開發人員指南

AWSMobile SDK for Unity

AWSMobile SDK for Unity: 開發人員指南

Table of Contents

所以此AWS適用於 Unity 的 Mobile SDK 現已包含在AWS SDK for .NET。本指南引用適用於 Unity 的 Mobile SDK 的存檔版本。如需詳細資訊，請參閱「」什麼是AWS適用於 Unity 的 Mobile SDK？

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。

什麼是AWS適用於 Unity 的 Mobile SDK？

所以此AWS適用於 Unity 的 Mobile SDK 現已包含在AWS SDK for .NET。如需詳細資訊，請參閱《AWS SDK for .NET 開發人員指南》。

本指南不再更新，它引用了適用於 Unity 的移動軟件開發工具包的存檔版本。

相關指南和主題

- 對於前端和移動應用程序開發，我們建議使用AWS Amplify。
- 有關使用AWS SDK for .NET的 Unity 應用程序，請參閱Unity 支持的特殊考量中的AWS SDK for .NET開發人員指南。
- 作為參考目的，您可以找到AWSMobile SDK for Unity上GitHub。

歸檔的參考資料內容

適用於 Unity 的歸檔移動軟件開發工具包包含一組 .NET 類，使用 Unity 編寫的遊戲能夠利用AWS服務。使用適用於 Unity 的移動 SDK 編寫的應用程序可以在 iOS 或安卓設備上運行。

支援AWS服務包括：

- Amazon Cognito
- Amazon DynamoDB
- AWS Identity and Access Management (IAM)
- Amazon Kinesis Data Streams
- AWS Lambda
- Amazon Mobile Analytics
- Amazon Simple Email Service (Amazon SES)
- Amazon Simple Notification Service (Amazon SNS)
- Amazon Simple Queue Service (Amazon SQS)
- Amazon Simple Storage Service (Amazon S3)

這些服務使您能夠對用户進行身份驗證、保存玩家和遊戲數據、在雲中保存對象、發送推送通知以及收集和分析使用數據。

相容性

適用於 Unity v3 的移動軟件開發工具包與 Unity 4.6 及更高版本兼容。

適用於 Unity 的最新版本的移動 SDK 引入了一些改進，這些改進可能要求您在合併到項目中時更改代碼。如需這些變更的詳細資訊，請參改進項目AWSMobile SDK for Unity在前端網絡和移動博客上。

下載適用於 Unity 的 Mobile SDK

您還可以將適用於 Unity 的移動軟件開發工具包下載為 .zip 文件這裡。

適用於 Unity 的 Mobile SDK 包含哪些項目？

如需完整清單NuGet軟件包、示例和其他文件，請參閱AWS SDK for .NET上GitHub。

設定適用於 Unity 的 AWS Mobile SDK

要開始使用適用於 Unity 的 AWS 移動開發工具包，您可以設置軟件開發工具包並開始構建新項目，也可以將開發工具包與現有項目集成。您也可以複製並執行樣本來瞭解 SDK 的工作原理。

先決條件

您需要有下列資訊，才能使用適用於 Unity 的 AWS Mobile SDK：

- 一個 AWS 帳户
- 統一版本 4.x 或 5.x（如果要編寫在 iOS 64 位上運行的應用程序，則需要統一 4.6.4p4 或統一 5.0.1p3）

完成先決條件之後，您需要執行以下動作：

1. 下載適用於 Unity 的 AWS Mobile SDK。
2. 配定適用於 Unity 的 AWS Mobile SDK。
3. 使用 Amazon Cognito 託獲取 AWS 證書。

步驟 1：下載適用於 Unity 的 AWS Mobile SDK

首先，下載適用於 Unity 的 AWS Mobile SDK。軟件開發工具包中的每個軟件包都需要根據軟件包的名稱使用相應的 AWS 服務。例如，aws-unity-sdk-DynamoDB 庫 -2.1.0.0.0.0 單元軟件包用於調用 AWS 動態數據庫服務。您可以導入所有軟件包或只導入您想要使用的軟件包。

1. 打開 Unity 編輯器並創建一個新的空項目，使用默認設置。
2. 選擇資產 >導入套件 >自訂套件。
3. 在 中匯入Package (套件) 對話框中，導覽到並選擇您想要使用的 .unityPackage (.unityPackage)。
4. 在 中匯入包對話框中，確保選中所有項目，然後單擊匯入。

步驟 2：配定適用於 Unity 的 AWS Mobile SDK

建立場景

使用適用於 Unity 的 AWS Mobile SDK 時，您可以在`Start`或者`Awake`單聲道行為類的方法：

```
UnityInitializer.AttachToGameObject(this.gameObject);
```

通過選擇新建場景來自File (檔案)選單。

適用於 Unity 的 AWS 開發工具包包含其支持的每項 AWS 服務的客户端類。這些客户端使用名為awsconfig.xml。下列部分將介紹awsconfig.xmlfile. 如需這些設定的詳細資訊，請參Unity SDK API 參考。

設置默認 AWS 服務區域

若要為所有服務用户端配定預設區域：

```
<aws region="us-west-2" />
```

這將為 Unity SDK 中的所有服務客户端設置默認區域。可通過在創建服務客户端實例時顯式指定區域來覆蓋此設置，如下所示：

```
IAmazonS3 s3Client = new AmazonS3Client(<credentials>,RegionEndpoint.USEast1);
```

設定記錄信息

日誌記錄設置指定如下：

```
<logging logTo="UnityLogger"
         logResponses="Always"
         logMetrics="true"
         logMetricsFormat="JSON" />
```

此設置用於在 Unity 中配置日誌記錄。當您登錄到UnityLogger，框架內部將輸出打印到調試日誌。如果要記錄 HTTP 響應，請設置 logResponses Resples 標誌-值可以是「始終」、「從不」或 OnError。您還可以使用 LogMetrics 屬性記錄 HTTP 請求的性能度量，可以使用LogMetrics格式屬性，有效值為 JSON 或標準值。

下列範例顯示 awsconfig.xml 文件中最常用的設定。有關特定服務設置的詳細信息，請參閱下面的服務部分：

```
<?xml version="1.0" encoding="utf-8"?>
<aws region="us-west-2"
    <logging logTo="UnityLogger"
```

```
            logResponses="Always"
            logMetrics="true"
            logMetricsFormat="JSON" />
/>
```

使用 link.xml 檔案

SDK 針對特定於平台的組件使用反射。如果您使用的是 IL2CPP 腳本後端，請`strip bytecode`始終在 iOS 上啟用，所以您需要有`link.xml`文件，其中包含以下條目：

```
<linker>
<!-- if you are using AWSConfigs.HttpClient.UnityWebRequest option-->
<assembly fullname="UnityEngine">
    <type fullname="UnityEngine.Networking.UnityWebRequest" preserve="all" />
    <type fullname="UnityEngine.Networking.UploadHandlerRaw" preserve="all" />
    <type fullname="UnityEngine.Networking.UploadHandler" preserve="all" />
    <type fullname="UnityEngine.Networking.DownloadHandler" preserve="all" />
    <type fullname="UnityEngine.Networking.DownloadHandlerBuffer" preserve="all" />
</assembly>
<assembly fullname="mscorlib">
    <namespace fullname="System.Security.Cryptography" preserve="all"/>
</assembly>
<assembly fullname="System">
    <namespace fullname="System.Security.Cryptography" preserve="all"/>
</assembly>
<assembly fullname="AWSSDK.Core" preserve="all"/>
<assembly fullname="AWSSDK.CognitoIdentity" preserve="all"/>
<assembly fullname="AWSSDK.SecurityToken" preserve="all"/>
add more services that you need here...
</linker>
```

步驟 3：使用 Amazon Cognito 獲取身份池 ID

要在移動應用程序中使用 AWS 服務，您必須使用 Amazon Cognito 身份獲取身份池 ID。使用 Amazon Cognito 獲取身份池 ID 允許您的應用程序訪問 AWS 服務，而無需在應用程序中嵌入您的私有證書。這還允許您設置權限，以控制用户有權訪問哪些 AWS 服務。

