行模擬](#)

政策最佳實務

身分型政策會判斷您帳戶中的某個人員是否可以建立、存取或刪除 SimSpace Weaver 資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管政策並朝向最低權限許可的目標邁進：如需開始授予許可給使用者和工作負載，請使用 AWS 受管政策，這些政策會授予許可給許多常用案例。它們可在您的 AWS 帳戶中使用。我

們建議您定義特定於使用案例的 AWS 客戶管理政策，以便進一步減少許可。如需更多資訊，請參閱 IAM 使用者指南中的 [AWS 受管政策](#) 或 [任務職能的 AWS 受管政策](#)。

- 套用最低許可許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的權限。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的更多相關資訊，請參閱 IAM 使用者指南中的 [IAM 中的政策和許可](#)。
- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。您也可以使用條件來授予對服務動作的存取權，前提是透過特定 AWS 服務 (例如 AWS CloudFormation) 使用條件。如需更多資訊，請參閱《IAM 使用者指南》中的 [IAM JSON 政策元素：條件](#)。
- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您編寫安全且實用的政策。如需更多資訊，請參閱 IAM 使用者指南中的 [IAM Access Analyzer 政策驗證](#)。
- 需要多重要素驗證 (MFA)：如果存在需要 AWS 帳戶中 IAM 使用者或根使用者的情況，請開啟 MFA 提供額外的安全性。如需在呼叫 API 操作時請求 MFA，請將 MFA 條件新增至您的政策。如需更多資訊，請參閱 [IAM 使用者指南](#) 中的設定 MFA 保護的 API 存取。

有關 IAM 中最佳實務的更多相關資訊，請參閱 IAM 使用者指南中的 [IAM 最佳安全實務](#)。

使用 SimSpace Weaver 主控台

若要存取 AWS SimSpace Weaver 主控台，您必須擁有最低的一組許可。這些許可必須允許您列出和檢視您 AWS 帳戶中 SimSpace Weaver 資源的詳細資訊。如果您建立比最基本必要許可更嚴格的身分型政策，則對於具有該政策的實體 (使用者或角色) 而言，主控台就無法如預期運作。

對於僅呼叫 AWS CLI 或 AWS API 的使用者，您不需要允許其最基本主控台許可。反之，只需允許存取符合他們嘗試執行之 API 操作的動作就可以了。

若要確保使用者和角色仍可使用 SimSpace Weaver 主控台，請同時將 SimSpace Weaver *ConsoleAccess* 或受 *ReadOnly* AWS 管理的原則附加至實體。如需詳細資訊，請參閱《IAM 使用者指南》中的 [新增許可到使用者](#)。

允許使用者檢視他們自己的許可

此範例會示範如何建立政策，允許 IAM 使用者檢視附加到他們使用者身分的內嵌及受管政策。此政策包含在主控台上，或是使用 AWS CLI 或 AWS API 透過編寫程式的方式完成此動作的許可。

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "ViewOwnUserInfo",
    "Effect": "Allow",
    "Action": [
      "iam:GetUserPolicy",
      "iam:ListGroupsWithUser",
      "iam:ListAttachedUserPolicies",
      "iam:ListUserPolicies",
      "iam:GetUser"
    ],
    "Resource": ["arn:aws:iam::*:user/${aws:username}"]
  },
  {
    "Sid": "NavigateInConsole",
    "Effect": "Allow",
    "Action": [
      "iam:GetGroupPolicy",
      "iam:GetPolicyVersion",
      "iam:GetPolicy",
      "iam:ListAttachedGroupPolicies",
      "iam:ListGroupPolicies",
      "iam:ListPolicyVersions",
      "iam:ListPolicies",
      "iam:ListUsers"
    ],
    "Resource": "*"
  }
]
}

```

允許使用者建立和執行模擬

此 IAM 政策範例提供在 SimSpace Weaver 中建立和執行模擬所需的基本許可。

```

{
  "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"
        }
    }
}
]
}

```

為您建SimSpace Weaver立的權限

建立SimSpace Weaver專案時，服務會建立名稱weaver-*project-name*-app-role和 IAM 信任政策的 AWS Identity and Access Management (IAM) 角色。信任政策SimSpace Weaver允許承擔角色，以便它可以為您執行操作。

應用程式角色權限原

模擬應用程式角色具有下列權限原則。

```
{
  "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": "*"
    }
  ]
}
```

應用程式角色信任原

SimSpace Weaver將信任關係新增至模擬應用程式角色做為[信任原則](#)。SimSpace Weaver為每個模擬建立信任原則，類似下列範例。

```
{
  "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

在此範例中，科目編號為111122223333，模擬名稱為MySimName。這些值在您的信任政策中有所不同。

預防跨服務混淆代理人

[混淆的副問題](#)是一個安全性問題，即沒有執行動作權限的實體可能會誘騙更具權限的實體執行動作。在AWS中，跨服務模擬可能會導致混淆代理人問題。在某個服務(呼叫服務)呼叫另一個服務(被呼叫服務)時，可能會發生跨服務模擬。可以操縱呼叫服務來使用其許可，以其不應有存取許可的方式對其他客戶的資源採取動作。為了預防這種情況，AWS提供的工具可協助您保護所有服務的資料，而這些服務主體已獲得您帳戶中資源的存取權。

若要限制AWS SimSpace Weaver為資源提供另一項服務的許可，我們建議在資源政策中使用[aws:SourceArn](#)和[aws:SourceAccount](#)全域條件內容索引鍵。如果aws:SourceArn值不包含帳戶ID(例如Amazon S3儲存貯體Amazon Resource Name (ARN))，則您必須使用這兩個全域條件

內容索引鍵來限制許可。如果同時使用這兩個全域條件內容索引鍵，且 `aws:SourceArn` 值包含帳戶 ID，則在相同政策陳述式中使用 `aws:SourceAccount` 值和 `aws:SourceArn` 值中的帳戶時，必須使用相同的帳戶 ID。如果您想要僅允許一個資源與跨服務存取相關聯，則請使用 `aws:SourceArn`。如果您想要允許該帳戶中的任何資源與跨服務使用相關聯，請使用 `aws:SourceAccount`。

`aws:SourceArn` 的值必須使用擴充功能的 ARN。

防範混淆代理人問題的最有效方法是使用 `aws:SourceArn` 全域條件內容索引鍵，以及資源的完整 ARN。如果您不知道擴充功能的完整 ARN，或者如果您指定了多個擴充功能，請使用 `aws:SourceArn` 全域內容條件索引鍵搭配萬用字元 (*) 來表示 ARN 的未知部分。例如 `arn:aws:simspaceweaver:*:111122223333:*`。

下列範例示範如何使用 SimSpace Weaver 中的 `aws:SourceArn` 和 `aws:SourceAccount` 全域條件內容金鑰，來預防混淆代理人問題。只有當要求 SimSpace Weaver 來自指定的來源帳戶並提供指定的 ARN 時，此原則才允許擔任角色。#####*SimSpace Weaver*##### (111122223333) ###### (#### -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/*"
        }
      }
    }
  ]
}
```

撰寫此原則的更安全的方法是在中包含模擬名稱 `aws:SourceArn`，如下列範例所示，這樣會將策略限制為名 `MyProjectSimulation_22-10-04_22_10_15` 為的模擬：


```
{
  "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"
        }
      }
    }
  ]
}
```

當您aws:SourceArn明確包含帳號時，您可以省略的Condition元素測試 aws:SourceAccount (如需詳細資訊，請參閱 [IAM 使用者指南](#))，例如下列簡化政策所示：

```
{
  "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"
        }
      }
    }
  ]
}
```

```
    }  
  }  
]  
}
```

對 AWS SimSpace Weaver 身分與存取進行疑難排解

請使用以下資訊來協助您診斷和修復使用 SimSpace Weaver 和 IAM 時發生的常見問題。

主題

- [我未獲授權，不得在 SimSpace Weaver 中執行動作](#)
- [我沒有授權執行 iam : PassRole](#)
- [我想要檢視我的存取金鑰](#)
- [我是管理員，並且想要允許其他人存取 SimSpace Weaver](#)
- [我想要允許 AWS 帳戶外的人員存取我的 SimSpace Weaver 資源](#)

我未獲授權，不得在 SimSpace Weaver 中執行動作

若 AWS Management Console 告知您並未獲得執行動作的授權，您必須聯絡您的管理員以取得協助。您的管理員是提供您使用者名稱和密碼的人員。

以下範例錯誤會在 mateojackson IAM 使用者嘗試使用主控台檢視虛構 *my-example-widget* 資源的詳細資訊，但卻沒有虛構 `simspaceweaver:GetWidget` 許可時發生。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:  
simspaceweaver:GetWidget on resource: my-example-widget
```

在此情況下，Mateo 會請求管理員更新他的政策，允許他使用 *my-example-widget* 動作存取 `simspaceweaver:GetWidget` 資源。

我沒有授權執行 iam : PassRole

如果您收到錯誤，告知您未獲授權執行 `iam:PassRole` 動作，您的政策必須更新，允許您將角色傳遞給 SimSpace Weaver。

有些 AWS 服務 允許您傳遞現有的角色至該服務，而無須建立新的服務角色或服務連結角色。如需執行此作業，您必須擁有將角色傳遞至該服務的許可。

名為 marymajor 的 IAM 使用者嘗試使用主控台在 SimSpace Weaver 中執行動作時，發生下列範例錯誤。但是，動作要求服務具備服務角色授予的許可。Mary 沒有將角色傳遞至該服務的許可。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

在這種情況下，Mary 的政策必須更新，允許她執行 iam:PassRole 動作。

如需任何協助，請聯絡您的 AWS 管理員。您的管理員提供您的登入憑證。

我想要檢視我的存取金鑰

在您建立 IAM 使用者存取金鑰後，您可以隨時檢視您的存取金鑰 ID。但是，您無法再次檢視您的私密存取金鑰。若您遺失了密碼金鑰，您必須建立新的存取金鑰對。

存取金鑰包含兩個部分：存取金鑰 ID (例如 AKIAIOSFODNN7EXAMPLE) 和私密存取金鑰 (例如 wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY)。如同使用者名稱和密碼，您必須一起使用存取金鑰 ID 和私密存取金鑰來驗證您的請求。就如對您的使用者名稱和密碼一樣，安全地管理您的存取金鑰。

Important

請勿將您的存取金鑰提供給第三方，甚至是協助[尋找您的標準使用者 ID](#)。透過執行此操作，可能會讓他人永久存取您的 AWS 帳戶。

建立存取金鑰對時，您會收到提示，要求您將存取金鑰 ID 和私密存取金鑰儲存在安全位置。私密存取金鑰只會在您建立它的時候顯示一次。若您遺失了私密存取金鑰，您必須將新的存取金鑰新增到您的 IAM 使用者。您最多可以擁有兩個存取金鑰。若您已有兩個存取金鑰，您必須先刪除其中一個金鑰對，才能建立新的金鑰對。若要檢視說明，請參閱《IAM 使用者指南》中的[管理存取金鑰](#)。

我是管理員，並且想要允許其他人存取 SimSpace Weaver

若要允許其他人存取 SimSpace Weaver，您必須針對需要存取的人員或應用程式建立 IAM 實體 (使用者或角色)。他們將使用該實體的憑證來存取 AWS。您接著必須將政策連接到實體，在 SimSpace Weaver 中授予他們正確的許可。

若要立即開始使用，請參閱《IAM 使用者指南》中的[建立您的第一個 IAM 委派使用者及群組](#)。

我想要允許 AWS 帳戶 外的人員存取我的 SimSpace Weaver 資源

您可以建立一個角色，讓其他帳戶中的使用者或您的組織外部的人員存取您的資源。您可以指定要允許哪些信任物件取得該角色。針對支援基於資源的政策或存取控制清單 (ACL) 的服務，您可以使用那些政策來授予人員存取您的資源的許可。

如需進一步了解，請參閱以下內容：

- 若要了解 SimSpace Weaver 是否支援這些功能，請參閱 [AWS SimSpace Weaver 搭配 IAM 的運作方式](#)。
- 若要了解如何存取您擁有的所有 AWS 帳戶 所提供的資源，請參閱《IAM 使用者指南》中的[將存取權提供給您所擁有的另一個 AWS 帳戶 中的 IAM 使用者](#)。
- 如需了解如何將資源的存取權提供給第三方 AWS 帳戶，請參閱《IAM 使用者指南》中的[將存取權提供給第三方擁有的 AWS 帳戶](#)。
- 如需了解如何透過聯合身分提供存取權，請參閱《IAM 使用者指南》中的[將存取權提供給在外部進行身分驗證的使用者 \(聯合身分\)](#)。
- 如需了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱 IAM 使用者指南中的 [IAM 角色與資源型政策的差異](#)。

