Guía para desarrolladores

AWS SDK móvil para Unity

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

AWS SDK móvil para Unity: Guía para desarrolladores

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

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

..... vi
¿Qué es el SDK AWS móvil para Unity? 1
Guías y temas relacionados 1
Contenido de referencia archivado 1
Compatibilidad 2
Descargar Mobile SDK para Unity 2
¿Qué se incluye en el Mobile SDK para Unity? 2
Configurar el AWS Mobile SDK para Unity 3
Requisitos previos 3
Paso 1: Descargar el AWS Mobile SDK para Unity 3
Paso 2: Configurar el AWS Mobile SDK para Unity 4
Crear una escena 4
Configurar la región predeterminada del servicio de AWS 4
Configurar la información de registro 4
Uso del archivo link.xml 5
Paso 3: Obtener el ID de grupo de identidades mediante Amazon Cognito 6
Siguientes pasos 7
Introducción al AWS Mobile SDK para Unity 8
Amazon Cognito Identity 8
Amazon Cognito Sync 8
Uso de la CognitoSyncManager muestra 9
Dynamo DB 9
Uso de la muestra de DynamoDB 10
Mobile Analytics 10
Configuración de Mobile Analytics 11
Uso de la muestra de Mobile Analytics 11
Amazon S3 12
Configuración de la firma de S3 predeterminada 12
Uso de la muestra de S3 13
Amazon Simple Notification Service 13
AWS Lambda 14
Amazon Cognito Identity 15
¿Qué es Amazon Cognito Identity? 15
Uso de un proveedor público para autenticar a los usuarios 15

Uso de identidades autenticadas por el desarrollador ... 15
Amazon Cognito Sync ... 16
Amazon Mobile Analytics ... 17
Integración de Amazon Mobile Analytics ... 17
Creación de una aplicación en la consola de Mobile Analytics ... 17
Integración de Mobile Analytics en una aplicación ... 17
Registro de eventos de monetización ... 18
Registro de eventos personalizados ... 19
Registro de sesiones ... 20
Amazon Simple Storage Service (S3) ... 21
Creación y configuración de un bucket de S3 ... 21
Creación de S3 Bucket ... 21
Definición de permisos de S3 ... 21
Carga de archivos desde la consola ... 22
(opcional) Configuración de la versión de Signature para las solicitudes de S3 ... 23
Creación del cliente de Amazon S3 ... 23
Lista de buckets ... 23
Lista de objetos ... 24
Descarga de un objeto ... 25
Carga de un objeto ... 26
Amazon DynamoDB ... 28
Integración de Amazon DynamoDB ... 28
Creación de una tabla de DynamoDB ... 29
Creación de un cliente de DynamoDB ... 30
Descripción de una tabla ... 30
Guardar un objeto ... 31
Creación de un objeto Book ... 32
Recuperación de un objeto Book ... 32
Actualización de un objeto Book ... 33
Eliminación de un objeto Book ... 33
Amazon Simple Notification Service ... 35
Requisitos previos ... 3
Establecer permisos de SNS ... 35
Requisitos previos de iOS ... 36
Requisitos previos de Android ... 36
Configurar la aplicación de ejemplo para iOS ... 36

Configuración de Unity ... 36
Configuración para iOS ... 37
Configuración de SNS ... 38
Uso de Xcode ... 39
Ejemplo de Unity (iOS) ... 40
Configurar la aplicación de ejemplo Unity para Android ... 40
Configuración de Unity ... 41
Configuración para Android ... 41
Configuración de SNS ... 42
Ejemplo de Unity (Android) ... 43
AWS Lambda ... 44
Permisos ... 44
Configuración del proyecto ... 45
Establecer permisos para AWS Lambda ... 45
Crear un nuevo rol de ejecución ... 45
Crear una función en AWS Lambda ... 46
Crear un cliente de Lambda ... 46
Crear un objeto de una solicitud ... 46
Invocar una función de Lambda ... 46
Solución de problemas ... 48
Comprobar que un rol de IAM tiene los permisos requeridos ... 48
Uso de un depurador de proxy HTTP ... 49

El SDK AWS móvil para Unity ahora está incluido en AWS SDK para .NET. Esta guía hace referencia a la versión archivada de Mobile SDK para Unity. Para obtener más información, consulte ¿Qué es el SDK AWS móvil para Unity?.

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.

¿Qué es el SDK AWS móvil para Unity?

El SDK AWS móvil para Unity ahora está incluido en SDK para .NET. Para obtener más información, consulte la Guía para desarrolladores de AWS SDK para .NET .

Esta guía ya no se actualiza; hace referencia a la versión archivada del Mobile SDK para Unity.

Guías y temas relacionados

- Para el desarrollo de aplicaciones móviles y de front-end, recomendamos utilizar AWS Amplify.
- Para conocer consideraciones especiales sobre el uso del mismo AWS SDK para .NET en tus aplicaciones de Unity, consulta Consideraciones especiales sobre el soporte de Unity en la Guía para AWS SDK para .NET desarrolladores.
- Como referencia, puedes encontrar la versión archivada del SDK AWS móvil para Unity en GitHub.

Contenido de referencia archivado

El SDK móvil archivado para Unity contiene un conjunto de clases.NET que permiten que los juegos escritos con Unity utilicen los AWS servicios. Las aplicaciones escritas con el Mobile SDK para Unity se pueden ejecutar en dispositivos iOS o Android.

AWS Los servicios compatibles incluyen:

- 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)

Estos servicios le permiten autenticar a los usuarios, guardar datos del juego y de los jugadores, guardar objetos en la nube, enviar notificaciones de inserción y recopilar y analizar datos de uso.

Compatibilidad

Mobile SDK para Unity v3 es compatible con las versiones 4.6 y posteriores de Unity.

La última versión de Mobile SDK para Unity introdujo mejoras que pueden requerir la modificación del código para incorporarlo en su proyecto. Para obtener más información sobre estos cambios, consulta Mejoras en el SDK AWS móvil para Unity en el blog Front-End Web & Mobile.

Descargar Mobile SDK para Unity

También puede descargar el Mobile SDK para Unity como un archivo.zip aquí.

¿Qué se incluye en el Mobile SDK para Unity?

Para ver la lista completa de NuGet paquetes, muestras y otros archivos del Mobile SDK for Unity, consulta AWS SDK para .NETen GitHub.

Configurar el AWS Mobile SDK para Unity

Para comenzar a utilizar el AWS Mobile SDK para Unity, puede configurar el SDK y comenzar a compilar un proyecto nuevo, o puede integrar el SDK en un proyecto existente. También puede clonar y ejecutar las muestras para hacerse una idea del funcionamiento del SDK.

Requisitos previos

Para poder usar el AWS Mobile SDK para Unity, necesitará lo siguiente:

- Una cuenta de AWS
- Unity versión 4.x o 5.x (se necesita Unity 4.6.4p4 o Unity 5.0.1p3 si quiere escribir aplicaciones que se ejecuten en iOS de 64 bits)

Después de completar los requisitos previos, tendrá que hacer lo siguiente para comenzar:

1. Descargar el AWS Mobile SDK para Unity.
2. Configurar el AWS Mobile SDK para Unity.
3. Obtener las credenciales de AWS con Amazon Cognito.

Paso 1: Descargar el AWS Mobile SDK para Unity

En primer lugar, descargue el AWS Mobile SDK para Unity. Cada paquete del SDK debe consumir el servicio de AWS correspondiente en función del nombre del paquete. Por ejemplo, el paquete aws-unity-sdk-dynamodb -2.1.0.0.unitypackage se usa para llamar al servicio AWS DynamoDB. Puede importar todos los paquetes o solo los que vaya a utilizar.

1. Abra el editor de Unity y cree un proyecto vacío. Utilice la configuración predeterminada.
2. Seleccione Assets (Activos) > Import Package (Importar paquete) > Custom Package (Paquete personalizado).
3. En el cuadro de diálogo Import (Importar), vaya a los archivos .unitypackage que desee usar y selecciónelos.
4. En el cuadro de diálogo Importing (Importación) de importación de paquetes, compruebe que todos los elementos están seleccionados y haga clic en Import (Importar).

Paso 2: Configurar el AWS Mobile SDK para Unity

Crear una escena

Al trabajar con el AWS Mobile SDK para Unity, puede comenzar incluyendo la siguiente línea de código en el método `Start` o `Awake` de la clase de comportamiento mono:

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

Elija New Scene (Nueva escena) en el menú File (Archivo) para crear la escena.

El AWS SDK para Unity contiene clases de cliente para cada servicio de AWS que admite. Estos clientes se configuran con un archivo denominado awsconfig.xml. En la siguiente sección se describen la configuración más utilizada en el archivo awsconfig.xml. Para obtener más información sobre la configuración, consulte la Referencia de la API del SDK de Unity.