要開始使用 Amazon Cognito，您必須創建一個身份池。身分集區是您的帳戶專屬的使用者身分資料存放區。每個身份池都有可配置的 IAM 角色，允許您指定應用程序用户可以訪問哪些 AWS 服務。通常，開發人員將使用每個應用程序一個身份池。如需身分集區的詳細資訊，請參Amazon Cognito 開發人員指南。

若要為應用程式建立身分集區：

1. 登錄到Amazon Cognito 主控台，然後單擊建立新的身分集區。
2. 輸入身分集區的名稱，並選中複選框，以允許對未驗證身分的存取。按一下建立集區以建立身分集區。
3. 按一下Allow，以建立與身分集區相關聯的兩個預設角色，一個用於未驗證的使用者。這兩個預設角色可讓您的身分集區存取 Cognito Sync 和 Mobile Analytics。

下一頁顯示創建憑據提供程序的代碼，以便您可以輕鬆地將 Cognito 身份與 Unity 應用程序集成。您可以將登入資料供應商傳遞至所使用的 AWS 用户端建構函式。程式碼看起來像這樣：

```
CognitoAWSCredentials credentials = new CognitoAWSCredentials (
    "IDENTITY_POOL_ID", // Identity Pool ID
    RegionEndpoint.USEast1 // Region
);
```

後續步驟

- 入門：閱讀適用於 Unity 的 AWS Mobile SDK 入門以獲取 SDK 中包含的服務的更詳細概述。
- 執行示範：查看我們的範例統一應用程式，展示常見的使用案例。要運行示例應用程序，請按照上述所述設置 SDK for Unity，然後按照單個示例的自述文件中包含的說明進行操作。
- 閱讀 API 參考：檢視API 參考，瞭解適用於 Unity 的 AWS Mobile SDK。
- 詢問問題：將問題發佈到AWS Mobile SDK 論壇或者在 Github 上打開一個問題。

適用於 Unity 的 AWS Mobile 開發套件入門

本頁面為您提供適用於 Unity 的 AWS 移動開發工具包中的每個 AWS 服務的概述，以及有關如何設置 Unity 示例的說明。您必須完成設定適用於 Unity 的 AWS Mobile SDK頁面，然後再開始使用以下服務。

Amazon Cognito 身分

對 AWS 進行的所有調用都需要 AWS 證書。我們建議您使用Amazon Cognito Identity為您的應用程式提供 AWS 憑證。請依照設定適用於 Unity 的 AWS Mobile SDK通過 Amazon Cognito 託獲取 AWS 證書。

Cognito 還允許您使用諸如亞馬遜、臉書、Twitter 和谷歌等公共登錄提供商對用户進行身份驗證，以及支持OpenID Connect。Cognito 也適用於未經身份驗證的用户。Cognito 提供了具有有限訪問權限的臨時證書，您可以使用Identity and Access Management(IAM) 角色。Cognito 通過創建與 IAM 角色相關聯的新身分池進行配置。IAM 角色指定您的應用程序可以訪問的資源/服務。

若要開始使用 Cognito 身分，請參Amazon Cognito 開發人員指南。

Amazon Cognito Sync

Cognito Sync使您可以輕鬆地將最終用户數據（如用户首選項或遊戲狀態）保存到 AWS 雲，以便用户無論用户使用何種設備，都可以使用這些數據。Cognito 還可以在本地保存這些數據，即使互聯網連接不可用，您的應用也能正常工作。當互聯網連接可用時，您的應用可以將其本地數據同步到雲。

若要開始使用 Cognito Sync，請參Amazon Cognito 開發人員指南。

使用CognitoSync管理器範例

在 中專案窗格中，前往資產/AWSSDK/例子/CognitoSync，然後在窗格右下方選取CognitoSync場景以打開場景。

若要運行範例，請按一下編輯器屏幕頂端的播放按鈕。當應用程序運行時，它會顯示幾個文本框和按鈕，允許您輸入一些玩家信息。下面有一系列按鈕，可以在本地保存玩家信息，將本地玩家信息與 Cognito Cloud 同步，從 Cognito Cloud 刷新玩家信息，以及刪除本地玩家信息。按下每個按鈕以執行操作。該示例在遊戲屏幕頂部顯示反饋。

若要設定CognitoSync管理器示例，您必須指定一個 Cognito 份池 ID。要指定此值，請在 Unity 編輯器中選擇SyncManager中的世襲主義窗格中，然後將其輸入標識池 ID文字框中Inspector 窗格。

> **Note**
>
> 所以此CognitoSync管理器示例包含的代碼說明如何使用臉書身份提供程序，搜索 "USE_FACEBOOK 登錄" 宏。這需要使用適用於統一的臉書軟件開發工具包。如需詳細資訊，請參閱「」適用於 Unity 的 Facebook SDK。

Dynamo DB

Amazon DynamoDB 是一種快速、可輕鬆擴展、高度可用、經濟實惠、非關聯式資料庫服務。DynamoDB 移除資料儲存體的傳統擴展性限制，而仍維持低延遲及可預期的效能。

適用於 Unity 的 AWS 開發套件提供可用於 DynamoDB 的低階和高階程式庫。高級別庫包括 DynamoDB 對象映射程式，可讓您將客户端類別映射到 DynamoDB 表、執行各種建立、讀取、更新和刪除 (CRUD) 操作，以及執行查詢。使用 DynamoDB 物件映射器，您可以撰寫簡單、易讀的程式碼，以便在雲端中存放物件。

如需 DynamoDB 的詳細資訊，請參DynamoDB 開發人員指南。

如需使用 Unity Dynamo DB 的詳細資訊，請參Amazon DynamoDB。

使用 DynamoDB 範例

在 中專案窗格中，前往資產/AWSSDK/ 例子/DynamoDB。此範例由下列場景組成：

- DynamoDBExample 應用程序的初始場景
- LowLevelDynamoDb示例-使用低級別動態應用程序 API 的示例
- TableQueryAndScan示例-顯示如何執行查詢的示例
- HighLevel示例-使用高級 DynamoDB API 的示例

通過使用「生成設置」對話框（通過選擇「文件 .Buile.Build 設置」打開）將這些場景添加到構建中（按照它們在上面顯示的順序）中。此示例創建四個表：ProductCatalog, 論壇, 主題, 回覆.

若要運行範例，請按一下編輯器屏幕頂端的播放按鈕。當應用程序運行時，它會顯示一些按鈕：

- 低級別表操作-說明如何創建、列出、更新、描述和刪除表。
- 中級查詢和掃描操作-說明如何執行查詢。
- 高階物件映射器-說明如何創建、更新和刪除物件。

Mobile Analytics

使用Amazon Mobile Analytics，您可以跟蹤客户行為、聚合指標、生成數據可視化以及識別有意義的模式。適用於 Unity 的 AWS 開發工具包提供與 Amazon Mobile Analytics 服務的集成。有關 Mobile Analytics 的信息，請參閱Mobile Analytics 用户指南。如需使用 Unity Mobile Analytics 的詳細資訊，請參Amazon Mobile Analytics。

配置 Mobile Analytics

Mobile Analytics 定義了一些可以在 awsconfig.xml 文件中配置的設置：

```
<mobileAnalytics sessionTimeout = "5"
                 maxDBSize = "5242880"
                 dbWarningThreshold = "0.9"
                 maxRequestSize = "102400"
                 allowUseDataNetwork = "false"/>
```

- sessionTimeout-這是應用程序進入後台之後的時間間隔和可以終止會話的時間間隔。
- maxDBSize-這是 SQLite 數據庫的大小。當數據庫達到最大大小時，將刪除任何其他事件。
- dbWarningThreshold-這是對數據庫大小的限制，一旦達到，將生成警告日誌。
- maxRequestSize-這是應在 HTTP 請求中傳輸到移動分析服務的請求的最大大小（以字節為單位）。
- allowUseData網絡-一個布爾值，它指定是否在數據網絡上發送會話事件。

使用 Mobile Analytics 範例

在 中專案窗格中，前往資產/AWSSDK/ 例子/Mobile Analytics，然後在窗格右下方選取Amazon Mobile Analytics 範例場景以打開場景。要使用示例，您需要使用Amazon Mobile Analytics 主控台。如需使用 Mobile Analytics 控台的詳細資訊，請參Amazon Mobile Analytics 使用者指南。

