Panduan Developer

AWS SDK Seluler untuk Unity

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS SDK Seluler untuk Unity: Panduan Developer

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan antara para pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

..... vi
Apa itu AWS SDK Seluler untuk Unity? 1
Panduan dan topik terkait 1
Isi referensi yang diarsipkan 1
Kompatibilitas 2
Mengunduh Mobile SDK for Unity 2
Apa saja yang disertakan dalam Mobile SDK for Unity? 2
Menyiapkan AWS Mobile SDK for Unity 3
Prasyarat 3
Langkah 1: Mengunduh AWS Mobile SDK for Unity 3
Langkah 2: Mengonfigurasi AWS Mobile SDK for Unity 4
Membuat Adegan 4
Mengatur Wilayah Layanan AWS Default 4
Mengatur Informasi Pencatatan 4
Cara menggunakan file link.xml 5
Langkah 3: Mendapatkan ID Kolam Identitas menggunakan Amazon Cognito 6
Langkah Berikutnya 7
Memulai AWS Mobile SDK for Unity 8
Identitas Amazon Cognito 8
Sinkronisasi Amazon Cognito 8
Menggunakan CognitoSyncManager sampel 8
Dynamo DB 9
Menggunakan Sampel DynamoDB 10
Mobile Analytics 10
Mengonfigurasi Mobile Analytics 10
Menggunakan Sampel Mobile Analytics 11
Amazon S3 12
Mengonfigurasi Tanda Tangan Default S3 12
Menggunakan Sampel S3 12
Amazon Simple Notification Service 13
AWS Lambda 13
Identitas Amazon Cognito 15
Apakah Identitas Amazon Cognito Itu? 15
Menggunakan Penyedia Publik untuk Melakukan Autentikasi Pengguna 15

Menggunakan Identitas Di-autentikasi Developer 15
Amazon Cognito Sync 16
Amazon Mobile Analytics 17
Mengintegrasikan Amazon Mobile Analytics 17
Membuat Aplikasi di Konsol Mobile Analytics 17
Mengintegrasikan Mobile Analytics ke Aplikasi Anda 17
Mencatat Peristiwa Monetisasi 18
Mencatat Peristiwa Kustom 19
Sesi Pecatatan 20
Amazon Simple Storage Service (S3) 21
Buat dan Lakukan Konfigurasi atas Bucket S3 21
Buat Bucket S3 21
Mengatur Izin untuk S3 21
Unggah File dari Konsol Tersebut 22
(opsional) Mengkonfigurasi Versi Tanda Tangan untuk Permintaan S3 23
Buat Klien Amazon S3 23
Buat Daftar Bucket 23
Daftar Objek 24
Mengunduh Objek 25
Meng-unggah Objek 25
Amazon DynamoDB 27
Mengintegrasikan Amazon DynamoDB 27
Membuat Tabel DynamoDB 28
Membuat Klien DynamoDB 28
Menjelaskan Tabel 29
Menyimpan Objek 30
Membuat Buku 30
Mengambil Buku 31
Memperbarui Buku 32
Menghapus Buku 32
Amazon Simple Notification Service 34
Prasyarat 3
Mengatur Izin untuk SNS 34
Prasyarat iOS 35
Prasyarat Android 35
Mengonfigurasi Aplikasi Sampel Unity untuk iOS 35

Konfigurasi Unity 35
Konfigurasi iOS 36
Konfigurasi SNS 37
Menggunakan Xcode 38
Sampel Unity (iOS) 38
Mengonfigurasi Aplikasi Sampel Unity untuk Android 39
Konfigurasi Unity 39
Konfigurasi Android 40
Konfigurasi SNS 40
Sampel Unity (Android) 41
AWS Lambda 42
Izin 42
Penyiapan Proyek 43
Mengatur Izin untuk AWS Lambda 43
Buat Peran Eksekusi Baru 43
Membuat Fungsi di AWS Lambda 44
Buat Klien Lambda 44
Buat Objek Permintaan 44
Melakukan Invokasi Fungsi Lambda 44
Pemecahan masalah 46
Pastikan IAM role Memiliki Izin yang Diperlukan 46
Menggunakan HTTP Proxy Debugger 47

SDK AWS Seluler untuk Unity sekarang disertakan dalam AWS SDK untuk .NET. Panduan ini menjadi referensi versi yang diarsipkan dari Mobile SDK for Unity. Untuk informasi selengkapnya, lihat Apa itu AWS SDK Seluler untuk Unity?

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.

Apa itu AWS SDK Seluler untuk Unity?

SDK AWS Seluler untuk Unity sekarang disertakan dalam SDK untuk .NET. Untuk informasi selengkapnya, lihat Panduan Developer AWS SDK untuk .NET .

Panduan ini tidak lagi diperbarui—ia menjadi referensi versi yang diarsipkan dari Mobile SDK for Unity.