Configurar la región predeterminada del servicio de AWS

Para configurar la región predeterminada de todos los clientes de servicios:

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

De este modo se establece la región predeterminada para todos los clientes de servicios en el SDK de Unity. Esta configuración se puede anular al especificar explícitamente la región en el momento de crear una instancia del cliente de servicio, del siguiente modo:

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

Configurar la información de registro

La configuración de registro se especifica de este modo:

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

Esta configuración se utiliza para establecer el registro en Unity. Al efectuar el registro en `UnityLogger`, el marco de trabajo imprime internamente la salida en los registros de depuración.

Si quieres registrar las respuestas HTTP, establece el indicador LogResponses; los valores pueden ser Always, Never o. OnError También puede registrar las métricas de rendimiento de las solicitudes HTTP mediante la propiedad LogMetrics, el formato de registro se puede especificar mediante la LogMetricsFormat propiedad, los valores válidos son JSON o estándar.

En el siguiente ejemplo se muestra la configuración más utilizada en el archivo awsconfig.xml. Para obtener más información sobre la configuración de servicios específicos, consulte la sección de servicios más adelante:

```
<?xml version="1.0" encoding="utf-8"?>
<aws region="us-west-2"
    <logging logTo="UnityLogger"
             logResponses="Always"
             logMetrics="true"
             logMetricsFormat="JSON" />
/>
```

Uso del archivo link.xml

El SDK utiliza la reflexión para los componentes específicos de plataforma. Si utiliza el backend de secuencias de comandos IL2 CPP, siempre `strip bytecode` está habilitado en iOS, por lo que debe tener un `link.xml` archivo en la raíz del ensamblaje con las siguientes entradas:

```
<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>
```

Paso 3: Obtener el ID de grupo de identidades mediante Amazon Cognito

Para utilizar los servicios de AWS en su aplicación móvil, debe obtener el ID de grupo de identidades mediante Amazon Cognito Identity. Si usa Amazon Cognito para obtener el ID de grupo de identidades, su aplicación podrá obtener acceso a los servicios de AWS sin tener que integrar sus credenciales privadas en la aplicación. También le permite establecer permisos para controlar a qué servicios de AWS tienen acceso sus usuarios.

Para comenzar a utilizar Amazon Cognito, debe crear un grupo de identidades. Un grupo de identidades es un almacén de datos de identidades de usuarios específicos de su cuenta. Cada grupo de identidades tiene roles de IAM configurables que le permiten especificar a qué servicios de AWS pueden obtener acceso los usuarios de su aplicación. Normalmente, un desarrollador usa un grupo de identidades por aplicación. Para obtener más información sobre los grupos de identidades, consulte la Guía para desarrolladores de Amazon Cognito.

Para crear un grupo de identidades nuevo para su aplicación:

1. Inicie sesión en la consola de Amazon Cognito y haga clic en Create new identity pool (Crear nuevo grupo de identidades).
2. Escriba un nombre para el grupo de identidades y marque la casilla de verificación para habilitar el acceso a las identidades sin autenticar. Haga clic en Create Pool (Crear grupo) para crear el grupo de identidades.
3. Haga clic en Allow (Permitir) para crear los dos roles predeterminados asociados al grupo de identidades: uno para los usuarios sin autenticar y otro para los usuarios autenticados. Estos roles predeterminados proporcionan al grupo de identidades acceso a Cognito Sync y Mobile Analytics.

En la página siguiente se muestra el código que crea un proveedor de credenciales para que pueda integrar fácilmente Cognito Identity en su aplicación de Unity. Pase el objeto de proveedor de credenciales al constructor del cliente de AWS que utilice. El código tiene este aspecto:

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

```
);
```

Siguientes pasos

- Introducción: lea Introducción al AWS Mobile SDK para Unity para obtener información más detallada de los servicios que incluye el SDK.
- Ejecute las demostraciones: vea nuestras aplicaciones Unity de muestra que demuestran los casos de uso comunes. Para ejecutar las aplicaciones de muestra, configure el SDK para Unity tal como se ha descrito anteriormente y siga las instrucciones incluidas en los archivos README de cada muestra.
- Lea la referencia de la API: consulte la Referencia de la API para el AWS Mobile SDK para Unity.
- Plantee preguntas: publique preguntas en los foros del AWS Mobile SDK o abrir una incidencia en Github.

Introducción al AWS Mobile SDK para Unity

Esta página le proporciona información general de cada servicio de AWS en el AWS Mobile SDK para Unity, así como instrucciones acerca de cómo configurar las muestras de Unity. Debe completar todas las instrucciones de la página Configurar el AWS Mobile SDK para Unity antes de comenzar a usar los servicios siguientes.

Amazon Cognito Identity

Todas llamadas efectuadas a AWS requieren credenciales de AWS. En vez de codificar de forma rígida las credenciales en sus aplicaciones, le recomendamos usar Amazon Cognito Identity para proporcionar las credenciales de AWS a su aplicación. Siga las instrucciones de Configurar el AWS Mobile SDK para Unity para obtener las credenciales de AWS a través de Amazon Cognito.

Cognito también le permite autenticar usuarios mediante proveedores de inicio de sesión públicos como Amazon, Facebook, Twitter y Google, así como proveedores que admiten OpenID Connect. Cognito también funciona con usuarios sin autenticar. Cognito proporciona credenciales temporales con derechos de acceso limitados que se especifican con un rol de Identity and Access Management (IAM). Cognito se configure mediante la creación de un nuevo grupo de identidades que está asociado a un rol de IAM. El rol de IAM especifica los recursos y servicios a los que puede obtener acceso la aplicación.

Para comenzar a utilizar Cognito Identity, consulte la Guía para desarrolladores de Amazon Cognito.

Amazon Cognito Sync

Cognito Sync le facilita la tarea de guardar los datos de los usuarios finales, como las preferencias de usuario o el estado del juego, en la nube de AWS a fin de que esté a disposición de los usuarios con independencia del dispositivo que utilicen. Cognito puede guardar estos datos localmente, lo que permite que sus aplicaciones funcionen aunque no esté disponible una conexión a Internet. Cuando esté disponible una conexión a Internet, las aplicaciones pueden sincronizar sus datos locales con la nube.

Para comenzar a utilizar Cognito Sync, consulte la Guía para desarrolladores de Amazon Cognito.

Uso de la CognitoSyncManager muestra

En el panel Proyecto, vaya a Assets AWSSDK//examples/y CognitoSync, en la parte derecha del panel, seleccione la CognitoSyncescena para abrirla.

Para ejecutar la muestra, haga clic en el botón de reproducción en la parte superior de la pantalla del editor. Cuando la aplicación se ejecuta, muestra cuadros de texto y botones que le permiten escribir información de los jugadores. Debajo hay una serie de botones que guardan localmente la información de los jugadores, sincronizan la información local de los jugadores con Cognito Cloud, actualizan la información de los jugadores a partir de Cognito Cloud y eliminan la información local de los jugadores. Pulse cada botón para realizar una operación. La muestra presenta comentarios en la parte superior de la pantalla de juego.

Para configurar el CognitoSyncManager ejemplo, debe especificar un ID de Cognito Identity Pool. Para especificar este valor, en el editor de Unity, selecciónelo SyncManageren el panel Jerarquía e introdúzcalo en el cuadro de texto IDENTITY_POOL_ID del Panel del Inspector.

> **Note**
>
> El CognitoSyncManager ejemplo contiene un código que ilustra cómo usar el proveedor de identidad de Facebook. Busca la macro «USE_FACEBOOK_LOGIN». Requiere el uso del SDK de Facebook para Unity. Para obtener más información, consulte SDK de Facebook para Unity.

Dynamo DB

Amazon DynamoDB es un servicio de base de datos no relacional rentable y rápido, de alta disponibilidad y de alta escalabilidad. DynamoDB elimina las limitaciones tradicionales de escalabilidad del almacenamiento de datos y, al mismo tiempo, mantiene una baja latencia y un desempeño previsible.

El SDK de AWS para Unity proporciona bibliotecas tanto de bajo nivel como de alto nivel para trabajar con DynamoDB. La biblioteca de alto nivel incluye DynamoDB Object Mapper, que permite mapear las clases del cliente a tablas de DynamoDB, efectuar varias operaciones de creación, lectura, actualización y eliminación (CRUD, por sus siglas en inglés), y ejecutar consultas. Con DynamoDB Object Mapper, puede escribir código sencillo y legible que almacena objetos en la nube.

Para obtener más información sobre DynamoDB, consulte la Guía para desarrolladores de DynamoDB.

Para obtener más información sobre el uso de Dynamo DB desde aplicaciones de Unity, consulte Amazon DynamoDB.