在運行之前，請依照下列步驟來配置範例：

1. 選取AmazonMobileAnalyticsSample遊戲物件。
2. 指定您的應用程序 ID（在Amazon Mobile Analytics 主控台) 在「應用程序 ID」字段中。
3. 指定您的 Cognito 份池 ID（使用Amazon Cognito 主控台）在「Cognito 份池 ID」字段中。
4. 確保您的經過身份驗證和未經身份驗證的角色具有訪問 Mobile Analytics 服務的權限。如需將策略應用於 IAM 角色的詳細資訊，請參管理角色。

運行示例應用程序時，請注意事件可能不會立即傳輸到後端服務。後台線程將在本地緩衝事件，並以定期間隔（默認值為 60 秒）將事件批量發送到 Amazon Mobile Analytics 後端，以確保您的遊戲性能不會受到負面影響。由於 Amazon Mobile Analytics 對您的數據執行了複雜的處理，所以提交的事件和相應報告可能在初始提交後 60 分鐘內可能無法在 AWS 控制台中看到。

有關 Amazon Mobile Analytics 提供的報告的更多信息，請參閱報告和移動指標。

Simple Storage Service (Amazon S3)

Amazon Simple Storage Service (Amazon S3) 為開發人員和 IT 團隊提供安全、耐用、高擴展性的資料元儲存。在 Unity 中，您可以使用 S3 存儲、列出和檢索遊戲使用的圖像、視頻、音樂和其他數據。

如需 S3 的詳細資訊，請參Amazon S3和S3 入門。

如需使用 Unity 應用程式中的 S3 的詳細資訊，請參Amazon Simple Storage Service (S3)。

配置 S3 默認簽名

默認 S3 簽名配置如下：

```
<s3 useSignatureVersion4="true" />
```

這用於指定是否應將簽名版本 4 用於 S3 請求。

使用 S3 示例

在 中專案窗格中，前往資產/AWSSDK/例子/S3，然後在窗格右下方選取S3 示例場景以打開場景。該示例說明如何列出存儲桶、列出存儲桶中的對象、將對象發佈到存儲桶以及從存儲桶下載數據元。在運行之前，請依照下列步驟來配置範例：

1. 選取S3遊戲對象Aynamo窗格中。
2. 在 中Inspector窗格輸入S3BucketName和SampleFile名稱。S3BucketName是範例使用的儲存貯體名稱，S3SampleFile名稱是示例將上傳到指定 S3 存儲桶的文件的名稱。
3. 確保您的經過身份驗證的角色和未經身份驗證的角色具有訪問賬戶中的 S3 存儲桶的權限。如需將策略應用於 IAM 角色的詳細資訊，請參管理角色。

若要運行範例，請按一下編輯器屏幕頂端的播放按鈕。當應用程序運行時，它會顯示一些按鈕：

- 獲取物件-獲取您 AWS 賬戶中所有儲存貯體中所有物件的列表。

- 獲取儲存貯體-獲取您 AWS 賬户中所有儲存貯體的列表。
- 發佈物件-上傳物件至指定的 S3 儲存貯體。
- 刪除對象-刪除指定 S3 存儲桶中的所有對象。

該示例在遊戲屏幕頂部顯示反饋。

Amazon Simple Notification Service

Amazon Simple Simple Storage Service 是一種快速、靈活、完全託管的推送通知服務，讓您發送單個郵件或向大量收件人發送信息。Amazon Simple Simple Storage Service 可讓您輕鬆、經濟高效地向移動設備用户、電子郵件收件人發送推送通知，甚至向其他分佈式服務發送消息。若要開始使用 Amazon Simple Simple Service，請參Amazon Simple Notification Service。

AWS Lambda

AWS Lambda 是一種運算服務，可執行程式碼來回應請求或事件並自動為您管理運算資源，讓您輕鬆建構可快速回應新資訊的應用程式。AWS Lambda 函數可直接從移動、IoT 和 Web 應用程式進行調用，並同步回應，讓您輕鬆建立可擴展、安全且可用性高的後端，無需預配或管理基礎設施。如需詳細資訊，請參閱「」AWS Lambda。

Amazon Cognito 身分

什麼是 Amazon Cognito Identity？

使用 Amazon Cognito 身分，您可以為使用者建立唯一的身分並驗證它們以安全存取您的 AWS 資源（例如 Amazon S3 或 Amazon DynamoDB）。Amazon Cognito Identity 支援公有身分供應商 (Amazon、Facebook、Twitt/ 數字)、或任何 OpenID 連接兼容的供應商, 以及未驗證的身分。Cognito 也支援開發人員驗證的身分，可讓您透過自己的後端身分驗證程序來註冊及驗證使用者，同時仍使用Amazon Cognito Sync來同步用户數據並訪問 AWS 資源。

如需 Cognito 身分的詳細資訊，請參Amazon Cognito 開發人員指南。

如需 Cognito 驗證區域可用性的相關資訊，請參Amazon Cognito Identity 區域可用性。

使用公共提供程序對用户進行身份驗證

有關使用亞馬遜、Facebook、Twitt/ 數字或谷歌等公共身份提供商對用户進行身份驗證的信息，請參閱外部供應商在 Amazon Cognito 開發人員指南。

使用開發人員驗證的身分

有關開發人員經過身份驗證的身份的信息，請參閱開發人員驗證的身份在 Amazon Cognito 開發人員指南。

Amazon Cognito Sync

Cognito Sync 是 AWS 服務和用户端程式庫，可讓您跨裝置同步應用程式相關的使用者資料。您可以使用 Cognito 同步 API 跨設備同步用户數據。要在應用程序中使用 Cognito Sync，您必須在項目中包含適用於 Unity 的 AWS 移動開發工具包。

如需如何將 Amazon Cogito Sync 集成到應用程式中的指示，請參Amazon Cognito Sync 開發者指南。

Amazon Mobile Analytics

使用 Amazon Mobile Analytics，您可以跟蹤買家行為、彙總指標、生成數據可視化以及識別有意義的模式。有關 Mobile Analytics 的信息，請參閱AWS Mobile Analytics。

集成 Amazon Mobile Analytics

以下部分介紹瞭如何將 Mobile Analytics 與您的應用集成。

在 Mobile Analytics 控制台中創建應用

前往Amazon Mobile Analytics並建立應用程式。請注意`appId`值，因為您稍後需要。

> **Note**
>
> 若要進一步了解在控台中工作，請參Amazon Mobile Analytics 使用者指南。

在 Mobile Analytics 控制台中創建應用時，您需要指定一個 Cognito 身份池 ID。要創建新的 Cognito 身份池並生成 ID，請參閱Cognito Identity Analytics 指南。

將 Mobile Analytics 集成到您的應用

要從 Unity 訪問 Mobile Analytics，您將需要以下使用語句：

```
using Amazon.MobileAnalytics.MobileAnalyticsManager;
using Amazon.CognitoIdentity;
```

最佳做法是使用 Amazon Cognito 來提供臨時 AWS 登入資料給您的應用程式。這些登入資料可讓應用程式存取您的 AWS 資源。若要建立登入資料供應商，請依照Amazon Cognito Identity。

實例化MobileAnalyticsManager 實例，包含下列資訊：

- cognitoIdentityPoolId-應用程式的 Cognito Identity Pool 的 ID
- cognitoRegion-您的 Cognito 身份池的區域，例如」RegionEndpoint.USAST1」
- 區域-Mobile Analytics 服務的區域，例如」RegionEndpoint.USAST1」

- appId-添加應用程序時 Mobile Analytics 控制台生成的值

使用MobileAnalyticsClientContextConfig 初始化 **`MobileAnalyticsManager`** 實例，如下列程式碼片段所示：

```
// Initialize the MobileAnalyticsManager
void Start()
{
    // ...
    analyticsManager = MobileAnalyticsManager.GetOrCreateInstance(
        new CognitoAWSCredentials(<cognitoIdentityPoolId>, <cognitoRegion>),
        <region>,
        <appId>);
    // ...
}
```

> **Note**
>
> 應用程序 ID 是在應用程序創建嚮導期間為您生成的。這兩個值都必須與 Mobile Analytics 控制台中的值匹配。

所以此appId用於在 Mobile Analytics 控制台中對您的數據進行分組。要在 Mobile Analytics 控制台中創建應用後查找應用 ID，請瀏覽至 Mobile Analytics 控制台，單擊屏幕右上角的齒輪圖標。這將顯示應用程序管理頁面，其中列出了所有已註冊的應用程序及其應用 ID。