安全事件記錄和監控 AWS SimSpace Weaver

監控是維護 SimSpace Weaver 及您 AWS 解決方案可靠性、可用性和效能的重要部分。您應該收集 AWS 解決方案全面的監控資料，以便在出現多點故障時更輕鬆地進行偵錯。

AWS 並 SimSpace Weaver 提供數種工具來監控您的模擬資源並回應潛在事件。

Amazon 日誌 CloudWatch

SimSpace Weaver 將其日誌存儲在中 CloudWatch。您可以使用這些記錄來監視模擬中的事件 (例如啟動和停止應用程式) 以及偵錯。如需詳細資訊，請參閱 [SimSpace Weaver Amazon 日誌中的 CloudWatch 日誌](#)。

Amazon CloudWatch 警報

使用 Amazon CloudWatch 警示，您可以在指定的時間段內觀看單一指標。如果指標超過指定臨界值，則會向 Amazon SNS 主題或 AWS Auto Scaling 政策傳送通知。CloudWatch 警示會在其狀態變更時觸發，並且會在指定的期間數目 (而非處於特定狀態) 維護警示。如需詳細資訊，請參閱 [SimSpace Weaver 使用 Amazon 監控 CloudWatch](#)。

AWS CloudTrail 日誌

CloudTrail 提供使用者、角色或AWS服務所採取之動作的記錄SimSpace Weaver。使用收集的資訊 CloudTrail，您可以判斷提出的要求SimSpace Weaver、提出要求的 IP 位址、提出要求的人員、提出要求的時間，以及其他詳細資訊。如需詳細資訊，請參閱 [使用 AWS CloudTrail 記錄 AWS SimSpace Weaver API 呼叫](#)。

AWS SimSpace Weaver 的合規驗證

SimSpace Weaver 不在任何 AWS 合規計畫範圍內。

協力廠商稽核人員會評估其他AWS服務的安全性與合規性，作為多項AWS合規計畫的一部分。這些計畫包括 SOC、PCI、FedRAMP、HIPAA 等等。

要了解 AWS 服務 是否在特定法規遵循方案範圍內，請參閱[法規遵循方案範圍內的 AWS 服務](#)，並選擇您感興趣的法規遵循方案。如需一般資訊，請參閱 [AWS 法規遵循方案](#)。

您可使用 AWS Artifact 下載第三方稽核報告。如需詳細資訊，請參閱 [AWS Artifact 中的下載報告](#)。

您使用 AWS 服務 時的法規遵循責任取決於資料的敏感度、您的公司的合規目標，以及適用的法律和法規。AWS 提供以下資源協助您處理法規遵循事宜：

- [安全與合規快速入門指南](#) – 這些部署指南討論在 AWS 上部署以安全及合規為重心的基準環境的架構考量和步驟。
- [Amazon Web Services 的 HIPAA 安全與法規遵循架構](#)：本白皮書說明公司可如何運用 AWS 來建立符合 HIPAA 規定的應用程式。

Note

並非全部的 AWS 服務 都符合 HIPAA 資格。如需詳細資訊，請參閱 [HIPAA 資格服務參照](#)。

- [AWS 合規資源](#)：這組手冊和指南可能適用於您的產業和位置。
- [AWS 客戶合規指南](#)：透過合規的角度了解共同的責任模式。這份指南橫跨多個架構 (包含國家標準技術研究所 (NIST)、支付卡產業安全標準委員會 (PCI) 和國際標準組織 (ISO))，總結保護 AWS 服務的最佳實務並將指導方針對應至安全控制。
- AWS Config 開發人員指南中的 [使用規則評估資源](#)：AWS Config 服務可評估您的資源組態對於內部實務、業界準則和法規的合規狀態。

- [AWS Security Hub](#) – 此 AWS 服務 可供您全面檢視 AWS 中的安全狀態。Security Hub 使用安全控制，可評估您的 AWS 資源並檢查您的法規遵循是否符合安全業界標準和最佳實務。如需支援的服務和控制清單，請參閱 [Security Hub controls reference](#)。
- [AWS Audit Manager](#) – 此 AWS 服務 可協助您持續稽核 AWS 使用情況，以簡化管理風險與法規與業界標準的法規遵循方式。

AWS SimSpace Weaver 中的恢復能力

AWS 全球基礎架構是以 AWS 區域 與可用區域為中心建置的。AWS 區域 提供多個分開且隔離的實際可用區域，並以具備低延遲、高輸送量和高度備援特性的聯網相互連結。透過可用區域，您可以設計與操作的應用程式和資料庫，在可用區域之間自動容錯移轉而不會發生中斷。可用區域的可用性、容錯能力和可擴展性能力，均較單一或多個資料中心的傳統基礎設施還高。

如需 AWS 區域 與可用區域的詳細資訊，請參閱[AWS全球基礎架構](#)。

AWS SimSpace Weaver 中的基礎設施安全

作為一種受管服務，AWS SimSpace Weaver 受 AWS 全域網路安全的保護。如需 AWS 安全服務以及 AWS 如何保護基礎設施的相關資訊，請參閱 [AWS 雲端安全](#)。若要使用基礎設施安全性的最佳實務以設計您的 AWS 環境，請參閱安全性支柱 AWS 架構良好的框架中的[基礎設施保護](#)。

您可使用 AWS 發佈的 API 呼叫，透過網路存取 SimSpace Weaver。用戶端必須支援下列項目：

- Transport Layer Security (TLS)。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 具備完美轉送私密(PFS)的密碼套件，例如 DHE (Ephemeral Diffie-Hellman)或 ECDHE (Elliptic Curve Ephemeral Diffie-Hellman)。現代系統(如 Java 7 和更新版本)大多會支援這些模式。

此外，請求必須使用存取金鑰 ID 和與 IAM 主體相關聯的私密存取金鑰來簽署。或者，您可以透過 [AWS Security Token Service](#) (AWS STS) 來產生暫時安全憑證來簽署請求。

網路連線安全性模型

您的模擬會在 Amazon VPC 內的運算執行個體上執行，位於您選取的AWS區域內。Amazon VPC 是 AWS雲端中的虛擬網路，可依工作負載或組織實體隔離基礎設施。Amazon VPC 內的運算執行個體之間的通訊會保留在AWS網路內，而且不會透過網際網路傳輸。一些內部服務通信穿過互聯網，並且是加密的。在相同區域中執行的所有客戶的模擬都會AWS共用相同的 Amazon VPC。不同客戶的模擬會在同一個 Amazon VPC 中使用不同的運算執行個體。

您的模擬用戶端與透過網際網路SimSpace Weaver旅行時執行的模擬之間的通訊。 SimSpace Weaver 不會處理這些連線。保護您的客戶連接是您的責任。

您透過網際網路與SimSpace Weaver服務的連線並經過加密。這包括使用AWS Management Console、AWS Command Line Interface (AWS CLI)、AWS軟體開發套件 (SDK) 和應用SimSpace Weaver程式 SDK 的連線。

AWS SimSpace Weaver 中的組態與漏洞分析

配置和 IT 控制是與您之間共同AWS的責任。如需詳細資訊，請參閱AWS[共用責任模型](#)。AWS處理基礎架構的基本安全性工作，例如在運算執行個體上修補作業系統、防火牆組態和基AWS礎結構災難復原。這些程序已由適當的第三方進行檢閱並認證。如需詳細資訊，請參閱[安全性、身分識別和合規性的最佳做法](#)。

您必須對模擬軟體的安全性負責：

- 維護您的應用程式碼，包括更新和安全性修補程式。
- 驗證和加密模擬用戶端與其連線到的應用程式之間的通訊。
- 更新您的模擬以使用最新的 SDK 版本，包括 AWS SDK 和SimSpace Weaver應用程式 SDK。

Note

SimSpace Weaver不支援執行中模擬中應用程式的更新。如果您需要更新應用程式，您必須停止並刪除模擬，然後使用更新的應用程式程式碼建立新模擬。建議您將模擬狀態儲存在外部資料倉庫中，以便在需要重新建立模擬時可以還原模擬。

SimSpace Weaver 的安全最佳實務

本節說明特定的安全性最佳作法SimSpace Weaver。若要深入了解中的安全性最佳做法AWS，請參閱[安全性、身分識別和合規性的最佳做法](#)。

主題

- [加密應用程式與其用戶端之間的通訊](#)
- [定期備份模擬狀態](#)
- [維護您的應用程式和 SDK](#)

加密應用程式與其用戶端之間的通訊

SimSpace Weaver不會管理您的應用程式與其用戶端之間的通訊。您應該為客戶端會話實施某種形式的身份驗證和加密。

定期備份模擬狀態

SimSpace Weaver不會儲存您的模擬狀態。停止的模擬 (由於 API 呼叫、主控台選項或系統當機) 不會儲存其狀態，也沒有固有的方法可以復原它們。停止的模擬無法重新啟動。執行等效重新啟動的唯一方法是使用相同的規劃和資料重新建立模擬。您可以使用模擬狀態的備份來初始化新模擬。AWS提供高度可靠且可用的雲端[儲存](#)和[資料庫](#)服務，您可以使用這些服務來儲存模擬狀態。

維護您的應用程式和 SDK

維護您的應用程式、AWS軟體開發套件 (SDK) 的本機安裝，以及SimSpace Weaver應用程式 SDK。您可以下載並安裝新版本的 AWS SDK。使用非生產SimSpace Weaver應用程式組建來測試新版應用程式 SDK，以確保您的應用程式繼續如預期般執行。您無法在執行中的模擬中更新應用程式。若要更新您的應用程式：

1. 在本地 (或在測試環境中) 更新和測試應用程序代碼。
2. 停止變更模擬狀態並將其儲存 (如有必要)。
3. 停止模擬 (一旦停止，就無法重新啟動)。
4. 刪除您的模擬 (未刪除的已停止模擬會計入您的服務限制)。
5. 使用相同的配置和更新的應用程序代碼重新創建模擬。
6. 使用儲存的狀態資料 (如果可用) 初始化模擬。
7. 開始您的新模擬。

Note

使用相同規劃建立的新模擬與舊模擬分開。它將有一個新的模擬 ID，並將日誌發送到 Amazon 中的新日誌流 CloudWatch。

SimSpace Weaver 中的日誌記錄和監控

監控是維護 SimSpace Weaver 及其他 AWS 解決方案的可靠性、可用性和效能的重要部分。AWS 提供以下監控工具，可讓您監看 SimSpace Weaver、在發現錯誤時回報，並適時採取自動動作。

- Amazon 會即時 CloudWatch 監控您的 AWS 資源和執行 AWS 的應用程式。您可以收集和追蹤指標、建立自訂儀板表，以及設定警示，在特定指標達到您指定的閾值時通知您或採取動作。如需詳細資訊，請參閱 [Amazon CloudWatch 使用者指南](#)。
- Amazon CloudWatch Logs 可讓您監控、存放和存取 SimSpace Weaver 員工和其他來源的日誌資料。CloudTrail CloudWatch 記錄檔可以監控記錄資料中的資訊，並在符合特定臨界值時通知您。您也可以將日誌資料存檔在高耐用性的儲存空間。如需詳細資訊，請參閱 [Amazon CloudWatch 日誌使用者指南](#)。
- AWS CloudTrail 擷取您 AWS 帳戶 發出或代表發出的 API 呼叫和相關事件，並傳送日誌檔案至您指定的 Amazon S3 儲存貯體。您可以找出哪些使用者和帳戶呼叫 AWS、發出呼叫的來源 IP 地址，以及呼叫的發生時間。如需詳細資訊，請參閱《AWS CloudTrail 使用者指南》<https://docs.aws.amazon.com/awscloudtrail/latest/userguide/>。