Uso de la muestra de DynamoDB

En el panel Proyecto, vaya a Assets//examples AWSSDK/DynamoDB. Esta muestra se compone de las siguientes escenas:

- DynamoDBExample : la escena inicial de la aplicación
- LowLevelDynamoDbExample - ejemplo de uso de la API DynamoDBD de bajo nivel
- TableQueryAndScanExample - ejemplo que muestra cómo realizar consultas
- HighLevelExample - ejemplo de uso de la API DynamoDB de alto nivel

Agregue estas escenas a la compilación (en el orden en que se muestran más arriba) con el cuadro de diálogo Build Settings (se abre seleccionando File.Build Settings). En este ejemplo se crean cuatro tablas: foro ProductCatalog, hilo y respuesta.

Para ejecutar la muestra, haga clic en el botón de reproducción en la parte superior de la pantalla del editor. Cuando la aplicación se ejecuta, muestra una serie de botones:

- Low Level Table Operations: ilustra cómo crear, enumerar, actualizar, describir y eliminar tablas.
- Mid Level Query & Scan Operations: ilustra cómo realizar consultas.
- High Level Object Mapper: ilustra cómo crear, actualizar y eliminar objetos.

Mobile Analytics

Amazon Mobile Analytics le permite realizar un seguimiento del comportamiento de los clientes, agrupar métricas, generar visualizaciones de datos e identificar patrones significativos. El SDK de AWS para Unity proporciona integración con el servicio Amazon Mobile Analytics. Para obtener información sobre Mobile Analytics, consulte Guía del usuario de Mobile Analytics. Para obtener más información sobre el uso de Mobile Analytics desde aplicaciones de Unity, consulte Amazon Mobile Analytics.

Configuración de Mobile Analytics

Mobile Analytics define algunos ajustes que se pueden configurar en el archivo awsconfig.xml:

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

- sessionTimeout: intervalo de tiempo después transcurrido el cual una aplicación cambia a segundo plano y la sesión se puede terminar.
- maxDBSize : este es el tamaño de la SQLIte base de datos. Cuando la base de datos alcanza el tamaño máximo, se eliminan los eventos adicionales.
- dbWarningThreshold - Este es el límite del tamaño de la base de datos que, una vez alcanzado, generará registros de advertencias.
- maxRequestSize - Es el tamaño máximo de la solicitud en bytes que debe transmitirse en una solicitud HTTP al servicio de análisis móvil.
- allowUseDataRed: booleano que especifica si los eventos de la sesión se envían a la red de datos.

Uso de la muestra de Mobile Analytics

En el panel Proyecto, vaya a Assets/AWSSDK/examples/Mobile Analytics y, en la parte derecha del panel, seleccione la escena de muestra de Amazon Mobile Analytics para abrirla. Para utilizar la muestra, debe agregar la aplicación mediante la consola de Amazon Mobile Analytics. Para obtener más información sobre la consola de Mobile Analytics, consulte la Guía del usuario de Amazon Mobile Analytics.

Siga estos pasos para configurar la muestra antes de ejecutarla:

1. Selecciona el objeto del AmazonMobileAnalyticsSample juego.
2. Especifique el ID de aplicación (creado en la consola de Amazon Mobile Analytics) en el campo "App Id" (ID de aplicación).
3. Especifique el ID de grupo de identidades de Cognito (creado en la consola de Amazon Cognito) en el campo "Cognito Identity Pool Id" (ID de grupo de identidades de Cognito).

4. Asegúrese de que sus roles autenticados y no autenticados tienen permisos para obtener acceso al servicio Mobile Analytics. Para obtener más información sobre la aplicación de la política a los roles de IAM, consulte Administración de roles.

Al ejecutar la aplicación de muestra, tenga en cuenta que es posible que los eventos no se transmitan inmediatamente al servicio backend. Un proceso en segundo plano guardará en búfer los eventos localmente y los enviará por lotes al backend de Amazon Mobile Analytics a intervalos regulares (el valor predeterminado es 60 segundos) para garantizar que el desempeño del juego no se ve afectado negativamente. Debido al procesamiento complejo que Amazon Mobile Analytics lleva a cabo en los datos, es posible que los eventos enviados y los informes correspondientes no estén visibles en la consola de AWS hasta que hayan transcurrido hasta 60 minutos después del envío inicial.

Para obtener más información sobre los informes que proporciona Amazon Mobile Analytics, consulte Métricas de informes y móviles.

Amazon S3

Amazon Simple Storage Service (Amazon S3) proporciona a los desarrolladores y equipos de TI un almacenamiento de objetos seguro, duradero y con un alto grado de escalabilidad. Desde Unity puede utilizar S3 para almacenar, enumerar y recuperar imágenes, vídeos, música y otros datos que utilizan sus juegos.

Para obtener más información sobre S3, consulte Amazon S3 e Introducción a S3.

Para obtener más información sobre el uso de S3 desde aplicaciones Unity, consulte Amazon Simple Storage Service (S3).

Configuración de la firma de S3 predeterminada

La firma de S3 predeterminada está configurada del siguiente modo:

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

Se utiliza para especificar si debe utilizar la versión 4 de la firma para las solicitudes de S3.

Uso de la muestra de S3

En el panel Proyecto, vaya a Assets/AWSSDK/examples/S3 y, en la parte derecha del panel, seleccione la escena S3Example para abrirla. La muestra ilustra cómo publicar buckets, enumerar objetos de un bucket, publicar objetos en un bucket y descargar los objetos de un bucket. Siga estos pasos para configurar la muestra antes de ejecutarla:

1. Seleccione el objeto de juego S3 en el panel Hierarchy (Jerarquía).
2. En el panel Inspector, introduzca los valores para S3 BucketName y SampleFileName. S3 BucketName es el nombre del depósito utilizado por la muestra y S3 SampleFileName es el nombre del archivo que la muestra cargará en el depósito de S3 especificado.
3. Asegúrese de que sus roles autenticados y no autenticados tienen permisos para obtener acceso a los buckets de S3 de su cuenta. Para obtener más información sobre la aplicación de la política a los roles de IAM, consulte Administración de roles.

Para ejecutar la muestra, haga clic en el botón de reproducción en la parte superior de la pantalla del editor. Cuando la aplicación se ejecuta, muestra una serie de botones:

- Get Objects: obtiene una lista de todos los objetos de todos los buckets de su cuenta de AWS.
- Get Buckets: obtiene una lista de todos los buckets de su cuenta de AWS.
- Post Object: carga un objeto en un bucket de S3 especificado.
- Delete Object: elimina todos los objetos de un bucket de S3 especificado.

La muestra presenta comentarios en la parte superior de la pantalla de juego.

Amazon Simple Notification Service

Amazon Simple Notification Service es un servicio de notificaciones push rápido, flexible y totalmente administrado que le permite enviar mensajes individuales o distribuir mensajes a un gran número de destinatarios. Amazon Simple Notification Service hace que enviar notificaciones de inserción a usuarios de dispositivos móviles o destinatarios de correo electrónico, o incluso enviar mensajes a otros servicios distribuidos, resulte sencillo y rentable. Para comenzar a utilizar Amazon Simple Notification Service, consulte Amazon Simple Notification Service.

AWS Lambda

AWS Lambda es un servicio informático que ejecuta el código en respuesta a solicitudes o eventos y administra de forma automática los recursos informáticos, lo que facilita la construcción de aplicaciones que respondan rápidamente a la nueva información. Las funciones de AWS Lambda se pueden llamar directamente desde dispositivos móviles, IoT y aplicaciones web. Envían una respuesta de forma sincrónica, facilitando así la creación de backends escalables, seguros y de alta disponibilidad para las aplicaciones móviles sin la necesidad de aprovisionar o administrar la infraestructura. Para obtener más información, consulte AWS Lambda.

Amazon Cognito Identity

¿Qué es Amazon Cognito Identity?

Con Amazon Cognito Identity puede crear identidades exclusivas para los usuarios y autenticarlos para un acceso seguro a los recursos de AWS, como Amazon S3 o Amazon DynamoDB. Amazon Cognito Identity es compatible con proveedores de identidades públicos (Amazon, Facebook, Twitter/Digits, Google o cualquier proveedor compatible con OpenID Connect) así como con identidades sin autenticar. Cognito también es compatible con las identidades autenticadas por el desarrollador, que le permiten registrar y autenticar usuarios mediante su propio proceso de autenticación backend, sin dejar de usar Amazon Cognito Sync para sincronizar los datos de los usuarios y obtener acceso a los recursos de AWS.

Para obtener más información sobre Cognito Identity, consulte la Guía para desarrolladores de Amazon Cognito.

Para obtener información acerca de la disponibilidad regional de Cognito Authentication, consulte la disponibilidad regional de Amazon Cognito Identity.