記錄獲利事件

適 SDK for Unity 提供了MonetizationEvent類，使您可以生成盈利事件以跟蹤在移動應用程序中進行的購買。下列程式碼片段演示如何建立獲利事件：

```
// Create the monetization event object
MonetizationEvent monetizationEvent = new MonetizationEvent();

// Set the details of the monetization event
monetizationEvent.Quantity = 3.0;
monetizationEvent.ItemPrice = 1.99;
monetizationEvent.ProductId = "ProductId123";
monetizationEvent.ItemPriceFormatted = "$1.99";
monetizationEvent.Store = "Your-App-Store";
```

```
monetizationEvent.TransactionId = "TransactionId123";
monetizationEvent.Currency = "USD";

// Record the monetiziation event
analyticsManager.RecordEvent(monetizationEvent);
```

記錄自訂事件

Mobile Analytics 允許您定義自定義事件。自定義事件完全由您定義；它們可幫助您跟蹤特定於應用或遊戲的用户操作。如需自訂事件的詳細資訊，請參自訂事件。在此示例中，假設您的應用程序是一個遊戲，並且您希望在用户完成關卡時記錄事件。建立」LevelComplete」事件，方法是創建一個新的AmazonMobileAnalyticsEvent實例：

```
CustomEvent customEvent = new CustomEvent("LevelComplete");

// Add attributes
customEvent.AddAttribute("LevelName","Level1");
customEvent.AddAttribute("CharacterClass","Warrior");
customEvent.AddAttribute("Successful","True");

// Add metrics
customEvent.AddMetric("Score",12345);
customEvent.AddMetric("TimeInLevel",64);

// Record the event
analyticsManager.RecordEvent(customEvent);
```

記錄會話

當應用程序失去焦點時，您可以暫停會話。InOnApplicationFocus檢查應用程序是否正在暫停。如果是這樣，調用PauseSession否則調用ResumeSession如下列程式碼片段所示：

```
void OnApplicationFocus(bool focus)
{
    if(focus)
    {
        analyticsManager.ResumeSession();
    }
    else
    {
        analyticsManager.PauseSession();
```

```
    }
}
```

默認情況下，如果用户將焦點從應用程序切換到不到 5 秒，並切換回應用程序，則會話將恢復。如果用户將焦點從應用程序切換為 5 秒或更長時間，將創建一個新會話。此設置可以在 awsconfig.xml 文件中進行配置。如需詳細資 Mobile Analytics 參適用於 Unity 的 AWS Mobile SDK 入門。

Amazon Simple Storage Service (S3)

Amazon Simple Storage Service (Amazon S3) 為開發人員和 IT 團隊提供安全、耐用、高可擴展性的對象儲存空間。Unity 開發人員可以利用 S3 動態加載遊戲使用的資產。這可以使遊戲最初從應用商店下載速度更快。

如需 S3 的詳細資訊，請參Amazon S3。

如需 AWS S3 區域可用性的資訊，請參AWS 服務區域可用性。

> **Note**
>
> 本文檔中的一些示例假定使用了名為ResultText以顯示跟蹤輸出。

建立和配置 S3 儲存貯體

Amazon S3 將您的資源存儲在 Amazon S3 存儲桶中-存儲在特定區域。每個 Amazon S3 儲存貯體都必須具有全域唯一的名稱。您可以使用Amazon S3 主控台建立儲存貯體。

建立 S3 儲存貯體

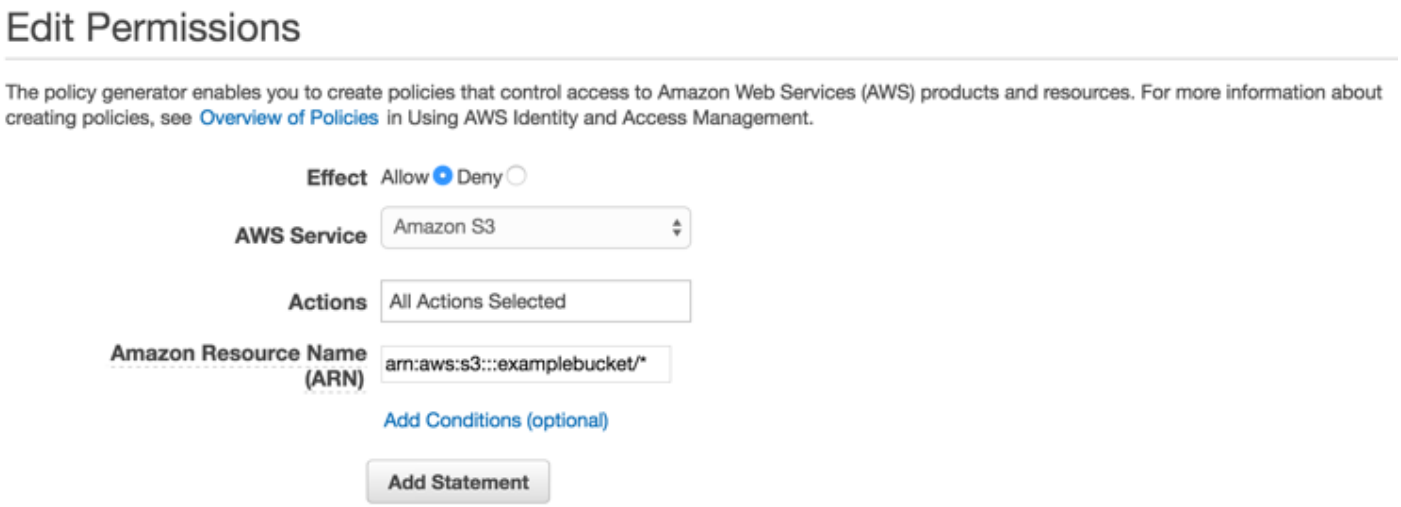
1. 登入Amazon S3 主控台，然後按一下建立儲存貯體。
2. 輸入儲存貯體名稱，選取區域，然後按一下建立。

設定 S3 的許可

默認 IAM 角色政策授予您的應用程序訪問 Amazon Mobile Analytics 和 Amazon Cognito Sync 的權限。要使您的 Cognito 身份池能夠訪問 Amazon S3，您必須修改身份池的角色。

1. 前往Identity and Access Management Console，然後按一下角色在左側窗格中。
2. 在搜尋方塊中輸入您的身分集區名稱。將會列出兩個角色：一個用於未經驗證的使用者，另一個用於通過驗證的使用者。
3. 按一下未經驗證用户的角色 (它會在您的身份池名稱後附加非授權)。
4. 按一下建立角色政策中，選取政策產生器，然後按一下選擇。
5. 在編輯許可頁面上，輸入以下圖片中顯示的設定，將 Amazon Resource Name (ARN) 替換為您自己的資源名稱 (ARN)。S3 儲存貯體的 ARN 如下所示`arn:aws:s3:::examplebucket/*`並由儲存

貯體所在的區域和儲存貯體的名稱組成。下面顯示的設置將為您的身份池提供完全訪問指定存儲桶的所有操作。



1. 按一下新增陳述式按鈕，然後單擊後續步驟。
2. 嚮導會顯示您產生的設定。按一下應用政策。

如需授予 S3 訪問權的詳細資訊，請參[授予 Amazon S3 儲存貯體的存取權](#)。

從控制台上傳文件

將測試文件上傳至儲存貯體：

1. 在 S3 控制台中的存儲桶視圖中，單擊上傳。
2. 按一下新增檔案，然後選擇要上傳的測試文件。在本教程中，我們假設您正在上傳一個名為`myImage.jpg`。
3. 選擇測試圖像後，單擊開始上傳。

（可選）配置 S3 請求的簽名版本

與 Amazon S3 的每次互動，可以經過驗證身份或是匿名進行。AWS 使用簽名版本 4 或簽名版本 2 算法對服務的調用進行身份驗證。

2014 年 1 月之後創建的所有新 AWS 區域僅支持簽名版本 4。但是，許多較舊的區域仍然支援 Signature 第 4 版和 Signature 第 2 版請求。