Panduan dan topik terkait

- Untuk pengembangan aplikasi seluler dan front-end, kami merekomendasikan untuk menggunakan AWS Amplify.
- Untuk pertimbangan khusus untuk menggunakan aplikasi Unity Anda, lihat Pertimbangan khusus untuk dukungan Unity di Panduan AWS SDK untuk .NET Pengembang. AWS SDK untuk .NET
- Untuk tujuan referensi, Anda dapat menemukan versi arsip AWS Mobile SDK untuk Unity di GitHub.

Isi referensi yang diarsipkan

Mobile SDK untuk Unity yang diarsipkan berisi serangkaian kelas.NET yang memungkinkan game yang ditulis dengan Unity untuk memanfaatkan AWS layanan. Aplikasi yang ditulis dengan Mobile SDK for Unity dapat berjalan di perangkat iOS atau Android.

Layanan AWS yang didukung meliputi:

- Amazon Cognito
- Amazon DynamoDB
- AWS Identity and Access Management (IAM)
- Aliran Data Amazon Kinesis
- AWS Lambda
- Analisis Seluler Amazon
- Amazon Simple Email Service (Amazon SES)
- Layanan Pemberitahuan Sederhana Amazon (Amazon SNS)
- Amazon Simple Queue Service (Amazon SQS)

- Amazon Simple Storage Service (Amazon S3)

Layanan ini memungkinkan Anda untuk meng-autentikasi pengguna, menyimpan data pemain dan game, menyimpan objek di cloud, mengirim notifikasi push, dan mengumpulkan serta menganalisis data penggunaan.

Kompatibilitas

Mobile SDK for Unity v3 kompatibel dengan Unity versi 4.6 dan versi yang lebih tinggi.

Mobile SDK for Unity versi terbaru memperkenalkan perbaikan yang mungkin mengharuskan Anda untuk mengubah kode Anda ketika Anda memasukkan ke dalam proyek Anda. Untuk informasi selengkapnya tentang perubahan ini, lihat Perbaikan di SDK Sel AWS uler untuk Unity di blog Front-End Web & Seluler.

Mengunduh Mobile SDK for Unity

Anda juga dapat mengunduh Mobile SDK for Unity sebagai file .zip Di sini.

Apa saja yang disertakan dalam Mobile SDK for Unity?

Untuk daftar lengkap NuGet paket, sampel, dan file lainnya di Mobile SDK for Unity, lihat AWS SDK untuk .NET di GitHub.

Menyiapkan AWS Mobile SDK for Unity

Untuk memulai AWS Mobile SDK for Unity, Anda dapat menyiapkan SDK dan mulai membangun sebuah proyek baru, atau Anda dapat mengintegrasikan SDK dengan proyek yang ada. Anda juga dapat melakukan klon dan menjalankan Sampel untuk mengetahui cara kerja SDK.

Prasyarat

Sebelum Anda dapat menggunakan AWS Mobile SDK for Unity, Anda akan memerlukan yang berikut ini:

- Akun AWS
- Unity versi 4.x atau 5.x (Unity 4.6.4p4 atau Unity 5.0.1p3 diperlukan jika Anda ingin tulis aplikasi yang berjalan di iOS 64-bit)

Setelah menyelesaikan prasyarat tersebut, Anda harus melakukan hal-hal berikut untuk memulai:

1. Mengunduh AWS Mobile SDK for Unity.
2. Mengonfigurasi AWS Mobile SDK for Unity.
3. Mendapatkan kredensial AWS dengan menggunakan Amazon Cognito.

Langkah 1: Mengunduh AWS Mobile SDK for Unity

Pertama, Mengunduh AWS Mobile SDK for Unity. Setiap paket dalam SDK tersebut diperlukan untuk menggunakan layanan AWS yang sesuai berdasarkan nama paket. Misalnya, paket aws-unity-sdk-dynamodb -2.1.0.0.unitypackage digunakan untuk memanggil layanan AWS DynamoDB. Anda dapat mengimpor semua paket atau hanya paket yang ingin Anda gunakan.

1. Buka editor Unity dan buat proyek kosong baru, gunakan pengaturan default.
2. Pilih Aset > Paket Impor > Paket Kustom.
3. Di dialog paket Impor, arahkan ke dan pilih file .unitypackage yang ingin Anda gunakan.
4. Di dialog paket Mengimpor, pastikan semua item dipilih dan klik Impor.

Langkah 2: Mengonfigurasi AWS Mobile SDK for Unity

Membuat Adegan

Ketika bekerja dengan AWS Mobile SDK for Unity, Anda bisa memulai dengan memasukkan baris kode berikut di metode `Start` atau `Awake` kelas perilaku mono Anda:

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

Buat adegan Anda dengan memilih Adegan Baru dari halaman menu File.

AWS SDK for Unity berisi kelas klien untuk masing-masing layanan AWS yang di-support-nya. Klien ini dikonfigurasi menggunakan file bernama awsconfig.xml. Bagian berikut menjelaskan pengaturan yang paling umum digunakan di file awsconfig.xml. Untuk informasi lebih lanjut tentang pengaturan ini, lihat Referensi Unity SDK API.