Uso de un proveedor público para autenticar a los usuarios

Para obtener información sobre el uso de proveedores de identidades públicos como Amazon, Facebook, Twitter/Digits o Google para autenticar usuarios, consulte Proveedores externos en la Guía para desarrolladores de Amazon Cognito.

Uso de identidades autenticadas por el desarrollador

Para obtener información sobre las identidades autenticadas, consulte Identidades autenticadas por el desarrollador en la Guía para desarrolladores de Amazon Cognito.

Amazon Cognito Sync

Cognito Sync es un servicio de AWS y una biblioteca de cliente que permite la sincronización entre dispositivos de los datos de usuarios relacionados con la aplicación. Puede utilizar la API de Cognito Sync para sincronizar los datos del usuario entre los distintos dispositivos. Para utilizar Cognito Sync en su aplicación, deberá incluir el SDK para móviles de AWS para Unity en su proyecto.

Para obtener más información sobre cómo integrar Amazon Cognito Sync en su aplicación, consulte la Amazon Cognito Sync Developer Guide.

Amazon Mobile Analytics

Amazon Mobile Analytics le permite realizar un seguimiento del comportamiento de los clientes, agrupar métricas, generar visualizaciones de datos e identificar patrones significativos. Para obtener información acerca de Mobile Analytics, consulte AWS Mobile Analytics.

Integración de Amazon Mobile Analytics

En las secciones siguientes se explica el procedimiento para integrar Mobile Analytics con una aplicación.

Creación de una aplicación en la consola de Mobile Analytics

Vaya a la consola de Amazon Mobile Analytics y cree una aplicación. Tome nota del valor de `appId`, ya que lo necesitará posteriormente.

> **Note**
>
> Para obtener más información acerca del trabajo en la consola, consulte Amazon Mobile Analytics User Guide.

Cuando cree una aplicación en la consola de Mobile Analytics, tendrá que especificar un ID de grupo de identidades de Cognito. Para crear un nuevo grupo de identidades de Cognito y generar un ID, consulte Cognito Identity Developer Guide.

Integración de Mobile Analytics en una aplicación

Para obtener acceso a Mobile Analytics desde Unity, debe utilizar las siguientes declaraciones:

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

La práctica recomendada consiste en usar Amazon Cognito para proporcionar credenciales temporales de AWS a su aplicación. Esas credenciales permiten a la aplicación obtener acceso a sus recursos de AWS. Para crear un proveedor de credenciales, siga las instrucciones que se describen en Amazon Cognito Identity.

Crea una MobileAnalyticsManager instancia con la siguiente información:

- cognitoIdentityPoolID: el ID del grupo de identidades de Cognito para tu aplicación
- CognitoRegion: la región de su grupo de identidades de Cognito, por ejemplo «. RegionEndpoint USEast1»
- región: la región del servicio Mobile Analytics, por ejemplo «RegionEndpoint. USEast1»
- appId: valor generado por la consola de Mobile Analytics al añadir una aplicación

MobileAnalyticsClientContextConfig Utilízala para inicializar una **`MobileAnalyticsManager`**instancia, tal y como se muestra en el siguiente fragmento de código:

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

> **Note**
>
> El ID de la aplicación se genera durante la ejecución del asistente de creación de aplicaciones. Ambos valores deben coincidir con los de la consola de Mobile Analytics.

El valor de `appId` se utiliza para agrupar los datos en la consola de Mobile Analytics. Para encontrar el ID de la aplicación después de crearla en la consola de Mobile Analytics, vaya a la consola de Mobile Analytics y haga clic en el icono del engranaje situado en la esquina superior derecha de la pantalla. Aparecerá la página de administración de aplicaciones, que muestra todas las aplicaciones registradas y sus aplicaciones. IDs

Registro de eventos de monetización

El SDK de Unity proporciona la clase `MonetizationEvent`, que permite generar eventos de monetización para realizar un seguimiento de las compras realizadas en las aplicaciones móviles. El siguiente fragmento de código muestra cómo crear un evento de monetización:

```
// 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);
```

Registro de eventos personalizados

Mobile Analytics le permite definir eventos personalizados. Los eventos personalizados están íntegramente definidos por usted y le ayudan a realizar un seguimiento de las acciones de los usuarios específicas de su aplicación o su juego. Para obtener más información acerca de los eventos personalizados, consulte Detalles de eventos personalizados. En este ejemplo, supongamos que su aplicación es un juego y que desea registrar un evento cuando un usuario complete un nivel. Cree un evento «LevelComplete» mediante la creación de una nueva `AmazonMobileAnalyticsEvent` instancia:

```
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);
```

Registro de sesiones

Cuando la aplicación deja de ser el elemento activo, puede hacer una pausa en la sesión. En `OnApplicationFocus`, compruebe si la aplicación está en pausa. De ser así, llame al evento `PauseSession`; de lo contrario, llame al evento `ResumeSession`, como se muestra en el siguiente fragmento de código:

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

De forma predeterminada, si el usuario cambia el foco a otro elemento distinto de la aplicación durante menos de 5 segundos y vuelve a la aplicación, la sesión se reanuda. Si el usuario cambia el foco a un elemento distinto de la aplicación durante 5 segundos o más, se crea una nueva sesión. Este ajuste se puede configurar en el archivo awsconfig.xml. Para obtener más información, consulte la sección Configuring Mobile Analytics de Introducción al AWS Mobile SDK para Unity.

Amazon Simple Storage Service (S3)

Amazon Simple Storage Service (Amazon S3) proporciona a los desarrolladores y equipos de TI un almacenamiento de objetos seguro, duradero y con un alto grado de escalabilidad. Los desarrolladores de Unity pueden utilizar S3 para cargar dinámicamente los recursos utilizados por sus juegos. Esto puede ayudar a que la descarga inicial de los juegos desde las tiendas de aplicaciones sea más rápida.

Para obtener más información acerca de S3, consulte Amazon S3.

Para obtener información acerca de la disponibilidad regional de AWS S3, consulte la disponibilidad regional de los servicios de AWS.

> **Note**
>
> Algunos de los ejemplos de este documento asumen el uso de una variable de cuadro de texto llamada ResultText para mostrar el resultado del rastreo.

Creación y configuración de un bucket de S3

Amazon S3 almacena los recursos en buckets de Amazon S3, es decir, en contenedores de almacenamiento en la nube que residen en una región concreta. Cada bucket de Amazon S3 debe tener un nombre único. Puede utilizar la consola de Amazon S3 para crear un bucket.

Creación de S3 Bucket

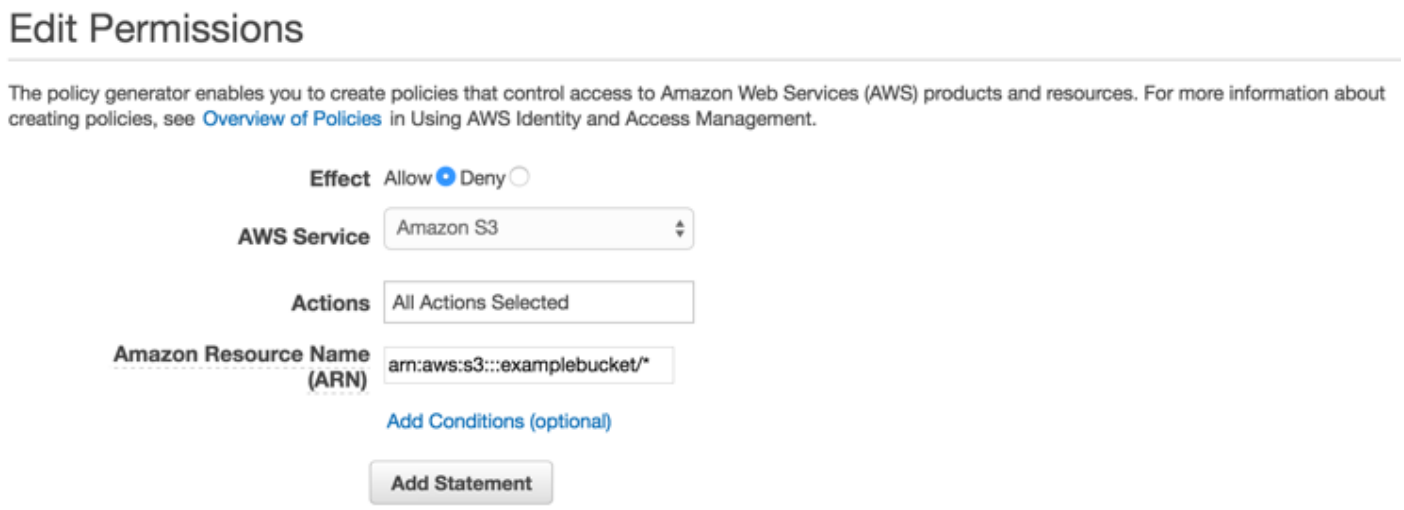
1. Inicie sesión en la consola de Amazon S3 y haga clic en Create Bucket (Crear bucket).
2. Escriba un nombre de bucket, seleccione una región y haga clic en Create (Crear).