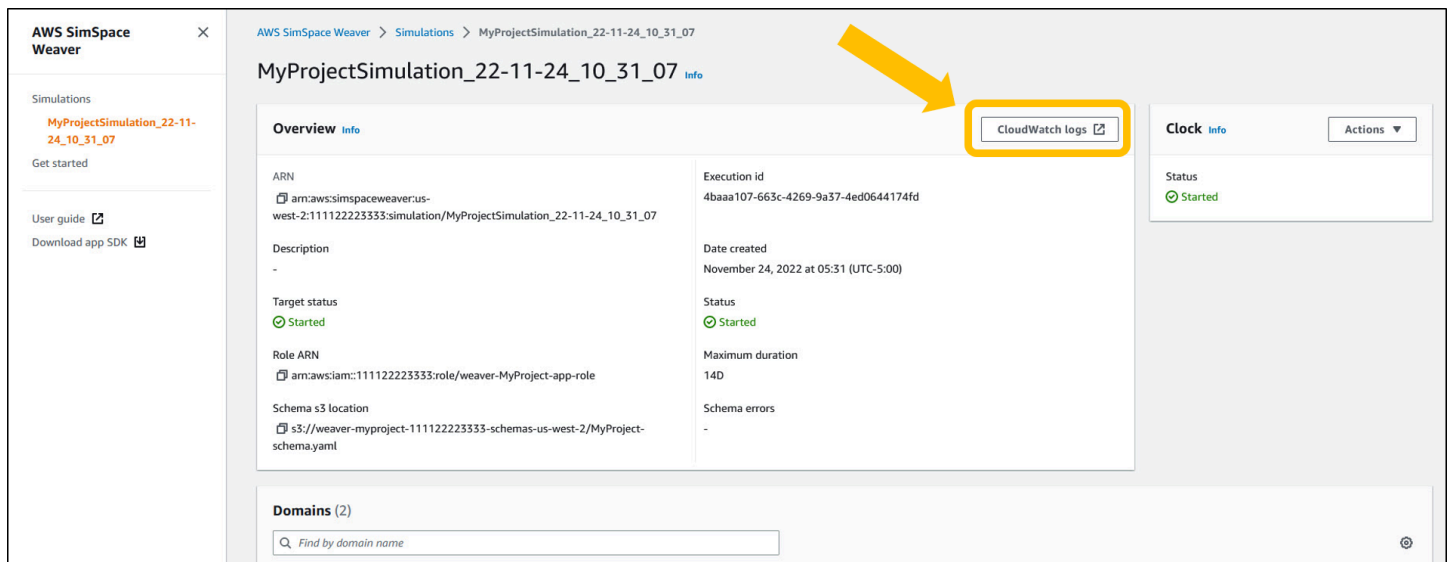
主題

- [SimSpace Weaver Amazon 日誌中的 CloudWatch 日誌](#)
- [SimSpace Weaver 使用 Amazon 監控 CloudWatch](#)
- [使用 AWS CloudTrail 記錄 AWS SimSpace Weaver API 呼叫](#)

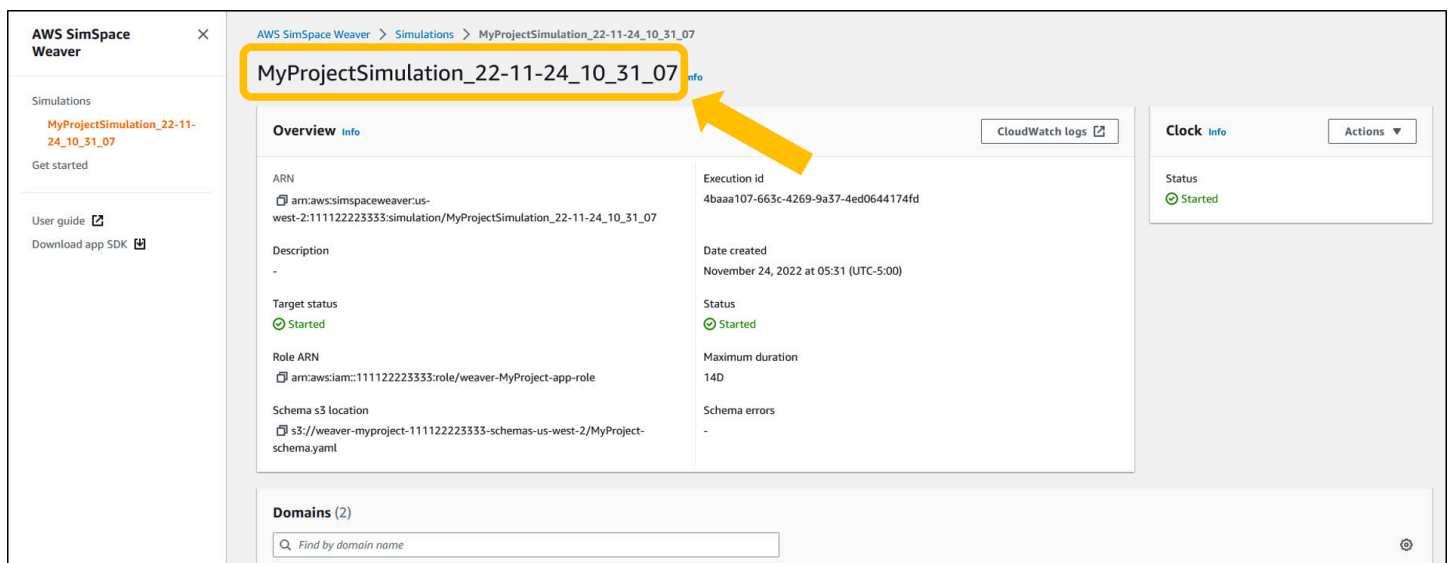
SimSpace Weaver Amazon 日誌中的 CloudWatch 日誌

存取您的 SimSpace Weaver 記錄

從您的 SimSpace Weaver 模擬產生的所有日誌都會儲存在 Amazon CloudWatch 日誌中。若要存取記錄，您可以在 SimSpace Weaver 主控台中使用模擬的「概觀」窗格中的「CloudWatch 記錄」按鈕，這會將您直接帶到該特定模擬的記錄檔。



您也可以透過 CloudWatch 主控台存取記錄檔。您需要模擬的名稱才能搜尋其記錄。



SimSpace Weaver 日誌

SimSpace Weaver 將模擬管理訊息和主控台輸出從您的應用程式寫入 Amazon CloudWatch 日誌。如需使用日誌的詳細資訊，請參閱 Amazon 日誌使用者指南中的使用日誌群組和 CloudWatch 日誌串流。

您建立的每個模擬在記錄檔中都有自己的 CloudWatch 記錄群組。記錄群組的名稱是在模擬結構描述中指定的。在下列結構描述程式碼片段中，的值 `log_destination_service` 為 `logs`。這表示的值 `log_destination_resource_name` 是記錄群組的名稱。在此情況下，記錄群組為 `MySimulationLogs`。

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

您也可以在此啟動後使用 DescribeSimulation API 來尋找用於模擬的記錄群組名稱。

Important

如果您為 AWS Command Line Interface (AWS CLI) 使用 AWS IAM Identity Center 或命名描述檔，您必須使用 SimSpace Weaver 應用程式 SDK 1.12.1 版或更新版本。最新的版本是 1.16.0。如需 SimSpace Weaver 版本的詳細資訊，請參閱 [SimSpace Weaver 版本](#)。SimSpace Weaver 應用程式 SDK 指令碼會使用 AWS CLI。如果您使用 IAM 身分中心，您可以將您的 IAM 身分中心設定檔複製到您的 default 設定檔，或將 IAM 身分中心設定檔的名稱提供給具有 `--profile cli-profile-name` 參數的 SimSpace Weaver 應用程式 SDK 指令碼。如需詳細資訊，請參閱 [《使 AWS CLI 用 AWS Command Line Interface 者指南》](#) AWS IAM Identity Center 中的 [〈配置使用指南〉](#) 和 [〈組態和認證檔案設 AWS Command Line Interface 定〉](#)。

Docker

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

WSL

Important

為了您的方便，我們提供這些說明。它們適用於搭配使用 Windows Subsystem for Linux (WSL)，且不受支援。如需詳細資訊，請參閱 [設定您的本機環境 SimSpace Weaver](#)。

```
project-folder/tools/linux/weaver-project-name-cli.sh describe-simulation --simulation simulation-name
```

下列範例會顯示描述記錄組態的輸出DescribeSimulation部分。記錄群組的名稱會顯示在的結尾LogGroupArn。

```

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

```

每個模擬記錄群組都包含數個記錄資料流：

- 管理日誌流 — SimSpace Weaver 服務產生的模擬管理消息。

```
/sim/management
```

- 錯誤記錄資料流 — SimSpace Weaver 服務產生的錯誤訊息。此記錄資料流只有在發生錯誤時才存在。SimSpace Weaver將您的應用程式寫入的錯誤存儲在其自己的應用程式日誌流中（請參閱以下日誌流）。

```
/sim/errors
```

- 空間應用程式日誌流（每個工作人員上的每個空間應用程式 1 個）-空間應用程式生成的控制台輸出。每個空間應用程式都會寫入自己的記錄資料流。*spatial-app-id*是結尾處結尾斜線之後的所有字元*worker-id*。

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

- 自定義應用程式日誌流（每個自定義應用實例 1 個）-自定義應用程式生成的控制台輸出。每個自定義應用程式實例都寫入自己的日誌流。

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

- 服務應用程式日誌流 (每個服務應用程式實例 1 個) - 服務應用程式生成的控制台輸出。每個服務應用程式都會寫入自己的記錄資料流。*service-app-id* 是結尾處結尾斜線之後的所有字元 *service-app-name*。

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

SimSpace Weaver 使用 Amazon 監控 CloudWatch

您可以 SimSpace Weaver 使用 Amazon 進行監控 CloudWatch，Amazon 會收集原始資料並將其處理為可讀且接近即時的指標。這些統計資料會保留 15 個月，以便您存取歷史資訊，並更清楚 Web 應用程式或服務的執行效能。您也可以設定留意特定閾值的警示，當滿足這些閾值時傳送通知或採取動作。如需詳細資訊，請參閱 [Amazon CloudWatch 使用者指南](#)。

SimSpace Weaver 服務會在 AWS/simspaceweaver 命名空間中報告下列度量。

SimSpace Weaver 帳戶層級的指標

命名空間 SimSpace Weaver 包含下列與 AWS 帳戶層級活動相關的量度。

指標	描述
SimulationCount	目前帳戶的模擬次數。 單位：計數 尺寸：無 統計：平均值、最小值、最大值

使用 AWS CloudTrail 記錄 AWS SimSpace Weaver API 呼叫

AWS SimSpace Weaver 與 (提供中的使用者 AWS CloudTrail、角色或服務所採取的動作記錄) 的 AWS 服務整合 SimSpace Weaver。CloudTrail 擷取 SimSpace Weaver 作為事件的所有 API 呼叫。擷取的呼叫包括從 SimSpace Weaver 主控台進行的呼叫，以及針對 SimSpace Weaver API 操作的程式碼呼叫。如果您建立追蹤，您可以啟用持續交付 CloudTrail 事件到 Amazon S3 儲存貯體，包括 SimSpace Weaver。如果您未設定追蹤，您仍然可以在中的 CloudTrail 主控台中檢視最近的事件 Event history。使

用收集的資訊 CloudTrail，您可以判斷提出的要求 SimSpace Weaver、提出要求的 IP 位址、提出要求的人員、提出要求的時間，以及其他詳細資訊。

若要進一步了解 CloudTrail，請參閱使 [AWS CloudTrail 用者指南](#)。

SimSpace Weaver 中的資訊 CloudTrail

CloudTrail 在您創建帳戶 AWS 帳戶時啟用。中發生活動時 SimSpace Weaver，該活動會與中的其他 AWS 服務 CloudTrail 事件一起記錄在事件中 Event history。您可以檢視、搜尋和下載 AWS 帳戶 的最新事件。如需詳細資訊，請參閱 [使用 CloudTrail 事件歷程記錄檢視事件](#)。

如需您 AWS 帳戶 帳戶中正在進行事件的記錄 (包含 SimSpace Weaver 的事件)，請建立追蹤。追蹤可 CloudTrail 將日誌檔交付到 Amazon S3 儲存貯體。依預設，當您在主控台中建立追蹤時，該追蹤會套用至所有的 AWS 區域。該追蹤會記錄來自 AWS 分割區中所有區域的事件，並將日誌檔案交付到您指定的 Amazon S3 儲存貯體。此外，您還可以設定其他 AWS 服務，以進一步分析 CloudTrail 記錄中收集的事件資料並採取行動。如需詳細資訊，請參閱下列內容：

- [建立追蹤的概觀](#)
- [CloudTrail 支援的服務與整合](#)
- [設定 Amazon SNS 通知 CloudTrail](#)
- [從多個區域接收 CloudTrail 日誌文件並從多個帳戶接收 CloudTrail 日誌文件](#)

所有 SimSpace Weaver 動作均由「API 參考」記錄 CloudTrail 並記錄在「[AWS SimSpace Weaver API 參考](#)」中。例如，呼叫 DescribeSimulation 和 DeleteSimulation 動作會 ListSimulations 在 CloudTrail 記錄檔中產生項目。

每一筆事件或日誌項目都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 該請求是否透過根或 AWS Identity and Access Management (IAM) 使用者憑證來提出。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 該請求是否由另一項 AWS 服務提出。

如需詳細資訊，請參閱 [CloudTrail userIdentity 元素](#)。

了解 SimSpace Weaver 日誌檔案項目

追蹤是一種組態，可讓事件以日誌檔的形式傳遞到您指定的 Amazon S3 儲存貯體。CloudTrail 記錄檔包含一或多個記錄項目。事件代表來自任何來源的單一請求，並包含有關請求動作的信息，例如操作的