Mengatur Wilayah Layanan AWS Default

Untuk mengonfigurasi wilayah default untuk semua klien layanan:

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

Ini akan menetapkan wilayah default untuk semua klien layanan di Unity SDK. Pengaturan ini dapat diganti dengan secara eksplisit menentukan wilayah pada saat membuat sebuah instans dari klien layanan, seperti berikut:

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

Mengatur Informasi Pencatatan

Pengaturan pencatatan ditentukan sebagai berikut:

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

Pengaturan ini digunakan untuk mengonfigurasi pencatatan di Unity. Saat Anda mencatat catatan ke `UnityLogger`, kerangka kerja secara internal mencetak keluaran ke Catatan Debug. Jika Anda ingin

mencatat respons HTTP, atur tanda logreSponses - nilainya bisa Selalu, Tidak Pernah, atau. OnError Anda juga dapat mencatat metrik kinerja untuk permintaan HTTP menggunakan properti LogMetrics, format log dapat ditentukan LogMetricsFormat menggunakan properti, nilai yang valid adalah JSON atau standar.

Contoh berikut menunjukkan pengaturan yang paling umum digunakan dalam file awsconfig.xml. Untuk informasi lebih lanjut tentang pengaturan layanan tertentu, lihat bagian layanan di bawah ini:

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

Cara menggunakan file link.xml

SDK tersebut menggunakan refleksi untuk komponen spesifik platform. Jika Anda menggunakan backend skrip IL2 CPP, selalu `strip bytecode` diaktifkan di iOS, jadi Anda harus memiliki `link.xml` file di root assembly Anda dengan entri berikut:

```
<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>
```

Langkah 3: Mendapatkan ID Kolam Identitas menggunakan Amazon Cognito

Untuk menggunakan layanan AWS dalam aplikasi seluler Anda, Anda harus mendapatkan ID Kolam Identitas menggunakan Identitas Amazon Cognito. Dengan menggunakan Amazon Cognito untuk mendapatkan ID Kolam Identitas akan memungkinkan aplikasi Anda untuk mengakses layanan AWS tanpa harus menanamkan kredensial privat Anda dalam aplikasi Anda. Hal ini juga akan memungkinkan Anda untuk mengatur izin untuk mengendalikan layanan AWS yang bisa diakses pengguna Anda.

Untuk memulai Amazon Cognito, Anda harus membuat kolam identitas. Kolam identitas adalah penyimpanan data identitas pengguna yang spesifik untuk akun Anda. Setiap kolam identitas memiliki IAM role yang dapat dikonfigurasi yang memungkinkan Anda untuk menentukan layanan AWS yang dapat diakses pengguna aplikasi Anda. Biasanya, developer akan menggunakan satu kolam identitas untuk setiap aplikasi. Untuk informasi lebih lanjut tentang kolam identitas, lihat Panduan Developer Amazon Cognito.

Untuk membuat kolam identitas untuk aplikasi Anda:

1. Masuk ke Konsol Amazon Cognito dan klik Membuat kolam identitas baru.
2. Masukkan nama untuk Kolam Identitas Anda dan beri tanda periksa pada kotak centang untuk mengaktifkan akses ke identitas yang tidak di-autentikasi. Klik Buat Kolam untuk membuat kolam identitas Anda.
3. Klik Izinkan untuk membuat dua peran default yang yang ter-associate dengan kolam identitas Anda—satu untuk pengguna yang tidak di-autentikasi dan satu untuk pengguna yang di-autentikasi. Peran default ini memberikan akses kolam identitas Anda ke Cognito Sync dan Mobile Analytics.

Halaman berikutnya menampilkan kode yang membuat penyedia kredensial sehingga Anda dapat dengan mudah mengintegrasikan Identitas Cognito dengan aplikasi Unity Anda. Anda memberikan objek penyedia kredensials untuk konstruktor dari klien AWS yang Anda gunakan. Kodenya terlihat seperti ini:

```
CognitoAWSCredentials credentials = new CognitoAWSCredentials (
```

```
    "IDENTITY_POOL_ID", // Identity Pool ID
    RegionEndpoint.USEast1 // Region
);
```

Langkah Berikutnya

- Memulai: Baca Memulai AWS Mobile SDK for Unity untuk mendapatkan gambaran umum yang lebih detail tentang layanan yang disertakan dalam SDK.
- Jalankan demo: Lihat Aplikasi Unity sampel kami yang menunjukkan kasus penggunaan umum. Untuk menjalankan aplikasi sampel, lakukan pengaturan atas SDK for Unity seperti yang dijelaskan di atas, dan kemudian ikuti petunjuk yang terdapat dalam file README dari setiap sampel.
- Baca Referensi API: Lihat Referensi API untuk AWS Mobile SDK for Unity.
- Ajukan pertanyaan: Posting pertanyaan di Forum AWS Mobile SDK atau buka masalah di Github.