如果您的存儲桶位於不支持簽名版本 2 請求的區域之一，如此頁面時，您必須設定AWSConfigsS3.UseSignature版本 4 屬性設置為「true」。

如需 AWS Signature 版本的詳細資訊，請參驗證請求（AWS Signature 第 4 版）。

建立 Amazon S3 用户端

要使用 Amazon S3，我們首先需要創建一個亞馬遜 3 客户端實例，該實例需要引用CognitoAWSCredentials實例：

```
AmazonS3Client S3Client = new AmazonS3Client (credentials);
```

所以此AmazonS3Client類別是高級 S3 API 的進入點。

列出儲存貯體

要列出 AWS 賬户中的存儲桶，請調用AmazonS3Client.ListBucketsAsync方法，如下列程式碼所示：

```
// ResultText is a label used for displaying status information
ResultText.text = "Fetching all the Buckets";
 Client.ListBucketsAsync(new ListBucketsRequest(), (responseObject) =>
 {
     ResultText.text += "\n";
     if (responseObject.Exception == null)
     {
         ResultText.text += "Got Response \nPrinting now \n";
         responseObject.Response.Buckets.ForEach((s3b) =>
         {
             ResultText.text += string.Format("bucket = {0}, created date = {1} \n",
             s3b.BucketName, s3b.CreationDate);
         });
     }
     else
     {
         ResultText.text += "Got Exception \n";
     }
 });
```

列出物件

要列出存儲桶中的所有對象，請調用AmazonS3Client.ListObjectsAsync方法，如下列程式碼所示：

```
// ResultText is a label used for displaying status information
ResultText.text = "Fetching all the Objects from " + S3BucketName;

var request = new ListObjectsRequest()
{
    BucketName = S3BucketName
};

Client.ListObjectsAsync(request, (responseObject) =>
{
    ResultText.text += "\n";
    if (responseObject.Exception == null)
    {
        ResultText.text += "Got Response \nPrinting now \n";
        responseObject.Response.S3Objects.ForEach((o) =>
        {
            ResultText.text += string.Format("{0}\n", o.Key);
        });
    }
    else
    {
        ResultText.text += "Got Exception \n";
    }
});
```

下載物件

若要下載物件，請建立GetObject請求，指定存儲桶名稱和密鑰，並將對象傳遞給客户端的調用。GetObject非同步：

```
private void GetObject()
{
    ResultText.text = string.Format("fetching {0} from bucket {1}",
    SampleFileName, S3BucketName);
    Client.GetObjectAsync(S3BucketName, SampleFileName, (responseObj) =>
    {
```

```
        string data = null;
        var response = responseObj.Response;
        if (response.ResponseStream != null)
        {
           using (StreamReader reader = new StreamReader(response.ResponseStream))
           {
              data = reader.ReadToEnd();
           }

           ResultText.text += "\n";
           ResultText.text += data;
        }
    });
}
```

GetObject異步採用GetObject請求、回調和AsyncOptions實例。回調必須為以下類型：AmazonServiceCallback<GetObjectRequest, GetObjectResponse>。 所以此AsyncOptions實例是可選的。如果指定，它將確定回調是否在主線程上運行。

上傳物件

要上傳對象，請將對象寫入流，創建一個新的PostObject請求並指定密鑰、存儲桶名稱和流數據。

適用於 Unity 的 AWS 開發工具包使用不支持 HTTP PUT 操作的 WWW HTTP 客户端。要將數據元上傳到 S3 存儲桶，您需要使用 S3 的瀏覽器帖子，如下所示。

```
public void PostObject(string fileName)
{
    ResultText.text = "Retrieving the file";

    var stream = new FileStream(Application.persistentDataPath +
    Path.DirectorySeparatorChar + fileName,
    FileMode.Open, FileAccess.Read, FileShare.Read);

    ResultText.text += "\nCreating request object";
    var request = new PostObjectRequest()
    {
        Bucket = S3BucketName,
        Key = fileName,
        InputStream = stream,
        CannedACL = S3CannedACL.Private
    };
```

```
    ResultText.text += "\nMaking HTTP post call";

    Client.PostObjectAsync(request, (responseObj) =>
    {
        if (responseObj.Exception == null)
        {
            ResultText.text += string.Format("\nobject {0} posted to bucket {1}",
            responseObj.Request.Key, responseObj.Request.Bucket);
        }
        else
        {
            ResultText.text += "\nException while posting the result object";
            ResultText.text += string.Format("\n receieved error {0}",
            responseObj.Response.HttpStatusCode.ToString());
        }
    });
}
```

Amazon DynamoDB

Amazon DynamoDB 是一種快速、可輕鬆擴展、高度可用、經濟實惠、非關聯式資料庫服務。DynamoDB 移除資料儲存體的傳統擴展性限制，而仍維持低延遲及可預期的效能。如需的有關 DynamoDB 的資訊，請參。Amazon DynamoDB。

適用於 Unity 的 AWS 移動軟件開發工具包提供了一個用於使用 DynamoDB 的高級庫。您也可以直接針對低級 DynamoDB API 發出請求，但對於大多數用例，建議使用高級庫。所以此 AmazonDynamoDBClient 是高級庫中特別有用的部分。使用此類別，您可以執行各種建立、讀取、更新和刪除 (CRUD) 操作，以及執行查詢。

> **Note**
>
> 本文檔中的一些示例假定使用了名為ResultText以顯示跟蹤輸出。

集成 Amazon DynamoDB

要在 Unity 應用程序中使用 DynamoDB，您需要將 Unity 軟件開發工具包添加到您的項目中。如果您尚未建立資料表，請先執行下載適 SDK for Unity，然後請遵循設定適用於 Unity 的 AWS Mobile SDK。我們建議您使用 Amazon Cognito 身分來提供臨時 AWS 資料給您的應用程式。這些證書允許您的應用程序訪問 AWS 服務和資源。

若要在應用程式中使用 DynamoDB，您必須設定正確的權限。以下 IAM 策略允許用户刪除、獲取、放置、掃描和更新特定 DynamoDB 表中的項目，該表由ARN：

```
{
"Statement": [{
    "Effect": "Allow",
    "Action": [
        "dynamodb:DeleteItem",
        "dynamodb:GetItem",
        "dynamodb:PutItem",
        "dynamodb:Scan",
        "dynamodb:UpdateItem"
    ],
    "Resource": "arn:aws:dynamodb:us-west-2:123456789012:table/MyTable"
}]
}
```

此政策應該適用於指派給 Cognito 身分集區的角色，但您需要將 **Resource** 值，使用適用於 DynamoDB 表的正確 ARN。Cognito 自動為您的新身份池創建角色，並且您可以在IAM 主控台。

您應該根據您的應用需要來添加或移除允許的操作。若要進一步了解 IAM 政策，請參閱使用 IAM。若要進一步了解 DynamoDB 特定的策略，請參。使用 IAM 控制對 DynamoDB 資源的存取。

建立 DynamoDB 資料表

現在我們已經設置了權限和憑據，讓我們為我們的應用程序創建一個 DynamoDB 表。若要建立資料表，請轉到DynamoDB 主控台，然後請遵循下列步驟：

1. 按一下 Create Table (建立資料表)。
2. EnterBookstore作為資料表的名稱。
3. 選擇雜湊作為主鍵類型。
4. 選擇數字並輸入id作為哈希屬性名稱。按一下 Continue (繼續)。
5. 按一下Continue跳過添加索引。
6. 將讀取容量設置為10和寫入容量5。按一下 Continue (繼續)。
7. 輸入通知電子郵件，然後單擊Continue創建吞吐量警報。
8. 按一下 Create (建立)。DynamoDB 將建立資料庫。

建立 DynamoDB 客户端

為了讓我們的應用程序與 DynamoDB 表進行交互，我們需要一個客户端。我們可以創建默認的 DDynamodDB 客户端，如下所示：

```
var credentials = new CognitoAWSCredentials(IDENTITY_POOL_ID, RegionEndpoint.USEast1);
AmazonDynamoDBClient client = new AmazonDynamoDBClient(credentials);
DynamoDBContext Context = new DynamoDBContext(client);
```

所以此AmazonDynamoDBClient 類別是 DynamoDB API 的入口點。該類提供了用於創建、描述、更新和刪除表以及其他操作的實例方法。上下文在客户端上添加了一個抽象層，並允許您使用其他功能，如對象持久化模型。