日期和時間，請求參數和其他詳細信息。CloudTrail 日誌文件不是公共 API 調用的有序堆棧跟踪，因此它們不會以任何特定順序顯示。

下列範例顯示示範ListSimulations動作的 CloudTrail 記錄項目。

```
{
  "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 韋弗端點和配額

下表說明 SimSpace Weaver 的服務端點和服務配額。服務配額 (也稱為限制) 是您的服務資源或作業數目上限AWS 帳戶。如需詳細資訊，請參閱 AWS 一般參考中的 [AWS 服務配額](#)。

服務端點

區域名稱	區域	端點	通訊協定
美國東部 (維吉尼亞北部)	us-east-1	simspaceweaver.us-east-1.amazonaws.com	HTTPS
美國東部 (俄亥俄)	us-east-2	simspaceweaver.us-east-2.amazonaws.com	HTTPS
美國西部 (奧勒岡)	us-west-2	simspaceweaver.us-west-2.amazonaws.com	HTTPS
亞太區域 (新加坡)	ap-southeast-1	simspaceweaver.ap-southeast-1.amazonaws.com	HTTPS
亞太區域 (雪梨)	ap-southeast-2	simspaceweaver.ap-southeast-2.amazonaws.com	HTTPS
歐洲 (斯德哥爾摩)	eu-north-1	simspaceweaver.eu-north-1.amazonaws.com	HTTPS
歐洲 (法蘭克福)	eu-central-1	simspaceweaver.eu-central-1.amazonaws.com	HTTPS

區域名稱	區域	端點	通訊協定
歐洲 (愛爾蘭)	eu-west-1	simspaceweaver.eu-west-1.amazonaws.com	HTTPS
AWS GovCloud (美國東部)	us-gov-east-1	simspaceweaver.us-gov-east-1.amazonaws.com	HTTPS
AWS GovCloud (美國西部)	us-gov-west-1	simspaceweaver.us-gov-west-1.amazonaws.com	HTTPS

Service Quotas

名稱	預設	可調整	描述
計算每個應用程式的資源單位	每個受支援的區域：4	否	您可以配置給每個應用程式的運算資源單位數目上限。
計算每個工作站的資源單位	每個支持地區：17	否	每個 Worker 可用的計算資源單位數目。
每個實體的資料欄位	每個支持地區：7	否	實體可以擁有的資料 (非索引) 欄位數目上限。
分割區中的實體	每個支持的地區：8,192	否	1 個分區中的最大實體數量。
實體資料欄位大小	每個支持的地區：1,024 字節	否	實體資料 (非索引) 欄位的大小上限。

名稱	預設	可調整	描述
員工之間的實體轉移	每個受支援的區域：25	否	每個分區和每個刻度之間的實體傳輸數目上限。
在同一個 Worker 上進行實體轉移	每個受支援的區域：500	否	相同 Worker、每個分割區和每個刻度的實體傳輸數目上限。
每個實體的索引欄位	每個受支援的區域：1	否	實體可以擁有的索引欄位數目上限。
模擬的最大持續時間上限 (以天為單位)	每個支持地區：14	否	您可以指定為模擬持續時間上限的最大天數。即使您未指定值，所有模擬都有最長持續時間。模擬達到其最長持續時間時會自動停止。
每個運算資源單位的記憶體	每個支持的地區：1 GB	否	應用程式為每個計算資源單位取得的隨機存取記憶體 (RAM) 數量。
每個工作人員的遠端訂閱	每個支持地區：24	否	每個 Worker 的遠端訂閱數目上限。
模擬計數	每個支持地區：2	<u>是</u>	帳戶中目標狀態為 STARE 的最大模擬次數。您最多可以要求增加 10 個配額。
工人, 為, a, 模擬	每個支持地區：2	<u>是</u>	您可以指定給 1 個模擬的最大工作者數目。您最多可以要求增加 10 個配額。

名稱	預設	可調整	描述
每個運算資源單位的 vCPUs	每個支持地區：2	否	應用程式為每個計算資源單元取得的虛擬中央處理單元 (vCPUs) 數目。

郵件配額

下列配額適用於 SimSpace Weaver Local 和中的應用程式至應用程式訊息傳送 AWS 雲端。

名稱	預設	可調整	描述
訊息大小上限	每個支援的地區：256 位元組	否	訊息承載的大小上限。
訊息傳送速率上限	每個支持地區：128	否	每個應用程序每個刻度可以發送的消息數量上限。

時脈速率

模擬資料架構指定模擬的時脈速率 (也稱為刻度速率)。下表指定您可以使用的有效時脈速率。

名稱	有效值	描述
時脈速率	每個支援地區：「10」、「15」、「30」、「無限」	模擬的有效時脈速率。
時脈速率 (版本 1.13 和 1.12 版本)	每個支持的地區：	模擬的有效時脈速率。

的服務配額 SimSpace Weaver Local

下列服務配額 SimSpace Weaver Local 僅適用於。所有其他配額也適用於 SimSpace Weaver Local。

名稱	預設	可調整	描述
最大分割區	SimSpace Weaver Local : 24	否	模擬的最大分割區數目。
最大的應用	SimSpace Weaver Local : 24	否	模擬應用程式 (任何類型) 的最大總數。
最大網域	SimSpace Weaver Local : 24	否	模擬的最大網域總數 (任何類型)。
每個分區的實體	SimSpace Weaver Local: 4,096	否	每個分區中的最大實體數量。
每個實體的欄位	SimSpace Weaver Local : 八	否	每個實體的欄位數目上限。
欄位大小	SimSpace Weaver Local: 1 個位元組	否	實體字段的最大大小。

疑難排解 SimSpace Weaver

主題

- [AssumeRoleAccessDenied](#)
- [InvalidBucketName](#)
- [ServiceQuotaExceededException](#)
- [TooManyBuckets](#)
- [模擬啟動期間拒絕權限](#)
- [與使用時間有關的問題 Docker](#)
- [PathfindingSample 主控台用戶端無法連線](#)
- [無AWS CLI法識別 simspaceweaver](#)

AssumeRoleAccessDenied

如果模擬無法啟動，您可能會收到下列錯誤：

```
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
```

如果您模擬的 AWS Identity and Access Management (IAM) 角色符合以下條件之一，則您可能會收到此錯誤：

- 亞馬遜資源名稱(ARN)是指不存在的 IAM 角色。
- IAM 角色的信任政策，不允許新模擬的名稱擔任該角色。

檢查以確定角色存在。如果角色存在，請檢查角色的信任原則。aws:SourceArn在下列範例中，信任原則只允許名稱開頭為角色的模擬 (帳戶 111122223333 中)。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*"
      }
    }
  }
]
}

```

若要允許名稱開頭為的MyOtherSimulation另一個模擬擔任該角色，必須修改信任原則，如下列編輯範例所示：

```

{
  "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

建立專案時，您可能會收到下列錯誤：

```
An error occurred (InvalidBucketName) when calling the CreateBucket operation: The specified bucket is not valid.
```

您收到此錯誤是因為 SimSpace Weaver 傳遞至 Amazon Simple Storage Service (Amazon S3) 的名稱違反了儲存貯體命名規則 (如需詳細資訊，請參閱 Amazon 簡單儲存服務使用者指南中的儲存貯體命名規則)。

應用 SimSpace Weaver 程式 SDK 中的 `create-project` 指令碼會使用您提供給指令碼的專案名稱來建立值區名稱。值區名稱使用下列格式：

- 版本 1.13.x 或更新版本
 - `weaver-lowercase-project-name-account-number-region`
- 版本 1.12 倍
 - `weaver-lowercase-project-name-account-number-app-zips-region`
 - `weaver-lowercase-project-name-account-number-schemas-region`

例如，給定以下項目屬性：

- 項目名稱：MyProject
- AWS 帳戶編號：111122223333
- AWS 區域：us-west-2

該項目將具有以下儲存桶：

- 版本 1.13.x 或更新版本
 - `weaver-myproject-111122223333-us-west-2`
- 版本 1.12 倍
 - `weaver-myproject-111122223333-app-zips-us-west-2`
 - `weaver-myproject-111122223333-schemas-us-west-2`

您的專案名稱不得違反 Amazon S3 命名規則。您還必須使用足夠短的專案名稱，以便 `create-project` 指令碼建立的儲存貯體名稱不會超過 Amazon S3 儲存貯體的名稱長度限制。

ServiceQuotaExceededException

啟動模擬時，您可能會收到下列錯誤：

```
An error occurred (ServiceQuotaExceededException) when calling the StartSimulation operation: Failed to start simulation due to: simulation quota has already been reached.
```

如果您嘗試啟動新模擬，但您的帳戶目前擁有目標狀態為「已啟動」的最大模擬數量，則會收到此錯誤訊息。這包括執行模擬、失敗的模擬以及因為達到最長持續時間而停止的模擬。您可以刪除已停止或失敗的模擬，以允許您啟動新模擬。如果所有模擬都在執行中，您可以停止並刪除執行模擬。如果您尚未達到請求限制，也可以要求提高服務配額。如需詳細資訊，請參閱[SimSpace 韋弗端點和配額](#)。若要取得有關如何清理不必要模擬的資訊，請參閱快速入門自學課程[步驟 6：停止並清理您的模擬](#)中的〈〉。

TooManyBuckets

建立專案時，您可能會收到下列錯誤：

```
An error occurred (TooManyBuckets) when calling the CreateBucket operation: You have attempted to create more buckets than allowed.
```

Amazon Simple Storage Service (Amazon S3) 對 AWS 帳戶中可容納的儲存貯體數量有限制 (如需詳細資訊，請參閱 [Amazon 簡單儲存服務使用者指南中的儲存貯體限制和限制](#))。

您必須先執行下列其中一項作業，才能繼續：

- 刪除 2 個或更多您不需要的現有 Amazon S3 儲存貯體。
- 請求提高 Amazon S3 限制 (如需詳細資訊，請參閱 [Amazon 簡單儲存服務使用者指南中的儲存貯體限制和限制](#))。
- 使用不同的 AWS 帳戶。

Note

中的 DeleteSimulation API SimSpace Weaver 不會刪除與您的模擬相關聯的 Amazon S3 資源。建議您在不再需要模擬時移除與模擬相關聯的所有資源。若要取得有關如何清理模擬的指導，請參閱快速[步驟 6：停止並清理您的模擬](#)入門自學課程中的〈〉。

模擬啟動期間拒絕權限

啟動模擬時，您可能會收到錯誤訊息，指出權限遭拒或存取應用程式成品時發生錯誤。當您為 SimSpace Weaver 未為您建立的模擬指定 Amazon S3 儲存貯體時 (透過主控台或 SimSpace Weaver 應用程式 SDK 指令碼)，就會發生此問題。

以下情況是最可能的根本原因：

- 該服務沒有存取您在模擬架構中指定的一或多個 Amazon S3 儲存貯體的權限 — 請檢查您的應用程式角色許可政策、Amazon S3 儲存貯體政策和 Amazon S3 儲存貯體許可，以確保 `sinspaceweaver.amazonaws.com` 具有存取儲存貯體的正確許可。如需應用程式角色權限原則的詳細資訊，請參閱 [為您建 SimSpace Weaver 立的權限](#)。
- 您的 Amazon S3 儲存貯體可能與模擬不 AWS 區域同 — 模擬人工品的 Amazon S3 儲存貯體必須與模擬位於 AWS 區域相同。檢查您的 Amazon S3 主控台，瞭解 AWS 區域您的儲存貯體所在。如果您的 Amazon S3 儲存貯體位於不同的儲存貯體 AWS 區域，請選取與模擬相 AWS 區域同的儲存貯體。

與使用時間有關的問題 Docker

如果您正在使用 Docker 並且在從 SimSpace Weaver 應用程式 SDK 執行指令碼時收到與時間有關的錯誤，原因可能是您的 Docker 虛擬機器時鐘不正確。如果您的計算機正在運行 Docker，然後從睡眠或休眠狀態恢復，則可能會發生這種情況。

嘗試的解決方案

- 重新啟動 Docker。
- 在 Windows PowerShell 以下位置停用然後重新啟用時間同步：

```
Get-VMIntegrationService -VMName DockerDesktopVM -Name "Time Synchronization" |  
  Disable-VMIntegrationService  
Get-VMIntegrationService -VMName DockerDesktopVM -Name "Time Synchronization" |  
  Enable-VMIntegrationService
```

PathfindingSample 主控台用戶端無法連線

當您連線至 [快速入門教學課程](#) 和 [詳細教學課程](#) 中描述的 PathfindingSample 模擬時，您可能會從主控台用戶端收到下列錯誤。之所以發生這個錯誤，是因為用戶端無法 ViewApp 在您提供的合併 IP 位址和連接埠號碼上開啟網路連線。