Memulai AWS Mobile SDK for Unity

Halaman ini memberikan Anda gambaran umum dari setiap layanan AWS di AWS Mobile SDK for Unity, serta memberikan petunjuk tentang cara untuk mengatur sampel Unity. Anda harus menyelesaikan semua instruksi yang ada di halaman Menyiapkan AWS Mobile SDK for Unity sebelum Anda mulai menggunakan layanan di bawah ini.

Identitas Amazon Cognito

Semua panggilan yang dibuat untuk AWS memerlukan kredensial AWS. Daripada melakukan hard-coding atas kredensial Anda ke aplikasi, sebaiknya gunakan Identitas Amazon Cognito untuk memberikan kredensial AWS untuk aplikasi Anda. Ikuti instruksi yang ada di Menyiapkan AWS Mobile SDK for Unity untuk mendapatkan kredensial AWS melalui Amazon Cognito.

Cognito juga mengizinkan Anda untuk meng-autentikasi pengguna menggunakan penyedia log-in publik seperti Amazon, Facebook, Twitter, dan Google serta penyedia yang men-support OpenID Connect. Cognito juga berfungsi dengan pengguna yang tidak di-autentikasi. Cognito memberikan kredensial sementara dengan hak akses terbatas yang Anda tentukan dengan peran Identity and Access Management (IAM) role. Cognito dikonfigurasi dengan membuat Kolam Identitas baru yang di-associate dengan IAM role. Peran IAM menentukan aplikasi resources/services Anda dapat mengakses.

Untuk memulai Identitas Cognito, lihat Panduan Developer Amazon Cognito.

Sinkronisasi Amazon Cognito

Cognito Sync memudahkan Anda untuk menyimpan data pengguna akhir seperti preferensi pengguna atau status game ke AWS Cloud sehingga dapat disediakan untuk pengguna terlepas dari perangkat yang mereka gunakan. Cognito juga dapat menyimpan data ini secara lokal, mengizinkan aplikasi Anda bekerja meskipun koneksi internet tidak tersedia. Ketika koneksi internet tersedia, aplikasi Anda dapat menyinkronkan data lokal mereka ke cloud.

Untuk memulai Cognito Sync, lihat Panduan Developer Amazon Cognito.

Menggunakan CognitoSyncManager sampel

Di panel Project, navigasikan ke Assets AWSSDK//examples/CognitoSync, dan di sisi kanan panel pilih CognitoSyncscene untuk membuka scene.

Untuk menjalankan contoh klik tombol putar yang ada di bagian atas layar editor. Ketika aplikasi berjalan, aplikasi itu akan menampilkan beberapa kotak teks dan tombol yang memungkinkan Anda untuk memasukkan beberapa informasi pemain. Di bawah ini, ada serangkaian tombol yang menyimpan info pemain secara lokal, sinkronkan info pemain lokal dengan Cognito Cloud, kemudian refresh info pemain dari Cognito Cloud, dan hapus info pemain lokal. Tekan masing-masing tombol untuk melakukan operasi. Sampel tersebut menampilkan umpan balik yang ada di bagian atas layar game.

Untuk mengonfigurasi CognitoSyncManager sampel, Anda harus menentukan ID Kumpulan Identitas Cognito. Untuk menentukan nilai ini, di editor Unity, pilih di panel Heirarchy, dan masukkan ke SyncManagerdalam kotak teks IDENTITY_POOL_ID di Inspector Pane.

> **Note**
>
> CognitoSyncManager Sampel berisi kode yang menggambarkan bagaimana menggunakan Penyedia identitas Facebook, cari makro “USE_FACEBOOK_LOGIN”. Hal ini mengharuskan penggunaan Facebook SDK for Unity. Untuk informasi selengkapnya, lihat Facebook SDK for Unity.

Dynamo DB

Amazon DynamoDB layanan basis data yang cepat, sangat dapat diskalakan, sangat tersedia, hemat biaya, dan bukan basis data relasional. DynamoDB menghilangkan keterbatasan skalabilitas tradisional pada penyimpanan data sekaligus mempertahankan performa latensi rendah dan dapat diprediksi.

AWS SDK for Unity menyediakan perpustakaan tingkat rendah dan perpustakaan tingkat tinggi untuk bekerja dengan DynamoDB. Perpustakaan tingkat tinggi termasuk Pemeta Objek DynamoDB, yang memungkinkan Anda memetakan kelas sisi klien ke tabel DynamoDB; melakukan operasi buat, baca, perbarui, dan hapus (CRUD); dan menjalankan kueri. Dengan menggunakan Pemeta Objek DynamoDB, Anda dapat tulis kode sederhana dan mudah dibaca yang menyimpan objek di cloud.

Untuk informasi selengkapnya tentang DynamoDB, lihat Panduan Developer DynamoDB.

Untuk informasi selengkapnya tentang menggunakan Dynamo DB dari Aplikasi Unity, lihat Amazon DynamoDB.