描述資料表

要獲取 DynamoDB 表的描述，我們可以使用以下代碼：

```
resultText.text +=("\n*** Retrieving table information ***\n");
       var request = new DescribeTableRequest
       {
           TableName = @"ProductCatalog"
       };
       Client.DescribeTableAsync(request, (result) =>
       {
               if (result.Exception != null)
               {
                       resultText.text += result.Exception.Message;
                       Debug.Log(result.Exception);
                       return;
               }
               var response = result.Response;
               TableDescription description = response.Table;
               resultText.text += ("Name: " + description.TableName + "\n");
               resultText.text += ("# of items: " + description.ItemCount + "\n");
               resultText.text += ("Provision Throughput (reads/sec): " +
                   description.ProvisionedThroughput.ReadCapacityUnits + "\n");
               resultText.text += ("Provision Throughput (reads/sec): " +
                   description.ProvisionedThroughput.WriteCapacityUnits + "\n");

       }, null);
   }
```

在此範例中，我們將建立用户端和DescribeTable請求對象，將我們的表的名稱分配給 **TableName** 屬性，然後將請求物件傳遞給DescribeTable異步方法AmazonDynamoDBClient 物件。DescribeTable異步還需要一個委託，該委託將在異步操作完成時調用。

> **Note**
>
> 上的所有異步方法AmazonDynamoDBClient 接受異步操作完成時調用的委託。

保存對象

要將對象保存到 DynamoDB，請使用SaveAsync<T>方 法AmazonDynamoDBClient 對象，其中 T 是要保存的對象的類型。

我們將我們的數據庫稱為「書店」，根據這個主題，我們將實現一個數據模型來記錄與書籍相關的屬性。以下是定義我們數據模型的類。

```
[DynamoDBTable("ProductCatalog")]
    public class Book
    {
        [DynamoDBHashKey]   // Hash key.
        public int Id { get; set; }
        [DynamoDBProperty]
        public string Title { get; set; }
        [DynamoDBProperty]
        public string ISBN { get; set; }
        [DynamoDBProperty("Authors")]    // Multi-valued (set type) attribute.
        public List<string> BookAuthors { get; set; }
    }
```

當然，對於真正的書店應用程序，我們需要額外的字段，例如作者和價格。書類使用 [DynamoDBTable] 屬性進行裝飾，這定義了將寫入 Book 類型的數據庫表對象。Book 類的每個實例的密鑰使用 [DynamoDBHashKey] 屬性。屬性使用 [DynamoDB 屬性] 屬性標識，這些屬性指定了數據庫表中要寫入屬性的列。建立模型後，我們可以編寫一些方法來建立、檢索、更新和刪除 Book 物件。

建立電子書

```
private void PerformCreateOperation()
{
    Book myBook = new Book
    {
        Id = bookID,
        Title = "object persistence-AWS SDK for.NET SDK-Book 1001",
        ISBN = "111-1111111001",
        BookAuthors = new List<string> { "Author 1", "Author 2" },
    };

    // Save the book.
    Context.SaveAsync(myBook,(result)=>{
        if(result.Exception == null)
            resultText.text += @"book saved";
    });
}
```

檢索電子書

```
private void RetrieveBook()
```

```
{
    this.displayMessage += "\n*** Load book**\n";
    Context.LoadAsync<Book>(bookID,
                            (AmazonDynamoResult<Book> result) =>
    {
        if (result.Exception != null)
        {

            this.displayMessage += ("LoadAsync error" +result.Exception.Message);
            Debug.LogException(result.Exception);
            return;
        }
        _retrievedBook = result.Response;
        this.displayMessage += ("Retrieved Book: " +
                                "\nId=" + _retrievedBook.Id +
                                "\nTitle=" + _retrievedBook.Title +
                                "\nISBN=" + _retrievedBook.ISBN);
        string authors = "";
        foreach(string author in _retrievedBook.BookAuthors)
            authors += author + ",";
        this.displayMessage += "\nBookAuthor= "+ authors;
        this.displayMessage += ("\nDimensions= "+ _retrievedBook.Dimensions.Length + "
 X " +
                                _retrievedBook.Dimensions.Height + " X " +
                                _retrievedBook.Dimensions.Thickness);

    }, null);
}
```

更新電子書

```
private void PerformUpdateOperation()
{
    // Retrieve the book.
    Book bookRetrieved = null;
    Context.LoadAsync<Book>(bookID,(result)=>
    {
        if(result.Exception == null )
        {
            bookRetrieved = result.Result as Book;
            // Update few properties.
            bookRetrieved.ISBN = "222-2222221001";
```

```
            // Replace existing authors list with this
            bookRetrieved.BookAuthors = new List<string> { "Author 1", "Author x" };
            Context.SaveAsync<Book>(bookRetrieved,(res)=>
            {
                if(res.Exception == null)
                     resultText.text += ("\nBook updated");
            });
        }
   });
}
```

刪除電子書

```
private void PerformDeleteOperation()
{
    // Delete the book.
    Context.DeleteAsync<Book>(bookID,(res)=>
    {
        if(res.Exception == null)
        {
            Context.LoadAsync<Book>(bookID,(result)=>
            {
                 Book deletedBook = result.Result;
                 if(deletedBook==null)
                     resultText.text += ("\nBook is deleted");
            });
        }
   });
}
```

Amazon Simple Notification Service

使用 Amazon Simple Notification Service (SNS) 和 Unity SDK，您可以編寫可接收移動推送通知的 iOS 和 Android 應用程序。如需 SNS 的詳細資訊，請參Amazon Simple Notification Service。

本主題將介紹如何配置適用於 Unity 的 AWS 開發工具包示例應用程序 SNSEXAMple.Unity，以便通過 Amazon SNS 接收移動推送通知。

您可以使用 Snsexample.Unity 示例創建 iOS 和安卓應用程序。iOS 和 Android 之間的配置步驟不同，請閱讀下面針對您目標平台的相應部分。

先決條件

使用此解決方案需要下列先決條件。

設定 SNS 的許可

創建 Cognito 份池時，將生成兩個 IAM 角色：

- 幹邑 /<Identity-Pool-Name> _ 授權DefaultRole— 經過身份驗證的用户的默認 IAM 角色
- 幹邑 <Identity-Pool-Name>/_DefaultRole— 未驗證使用者的默認 IAM 角色

您必須向這些角色添加訪問 Amazon SNS 服務的權限。若要執行此作業：

1. 瀏覽至IAM 主控台，然後選擇要配置的 IAM 角色。
2. 按一下連接政策，請選取卓越亞馬遜FullAccess策略，然後單擊連接政策。

> **Note**
>
> 使用卓越亞馬遜 SNSFullAccess在生產環境中，我們會使用它來讓您快速啟動和運行。如需為 IAM 角色指定許可的詳細資訊，請參IAM 角色許可概觀。

iOS 先決條件

- 蘋果 iOS 開發者計劃會員
- 生成簽名標識

- 創建為推送通知配置的置備配置文件

您需要在物理設備上運行應用程序才能接收推送通知。要在設備上運行應用程序，您必須具有蘋果 iOS 開發者計劃會員。成為會員資格後，您可以使用 Xcode 來生成簽署身份。如需詳細資訊，請參 Apple 的應用程式分發快速入門文件中)。接下來，您需要為推送通知配置配置文件，瞭解更多信息，請參閱 Apple 的配置推送通知通知文件中)。

Android 先決條件

- 安裝 Android SDK
- 安裝 JDK
- android-support-v4.jar
- google-play-services.jar

配置 iOS 的統一示例應用

開啟 Unity (Unity) 編輯器並建立新專案。導入適用於 Unity 的 AWS 開發工具包包，方法是選擇資產/導入套件/自訂套件並選取aws-unity-sdk-SN-2.0.0.1. 單元包裝。確保導入套件對話框，然後單擊匯入。