Definición de permisos de S3

La política predeterminada de roles de IAM da a su aplicación acceso a Amazon Mobile Analytics y Amazon Cognito Sync. Para que el grupo de identidades de Cognito obtenga acceso a Amazon S3, debe modificar los roles del grupo de identidades.

1. Vaya a la consola de Identity and Access Management y haga clic en Roles (Roles) en el panel izquierdo.

2. Escriba el nombre del grupo de identidades en el campo de búsqueda. Se muestran dos roles: uno para los usuarios no autenticados y otro para los usuarios autenticados.
3. Haga clic en el rol de los usuarios no autenticados (el nombre del grupo de identidades tendrá añadido el texto "unauth").
4. Haga clic en Create Role Policy (Crear política de roles), seleccione Policy Generator (Generador de políticas) y haga clic en Select (Seleccionar).
5. En la página Edit Permissions (Editar permisos), escriba la configuración que se muestra en la siguiente imagen y reemplace el nombre de recurso de Amazon (ARN) por el suyo. El ARN de un bucket de S3 es parecido a `arn:aws:s3:::examplebucket/*` y está integrado por la región en la que se encuentra el bucket y por el nombre del bucket. La configuración que se muestra a continuación dará a su grupo de identidades pleno acceso a todas las acciones para el bucket especificado.



1. Haga clic en el botón Add Statement (Agregar instrucción) y, a continuación, haga clic en Next Step (Paso siguiente).
2. El asistente le mostrará la configuración que ha generado. Haga clic en Apply Policy (Aplicar política).

Para obtener más información acerca del procedimiento para conceder acceso a S3, consulte Granting Access to an Amazon S3 Bucket.

Carga de archivos desde la consola

Para cargar un archivo de prueba en un bucket:

1. En la consola de S3, en la vista del bucket, haga clic en Upload (Cargar).
2. Haga clic en Add Files (Añadir archivos) y seleccione un archivo de prueba para cargarlo. En este tutorial, supondremos que está cargando una imagen llamada `myImage.jpg`.
3. Con la imagen de prueba seleccionada, haga clic en Start Upload (Comenzar carga).

(opcional) Configuración de la versión de Signature para las solicitudes de S3

Toda interacción con Amazon S3 es o autenticada o anónima. AWS utiliza los algoritmos de Signature Version 4 o Signature Version 2 para autenticar las llamadas al servicio.

Las nuevas regiones de AWS creadas después de enero de 2014 admiten únicamente Signature Version 4. Sin embargo, muchas de las regiones más antiguas todavía admiten solicitudes de Signature Version 4 y Signature Version 2.

Si su depósito se encuentra en una de las regiones que no admiten las solicitudes de la versión 2 de Signature, tal como se indica en esta página, debe configurar el AWSConfigs S3. UseSignatureVersionLa propiedad 4 es «true».

Para obtener más información acerca de las versiones de AWS Signature, consulte Authenticating Requests (AWS Signature Version 4).

Creación del cliente de Amazon S3

Para usar Amazon S3, primero necesitamos crear una instancia de AmazonS3Client que haga referencia a la instancia de Cognito que creó anteriormenteAWSCredentials :

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

La clase `AmazonS3Client` es el punto de entrada a la API de S3 de alto nivel.

Lista de buckets

Para ver los buckets de una cuenta de AWS, llame al método `AmazonS3Client.ListBucketsAsync` como se indica en el siguiente código de muestra:

```
// 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";
    }
});
```

Lista de objetos

Para ver una lista de todos los objetos de un bucket, llame al método `AmazonS3Client.ListObjectsAsync` como se indica en el siguiente código de muestra:

```
// 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";
    }
});
```

Descarga de un objeto

Para descargar un objeto, cree un GetObjectRequest, especifique el nombre y la clave del bucket y pase el objeto a una llamada a Client. GetObjectAsync:

```
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;
       }
  });
}
```

GetObjectAsync toma una instancia de GetObjectRequest, una devolución de llamada y una AsyncOptions instancia. La devolución de llamada debe ser de tipo: `AmazonServiceCallback<GetObjectRequest, GetObjectResponse>`. La AsyncOptions instancia es opcional. Si se especifica, determina si la devolución de llamada se ejecutará en el hilo principal.

Carga de un objeto

Para cargar un objeto, escríbelo en una secuencia, crea uno nuevo PostObjectRequest y especifica la clave, el nombre del depósito y los datos de la transmisión.

AWS SDK para Unity usa el cliente HTTP WWW, que no admite la operación HTTP PUT. Para cargar un objeto en el bucket de S3, tendrá que utilizar Browser Post de S3, como se muestra a continuación.

```
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 es un servicio de base de datos no relacional rentable y rápido, de alta disponibilidad y de alta escalabilidad. DynamoDB elimina las limitaciones tradicionales de escalabilidad del almacenamiento de datos y, al mismo tiempo, mantiene una baja latencia y un desempeño previsible. Para obtener más información acerca de DynamoDB, consulte Amazon DynamoDB.

AWS Mobile SDK para Unity proporciona una biblioteca de alto nivel para trabajar con DynamoDB. También se pueden realizar solicitudes directamente en la API de DynamoDB de bajo nivel, pero para la mayoría de los casos de uso se recomienda trabajar con la biblioteca de alto nivel. AmazonDynamoDBClient Es una parte especialmente útil de la biblioteca de alto nivel. Con esta clase, puede realizar varias operaciones de creación, lectura, actualización y eliminación (CRUD, Create, Read, Update and Delete) y ejecutar consultas.

> **Note**
>
> Algunos de los ejemplos de este documento asumen el uso de una variable de cuadro de texto llamada ResultText para mostrar el resultado del rastreo.

Integración de Amazon DynamoDB

Para utilizar DynamoDB en una aplicación de Unity, tendrá que añadir el SDK de Unity al proyecto. Si aún no lo ha hecho, descargue el SDK de Unity y siga las instrucciones que se detallan en Configurar el AWS Mobile SDK para Unity. Le recomendamos que utilice Amazon Cognito Identity para proporcionar credenciales temporales de AWS para sus aplicaciones. Estas credenciales permiten a la aplicación obtener acceso a los servicios y recursos de AWS.

Para utilizar DynamoDB en una aplicación, debe definir los permisos correctos. La siguiente política de IAM permite al usuario eliminar, obtener, colocar, examinar y actualizar elementos en una tabla de DynamoDB, que se identifica mediante un 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"
}]
}
```

Esta política se debe aplicar a los roles asignados al grupo de identidades de Cognito, pero debe sustituir el valor de **`Resource`** por el ARN correcto para su tabla de DynamoDB. Cognito crea automáticamente un rol para el nuevo grupo de identidades y el usuario puede aplicar las políticas a ese rol en la consola de IAM.

Debe añadir o eliminar acciones permitidas en función de las necesidades de su aplicación. Para obtener más información acerca de las políticas de IAM, consulte cómo usar IAM. Para obtener más información acerca de las políticas específicas de DynamoDB, consulte cómo usar IAM para controlar el acceso a los recursos de DynamoDB.

Creación de una tabla de DynamoDB

Ahora que tenemos los permisos y las credenciales configurados, vamos a crear una tabla de DynamoDB para su aplicación. Para crear una tabla, vaya a la consola de DynamoDB y siga estos pasos:

1. Haga clic en Create Table (Crear tabla).
2. Escriba `Bookstore` como nombre de la tabla.
3. Seleccione Hash (Hash) como tipo de clave principal.
4. Seleccione Number (Número) y escriba `id` en el nombre del atributo hash. Haga clic en Continue (Continuar).
5. Haga clic de nuevo en Continue (Continuar) para omitir la adición de índices.
6. Establezca la capacidad de lectura en `10` y la capacidad de escritura en `5`. Haga clic en Continue (Continuar).
7. Escriba un correo electrónico de notificación y haga clic en Continue (Continuar) para crear alarmas de rendimiento.
8. Haga clic en Crear. DynamoDB creará la base de datos.

Creación de un cliente de DynamoDB

Para que la aplicación interactúe con una tabla de DynamoDB, necesitamos un cliente. Podemos crear un cliente predeterminado de DynamodDB de la siguiente forma:

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

La AmazonDynamo DBClient clase es el punto de entrada a la API de DynamoDB. La clase proporciona métodos de instancia para crear, describir, actualizar y eliminar tablas, entre otras operaciones. Context añade una nueva capa de abstracción en el cliente y permite utilizar funcionalidades adicionales como el modelo de persistencia de objetos.