Menggunakan Sampel DynamoDB

Di panel Project, navigasikan ke Assets//examples AWSSDK/DynamoDB. Sampel ini terdiri atas adegan-adegan berikut:

- Dynamo DBExample - adegan awal aplikasi
- LowLevelDynamoDbExample - contoh menggunakan API DynamoDBD tingkat rendah
- TableQueryAndScanExample - Contoh yang menunjukkan bagaimana melakukan query
- HighLevelExample - contoh menggunakan API DynamoDB tingkat tinggi

Menambahkan adegan ini ke dalam membangun (berdasarkan urutannya mereka muncul di atas) dengan menggunakan dialog Membangun Pengaturan (buka dengan memilih File.Build Pengaturan). Contoh ini membuat empat tabel: ProductCatalog, Forum, Thread, Reply.

Untuk menjalankan contoh klik tombol putar yang ada di bagian atas layar editor. Saat aplikasi menjalankannya, ia akan menampilkan sejumlah tombol:

- Operasi Tabel Tingkat Rendah - menggambarkan cara membuat, men-daftar, update, menjelaskan, dan menghapus tabel.
- Operasi Kueri Tingkat Menengah & Pindai - menggambarkan bagaimana melakukan kueri.
- Pemeta Objek Tingkat tinggi - menggambarkan cara membuat, memperbarui, dan menghapus objek.

Mobile Analytics

Dengan menggunakan Amazon Mobile Analytics, Anda dapat melacak perilaku pelanggan, melakukan penggabungan metrik, membuat visualisasi data, dan mengidentifikasi pola yang berarti. AWS SDK for Unity menyediakan integrasi dengan layanan Amazon Mobile Analytics. Untuk informasi tentang Mobile Analytics, lihat Panduan Pengguna Mobile Analytics. Untuk informasi selengkapnya tentang menggunakan Mobile Analytics dari Aplikasi Unity, lihat Amazon Mobile Analytics.

Mengonfigurasi Mobile Analytics

Mobile Analytics menentukan beberapa pengaturan yang dapat dikonfigurasi dalam file awsconfig.xml:

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

- SessionTimeout - ini adalah interval waktu setelah aplikasi pergi ke latar belakang dan ketika sesi dapat diakhiri.
- max DBSize - Ini adalah ukuran SQLIte Database. Ketika basis data mencapai ukuran maksimum, setiap peristiwa tambahan dihapus.
- dbWarningThreshold - Ini adalah batas ukuran database yang, setelah tercapai, akan menghasilkan log peringatan.
- maxRequestSize - Ini adalah ukuran maksimum permintaan dalam Bytes yang harus dikirimkan dalam permintaan HTTP ke layanan analitik seluler.
- allowUseDataJaringan - Sebuah boolean yang menentukan apakah peristiwa sesi dikirim pada jaringan data.

Menggunakan Sampel Mobile Analytics

Di panel Project, navigasikan ke Assets AWSSDK//examples/Mobile Analytics, dan di sisi kanan panel pilih adegan Sampel Amazon Mobile Analytics untuk membuka adegan. Untuk menggunakan sampel tersebut, Anda perlu menambahkan aplikasi Anda menggunakan Konsol Amazon Mobile Analytics. Untuk informasi selengkapnya tentang menggunakan konsol Mobile Analytics, lihat Panduan Pengguna Amazon Mobile Analytics.

Ikuti langkah-langkah berikut ini untuk mengonfigurasi sampel sebelum menjalankannya:

1. Pilih objek AmazonMobileAnalyticsSample permainan.
2. Tentukan Id Aplikasi Anda (dibuat di Konsol Amazon Mobile Analytics) dalam bidang “Id Aplikasi”.
3. Tentukan Id Kolam Identitas Cognito Anda (dibuat menggunakan Konsol Amazon Cognito di) di bidang “Id Kolam Identitas Cognito”.
4. Pastikan peran Anda yang di-autentikasi dan tidak di-autentikasi memiliki izin untuk mengakses layanan Mobile Analytics. Untuk informasi lebih lanjut tentang menerapkan kebijakan untuk IAM role, lihat Mengelola Peran.

Ketika menjalankan aplikasi sampel, perhatikan bahwa peristiwa mungkin tidak langsung ditransmisikan ke layanan backend. Utas latar belakang akan menyangga peristiwa secara lokal dan mengirimkannya dalam beberapa batch ke backend Amazon Mobile Analytics dengan interval reguler (nilai default-nya adalah 60 detik) untuk memastikan performa game Anda tidak terpengaruh secara negatif. Karena pemrosesan Amazon Mobile Analytics yang kompleks yang dilakukan pada data Anda, peristiwa yang dikirimkan dan laporan yang sesuai mungkin tidak terlihat di konsol AWS sampai 60 menit setelah penyerahan awal.

Untuk informasi selengkapnya tentang laporan yang disediakan oleh Amazon Mobile Analytics, lihat Metrik Laporan dan Mobile.