Unity 配置

執行下列步驟來設定 Unity 專案：

1. 在 中專案窗格中，前往資產/AWSSDK/例子，然後打開 SNSample 場景。
2. 在 中IARCHY窗格中，選擇「信號採樣」。
3. 在 中Inspector窗格中指定您的 Cognito 份池 ID。
4. 請注意，有一個文本框iOS 平台應用程序 ARN，稍後將生成該信息。
5. 選擇File (檔案)/建置設定，在建置設定對話框中，單擊添加當前按鈕下方組態建置中的場景列表框將當前場景添加到構建中。
6. UNDER平台選取iOS並按一下播放器設定...按鈕，在Inspector 窗格，單擊 iPhone 圖標，然後向下滾動到識別部分並指定Bundle Identifier (套件組合識別碼)。

iOS 設定

執行下列步驟來設定示例，以配置 iOS 特定設定：

1. 在 Web 瀏覽器中，前往蘋果開發人員會員中心中，按一下證書、標識符和配置文件。
2. 按一下識別碼下iOS 應用程式下，按一下位於 Web 頁面右上角的加號按鈕，以新增 iOS App ID，然後輸入 App ID 描述。
3. 向下捲動到添加 ID 後綴部分，然後選擇明確應用程式 ID並輸入捆綁標識符。
4. 向下捲動到App Services部分，然後選擇推送通知。
5. 按一下Continue按鈕。
6. 按一下提交按鈕。
7. 按一下完成按鈕。
8. 選擇剛剛創建的應用程序 ID，然後單擊Edit (編輯)按鈕。
9. 向下捲動到推送通知部分。
10. 按一下建立憑證按鈕SSL 憑證。
11. 按照說明創建證書簽名請求 (CSR)、上傳請求並下載將用於與 Apple 通知服務 (APNS) 通信的 SSL 證書。
12. 返回證書、標識符和配置文件網頁上，按一下All (全部)下設定檔。
13. 按一下位於右上角的加號按鈕，以添加新的預配概要文件。
14. 選擇iOS 應用程式開發並按一下Continue按鈕。
15. 選擇您的應用程序 ID，然後單擊Continue按鈕。
16. 選擇您的開發者證書，然後單擊Continue按鈕。
17. 選擇您的設備，然後單擊Continue按鈕。
18. 輸入配置文件名稱，然後單擊Generate按鈕。
19. 下載並雙擊置備文件以安裝置配置文件。

新增後，您可能需要在 Xcode 中刷新 Project 中。Xcode 中：

1. 選取Xcode/偏好設定選單項目。
2. 選取帳戶選項卡上，選擇您的 Apple ID，然後單擊查看詳細資訊。
3. 單擊對話框左下角的刷新按鈕可刷新您的配置配置文件，並確保顯示新配置文件。

SNS 配置

1. 執行KeyChain訪問應用程序，選擇我的憑證，右鍵單擊您生成的 SSL 證書以連接到 APNS，然後選擇匯出，系統將提示您指定文件的名稱和密碼以保護證書。證書將保存在 P12 文件中。

2. 在 Web 瀏覽器中，前往SNS 控制台並按一下應用程式在畫面左側。
3. 按一下建立平台應用程式來建立新 SNS 平台應用程式。
4. 輸入應用程式名稱。
5. 選擇蘋果推送通知服務沙箱 (APNS_SANDBOX)為了推送通知平台。
6. 按一下選取檔案，然後選擇導出 SSL 證書時創建的 P12 文件。
7. 輸入導出 SSL 憑證時指定的密碼，然後單擊從檔案載入憑據。
8. 按一下建立平台應用程式。
9. 選擇剛剛創建的平台應用程序並複製應用程序 ARN。
10. 在 Unity 編輯器中返回到您的項目，選擇故障採樣中的IARCHY窗格中的Inspector窗格中，然後將平台應用程式 ARN 貼入標記為iOS 平台應用程序 ARN。
11. 選擇File (檔案)/建置設定並按一下建置按鈕，這將創建一個 Xcode 項目。

使用 Xcode

1. 打開 Xcode 項目，然後在項目導航器中選擇項目。
2. 驗證是否正確設置了捆綁標識符
3. 驗證您的 Apple 開發者帳戶是否在團隊— 預配置文件需要這樣做，才會生效。
4. 構建項目並在您的設備上運行它。
5. 按一下註冊通知中，按一下OK (OK)以允許通知，應用程序將顯示您的設備令牌

在 中SNS 控制台中，按一下應用程式，選擇您的平台應用程序，然後單擊建立平台端點，然後輸入應用程序顯示的設備令牌。

此時，您的應用程序、APNS 和 NSN 已完全配置。您可以選擇平台應用程序，選擇終端節點，然後單擊發佈至端點向您的設備發送推送通知。

統一示例 (iOS)

該樣本創建了 CognitoAWSCredentials實例生成允許應用程序調用 AWS 服務的臨時有限範圍證書。它還會建立一個AmazonSimpleNotificationService客户端與 SNS 進行通信。該應用程序顯示兩個標記為註冊通知和取消註冊。

當註冊獲取通知按鈕時，`RegisterDevice()`方法被調用。`RegisterDevice()`呼叫`UnityEngine.iOS.NotificationServices.RegisterForNotifications`，它指定將使用

哪些通知類型（警報、聲音或徽章）。它還對 APNS 進行異步調用以獲取設備令牌。由於沒有定義回調，CheckForDeviceToken被重複調用（最多 10 次）來檢查設備令牌。

檢索令牌
時AmazonSimpleNotificationServiceClient.CreatePlatformEndpointAsync()為 SNS 平台應用程序創建終端節點。

現在，該示例配置為接收推式通知。您可以瀏覽到SNS 控制台中，按一下應用程式，選擇平台應用程式，選擇終端節點，然後單擊發佈至端點。選擇要使用的終端節點，然後單擊發佈至端點。在文字方塊中輸入文字消息，然後單擊發佈訊息發佈訊息。

配置適用於安卓系統的統一示例應用

開啟 Unity (Unity) 編輯器並建立新專案。導入適用於 Unity 的 AWS 開發工具包包，方法是選擇資產/導入套件/自訂套件並選取aws-unity-sdk-SN-2.0.0.1. 單元包裝。確保導入套件對話框，然後單擊匯入。

Unity 配置

執行下列步驟來設定 Unity 專案：

1. 在 中專案窗格中，前往資產/AWSSDK/例子，然後打開 SNSample 場景。
2. 在 中IARCHY窗格中，選擇「信號採樣」。
3. 在 中Inspector窗格中指定您的 Cognito 份池 ID。
4. 請注意，有一個文本框安卓平台應用程序 ARN和Google 主控台專案 ID，稍後將生成該信息。
5. 選擇File (檔案)/建置設定，在建置設定對話框中，單擊添加當前按鈕下方組態建置中的場景列表框將當前場景添加到構建中。
6. UNDER平台選取安卓，然後按一下播放器設定...按鈕，在Inspector 窗格，單擊 Android 圖標，然後向下滾動到識別部分並指定Bundle Identifier (套件組合識別碼)。
7. 複製android-support-v4.jar 和google-play-services.jar 放入資產/外掛程式/安卓目錄中專案窗格。

如需其位於何處的詳細資訊android-support-v4.jar，請參閱Android Support 庫設定。如需如何找到google-play-services.jar，請參閱適用於安卓設置的谷歌 API。

Android 設定

首先添加一個新的谷歌 API 項目：

1. 在 Web 瀏覽器中，前往Google 開發者控制台中，按一下建立專案。
2. 在 中新建工程框中，輸入項目名稱，記下項目編號（稍後需要），然後單擊建立。

接下來，啟用 Google Cloud Message (GCM) 服務，請執行以下步驟：

1. 在 Google 開發者控制台中，您的新項目應該已被選中，如果沒有，請在頁面頂部的下拉菜單中選擇它。
2. 選擇API & 授權在頁面左側側欄中輸入。
3. 在搜尋方塊中，輸入 Google Cloud Message (Google Cloud Message)，然後按一下位於適用於安卓的谷歌雲消息鏈接。
4. 按一下啟用 API。

最後獲取 API 密鑰：

1. 在谷歌開發人員控制台中，選擇API & 授權 >登入資料。
2. UNDER公用 API 訪問中，按一下建立新的密鑰。
3. 在 中建立新的金鑰對話框中，單擊服務器密鑰。
4. 在生成的對話框中，單擊建立並複製顯示的 API 密鑰。