```
Fatal error in function nng_dial. Error code: 268435577. Error message: no link
```

對於中的模擬 AWS 雲端

- 您的網路連線是否正常運作？確認您可以連線到應該運作的其他 IP 位址或網站。請確定您的網頁瀏覽器並未從快取載入網站。
- 您的模擬是否正在執行？您可以使用 ListSimulationsAPI 取得模擬的狀態。如需詳細資訊，請參閱[步驟 4：獲取您的 IP 地址和端口號](#)。您也可以使用[SimSpace Weaver控制台](#)檢查模擬的狀態。
- 您的應用程式在執行嗎？您可以使用 DescribeAppAPI 取得應用程式的狀態。如需詳細資訊，請參閱[步驟 4：獲取您的 IP 地址和端口號](#)。您也可以使用[SimSpace Weaver控制台](#)檢查模擬的狀態。
- 您的應用程式在執行嗎？您可以使用 DescribeAppAPI 取得應用程式的狀態。如需詳細資訊，請參閱[步驟 4：獲取您的 IP 地址和端口號](#)。您也可以使用[SimSpace Weaver控制台](#)檢查模擬的狀態。
- 您是否使用了正確的 IP 地址和端口號？當您透過網際網路連線時，您必須使用的 IP 位址和連接埠號碼。您可以在 DescribeAppAPI 輸出的 EndpointInfo 塊中找到 IP Address 和 Actual 端口號。您也可以使用[SimSpace Weaver主控台](#)在 MyViewDomain 詳細資料頁面中尋找您的 IP 位址 (URI) 和連接埠號碼 (輸入連接埠)。
- 您的網路連接是否通過防火牆？您的防火牆可能會封鎖您與 IP 位址或連接埠號碼 (或兩者) 的連線。請檢查您的防火牆設定或洽詢防火牆管理員。

對於本端模擬

- 您可以連接到您的環回地址 (127.0.0.1) 嗎？如果您在 Windows 中使用 ping 命令行工具，則可以打開命令提示符窗口並嘗試 ping 127.0.0.1。按 Ctrl-C 結束 ping。

```
ping 127.0.0.1
```

Example 平輸出

```
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:\>
```

如果 ping 顯示遺失封包，您可能有其他軟體 (例如本機防火牆、安全性設定或反惡意程式碼程式) 阻擋您的連線。

- 您的應用程式在執行嗎？您的本端模擬會為每個應用程式以獨立的視窗執行。確保空間應用程序的窗口ViewApp已打開。如需詳細資訊，請參閱[本機開發](#)。
- 您是否使用了正確的 IP 地址和端口號？連接至本端模擬tcp://127.0.0.1:7000時必須使用。如需詳細資訊，請參閱[本機開發](#)。
- 您是否有可能阻止連接的本地安全軟件？檢查您的安全設置，本地防火牆或反惡意軟件程序，以查看它們是否阻止了 TCP 端口127.0.0.1上的連接7000。

無AWS CLI法識別 **simspaceweaver**

如果AWS CLI給你錯誤，表明它不知道SimSpace Weaver，運行以下命令。

```
aws simspaceweaver help
```

如果您收到以下幾行開頭的錯誤並列出了所有可用選項，那麼您AWS CLI可能是舊版本。

```
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:
```

執行下列命令來檢查您的AWS CLI。

```
aws --version
```

如果版本號早於 2.9.19，則您必須更新。AWS CLI 請注意，目前的版本晚AWS CLI於 2.9.19。

若要更新您的AWS CLI，請參閱第 2 [版的AWS Command Line Interface使用者指南AWS CLI中的安裝或更新](#)最新版本的。

SimSpace Weaver 模擬綱要參考

SimSpace Weaver 使用 YAML 檔案來規劃模擬的性質。此檔案稱為模擬資料架構 (或只是架構)。SimSpace Weaver 應用程式 SDK 中包含的範例模擬包含一個結構描述，您可以針對自己的模擬進行複製和編輯。

主題

- [一個完整的模式的例子](#)
- [架構格式](#)

一個完整的模式的例子

以下範例展示了描述 SimSpace Weaver 模擬的 YAML 格式文字檔。這些範例包括屬性的虛擬值。檔案的格式會根據檔案中 `sdk_version` 指定的值而有所不同。如 [架構格式](#) 需性質及其有效值的完整描述，請參閱。

```
sdk_version: "1.16"
simulation_properties:
  log_destination_resource_name: "MySimulationLogs"
  log_destination_service: "logs"
  default_entity_index_key_type: "Vector3<f32>"
  default_image: "111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-
repository:latest"
workers:
  MyComputeWorkers:
    type: "sim.c5.24xlarge"
    desired: 3
clock:
  tick_rate: "30"
partitioning_strategies:
  MyGridPartitioning:
    topology: "Grid"
    aabb_bounds:
      x: [-1000, 1000]
      y: [-1000, 1000]
    grid_placement_groups:
      x: 3
      y: 3
domains:
```

```
MyCustomDomain:
  launch_apps_via_start_app_call: {}
  app_config:
    package: "s3://weaver-myproject-111122223333-us-west-2/MyViewApp.zip"
    launch_command: ["MyViewApp"]
    required_resource_units:
      compute: 1
    endpoint_config:
      ingress_ports: [9000, 9001]
MyServiceDomain:
  launch_apps_per_worker:
    count: 1
  app_config:
    package: "s3://weaver-myproject-111122223333-us-west-2/
MyConnectionServiceApp.zip"
    launch_command: ["MyConnectionServiceApp"]
    required_resource_units:
      compute: 1
    endpoint_config:
      ingress_ports:
        - 9000
        - 9001
MySpatialDomain:
  launch_apps_by_partitioning_strategy:
    partitioning_strategy: "MyGridPartitioning"
    grid_partition:
      x: 6
      y: 6
  app_config:
    package: "s3://weaver-myproject-111122223333-us-west-2/MySpatialApp.zip"
    launch_command: ["MySpatialApp"]
    required_resource_units:
      compute: 1
MySpatialDomainWithCustomContainer:
  launch_apps_by_partitioning_strategy:
    partitioning_strategy: "MyGridPartitioning"
    grid_partition:
      x: 6
      y: 6
  app_config:
    package: "s3://weaver-myproject-111122223333-us-west-2/MySpatialApp2.zip"
    launch_command: ["MySpatialApp2"]
    required_resource_units:
      compute: 1
```

```
image: "111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest"
placement_constraints:
  - placed_together: ["MySpatialDomain", "MySpatialDomainWithCustomContainer"]
    on_workers: ["MyComputeWorkers"]
```

架構格式

下列範例顯示結構描述的整體結構。只要父子關係相同，結構描述每個層級的屬性順序並不重要。順序對於數組中的元素很重要。

```
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
```

章節

- [SDK 版本](#)
- [模擬性質](#)
- [工作程序](#)
- [時鐘](#)
- [分割策略](#)
- [網域](#)
- [位置限制](#)

SDK 版本

sdk_version 區段 (必要) 會識別支援此結構描述的 SimSpace Weaver 應用程式 SDK 版本。有效值：1.16、1.15、1.14、1.13、1.12

⚠ Important

的值 `sdk_version` 僅包括主要版本號碼和第一個次要版本號碼。例如，值會 `1.12` 指定所有版本 `1.12.x`，例如 `1.12.0`、`1.12.1`、和 `1.12.2`。

```
sdk_version: "1.16"
```

模擬性質

該 `simulation_properties` 區段 (必填) 指定模擬的各種性質。您可以在此段落設定記錄日誌並指定預設容器影像。即使您未設定記錄或選擇指定預設容器映像檔，此區段也是必要的。

```
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"
```

屬性

`log_destination_resource_name`

指定 SimSpace Weaver 將寫入日誌的資源。

需要：否。如果未包含此屬性，SimSpace Weaver 將不會為模擬寫入記錄。

類型：字串

有效值：

- CloudWatch 記錄檔記錄群組的名稱 (例如，`MySimulationLogs`)
- CloudWatch 日誌日誌群組的 Amazon 資源名稱 (ARN) (例如，`arn:aws:logs:us-west-2:111122223333:log-group/MySimulationLogs`)

📘 Note

SimSpace Weaver 僅支援相同帳戶和 AWS 區域模擬中的記錄目標。

log_destination_service

指出當您指定不是 ARN 的記錄目標資源的類型。logging_destination_resource_name

必要：如果已指定且不log_destination_resource_name是 ARN，則必須指定此內容。如果未指定或為 ARN，則log_destination_resource_name無法指定此屬性。

類型：字串

有效值：

- logs：記錄目的地資源是記錄群組。

default_entity_index_key_type

指定模擬圖元索引鍵欄位的資料類型。

必要：是

類型：字串

有效值：Vector3<f32>

default_image

指定模擬的預設容器影像 (版本1.13和不支援1.12)。如果指定此屬性，未指定的網域會image使用default_image。

必要：否

類型：字串

有效值：

- Amazon Elastic Container Registry (Amazon ECR) 中存儲庫的 URI (例如，111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest)

工作程序

workers區段 (必要) 會指定背景工作群組 (Worker 群組) 的組態。SimSpace Weaverplacement_constraints將此資訊與配置模擬的基礎架構一起使用。目前僅支援 1 個工作者群組。

若要指定 Worker 群組的內容，請 *worker-group-name* 以您選擇的名稱取代。名稱長度必須為 3-64 個字元，且可包含字元 A-Z、a-z、0-9 和 `_` (連字號)。在名稱後指定 Worker 群組的內容。

```
workers:  
  worker-group-name:  
    type: "sim.c5.24xlarge"  
    desired: number-of-workers
```

屬性

type

指定工作者類型。

必要：是

類型：字串

有效值：sim.c5.24xlarge

desired

指定此工作者群組所需的 Worker 數目。

必要：是

類型：整數

有效值：1-3。您模擬的 Worker 數量的服務配額 (限制) 會決定此屬性的最大值。例如，如果您的服務配額是2此屬性的最大值2。您可以要求提高服務配額。如需詳細資訊，請參閱 [SimSpace 韋弗端點和配額](#)。

時鐘

clock區段 (必填) 指定模擬時鐘的性質。

```
clock:  
  tick_rate: tick-rate
```

屬性

tick_rate

指定時鐘發佈至應用程式的每秒刻度數。

必要：是

類型：

- 版本1.14和1.15：字符串
- 版本1.13和1.12：整數

有效值：

- 版本1.14與1.15下列功能："10"|"15"|"30"|"unlimited"
 - "unlimited"：一旦所有應用程序完成當前刻度的提交操作，時鐘就會發送下一個刻度。
- 版本1.13與1.12下列項目:10| 15 | 30

分割策略

區partitioning_strategies段 (必要) 指定空間網域的分割區組織。

Note

SimSpace Weaver僅支持 1 個分區策略。

若要指定分割策略的屬性，請`partitioning-strategy-name`以您選擇的名稱取代。名稱長度必須為 3-64 個字元，且可包含字元 A-Z、a-z、0-9 和 _ (連字號)。在名稱後指定分割策略的屬性。

```
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
```

屬性

topology

指定此分割策略的拓撲 (資料分割排列配置)。

必要：是

類型：字串

有效值："Grid"

aabb_bounds

為模擬指定主軸對齊邊界框(AABB)的邊界。您可以將邊界指定為 2 個元素排序的陣列，用來描述每個軸 (和) 的最小值x和最大值 (依該順序y)。

必要：有條件限制。如果拓撲設定為，則此性質為必要 (且只能指定) "Grid"。

類型：Float陣列 (針對每個軸)

有效值：-3.4028235e38-3.4028235e38

grid_placement_groups

指定網格拓撲中沿每個軸 (x 和 y) 的放置群組數目。置放群組是空間上相鄰的分割區集合 (位於相同網域中)。

必要：有條件限制。如果拓撲設定為，則此性質為必要 (且只能指定) "Grid"。如果您沒有指定放置群組設定，系統SimSpace Weaver會為您計算一個設定。任何使用磁碟分割策略而不使用置放群組組態的網域都必須指定 grid_partition (請參閱[空間域分區策略](#))。

類型：整數 (每個軸)

有效值：1-20。我們建議 $x * y$ 等於所需的工作人數。否則，SimSpace Weaver會嘗試在可用員工間平衡您的放置群組。

網域

domains區段 (必要) 會指定每個網域的屬性。對於空間領域，所有模擬都必須至少有一個區段。您可以為其他網域建立多個區段。每種類型的域都有自己的配置格式。

Important

版本1.13和1.12不支持多個空間域。

⚠ Important

SimSpace Weaver 每個模擬最多支援 5 個網域。這包括所有空間、自訂和服務網域。

```
domains:
  domain-name:
    domain-configuration
  domain-name:
    domain-configuration
  ...
```

網域組態

- [空間域配置](#)
- [自訂網域組態](#)
- [服務網域組態](#)

空間域配置

若要指定空間網域的屬性，請 *spatial-domain-name* 以您選擇的名稱取代。名稱長度必須為 3-64 個字元，且可包含字元 A-Z、a-z、0-9 和 _ (連字號)。在名稱後指定空間網域的屬性。