Amazon S3

Amazon Simple Storage Service (Amazon S3), menyediakan developer dan tim IT dengan penyimpanan objek yang aman, tahan lama, dan sangat dapat diskalakan. Dari Unity Anda dapat menggunakan S3 untuk menyimpan, mendaftar, dan mengambil gambar, video, musik, dan data lain yang digunakan oleh game Anda.

Untuk informasi selengkapnya tentang S3, lihat Amazon S3 dan Memulai S3.

Untuk informasi selengkapnya tentang menggunakan S3 dari aplikasi Unity, lihat Amazon Simple Storage Service (S3).

Mengonfigurasi Tanda Tangan Default S3

Tanda tangan S3 default dikonfigurasi sebagai berikut:

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

Hal ini digunakan untuk menentukan apakah Anda harus menggunakan tanda tangan versi 4 untuk permintaan S3.

Menggunakan Sampel S3

Di panel Project, navigasikan ke Assets AWSSDK//examples/S3, dan di sisi kanan panel pilih adegan S3Example untuk membuka adegan. Sampel ini menggambarkan bagaimana cara men-daftar bucket, men-daftar objek dalam bucket, mengirim objek ke bucket dan mengunduh objek dari bucket. Ikuti langkah-langkah berikut ini untuk mengonfigurasi sampel sebelum menjalankannya:

1. Pilih objek game S3 di panel Hirarki.
2. Di panel Inspector masukkan nilai untuk S3 dan. BucketName SampleFileName S3 BucketName adalah nama bucket yang digunakan oleh sampel dan S3 SampleFileName adalah nama file yang akan diunggah sampel ke dalam bucket S3 yang ditentukan.
3. Pastikan peran Anda yang telah di-autentikasi dan tidak di-autentikasi memiliki izin untuk mengakses bucket S3 di akun Anda. Untuk informasi lebih lanjut tentang menerapkan kebijakan untuk IAM role, lihat Mengelola Peran.

Untuk menjalankan contoh klik tombol putar yang ada di bagian atas layar editor. Saat aplikasi menjalankannya, ia akan menampilkan sejumlah tombol:

- Get Objects - Untuk mendapatkan daftar semua objek di semua bucket yang ada di akun AWS Anda.
- Get Bucket - Untuk mendapatkan daftar semua bucket yang ada di akun AWS Anda.
- Post Object - Meng-unggah objek ke bucket S3 yang ditentukan.
- Delete Object - Menghapus semua objek dari bucket S3 yang ditentukan.

Sampel tersebut menampilkan umpan balik yang ada di bagian atas layar game.

Amazon Simple Notification Service

Amazon Simple Notification Service adalah layanan notifikasi push yang cepat, fleksibel, dan terkelola penuh yang memungkinkan Anda mengirim pesan individu atau pesan fan-out ke sejumlah besar penerima. Amazon Simple Notification Service membuat pengiriman notifikasi push ke pengguna perangkat mobile, penerima email atau bahkan pengiriman pesan ke layanan terdistribusi lainnya menjadi sederhana dan hemat biaya. Untuk memulai dengan Amazon Simple Notification Service, lihat Amazon Simple Notification Service.

AWS Lambda

AWS Lambda adalah layanan komputasi yang menjalankan kode Anda sebagai respons atas permintaan atau peristiwa dan secara otomatis mengelola sumber daya komputasi untuk Anda, sehingga membangun aplikasi yang merespons informasi baru dengan cepat menjadi mudah. Fungsi AWS Lambda dapat dipanggil langsung dari aplikasi mobile, IoT, dan Web dan mengirimkan respon kembali secara serentak, sehingga mudah untuk membuat backend yang dapat diskalakan, aman,

dan sangat tersedia untuk aplikasi seluler Anda tanpa perlu penyediaan atau mengelola infrastruktur. Untuk informasi lebih lanjut, lihat AWS Lambda.

Identitas Amazon Cognito

Apakah Identitas Amazon Cognito Itu?

Dengan Menggunakan Identitas Amazon Cognito, Anda dapat membuat identitas unik untuk pengguna Anda dan meng-autentikasi-nya untuk akses aman ke sumber daya AWS Anda seperti Amazon S3 atau Amazon DynamoDB. Identitas Amazon Cognito men-support penyedia identitas publik—Amazon, Facebook, Twitter/Digits, Google, atau penyedia OpenID mana pun yang kompatibel dengan Connect—serta identitas yang tidak diautentikasi. Cognito juga men-support identitas ter-autentikasi developer, yang memungkinkan Anda mendaftar dan meng-autentikasi pengguna menggunakan proses autentikasi backend Anda sendiri, sekaligus masih tetap menggunakan Amazon Cognito Sync untuk menyinkronkan data pengguna dan mengakses sumber daya AWS.

Untuk informasi selengkapnya tentang Identitas Cognito, lihat Panduan Developer Amazon Cognito.

Untuk informasi tentang ketersediaan Wilayah Autentikasi Cognito, lihat Ketersediaan Wilayah Identitas Amazon Cognito.