稍後您將使用 API 密鑰執行身份驗證。

SNS 配置

1. 在 Web 瀏覽器中，前往SNS 控制台並按一下應用程式在畫面左側。
2. 按一下建立平台應用程式來建立新 SNS 平台應用程式。
3. 輸入應用程式名稱
4. 選擇Google Cloud Message (GCM)為了推送通知平台
5. 將 API 密鑰貼入標記為API 金鑰。
6. 按一下建立平台應用程式
7. 選擇剛剛創建的平台應用程序並複製應用程序 ARN。
8. 在 Unity 編輯器中返回到您的項目，選擇故障採樣中的IARCHY窗格中的Inspector窗格中，然後將平台應用程式 ARN 貼入標記為安卓平台應用程序 ARN將您的專案編號放入標記為Google 主控台專案 ID。
9. Connect 您的 Android 裝置至電腦，選擇File (檔案)/建置設定，然後按一下建置並執行。

統一示例 (安卓系統)

該樣本創建了 CognitoAWSCredentials實例生成允許應用程序調用 AWS 服務的臨時有限範圍證書。它還會建立一個AmazonSimpleNotificationService客户端與 SNS 進行通信。

該應用程序顯示兩個標記為註冊通知和取消註冊。當註冊獲取通知按鈕時，`RegisterDevice()`方法被調用。`RegisterDevice()`呼叫`GCM.Register`，它將應用程序註冊到 GCM。GCM 是在示例代碼中定義的類。它進行異步調用以向 GCM 註冊應用程序。

當回調被調用`AmazonSimpleNotificationServiceClient.CreatePlatformEndpointAsync`來創建接收 SNS 消息的平台終端節點。

現在，該示例配置為接收推式通知。您可以瀏覽到SNS 控制台中，按一下應用程式，選擇平台應用程式，選擇終端節點，然後單擊發佈至端點。選擇要使用的終端節點，然後單擊發佈至端點。在文字方塊中輸入文字消息，然後單擊發佈訊息發佈訊息。

AWS Lambda

AWS Lambda 是一種運算服務，可執行程式碼來回應請求或事件並自動為您管理運算資源，讓您輕鬆建構可快速回應新資訊的應用程式。AWS Lambda 函數可直接從移動、IoT 和 Web 應用程式呼叫，並同步回應，讓您輕鬆地為行動應用程式建立可擴展、安全且可用性高的後端，不需要佈建或管理基礎設施。

AWS Lambda 可以執行您的 Lambda 函數以響應以下情況之一：

- 事件，例如離散更新（例如，Amazon S3 中的對象創建事件或CloudWatch警報）或流式更新（例如，網站點擊流或連接設備的輸出）。
- JSON 從自定義應用程序輸入或 HTTPS 命令。

AWS Lambda 只有在需要時才會執行程式碼，可自動從每天數項請求擴展成每秒數千項請求。透過這些功能，您就能使用 Lambda 輕鬆建置處理 AWS 服務 (如 Amazon S3 和 Amazon DynamoDB) 的觸發、處理儲存在 Amazon Kinesis 中的串流資料，或建立自己的後端，在 AWS 規模、效能及安全性中執行。

若要進一步了解 AWS Lambda 工作原理，請參AWS Lambda：運作方式。

許可

Lambda 函數有兩種類型的許可：

- 執行許可— Lambda 函數訪問您賬户中其他 AWS 資源所需的許可。您可建立 IAM 角色 (稱為執行角色) 授予這些許可。
- 叫用許可— 事件源與 Lambda 函數通信所需的權限。根據調用模型（推或拉模型），您可以使用執行角色或資源策略（與 Lambda 函數關聯的訪問策略）授予這些權限。

項目設定

設定 AWS Lambda 的許可

1. 開啟AWS IAM 主控台。
2. 將此自定義策略附加到您的角色，以允許您的應用程序調用 AWS Lambda。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lambda:*"
      ],
      "Resource": "*"
    }
  ]
}
```

建立新執行角色

此角色適用於您將在下一步中創建的 Lambda 函數，並確定該函數可以訪問哪些 AWS 資源。

1. 開啟AWS IAM 主控台。
2. 按一下角色。
3. 按一下創建新角色。
4. 按照屏幕上的說明選擇 Lambda 函數需要訪問的服務和相應策略。例如，如果您希望 Lambda 函數建立 S3 儲存貯體，則您的政策將需要 S3 的寫入許可。
5. 按一下建立角色。

在 AWS Lambda 中創建函數

1. 開啟AWS Lambda 主控台。
2. 按一下建立 Lambda 函數。
3. 按一下略過以略過建立藍圖。
4. 在下個畫面上配置自己的功能。輸入函數名稱、描述，然後選擇您的運行時間。根據您選擇的運行時遵循螢幕上的指示。通過將新創建的執行角色分配給函數來指定執行權限。
5. 完成後，按一下下一頁。
6. 按一下建立函數。

建立 Lambda 客户端

```
var credentials = new CognitoAWSCredentials(IDENTITY_POOL_ID, RegionEndpoint.USEast1);
var Client = new AmazonLambdaClient(credentials, RegionEndpoint.USEast1);
```

建立請求物件

創建請求對象以指定調用類型和函數名稱：

```
var request = new InvokeRequest()
{
    FunctionName = "hello-world",
    Payload = "{\"key1\" : \"Hello World!\"}",
    InvocationType = InvocationType.RequestResponse
};
```

叫用 Lambda 函數

調用調用，傳遞請求對象：

```
Client.InvokeAsync(request, (result) =>
{
    if (result.Exception == null)
    {
        Debug.Log(Encoding.ASCII.GetString(result.Response.Payload.ToArray()));
    }
    else
    {
        Debug.LogError(result.Exception);
    }
});
```

故障診斷

由於適用於 Unity 的 AWS 開發工具包使用的 Unity.www 類的限制，因此在調用 AWS 服務時出現問題時不會返回詳細的錯誤消息。本主題介紹瞭如何解決此類問題的一些想法。

確保 IAM 角色具有所需權限

調用 AWS 服務時，您的應用程序使用 Cognito 身份池中的身份。池中的每個身份都與 IAM（Identity and Access Management）角色相關聯。角色具有一個或多個與其關聯的策略文件，用於指定分配給該角色的用户有權訪問哪些 AWS 資源。默認情況下，將會創建兩個角色，一個用於通過驗證的使用者，另一個用於未驗證的使用者。您需要修改現有策略文件，或者將新策略文件與應用程序所需的權限相關聯。如果您的應用程序允許經過身份驗證的用户和未經身份驗證的用户，則必須向這兩個角色授予訪問您的應用程序所需的 AWS 資源的權限。

以下政策文件説明如何允許存取 S3 存儲體：

```
{
  "Statement": [
    {
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::MYBUCKETNAME/*",
      "Principal": "*"
    }
  ]
}
```

以下策略文件顯示瞭如何授予對 DynamoDB 數據庫的訪問權限：

```
{
"Statement": [{
    "Effect": "Allow",
    "Action": [
        "dynamodb:DeleteItem",
        "dynamodb:GetItem",
```

```
        "dynamodb:PutItem",
        "dynamodb:Scan",
        "dynamodb:UpdateItem"
    ],
    "Resource": "arn:aws:dynamodb:us-west-2:123456789012:table/MyTable"
}]
}
```

如需政策的詳細資訊，請參IAM 政策。

使用 HTTP 代理調試器

如果您的應用程序調用的 AWS 服務具有 HTTP 或 HTTPS 終端節點，則可以使用 HTTP/HTTPS 代理調試器查看請求和響應，以更深入地瞭解正在發生的內容。有許多 HTTP 代理調試器可用，例如：

- 查爾斯-OSX 的 Web 調試代理
- 提琴手-一個網絡調試代理軟件

> **Important**
>
> 在運行 Charles Web 調試代理時，Cognito 憑據提供程序存在一個已知問題，這會阻止憑據提供程序正常工作。

查爾斯和 Fiddler 都需要一些配置才能查看 SSL 加密流量，請閱讀這些工具的文檔以瞭解更多信息。如果您使用的 Web 調試代理無法配置為顯示加密流量，請打開 aws_endpoint _json 文件（位於 AWSUnitySDK/AWSCore/資源），並將需要調試的 AWS 服務的 HTTP 標籤設置為 true