Descripción de una tabla

Para obtener una descripción de la tabla de DynamoDB, podemos utilizar el código siguiente:

```
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);
```

```
    }
```

En este ejemplo, creamos un cliente y un DescribeTableRequest objeto, asignamos el nombre de nuestra tabla a la **`TableName`**propiedad y, a continuación, pasamos el objeto de solicitud al DescribeTableAsync método del objeto. AmazonDynamo DBClient DescribeTableAsync también utiliza un delegado al que se llamará cuando se complete la operación asíncrona.

> **Note**
>
> Todos los métodos asíncronos de la operación asíncrona AmazonDynamo DBClient toman delegados que se invocan cuando se completa la operación asíncrona.

Guardar un objeto

Para guardar un objeto en DynamoDB, utilice SaveAsync <T>el método del objeto, donde T es AmazonDynamo DBClient el tipo de objeto que va a guardar.

Hemos llamado a la base de datos "Bookstore" y en consonancia con ese tema vamos a implementar un modelo de datos que registra atributos relacionados con libros. Estas son las clases que definen nuestro modelo de datos.

```
[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; }
    }
```

Por supuesto, en una aplicación de librería real necesitaríamos campos adicionales para cosas como el autor y el precio. La clase Book está decorada con el atributo [DynamoDBTable], que define en qué tabla de base de datos se escribirán los objetos de tipo Book. La clave de cada instancia de la clase Book se identifica mediante el atributo [Dynamo DBHash Key]. Las propiedades se identifican

con el atributo [DynamoDBProperty], que especifica la columna de la tabla de la base de datos en la que se escribirá la propiedad. Con el modelo configurado, podemos escribir algunos métodos para crear, recuperar, actualizar y eliminar objetos Book.

Creación de un objeto Book

```
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";
    });
}
```

Recuperación de un objeto Book

```
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);
}
```

Actualización de un objeto Book

```
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");
            });
        }
   });
}
```