```
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"
```

空間域分區策略

區 `launch_apps_by_partitioning_strategy` 段 (必要) 指定分割策略和模擬空間的維度 (以分割區數目為單位)。

```
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
```

屬性

partitioning_strategy

指定此空間網域的分割策略。

必要：是

類型：字串

有效值：此特性的值必須與partitioning_strategies段落中定義之分割策略的名稱相符。如需詳細資訊，請參閱 [分割策略](#)。

grid_partition

指定網格拓樸中沿每個軸 (x 和 y) 的分割區數目。這些維度描述了此領域的總模擬空間。

必要：有條件限制。僅當拓樸設定為時，才能指定此性質"Grid"。此屬性取決於此網域的指定磁碟分割策略的grid_placement_groups屬性：

- 如果此網域的分割策略未指定grid_placement_groups組態，則需要此屬性。
- 如果有grid_placement_groups組態但您沒有指定grid_partition，則SimSpace Weaver會使用與grid_placment_groups組態相同的維度。
- 如果同時指定grid_placement_groups和grid_partition，則的維度grid_partition必須是維度的倍數 grid_placement_groups (例如，如果您的grid_placement_groups維度為 2x2，則某些有效維度為 grid_partition 2x2、4x4、6x6、8x8、10x10)。

類型：整數 (每個軸)

有效值：1-20

空間應用配置

app_config區段 (必要) 會指定此網域中應用程式的套件、啟動設定和資源需求。

```
app_config:
  package: "app-package-s3-uri"
  launch_command: ["app-launch-command", "parameter1", ...]
  required_resource_units:
    compute: app-resource-units
```

屬性

package

指定包含應用程式可執行文件/二進製文件的包 (zip 文件)。套件必須存放在 Amazon S3 儲存貯體中。僅支持 zip 文件格式。

必要：是

類型：字串

有效值：Amazon S3 儲存貯體中套件的 Amazon S3 URI。例如 s3://example-bucket/MySpatialApp.zip。

launch_command

指定可執行文件/二進製文件名和命令行參數以啟動應用程式。每個命令行字符串令牌都是數組中的元素。

必要：是

類型：字符串數組

required_resource_units

指定 SimSpace Weaver 應配置給此應用程式每個執行個體的資源單位數量。資源單位是 Worker 上固定數量的虛擬中央處理單元(vCPUs)和隨機存取記憶(RAM)體。如需有關資源單位的詳細資訊，請參閱[端點和服務配額](#)。此內 compute 容會指定 Worker compute 系列的資源單位配置，並且是目前唯一有效的配置類型。

必要：是

類型：整數

有效值：1-4

自定義容器映像

`image` 屬性 (選用) 會指定 SimSpace Weaver 用來執行此網域中應用程式之容器映像檔的位置 (版本 1.13 和不支援 1.12)。將 URI 提供給 Amazon Elastic Container Registry (Amazon ECR) 中包含映像的存儲庫。如果未指定此屬性，但 `default_image` 已在頂層 `simulation_properties` 區段中指定，則此網域中的應用程式會使用 `default_image`。如需詳細資訊，請參閱 [自訂容器](#)。

```
image: "ecr-repository-uri"
```

屬性

`image`

指定要在此網域中執行應用程式的容器映像檔位置。

必要：否

類型：字串

有效值：

- Amazon Elastic Container Registry (Amazon ECR) 中存儲庫的 URI (例如，`111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest`)

自訂網域組態

若要指定自訂網域的內容，請 `custom-domain-name` 以您選擇的名稱取代。名稱長度必須為 3-64 個字元，且可包含字元 A-Z、a-z、0-9 和 `_` (連字號)。在名稱後指定自訂網域的內容。針對每個自訂網域重複此程序。

```
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"
```

屬性

launch_apps_via_start_app_call

使用 StartApp API 啟動自訂應用程式需要此屬性。

必要：是

類型：

有效值：{}

自定義應用配置

`app_config` section(必要) 指定此自訂網域中應用程式的套件、啟動設定、資源需求和網路連接埠。

```
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, ...]
```

屬性

package

指定包含應用程式可執行文件/二進製文件的包 (zip 文件)。套件必須存放在 Amazon S3 儲存貯體中。僅支持 zip 文件格式。

必要：是

類型：字串

有效值：Amazon S3 儲存貯體中套件的 Amazon S3 URI。例如 `s3://example-bucket/MyCustomApp.zip`。

launch_command

指定可執行文件/二進製文件名和命令行參數以啟動應用程式。每個命令行字符串令牌都是數組中的元素。

必要：是

類型：字符串數組

required_resource_units

指定SimSpace Weaver應配置給此應用程式每個執行個體的資源單位數量。資源單位是 Worker 上固定數量的虛擬中央處理單元(vCPUs)和隨機存取記憶(RAM)體。如需有關資源單位的詳細資訊，請參閱[端點和服務配額](#)。此內compute容會指定 Worker compute 系列的資源單位配置，並且是目前唯一有效的配置類型。

必要：是

類型：整數

有效值：1-4

endpoint_config

指定此網域中應用程式的網路端點。的值會ingress_ports指定自訂應用程式針對內送用戶端連線繫結的連接埠。 SimSpace Weaver將動態配置的連接埠對應至指定的輸入連接埠。輸入連接埠都是 TCP 和 UDP。使用 DescribeApp API 尋找連接用戶端的實際連接埠號碼。

需要：否。如果您未指定端點設定，則此網域中的自訂應用程式將不會有網路端點。

類型：整數數組

有效值：1024-49152。值必須是唯一的。

自定義容器映像

image屬性 (選用) 會指定SimSpace Weaver用來執行此網域中應用程式之容器映像檔的位置 (版本1.13和不支援1.12)。將 URI 提供給 Amazon Elastic Container Registry (Amazon ECR) 中包含映像的存儲庫。如果未指定此屬性，但default_image已在頂層simulation_properties區段中指定，則此網域中的應用程式會使用default_image。如需詳細資訊，請參閱 [自訂容器](#)。

```
image: "ecr-repository-uri"
```

屬性

image

指定要在此網域中執行應用程式的容器映像檔位置。

必要：否

類型：字串

有效值：

- Amazon Elastic Container Registry (Amazon ECR) 中存儲庫的 URI (例如 , 111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest)

服務網域組態

若要指定服務網域的屬性，請 *service-domain-name* 以您選擇的名稱取代。名稱長度必須為 3-64 個字元，且可包含字元 A-Z、a-z、0-9 和 _ (連字號)。在名稱後指定服務網域的內容。針對每個服務網域重複此程序。

```
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"
```

每位工作者啟動應用

`launch_apps_per_worker` 區段 (必要) 表示這是服務網域設定，並指定每個 Worker 要啟動的服務應用程式數目。

```
launch_apps_per_worker:
  count: number-of-apps-to-launch
```

屬性

count

此內容會指定每個 Worker 要啟動的服務應用程式數目。

必要：是

類型：整數

有效值：{}| 1 | 2。的值{}指定的預設值1。

服務應用配置

`app_config` section(必要) 指定此服務網域中應用程式的套件、啟動設定、資源需求和網路連接埠。

```
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, ...]
```

屬性

package

指定包含應用程序可執行文件/二進製文件的包 (zip 文件)。套件必須存放在 Amazon S3 儲存貯體中。僅支持 zip 文件格式。

必要：是

類型：字串

有效值：Amazon S3 儲存貯體中套件的 Amazon S3 URI。例如 `s3://example-bucket/MyServiceApp.zip`。

launch_command

指定可執行文件/二進製文件名和命令行參數以啟動應用程序。每個命令行字符串令牌都是數組中的元素。

必要：是

類型：字符串數組

required_resource_units

指定SimSpace Weaver應配置給此應用程式每個執行個體的資源單位數量。資源單位是 Worker 上固定數量的虛擬中央處理單元(vCPUs)和隨機存取記憶(RAM)體。如需有關資源單位的詳細資訊，請參閱[端點和服務配額](#)。此內compute容會指定 Worker compute 系列的資源單位配置，並且是目前唯一有效的配置類型。

必要：是

類型：整數

有效值：1-4

endpoint_config

指定此網域中應用程式的網路端點。的值會ingress_ports指定您的服務應用程式針對內送用戶端連線繫結的連接埠。 SimSpace Weaver將動態配置的連接埠對應至指定的輸入連接埠。輸入連接埠都是 TCP 和 UDP。使用 DescribeApp API 尋找連接用戶端的實際連接埠號碼。

需要：否。 如果您未指定端點設定，則此網域中的服務應用程式將不會有網路端點。

類型：整數數組

有效值：1024-49152。值必須是唯一的。

自定義容器映像

image屬性 (選用) 會指定SimSpace Weaver用來執行此網域中應用程式之容器映像檔的位置 (版本1.13和不支援1.12)。將 URI 提供給 Amazon Elastic Container Registry (Amazon ECR) 中包含映像的存儲庫。如果未指定此屬性，但default_image已在頂層simulation_properties區段中指定，則此網域中的應用程式會使用default_image. 如需詳細資訊，請參閱 [自訂容器](#)。

```
image: "ecr-repository-uri"
```

屬性

image

指定要在此網域中執行應用程式的容器映像檔位置。

必要：否

類型：字串

有效值：

- Amazon Elastic Container Registry (Amazon ECR) 中存儲庫的 URI (例如, 111122223333.dkr.ecr.us-west-2.amazonaws.com/my-ecr-repository:latest)

位置限制

`placement_constraints` 區段 (選用) 指定 SimSpace Weaver 應將哪些空間網域放置在相同 Worker 上。如需詳細資訊，請參閱 [配置空間領域](#)。

Important

版本 1.13 和 1.12 不支持 `placement_constraints`。