Menggunakan Penyedia Publik untuk Melakukan Autentikasi Pengguna

Untuk informasi tentang cara menggunakan penyedia identitas publik seperti Amazon, Facebook, Twitter/Digits, atau Google untuk melakukan autentikasi pengguna, lihat Penyedia Eksternal dalam Panduan Developer Amazon Cognito.

Menggunakan Identitas Di-autentikasi Developer

Untuk informasi tentang identitas di-autentikasi developer, lihat Identitas Di-autentikasi Developer dalam Panduan Developer Amazon Cognito.

Amazon Cognito Sync

Cognito Sync adalah sebuah layanan AWS dan perpustakaan klien yang memungkinkan sinkronisasi lintas-perangkat data pengguna terkait aplikasi. Anda dapat menggunakan API Cognito Sync untuk menyinkronkan data pengguna di seluruh perangkat. Untuk menggunakan Cognito Sync di aplikasi Anda, Anda harus menyertakan AWS Mobile SDK for Unity dalam proyek Anda.

Untuk petunjuk tentang cara mengintegrasikan Amazon Coginto Sync di aplikasi Anda, lihat Panduan Developer Amazon Cognito Sync.

Amazon Mobile Analytics

Dengan menggunakan Amazon Mobile Analytics, Anda dapat melacak perilaku pelanggan, melakukan penggabungan metrik, membuat visualisasi data, dan mengidentifikasi pola yang berarti. Untuk informasi tentang Mobile Analytics lihat AWS Mobile Analytics.

Mengintegrasikan Amazon Mobile Analytics

Bagian di bawah ini menjelaskan cara mengintegrasikan Mobile Analytics dengan aplikasi Anda.

Membuat Aplikasi di Konsol Mobile Analytics

Buka Konsol Amazon Mobile Analytics dan buat aplikasi. Catat nilai `appId`, karena Anda akan membutuhkannya nanti.

> **Note**
>
> Untuk mem-pelajari selengkapnya tentang cara bekerja di konsol tersebut, lihat Panduan Pengguna Amazon Mobile Analytics.

Saat membuat Aplikasi di Konsol Mobile Analytics, Anda harus menentukan ID Kolam Identitas Cognito. Untuk membuat Kolam Identitas Cognito baru dan membuat ID, lihat Panduan Developer Cognito.

Mengintegrasikan Mobile Analytics ke Aplikasi Anda

Untuk mengakses Mobile Analytics dari Unity Anda akan memerlukan hal berikut ini menggunakan pernyataan:

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

Praktik terbaik adalah dengan menggunakan Amazon Cognito Amazon Cognito untuk memberikan kredensial AWS sementara untuk aplikasi Anda. Kredensial ini membiarkan aplikasi mengakses sumber daya AWS Anda. Untuk membuat penyedia kredensial, ikuti petunjuknya di Identitas Amazon Cognito.

Buat MobileAnalyticsManager instance dengan informasi berikut:

- cognitoIdentityPoolId - ID Cognito Identity Pool untuk aplikasi Anda
- CognitoRegion - Wilayah untuk Kumpulan Identitas Cognito Anda, misalnya “. RegionEndpoint USEast1”
- region - Wilayah untuk layanan Mobile Analytics, misalnya “RegionEndpoint. USEast1”
- appId - Nilai yang dihasilkan oleh Konsol Mobile Analytics saat Anda menambahkan aplikasi

Gunakan MobileAnalyticsClientContextConfig untuk menginisialisasi **`MobileAnalyticsManager`**instance seperti yang ditunjukkan dalam cuplikan kode berikut:

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

> **Note**
>
> ID aplikasi dibuat untuk Anda selama penuntun pembuatan aplikasi. Kedua nilai ini harus sesuai dengan nilai yang ada di Konsol Mobile Analytics.

`appId` digunakan untuk mengelompokkan data Anda dalam grup di konsol Mobile Analytics. Untuk menemukan ID aplikasi setelah membuat aplikasi di konsol Mobile Analytics, telusuri Konsol Mobile Analytics, klik ikon roda gigi yang ada di sudut kanan atas layar. Ini akan menampilkan halaman Manajemen Aplikasi yang mencantumkan semua aplikasi terdaftar dan aplikasinya IDs.

Mencatat Peristiwa Monetisasi

SDK for Unity menyediakan kelas `MonetizationEvent`, yang memungkinkan Anda membuat peristiwa monetisasi untuk melacak pembelian yang dilakukan dalam aplikasi mobile. Potongan kode berikut menunjukkan cara membuat peristiwa monetisasi:

```
// 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);
```

Mencatat Peristiwa Kustom

Mobile Analytics memungkinkan Anda untuk menentukan peristiwa kustom. Peristiwa kustom ditentukan sepenuhnya oleh Anda; mereka membantu Anda melacak tindakan pengguna tertentu untuk aplikasi atau game Anda. Untuk informasi selengkapnya tentang peristiwa kustom lihat Peristiwa-Kustom. Untuk contoh ini, katakanlah aplikasi Anda adalah game, dan Anda ingin mencatat peristiwa ketika pengguna menyelesaikan sebuah tingkat. Buat acara “LevelComplete” dengan membuat `AmazonMobileAnalyticsEvent` instance baru:

```
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);
```