Eliminación de un objeto Book

```
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

Puede escribir aplicaciones de iOS y Android capaces de recibir notificaciones de inserción a móviles utilizando Amazon Simple Notification Service (SNS) y el SDK de Unity. Para obtener más información, consulte Amazon Simple Notification Service.

En este tema se explica cómo configurar la aplicación de ejemplo de AWS SDK for Unity, SNSExample .unity, para recibir notificaciones push móviles a través de Amazon SNS.

Puedes crear aplicaciones para iOS y Android con el ejemplo SNSExample .unity. Los pasos de configuración son diferentes para iOS y Android; lea la sección correspondiente a la plataforma a la que va destinada la aplicación.

Requisitos previos

A continuación indicamos los requisitos previos para usar esta solución.

Establecer permisos de SNS

Al crear un grupo de identidades de Cognito se generan dos roles de IAM:

- Cognito/_ <Identity-Pool-Name>Auth_DefaultRole : la función de IAM predeterminada para los usuarios autenticados
- Cognito/_ <Identity-Pool-Name>Unauth_DefaultRole : la función de IAM predeterminada para los usuarios no autenticados

Debe añadir a esas funciones permisos para acceder al servicio Amazon SNS. Para ello:

1. Busque la consola de IAM y seleccione el rol de IAM que va a configurar.
2. Haz clic en Adjuntar política, selecciona la política de Amazon SNSFull Access y haz clic en Adjuntar política.

> **Note**
>
> No se recomienda utilizar Amazon SNSFull Access en un entorno de producción; lo utilizamos aquí para que pueda empezar a trabajar rápidamente. Para obtener más

información acerca de la especificación de los permisos de un rol de IAM, consulte la información general sobre los permisos de los roles de IAM.

Requisitos previos de iOS

- Pertenecer al Apple iOS Developer Program
- Generar una identidad de firma
- Crear un perfil de aprovisionamiento configurado para notificaciones de inserción

Tendrá que ejecutar su aplicación en un dispositivo físico para recibir notificaciones de inserción. Para ejecutar la aplicación en un dispositivo debe estar suscrito a Apple iOS Developer Program. Una vez que se haya suscrito, podrá utilizar Xcode para generar una identidad de firma. Para obtener más información, consulte la documentación App Distribution Quick Start de Apple. Necesitará un perfil de aprovisionamiento configurado para notificaciones de inserción. Para obtener más información consulte la documentación de Apple Configuring Push Notifications.

Requisitos previos de Android

- Instalación de Android SDK
- Instalación del JDK
- android-support-v4.jar
- google-play-services.jar

Configurar la aplicación de ejemplo para iOS

Abra el editor Unity y cree un nuevo proyecto. Importe el paquete AWS SDK for Unity seleccionando Assets/Import Package/Custom Package y seleccionando aws-unity-sdk-sns -2.0.0.1.unitypackage. Asegúrese de que todos los elementos del cuadro de diálogo Importing Package (Importar paquete) están seleccionados y haga clic en Import (Importar).

Configuración de Unity

Realice los siguientes pasos para configurar el proyecto Unity:

1. En el panel de proyectos, vaya a Assets/AWSSDK/examples y abra la escena. SNSExample

2. En el panel Jerarquía, seleccione SNSExample.
3. En el panel Inspector(Inspector), especifique su identidad del grupo de identidades de Cognito.
4. Observe que existe un cuadro de texto para iOS Platform Application ARN (ARN de aplicación de plataforma iOS); generará esa información más adelante.
5. Seleccione File (Archivo)/Build Settings (Configuración de la compilación). En el cuadro de diálogo Build Settings (Configuración de la compilación), haga clic en el botón Add Current (Agregar actual) situado en la lista desplegable Scenes in Build (Escenas en compilación) para añadir la escena actual.
6. En Platform (Plataforma) seleccione iOS y haga clic en el botón Player Settings (Configuración de reproductor), en Inspector Pane (Panel de inspector) del editor Unity, haga clic en el icono de iPhone y desplácese hasta la sección Identification (Identificación) y especifique un Bundle Identifier (Identificador de paquete).

Configuración para iOS

Realice los siguientes pasos para configurar la muestra a fin de establecer ajustes específicos de iOS:

1. En un navegador web, vaya al centro para desarrolladores de Apple y haga clic en Certificates, Identifiers & Profiles (Certificados, identificadores y perfiles).
2. Haga clic en Identifiers (Identificadores) en iOS Apps (Aplicaciones iOS), haga clic en el botón más situado en la esquina superior derecha de la página web para añadir un nuevo ID de aplicación de iOS y escriba una descripción de ID de aplicación.
3. Desplácese hasta la sección Add ID Suffix (Agregar sufijo de ID), seleccione Explicit App ID (ID de aplicación explícito) y escriba el identificador de su paquete.
4. Desplácese hacia abajo hasta la sección App Services (Servicios de aplicación) y seleccione Push Notifications (Notificaciones de inserción).
5. Haga clic en el botón Continue (Continuar).
6. Haga clic en el botón Submit (Enviar).
7. Haga clic en el botón Done (Listo).
8. Seleccione el ID de aplicación que acaba de crear y, a continuación, haga clic en el botón Edit (Editar).
9. Desplácese hasta la sección Push Notifications (Notificaciones de inserción).

10. Haga clic en el botón Create Certificate (Crear certificado) bajo Development SSL Certificate (Certificado SSL de desarrollo).
11. Siga las instrucciones para crear una solicitud de firma del certificado (CSR), cargar la solicitud y descargar un certificado SSL que se utilizará para la comunicación con Apple Push Notification Service (APNS).
12. En la página Certificates, Identifiers & Profiles (Certificados, identificadores y perfiles), haga clic en All (Todo) en Provisioning Profiles (Perfiles de aprovisionamiento).
13. Haga clic en el botón más situado en la esquina superior derecha para añadir un nuevo perfil de aprovisionamiento.
14. Seleccione iOS App Development (Desarrollo de aplicación iOS) y haga clic en el botónContinue (Continuar).
15. Seleccione su ID de aplicación y haga clic en el botón Continue (Continuar).
16. Seleccione su certificado de desarrollador y haga clic en el botón Continue (Continuar).
17. Seleccione su dispositivo y haga clic en el botón Continue (Continuar).
18. Introduzca un nombre de perfil y haga clic en el botón Generate (Generar).
19. Para instalar el perfil de aprovisionamiento, descargue el archivo de aprovisionamiento y haga doble clic en él.

Es posible que necesite actualizar los perfiles de aprovisionamiento en Xcode después de añadir uno nuevo. En Xcode:

1. Seleccione el elemento de menú Xcode (Xcode)/Preferences (Preferencias).
2. Seleccione la pestaña Accounts (Cuentas), seleccione su ID de Apple y haga clic en View Details (Ver detalles).
3. Haga clic en el botón de actualización que aparece en la esquina inferior izquierda del cuadro de diálogo para actualizar los perfiles de aprovisionamiento y asegurarse de que se muestra el nuevo perfil.

Configuración de SNS

1. Ejecuta la aplicación de KeyChain acceso, selecciona Mis certificados en la parte inferior izquierda de la pantalla, haz clic con el botón derecho en el certificado SSL que generaste para conectarte a APNS y selecciona Exportar. Se te pedirá que especifiques un nombre para el archivo y una contraseña para proteger el certificado. El certificado se guardará en un archivo P12.

2. En un navegador web, vaya a SNS Console (Consola de SNS) y haga clic en Applications (Aplicaciones) en la parte izquierda de la pantalla.
3. Haga clic en Create platform application (Crear aplicación de plataforma) para crear una nueva aplicación de plataforma de SNS.
4. Introduzca un nombre de aplicación en Application Name (Nombre de aplicación).
5. Seleccione Apple Push Notification Service Sandbox (APNS_SANDBOX)(Entorno de pruebas del servicio de notificaciones de inserción de Apple, APNS_SANDBOX) en Push notification platform (Plataforma de notificaciones de inserción).
6. Haga clic en Choose File (Elegir archivo) y seleccione el archivo P12 que creó al exportar el certificado SSL.
7. Escriba la contraseña que especificó cuando exportó el certificado SSL y haga clic en Load Credentials From File (Cargar credenciales desde archivo).
8. Haga clic en Create platform application (Crear aplicación de plataforma).
9. Seleccione la aplicación de plataforma que acaba de crear y copie el ARN de la aplicación.
10. Regrese a su proyecto en el Editor de Unity, seleccione SNSExampleen el panel Jerarquía, en el panel Inspector y pegue el ARN de la aplicación de plataforma en el cuadro de texto denominado ARN de la aplicación de plataforma iOS.
11. Seleccione File (Archivo)/Build Settings (Configuración de la compilación) y haga clic en el botón Build (Compilar) para crear un proyecto Xcode.

Uso de Xcode

1. Abra el proyecto Xcode y selecciónelo en el explorador de proyectos.
2. Verifique que el identificador del paquete está establecido correctamente
3. Verifique que se ha especificado su cuenta de Apple Developer en Team (Equipo): esto es necesario para que se aplique su perfil de aprovisionamiento.
4. Cree el proyecto y ejecútelo en su dispositivo.
5. Pulse Register for Notification (Registrar para notificaciones) y pulse OK (Aceptar) para permitir las notificaciones. La aplicación mostrará el token de dispositivo.

En la consola de SNS, haga clic en Applications (Aplicaciones), seleccione la aplicación de plataforma y haga clic en Create Platform Endpoint (Crear punto de enlace de plataforma). A continuación, introduzca el token de dispositivo que muestra la aplicación.

En este momento, su aplicación, APNS y NSN están totalmente configurados. Puede seleccionar la aplicación de plataforma, seleccionar el punto de enlace y hacer clic en Publish to endpoint (Publicar en punto de enlace) para enviar una notificación de inserción a su dispositivo.

Ejemplo de Unity (iOS)

En el ejemplo, se crea una AWSCredentials instancia de Cognito para generar credenciales temporales de alcance limitado que permiten a la aplicación llamar a los servicios de AWS. También crea una instancia AmazonSimpleNotificationServiceClient para comunicarse con SNS. La aplicación muestra dos botones, denominados Register for Notification (Registrar para notificación= y Unregister (Cancelar registro).

Cuando se pulsa el botón Register for Notifications (Registrar para notificaciones), se llama al método `RegisterDevice()`. A su vez, el método `RegisterDevice()` llama a `UnityEngine.iOS.NotificationServices.RegisterForNotifications`, que especifica qué tipos de notificaciones (alertas, sonido o insignia) se van a utilizar. También hace una llamada a asíncrona a APNS para obtener un token de dispositivo. Dado que no hay ninguna devolución de llamada definida, se contacta con `CheckForDeviceToken` repetidamente (hasta 10 veces) para comprobar el token de dispositivo.

Cuando se recupera un token, se llama a `AmazonSimpleNotificationServiceClient.CreatePlatformEndpointAsync()` para crear un punto de enlace para la aplicación de plataforma de SNS.

El ejemplo ya está configurado para recibir notificaciones de inserción. Puede ir a la consola de SNS, hacer clic en Applications (Aplicaciones), seleccionar la aplicación de plataforma en la parte izquierda de la página, seleccionar un punto de enlace y hacer clic en Publish to Endpoint (Publicar en punto de enlace). Seleccione el punto de enlace que desea utilizar y haga clic en Publish to Endpoint (Publicar en punto de enlace). Escriba un mensaje en el cuadro de texto y haga clic en Publish message (Publicar mensaje) para publicarlo.

Configurar la aplicación de ejemplo Unity para Android

Abra el editor Unity y cree un nuevo proyecto. Importe el paquete AWS SDK for Unity seleccionando Assets/Import Package/Custom Package y seleccionando aws-unity-sdk-sns -2.0.0.1.unitypackage. Asegúrese de que todos los elementos del cuadro de diálogo Importing Package (Importar paquete) están seleccionados y haga clic en Import (Importar).

Configuración de Unity

Realice los siguientes pasos para configurar el proyecto Unity:

1. En el panel de proyectos, vaya a Assets/AWSSDK/examples y abra la escena. SNSExample
2. En el panel Jerarquía, seleccione SNSExample.
3. En el panel Inspector(Inspector), especifique su identidad del grupo de identidades de Cognito.
4. Observe que existe un cuadro de texto para Android Platform Application ARN (ARN de aplicación de plataforma Android) y Google Console Project ID (ID de proyecto de consola de Google); generará dicha información más adelante.
5. Seleccione File (Archivo)/Build Settings (Configuración de la compilación). En el cuadro de diálogo Build Settings (Configuración de la compilación), haga clic en el botón Add Current (Agregar actual) situado en la lista desplegable Scenes in Build (Escenas en compilación) para añadir la escena actual.
6. En Platform (Plataforma) seleccione Android y haga clic en el botón Player Settings (Configuración de reproductor) en el Inspector Pane (Panel de inspector) del editor de Unity, haga clic en el icono de Android, desplácese hasta la sección Identification (Identificación) y especifique un Bundle Identifier (Identificador de paquete).
7. Copie los archivos android-support-v 4.jar y google-play-services .jar en el directorio Assets/Plugins/Android del panel Proyecto.

Para obtener más información sobre dónde encontrar el android-support-v archivo 4.jar, consulta Configuración de la biblioteca de soporte de Android. Para obtener más información sobre cómo encontrar google-play-services .jar, consulta la configuración de Google APIs para Android.

Configuración para Android

En primer lugar, añada un nuevo proyecto de API de Google:

1. En un navegador web, vaya a la consola para desarrolladores de Google y haga clic en Create Project (Crear proyecto).
2. En el cuadro New Project (Nuevo proyecto), escriba el nombre de proyecto, tome nota del número de proyecto (lo necesitará más adelante) y haga clic en Create (Crear).

A continuación, habilite el servicio Google Cloud Messaging (GCM) para el proyecto:

1. El nuevo proyecto debe aparecer seleccionado en Google Developers Console. De lo contrario, selecciónelo en el menú desplegable en la parte superior de la página.
2. Selecciona APIs & auth en la barra lateral de la parte izquierda de la página.
3. En el cuadro de búsqueda, escriba "Google Cloud Messaging for Android" y haga clic en el enlace Google Cloud Messaging for Android.
4. Haga clic en Enable API (Habilitar API).

Por último, obtenga una clave de API:

1. En la consola de desarrolladores de Google, selecciona APIs & auth > Credenciales.
2. En Public API access (Acceso a API pública), haga clic en Create new key (Crear nueva clave).
3. En el cuadro de diálogo Create a new key (Crear nueva clave), haga clic en Server key (Clave de servidor).
4. En el cuadro de diálogo resultante, haga clic en Create (Crear) y copie la clave de API que se muestra.

Utilizará la clave de API para realizar la autenticación más adelante.

Configuración de SNS

1. En un navegador web, vaya a SNS Console (Consola de SNS) y haga clic en Applications (Aplicaciones) en la parte izquierda de la pantalla.
2. Haga clic en Create platform application (Crear aplicación de plataforma) para crear una nueva aplicación de plataforma de SNS.
3. Introduzca un nombre de aplicación en Application Name (Nombre de aplicación).
4. Seleccione Google Cloud Messaging (GCM) en Push notification platform (Plataforma de notificaciones de inserción)
5. Pegue la clave de API en el cuadro de texto API key (Clave de API).
6. Haga clic en Create platform application (Crear aplicación de plataforma).
7. Seleccione la aplicación de plataforma que acaba de crear y copie el ARN de la aplicación.
8. Vuelva a su proyecto en el Editor de Unity, selecciónelo SNSExampleen el panel Jerarquía, en el panel Inspector y pegue el ARN de la aplicación de plataforma en el cuadro de texto denominado ARN de la aplicación de la plataforma Android y el número de proyecto en el cuadro de texto denominado ID de proyecto de la consola de Google.

9. Conecte el dispositivo Android a su equipo, seleccione File (Archivo)/Build Settings (Configuración de la compilación) y haga clic en Build and Run (Compilar y ejecutar).

Ejemplo de Unity (Android)

En el ejemplo, se crea una AWSCredentials instancia de Cognito para generar credenciales temporales de alcance limitado que permiten a la aplicación llamar a los servicios de AWS. También crea una instancia AmazonSimpleNotificationServiceClient para comunicarse con SNS.

La aplicación muestra dos botones, denominados Register for Notification (Registrar para notificación= y Unregister (Cancelar registro). Al pulsar el botón Register for Notifications (Registrarse para notificaciones), se llama al método `RegisterDevice()`. A su vez, el método `RegisterDevice()` llama a `GCM.Register`, que registra la aplicación en GCM. GCM es una clase definida en el código de ejemplo. Hace una la llamada asíncrona para registrar la aplicación en GCM.

Cuando se llama a la devolución de llamada, se llama a `AmazonSimpleNotificationServiceClient.CreatePlatformEndpointAsync` para crear un extremo de plataforma que reciba los mensajes de SNS.

El ejemplo ya está configurado para recibir notificaciones de inserción. Puede ir a la consola de SNS, hacer clic en Applications (Aplicaciones), seleccionar la aplicación de plataforma en la parte izquierda de la página, seleccionar un punto de enlace y hacer clic en Publish to Endpoint (Publicar en punto de enlace). Seleccione el punto de enlace que desea utilizar y haga clic en Publish to Endpoint (Publicar en punto de enlace). Escriba un mensaje en el cuadro de texto y haga clic en Publish message (Publicar mensaje) para publicarlo.

AWS Lambda

AWS Lambda es un servicio informático que ejecuta el código en respuesta a solicitudes o eventos y administra de forma automática los recursos informáticos, lo que facilita la construcción de aplicaciones que respondan rápidamente a la nueva información. Las funciones de AWS Lambda se pueden llamar directamente desde dispositivos móviles, IoT y aplicaciones web. Envían una respuesta de forma sincrónica, facilitando así la creación de backends escalables, seguros y de alta disponibilidad para las aplicaciones móviles sin la necesidad de aprovisionar o administrar la infraestructura.

AWS Lambda puede ejecutar las funciones de Lambda en respuesta a una de las siguientes situaciones:

- Eventos, como actualizaciones discretas (por ejemplo, eventos creados por objetos en Amazon S3 o CloudWatch alertas) o actualizaciones en streaming (por ejemplo, secuencias de clics en sitios web o salidas de dispositivos conectados).
- Entradas JSON o comandos HTTPS desde las aplicaciones personalizadas.

AWS Lambda ejecuta el código solo cuando es necesario y amplía de manera automática, pasando de pocas solicitudes al día a miles por segundo. Con estas capacidades puede usar Lambda para crear disparadores para servicios de AWS como Amazon S3 y Amazon DynamoDB, procesar datos en streaming almacenados en Amazon Kinesis o crear su propio backend que opera con el nivel de seguridad, desempeño y escala de AWS.

Para obtener más información sobre cómo funciona AWS Lambda, consulte AWS Lambda: How It Works.

Permisos

Hay dos tipos de permisos relacionados con las funciones de Lambda:

- Permisos de ejecución: los permisos que necesita la función Lambda para acceder a otros recursos de AWS en su cuenta. Puede conceder estos permisos creando un rol de IAM, denominado rol de ejecución.
- Permisos de invocación: permisos que el origen de eventos necesita para comunicarse con la función de Lambda. Según el modelo de invocación (de inserción o extracción), puede conceder

estos permisos utilizando el rol de ejecución o las políticas de recursos (la política de acceso asociada con su función de Lambda).

Configuración del proyecto

Establecer permisos para AWS Lambda

1. Abra la consola de AWS IAM.
2. Vincule esta política personalizada a sus roles, para que su aplicación pueda realizar llamadas a AWS Lambda.

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

Crear un nuevo rol de ejecución

Este rol se aplica a la función de Lambda que creará en el siguiente paso y que determina a qué recursos de AWS puede acceder esa función.

1. Abra la consola de AWS IAM.
2. Haga clic en Roles (Roles).
3. Haga clic en Create New Roles (Crear nuevos roles).
4. Siga las instrucciones que aparecen en pantalla para seleccionar los servicios y las políticas correspondientes a las que necesita acceder la función de Lambda. Por ejemplo, si desea que la función de Lambda cree un bucket de S3, su política necesitará acceso de escritura a S3.
5. Haga clic en Create Role (Crear rol).

Crear una función en AWS Lambda

1. Abra la consola de AWS Lambda.
2. Haga clic en Create a Lambda function (Crear una función de Lambda).
3. Haga clic en Skip (Omitir) para omitir la creación de un proyecto.
4. Configure su propia función en la siguiente pantalla. Introduzca el nombre de la función, una descripción y elija el tiempo de ejecución. Siga las instrucciones que aparecen en pantalla (según el tiempo de ejecución que ha seleccionado). Especifique los permisos de ejecución asignando a su función un rol de ejecución recién creado.
5. Cuando haya terminado, haga clic en Next (Siguiente).
6. Haga clic en Create Function (Crear función).

Crear un cliente de Lambda

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

Crear un objeto de una solicitud

Cree un objeto de solicitud para especificar el tipo de invocación y el nombre de la función:

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

Invocar una función de Lambda

Llame a invoke y transfiera el objeto de la solicitud:

```
Client.InvokeAsync(request, (result) =>
{
    if (result.Exception == null)
    {
```

```
        Debug.Log(Encoding.ASCII.GetString(result.Response.Payload.ToArray()));
    }
    else
    {
        Debug.LogError(result.Exception);
    }
});
```

Solución de problemas

Debido a las limitaciones de la clase Unity.WWW utilizada por AWS SDK para Unity, no se devuelven mensajes de error detallados cuando se produce un problema al llamar a un servicio de AWS. En este tema se describen algunas ideas para solucionar esos problemas.

Comprobar que un rol de IAM tiene los permisos requeridos

Cuando llama a los servicios de AWS, su aplicación usa una identidad de un grupo de identidades de Cognito. Cada identidad del grupo está asociada a un rol de IAM (Identity and Access Management). El rol tiene uno o varios archivos de políticas asociados que especifican a qué recursos de AWS tienen acceso los usuarios asignados al rol. De forma predeterminada, se crean dos roles: uno para los usuarios autenticados y otro para los usuarios sin autenticar. Tendrá que modificar el archivo de política existente o asociar un nuevo archivo de política con los permisos requeridos por la aplicación. Si su aplicación admite usuarios autenticados y no autenticados, se deben conceder a los dos roles permisos para obtener acceso a los recursos de AWS que necesita la aplicación.

El archivo de política siguiente muestra cómo dar acceso a un bucket de S3:

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

El archivo de política siguiente muestra cómo dar acceso a una base de datos de DynamoDB:

```
{
"Statement": [{
```

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

Para obtener más información acerca de la especificación de políticas, consulte Políticas de IAM.

Uso de un depurador de proxy HTTP

Si el servicio de AWS al que llama la aplicación tiene un punto de enlace HTTP o HTTPS, puede utilizar un depurador de proxy HTTP/HTTPS para ver las solicitudes y las respuestas con el fin de obtener más información acerca de lo que está sucediendo. Hay diversos depuradores de proxy HTTP disponibles, como:

- Charles: un proxy de depuración web para OSX
- Fiddler: un proxy de depuración web para Windows

> **Important**
>
> Cuando se ejecuta el proxy de depuración web Charles, hay un problema conocido con el proveedor de credenciales de Cognito que impide que el proveedor de credenciales funcione correctamente.

Tanto Charles como Fiddler requieren algunos ajustes para poder ver el tráfico cifrado SSL. Lea la documentación de estas herramientas para obtener más información. Si utiliza un proxy de depuración web que no se puede configurar para mostrar tráfico cifrado, abra el archivo aws_endpoints_json (ubicado en AWSUnitySDK/AWSCore/Resources) y establezca la etiqueta HTTP del servicio de AWS que necesita depurar en true