```
placement_constraints:  
  - placed_together: ["spatial-domain-name", "spatial-domain-name", ...]  
    on_workers: ["worker-group-name"]
```

屬性

`placed_together`

指定 SimSpace Weaver 應放在一起的空間領域。

必要：是

類型：字符串數組

有效值：結構描述中指定的空間網域名稱

`on_workers`

指定 SimSpace Weaver 應放置網域的背景工作群組。

必要：是

類型：1 元素字符串數組

有效值：綱要中指定之背景工作群組的名稱

SimSpace WeaverAPI 參考資料

SimSpace Weaver有 2 組不同的應用程式編程接口 (API) :

- 服務 API — 這些 API 控制服務和服務資源，例如您的模擬、時鐘和應用程式。它們是主要AWS軟件開發工具包 (SDK) 的一部分，您可以使用AWS命令行界面 (CLI) 調用它們。您也可以使用專案和平台的工具資料夾中的便利指令碼 (例如，`#####\工具\視窗\weaver-####-cli.bat`) 來呼叫這些 API。如需有關伺服器 API 的詳細資訊，請參閱 [SimSpace WeaverAPI 參考資料](#)。
- 應用程式 SDK API — 這些 API 可控制您模擬中的資料。您可以在應用程式代碼中使用它們來執行諸如讀取和寫入實體字段數據，使用訂閱以及監視模擬中的事件等操作。有關更多信息，請參閱解壓縮的SimSpace Weaver應用程式 SDK 文件夾中的應用程式 SDK 文檔：`sdk-folder\SimSpaceWeaverAppSdk-1.16.0\documentation`

Note

`sdk ###`是您解壓縮包的文件夾。 `SimSpaceWeaverAppSdkDistributable` 如果`sdk-folder\SimSpaceWeaverAppSdk-1.16.0`不存在，請確保您擁有SimSpace Weaver應用程式 SDK 的 1.16.0 版本。 `SimSpaceWeaverAppSdkDistributable` 不包含整個SimSpace Weaver應用程式 SDK。當您第一次使用SimSpace Weaver應用程式 SDK 指令碼建立專案時，指令碼會在 SDK 資料夾內建立 `SimSpaceWeaverAppSdk -1.16.0` 資料夾，並將其餘的應用程式 `SDK` 下載到其中。 SimSpace Weaver

AWS SimSpace Weaver 版本

我們不斷改善AWS SimSpace Weaver。如果您想利用SimSpace Weaver用新功能和功能更新，則必須在我們發布新版本時下載最新的應用程式 SDK。若要使用較新版本執行既有模擬，您可能必須更新其資料架構和程式碼，然後啟動模擬的新例證。您不需要升級，而且可以繼續使用舊版執行現有模擬。您可以查看此頁面以查看版本之間的不同之處。目前支援所有版本。

Important

最新版本的[AWS SimSpace Weaver用戶指南](#)僅涵蓋最新版本的服務。您可以在[AWS SimSpace Weaver指南目錄](#)中找到舊版的文件，可從[主要文件登陸頁面](#)取得。

最新版本

最新的版本是:1.16.0

如何找到您目前的版本

如果您使用SimSpace Weaver用應用程式 SDK 建立模擬，則create-project指令碼會將 SDK 程式庫的版本下載到. *sdk-folder* 包含 SDK 程式庫的子目錄名稱包含 SDK 版本號碼:SimSpaceWeaverAppSdk-*sdk-version*。例如，版本 1.15.3 的程式庫位於中SimSpaceWeaverAppSdk-1.15.3。

您也可以在您*sdk-folder*的. SimSpace Weaver app_sdk_distributable_version.txt

下載最新版本

使用下列其中一個連結下載最新版本。

- [完整的應用程式 SDK 可分發包](#)
- [只有應用程式 SDK 程式庫](#)

您也可以從中的[SimSpace Weaver主控台](#)下載完整的SimSpace Weaver應用程式 SDK 可散發套件。AWS Management Console從導航窗格中選擇「下載應用程式 SDK」。

Warning

請勿使用下AWS CLI載任何似乎是應用SimSpace Weaver程式 SDK 可散發套件的内容。請僅使用此頁面上的下載連結或主控台中的下載連結。不支援任何其他下載方法或位置，且可能包含過時、不正確或惡意程式碼。

排解應用程式 SDK 下載

我們使用 Amazon CloudFront (CloudFront) 來分發應用程序 SDK .zip 文件。您可能會遇到下列某些情況。

- 下載的軟件包不是最新版本
 - 如果您下載的 .zip 檔案不包含最新版本，可能是您 CloudFront 邊緣位置的快取尚未更新。24 小時後再次下載。
- 您使用下載連結收到 HTTP 4xx 或 5xx 錯誤訊息
 - 請在 24 小時後再試一次。如果您看到相同的錯誤訊息，請使用[SimSpace Weaver主控台](#)底部的 [意見反應] 連結回報問題。選取 [回報問題] 做為 [意見反應類型]。
- 您的瀏覽器報告無法加載頁面
 - 您可能在區域網路或瀏覽器設定問題。確認您可以載入其他頁面。請清除瀏覽器快取，然後再試一次。請確定您沒有可能封鎖下載 URL 的防火牆規則。
- 嘗試保存文件時收到錯誤
 - 請檢查您的本機檔案系統權限，以確定您擁有儲存檔案的正確權限。
- 您的瀏覽器顯示 AccessDenied
 - 如果您在瀏覽器中手動輸入 URL，請檢查網址是否正確。如果您使用了下載連結，請確定瀏覽器中的 URL 沒有干擾；請再次使用該連結。

安裝最新版本

若要安裝最新版本

1. [下載最新版本](#)。
2. 將 SimSpaceWeaverAppSdkDistributable .zip 解壓縮到一個文件夾。
3. 從解壓縮的最新版本 SimSpace Weaver 應用程式 SDK 資料夾執行 `docker-create-image.bat` (或 `docker-create-image.sh` 針對 WSL)。

4. 使用解壓縮的最新版本SimSpace Weaver應用程式 SDK 文件夾，而不是以前的版本。

服務版本

版本	備註	版本日期	文件	應用程式 SDK 下載
1.16.0	<p>新功能：</p> <ul style="list-style-type: none"> 現在，您可以在應用SimSpace Weaver程序 SDK 中使用消息傳遞 API 在應用程序之間發送和接收消息。此功能適用於 C++，Python 和統一和虛幻引擎 5 集成。如需詳細資訊，請參閱 簡訊。 	2024年2月12日	本指南	<ul style="list-style-type: none"> 完整的包裝 僅圖書館 <p>請參閱疑難排解</p>
1.15.3	<p>SimSpace Weaver Local更新：</p> <ul style="list-style-type: none"> 我們SimSpace Weaver Local更改為更緊密地將其與AWS 雲端。這些變更會影響 C++、Python、統一和虛幻引擎的專案和工作流程SimSpace Weaver Local。如需詳細資訊，請參閱 1.15.3 版 	2023 年 12 月 4 日	請參閱 AWS SimSpace Weaver 指南目錄 。	不提供下載

版本	備註	版本日期	文件	應用程式 SDK 下載
	本中的本機開發差異 。			
1.15.2	<p>應用程式 SDK 可散發套件更新：</p> <ul style="list-style-type: none"> 我們更新了 Dockerfile 使用的特定所需版本cmake。如果沒有此變更， Docker 容器組建可能會失敗。 	2023 年 11 月 2 日	請參閱 AWS SimSpace Weaver 指南目錄 。	不提供下載
1.15.1	<p>功能更新：</p> <ul style="list-style-type: none"> Python SDK：此版本修正了造成以 Python 為基礎的模擬在 AWS 雲端 如需詳細資訊，請參閱版本說明。 	2023 年 9 月 22 日	請參閱 AWS SimSpace Weaver 指南目錄 。	不提供下載

版本	備註	版本日期	文件	應用程式 SDK 下載
1.15.0	<p>新功能：</p> <ul style="list-style-type: none">開發套件：您現在可以使用 Python 開發模擬。SimSpace Weaver 應用程式 SDK 可散發套件包含範例 Python 專案及其 Python 檢視用戶端的範本。如需詳細資訊，請參閱 使用 Python。	2023 年 8 月 31 日	請參閱 AWS SimSpace Weaver 指南目錄 。	不提供下載

版本	備註	版本日期	文件	應用程式 SDK 下載
1.14.0	<p>新功能：</p> <ul style="list-style-type: none">• 自訂容器：建立您自己的 Amazon Linux 2 (AL2) 型容器映像，將其存放在亞馬遜彈性容器登錄檔 (Amazon ECR) 中，然後使用它在中執行您的 SimSpace Weaver 應用程式。AWS 雲端如需詳細資訊，請參閱 自訂容器。• 多個空間網域：在模擬中建立 1 個以上的空間網域。單獨的模擬邏輯，而不是將它們全部合併到一個空間應用程式中。根據空間領域的需求將不同的資源分配給空間領域。如需詳細資訊，請參閱 配置空間領域。• 無限點擊率：讓您的模擬執行速度與程式碼執行的速度一樣快。	2023 年 7 月 26 日	請參閱 AWS SimSpace Weaver 指南目錄 。	不提供下載

版本	備註	版本日期	文件	應用程式 SDK 下載
	<p>設置模擬的時鐘，以便在所有應用程式完成當前刻度的提交操作後立即發送下一個刻度。如需詳細資訊，請參閱 時鐘。</p> <p>SimSpace Weaver 應用程式 SDK：</p> <ul style="list-style-type: none"> 的現值在 <code>tick_rate</code> 是字串。值必須包含雙引號 (")。早期版本的價格點率仍然是一個整數。如需詳細資訊，請參閱 時鐘。 			
1.13.1	<p>SimSpace Weaver 應用程式 SDK：</p> <ul style="list-style-type: none"> 功能更新：專案建立現在可以正確使用 <code>PathfindingSampleUnreal</code> 範本。 	2023 年 6 月 7 日	請參閱 AWS SimSpace Weaver 指南目錄 。	不提供下載

版本	備註	版本日期	文件	應用程式 SDK 下載
1.13.0	<p>SimSpace Weaver 服務 API :</p> <ul style="list-style-type: none"> • 新 CreateSnapshot 動作 • StartSimulation 動作的變更： <ul style="list-style-type: none"> • 已新增從快照開始的 SnapshotS3Location 參數。 • 該 SchemaS3Location 參數現在是可選的。 • 更改 DescribeSimulation 輸出： <ul style="list-style-type: none"> • SchemaError 已棄用。 • 添加了一個 StartError 字段。 • 添加了一個 SnapshotS3Location 字段。 • 新增模 SNAPSHOT_ 	2023 年 4 月 29 日	請參閱 AWS SimSpace Weaver 指南目錄 。	不提供下載

版本	備註	版本日期	文件	應用程式 SDK 下載
	<p>IN_PROGRESS 擬狀態。</p> <ul style="list-style-type: none"> • 新S3Destination 資料類型 <p>SimSpace Weaver 主控台：</p> <ul style="list-style-type: none"> • 建立快照的新功能。 • 從快照啟動模擬的新功能。 <p>SimSpace Weaver 應用程式 SDK：</p> <ul style="list-style-type: none"> • 支援快照的新指令碼 <ul style="list-style-type: none"> • create-snapshot- <i>project-name</i> .bat • start-from-snapshot- <i>project-name</i> .bat • quick-start-from-snapshot- <i>project-name</i> - cli.bat 			

版本	備註	版本日期	文件	應用程式 SDK 下載
	<ul style="list-style-type: none">• <code>list-snapshots- <i>project-name</i> .bat</code>• <code>##### # Amazon S3 ####weaver -#### lowercase -project-name -##</code> <p>如需快照的相關資訊，請參閱 快照。</p>			

版本	備註	版本日期	文件	應用程式 SDK 下載
1.12.3	<p>SimSpace Weaver 應用程式 SDK :</p> <ul style="list-style-type: none"> • 下列指令碼現在支援 <code>--maximum-duration</code> 參數 : • <code>quick-start-<i>project-name</i> - cli.bat</code> • <code>quick-start-<i>project-name</i> - cli.sh</code> • <code>start-simulation-<i>project-name</i> .bat</code> • <code>start-simulation-<i>project-name</i> .sh</code> • <code>run-<i>project-name</i> .bat</code> • <code>run-<i>project-name</i> .sh</code> <p>如需詳細資訊，請參閱 模擬持續時間上限。</p>	2023 年 3 月 27 日	請參閱 AWS SimSpace Weaver 指南目錄 。	不提供下載

版本	備註	版本日期	文件	應用程式 SDK 下載
1.12.2	<p>SimSpace Weaver 應用程式 SDK :</p> <ul style="list-style-type: none"> 錯誤修復 : docker-create-image.bat 現在正確運行。 	2023 年 3 月 1 日	請參閱 AWS SimSpace Weaver 指南目錄 。	不提供下載
1.12.1	<p>SimSpace Weaver 應用程式 SDK :</p> <ul style="list-style-type: none"> 指令碼現在接受用於AWS驗證的AWS CLI設定檔。 腳本現在支持AWS身AWS IAM Identity Center份驗證。 <p>SimSpace Weaver Local:</p> <ul style="list-style-type: none"> 錯誤修復 : Api::BeginUpdateWillBlock 現在 , true如果所有的空間應用程序都沒有加入模擬 , 現在會正確返回。 	2023 年 2 月 28 日	請參閱 AWS SimSpace Weaver 指南目錄 。	不提供下載

版本	備註	版本日期	文件	應用程式 SDK 下載
1.12.0	正式上市的發行版本 (GA)	2022 年 11 月 29 日	請參閱 AWS SimSpace Weaver 指南目錄 。	不提供下載

AWS SimSpace Weaver 1.15.1 版本

對於最初在 1.15.0 版中發行的 Python SDK，此發行 SimSpace Weaver 版本是必要的更新。它修復了一個版本不匹配問題，該問題導致基於 Python 的模擬在 AWS 雲端 請使用此版本，而不是 1.15.0。

將現有的 Python 項目更新為 1.15.1

如果您有使用版本 1.15.0 Python SDK 建立的現有 Python 專案，您必須執行下列步驟將其更新至 1.15.1，以便在 AWS 雲端

您也可以使用 1.15.1 Python SDK 建立新的 Python 專案，並將您的自訂程式碼移至新專案，而不必遵循這個程序。

若要將一個 1.15.0 Python 專案更新至 1.15.1

- 轉到您的 Python 項目的文件夾。
- 在 `src/PythonBubblesSample/bin/run-python` 更改以下行中：

```
export PYTHONPATH=$PYTHONPATH:/roapp/lib
```

變更為：

```
export PYTHONPATH=$PYTHONPATH:$LD_LIBRARY_PATH:/roapp/lib
```

- 在 `CMakeLists.txt` 刪除以下幾行：

- ```
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)
```

1.15.1 版的疑難排解

在更新 1.15.0 Python 模擬之後，它無法在 AWS 雲端

症狀：在您開始模擬後大約 5-10 分鐘後，模擬管理記錄會報告，`internal error`且模擬狀態為 `FAILED`。

如果來自 1.15.0 Python SDK 的程式庫檔案包含在應用程式壓縮檔案中，就可能會發生這種情況。確保您已完成更新項目的步驟，並確保該步驟 `libweaver_app_sdk_python_v1.so` 不在您的 zip 文件中或以任何方式引用。

關於版本 1.15.1 的常見問題

此版本是否會影響 Python SDK 以外的其他任何內容？

否。

我是否必須更新至版本 1.15.1？

如果您不打算將 Python 用於空間應用程式，則不必更新為 1.15.1。如果您更新為 1.15.0，則您的 Python 模擬將不會在 AWS 雲端。如果您使用 1.15.0，建議您更新為 1.15.1。

什麼是 `$LD_LIBRARY_PATH`？

它是 Python SDK 的位置，當您的模擬在 AWS 雲端。這是 1.15.1 版的新功能。我們進行了此更改是為了避免 future 發生 Python 版本問題。鏈接到該目錄在功能上與 `libweaver_app_sdk_python_v1.so` 在 1.15.0 中鏈接相同。

AWS SimSpace Weaver 的文件歷程記錄

下表說明 SimSpace Weaver 文件的重要變更。

日期	變更	說明文件更新	API 版本已更新
2024年2月 12日	已更新內容	更新了 1.16.0 版本的 AWS SimSpace Weaver 版本 章節。	N/A
2024年2月 12日	新內容	已新增 簡訊 區段，做為 1.16.0 版本發行版本的一部分。本節說明新增至 SimSpace Weaver 應用程式 SDK 的訊息傳遞 API。您可以使用這些 API 在應用程式之間傳送和接收訊息。	N/A
2024年2月 12日	已更新內容	更新了 1.16.0 版本的 SimSpace Weaver 模擬綱要參考 章節。	N/A
2024年2月 12日	已更新內容	將訊息傳送的服務配額新增至 SimSpace 韋弗端點和配額 本章。	N/A
2024年2月 12日	新指南	將 1.16.0 之前版本的內容分割為單獨的指南。新增指 AWS SimSpace Weaver 南目錄 (可從 主要文件登陸頁面 取得) 以存取舊版的指南。	N/A
2023 年 12 月 4 日	已更新內容	更新了 1.15.3 版本的 AWS SimSpace Weaver 版本 章節。	N/A

日期	變更	說明文件更新	API 版本已更新
2023 年 12 月 4 日	已更新內容	已更新本 AWS SimSpace Weaver 版本 章以包含最新版本的安裝指示。	N/A
2023 年 12 月 4 日	已更新內容	更新了 的服務配額 SimSpace Weaver Local 。	N/A
2023 年 12 月 4 日	新的與更新的內容	重新架構 本機開發 區段，並新增描述 1.15.3 版中 SimSpace Weaver Local 引入之差異的新頁面。	N/A
2023 年 11 月 7 日	已更新內容	更新了設置 Docker 和 WSL 以使用應用程式 SDK 的直接下載鏈接/URL 的說明。如需詳細資訊，請參閱 設定您的本機環境 SimSpace Weaver 。	N/A
2023 年 11 月 2 日	已更新內容	更新了 1.15.2 版本的服務版本頁面。如需詳細資訊，請參閱 服務版本 。	N/A
2023 年 10 月 23 日	已更新內容	更新了服務版本頁面，其中包含下載應用程式 SDK 可分發包的新說明。客戶現在應該只使用我們核准的其中一個直接下載連結，而不 AWS CLI 要使用下載應用程式 SDK 可散發套件。如需詳細資訊，請參閱 下載最新版本 。	N/A

日期	變更	說明文件更新	API 版本已更新
2023 年 9 月 22 日	新內容	增加了版本說明頁面，其中包含 1.15.1 版本的更新說明。如需詳細資訊，請參閱 AWS SimSpace Weaver 1.15.1 版本 。	N/A
2023 年 9 月 10 日	新內容	針對無法辨識的情況新增疑難排解區段 SimSpace Weaver。AWS CLI 如需詳細資訊，請參閱 無 AWS CLI 法識別 simspaceweaver 。	N/A
2023 年 9 月 10 日	已更新內容	已更新 WSL AWS CLI 中的安裝指示。如需詳細資訊，請參閱 在中設定 Amazon Linux 2 (AL2) Windows Subsystem for Linux (WSL) 。	N/A
2023 年 9 月 7 日	API 更新	BucketName 和現 ObjectKey 在需要 S3 位置 資料類型。BucketName 現在需要 S3 目標 資料類型。	AWS 軟體開發套件： 2023-09-07
2023 年 8 月 31 日	新內容	增加了 1.15.0 版本的新章節： 使用 Python	N/A

日期	變更	說明文件更新	API 版本已更新
2023 年 8 月 15 日	已更新內容	已更新中的下載說明， AWS SimSpace Weaver 版本 以僅列出官方的 SimSpace Weaver Amazon S3 儲存貯體。其他下載位置不受惡意程式碼控制，AWS 且可能包含惡意程式碼。	N/A
2023 年 7 月 26 日	已更新內容	已更新 時鐘 。	N/A
2023 年 7 月 26 日	已更新內容	已更新 配置空間領域 。	N/A
2023 年 7 月 26 日	新內容	新增章節：「 自訂容器 」。	N/A
2023 年 7 月 26 日	已更新內容	已 AWS SimSpace Weaver 版本 針對 1.14.0 版本進行了更新。	N/A
2023 年 7 月 6 日	新內容	新增章節：「 PathfindingSample 主控台用戶端無法連線 」。	N/A
2023 年 6 月 7 日	已更新內容	已 AWS SimSpace Weaver 版本 針對 1.13.1 版本進行了更新。	N/A
2023 年 5 月 15 日	新內容	新增章節：「 搭配使用快照 AWS CloudFormation 」。	N/A

日期	變更	說明文件更新	API 版本已更新
2023 年 4 月 29 日	新內容	已新增 1.13.0 版本的內容。如需詳細資訊，請參閱 AWS SimSpace Weaver 版本 。	AWS 軟體開發套件：
2023 年 3 月 27 日	新內容	新增描述模擬最長持續時間的區段。在 1.12.3 版的教學課程中新增附註，新增了對 SimSpace Weaver 應用程式 SDK 指令碼 -- maximum-duration 參數的支援。	N/A
2023 年 3 月 9 日	變更的內容	澄清我們只提供 Docker 上 Windows 和的指示 Windows Subsystem for Linux (WSL)，而 WSL (以及任何其他 Linux 環境) 不受支援。	N/A
2023 年 2 月 28 日	新內容	已新增說明 SimSpace Weaver 版本的章節。	N/A
2023 年 2 月 28 日	變更的內容	已變更有關驗證的內容，以包含 AWS Command Line Interface (AWS CLI) 的使用 AWS IAM Identity Center 和已命名設定檔。	N/A
2023 年 2 月 17 日	新內容	已新增有關使用管理資源的章節 AWS CloudFormation。	N/A
2023 年 1 月 23 日	新內容	已新增偵錯本機模擬的指示。	N/A

日期	變更	說明文件更新	API 版本已更新
2022 年 11 月 29 日	服務啟動	發布了. 的用戶指南和 API 參 SimSpace Weaver 考	AWS 軟體開發套件 :

詞彙表

此詞彙表定義的特定詞彙AWS SimSpace Weaver。

如需最新 AWS 術語的清單，請參閱 AWS 一般參考中的 [AWS 詞彙表](#)。

A

應用 您建立的可執程式碼 (也稱為二進位檔案)。術語應用程序可以引用該代碼的代碼或正在運行的實例。應用程式封裝模擬行為。應用程式會建立、刪除、讀取和更新[實體](#)。

應用程式 SDK 您用來整合應用程式的軟體開發套件 (SDK) SimSpace Weaver。SDK 提供用於讀取和寫入[實體](#)資料和追蹤模擬時間的 API。如需詳細資訊，請參閱[SimSpace Weaver應用程式 SDK](#)。

C

用戶端 存在於模擬之外的程序 (或其定義)，SimSpace Weaver並透過[自訂應用程式](#)或[服務應用程式](#)與模擬互動。您可以使用用戶端來檢視或變更模擬狀態。

時鐘 內部排程程序SimSpace Weaver的抽象。時鐘會將[刻度](#)發佈至[應用程式](#)，以維持時間同步。每個模擬都有自己的時鐘。

時脈速率 [時鐘](#)發佈至[應用程式](#)的每秒[刻度](#)數。如需支援時脈速率的詳細資訊，請參閱[SimSpace 韋弗端點和配額](#)。

時鐘刻度速率 請參閱[時脈速率](#)。

計算資源單位 [Worker](#) 上的運算資源單位 (處理器和記憶體)。[應用程式](#)的單一執行個體通常會配置 1 個運算資源單位。您可以為每個應用程式配置一個以上的運算資源單位。

自定義應用 一種可用來讀取模擬狀態並與其互動的[應用程式](#)類型。自訂應用程式可以在模擬中建立實體，但不擁有它們。自訂應用程式建立實體時，必須將實體轉移至[空間網域](#)。您可以使用應用程式 API 控制自訂應用程式的生命週期。如需 SimSpace Weaver API 的詳細資訊，請參閱[SimSpace WeaverAPI 參考資料](#)。

自訂網域 包含 [自訂應用程式的網域](#)。

自訂分割區 [自訂應用程式的分割區](#)。

D

截止 作業 (例如處理 [刻度](#)) 應完成的 [實際時間](#)。

網域 一組運行相同的可執行代碼 ([應用程序二進制](#)) 並具有相同的啟動選項的應用程序實例。

E

端點 (服務) 程式 (例如) 用來連線至服務的完整網域名稱 (FQDN AWS Command Line Interface)。SimSpace Weaver

端點 (模擬) 用戶端用來連線至模擬的 IP 位址和連接埠號碼。您可以在 [自訂應用程式和服務應用程式](#) 上設定端點。

實體 客戶數據對象 (或其定義) 。圖元可以是靜態的 (保留在一個位置) 或動態 (在模擬空間中移動)。例如，模擬中的人和建築物。

I

指數 (模擬) 模擬的空間性質的描述，包括其空間邊界和座標系統。

L

生命週期 (應用程式) [應用程式](#) 在模擬期間所經歷的預期邏輯步驟的說明。生命週期可能是受管理 (SimSpace Weaver 啟動和停止應用程式) 或未受管理 (您啟動和停止應用程式)。

載入 (實體欄位資料) 從讀取 [實體欄位資料 State Fabric](#)。

P

分割區 [Worker](#) 上的共用記憶體區段。每個分區都包含一個 [域](#) 中的 [實體](#) 的離散子集。每個 [應用程式](#) 都有分配的分區。一個應用程式擁有其分區中的所有實體。當應用程式創建一個實體時，它會在其分區中創建它。當實體從一個分

區移動到另一個分區時，所有權將從源分區的應用程序轉移到目標分區的應用程序。

R

資源單位 [請參閱 ???](#)。

S

架構 描述模擬設定的 YAML 或 JSON 文件。SimSpace Weaver使用資料架構來建立[模擬資源](#)。

服務應用 一種可用來讀取模擬狀態並與其互動的[應用程式](#)類型。服務應用程式可以在模擬中建立實體，但必須將其傳輸到空間[網域](#)。SimSpace Weaver管理服務應用程式的[生命週期](#)，並在模擬中的每個[Worker](#)上啟動 1 個 (或更多，如[模擬結構描述](#)中所指定)。

服務網域 包含[服務應用程式](#)的[網域](#)。

服務分區 [服務應用程式](#)的[分割區](#)。

模擬 (資源) 執行模擬虛擬空間之運算叢集的抽象化。您可以有多個模擬。您可以使用[資料架構](#)規劃模擬。

空間應用 封裝核心模擬邏輯的一種[應用程式](#)。每個空間應用程序擁有 1 個 (並且只有 1 個) [分區](#)。

空間域 包含[空間應用程式](#)的[網域](#)。

空間分區 [空間應用程式](#)的[分割區](#)。

State Fabric SimSpace Weaver的記憶體內資料庫。State Fabric儲存模擬的狀態，包括實體和內部SimSpace Weaver資料。

存儲 (實體字段數據) 將實體欄位資料寫入 [State Fabric](#)。

訂閱 對特定[應用程式](#)實例的長時間運行請求，以從[訂閱區域](#)接收數據。訂閱應用程式使用訂閱來探索訂閱區域內[實體](#)的變更。

訂閱區 模擬空間的二維區域。[訂閱](#)是指訂閱區域。一個訂閱區域可以跨越一個以上的[分區](#)，並且還包括分區的一部分。訂閱區域在其定義的範圍內是連續的。

T

蜚	時間的離散值 (壁鐘時間或模擬時間)。 應用程式 的迭代速度可以比刻度持續時間更快，但預計會在特定的截止日期內編寫指定的刻度 給定刻度的所有應用程式的所有操作必須完成，然後下一個刻度才能開始。
刻度	請參閱時脈速率。
時間 (實際)	從現實的角度看當前的時間。 SimSpace Weaver 使用 64 位 POSIX 時間戳，該時間戳是自紀元以來的納秒數 Unix (January 1, 1970, 00:00:00 UTC)。
時間 (模擬)	從模擬的角度來看的目前時間。 SimSpace Weaver 使用 64 位整數邏輯刻度計數器，該計數器可能與實際時間不直接對應。

W

工人	可執行模擬程式碼的亞馬遜彈性運算雲端 (Amazon EC2) 執行個體。
----	---------------------------------------

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。