Sesi Pecatatan

Ketika aplikasi kehilangan fokus Anda dapat menjeda sesi. Dalam `OnApplicationFocus`, periksa untuk melihat apakah aplikasi sedang dijeda. Jika ya, maka panggil `PauseSession`, jika tidak, maka panggil `ResumeSession`, seperti yang ditunjukkan dalam potongan kode berikut:

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

Secara default, jika pengguna mengalihkan fokus dari aplikasi selama kurang dari 5 detik, dan beralih kembali ke aplikasi, maka sesi akan dilanjutkan. Jika pengguna mengalihkan fokus dari aplikasi selama 5 detik atau lebih, maka sesi baru akan dibuat. Pengaturan ini dapat dikonfigurasi dalam file awsconfig.xml. Untuk informasi selengkapnya, lihat referensi bagian Mengonfigurasi Mobile Analytics di Memulai AWS Mobile SDK for Unity.

Amazon Simple Storage Service (S3)

Amazon Simple Storage Service (Amazon S3), menyediakan developer dan tim IT dengan penyimpanan objek yang aman, tahan lama, dan sangat dapat diskalakan. Developer Unity dapat memanfaatkan S3 untuk memuat aset yang digunakan oleh game mereka secara dinamis. Hal ini dapat membuat game awalnya men-download lebih cepat dari toko aplikasi.

Untuk informasi lebih lanjut tentang S3, lihat Amazon S3.

Untuk informasi tentang ketersediaan Wilayah AWS S3, lihat Ketersediaan Wilayah Layanan AWS.

> **Note**
>
> Beberapa sampel dalam dokumen ini mengasumsikan penggunaan variabel kotak teks yang dipanggil ResultText untuk menampilkan keluaran jejak.

Buat dan Lakukan Konfigurasi atas Bucket S3

Amazon S3 menyimpan sumber daya Anda di bucket S3 Amazon - kontainer penyimpanan cloud yang berlokasi di wilayah tertentu. Setiap bucket Amazon S3 harus memiliki nama yang unik secara global. Anda dapat menggunakan Konsol Amazon S3 untuk membuat bucket.

Buat Bucket S3

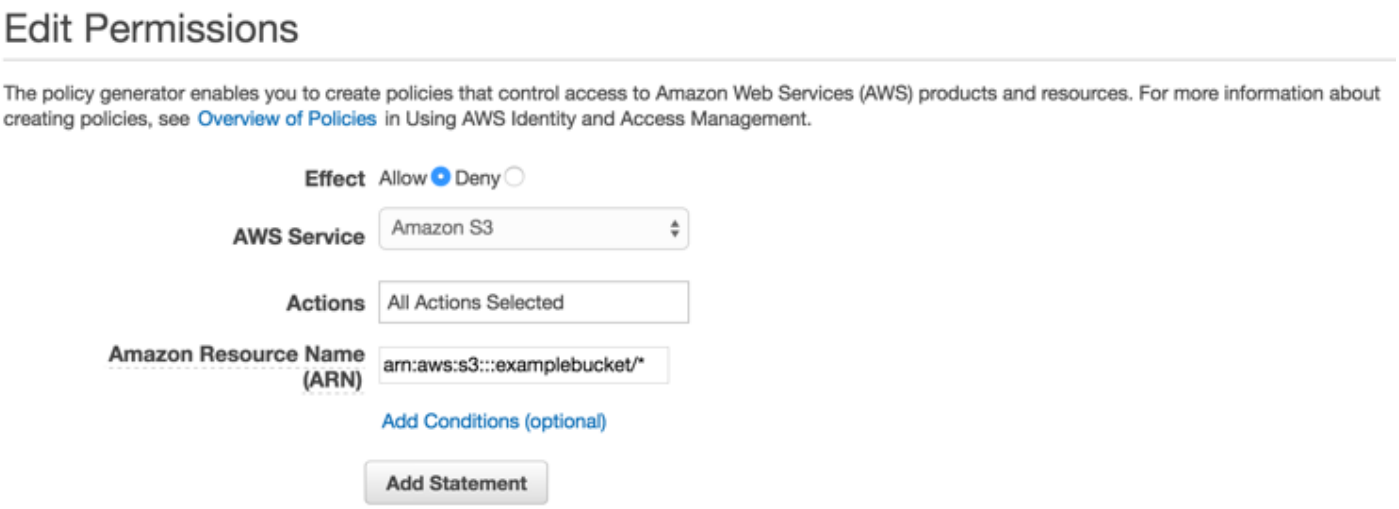
1. Masuklah ke Konsol Amazon S3 dan klik Buat Bucket.
2. Masukkan nama bucket, pilih wilayah, dan klik Buat.

Mengatur Izin untuk S3

Kebijakan IAM role default memberikan akses aplikasi Anda ke Amazon Mobile Analytics dan Amazon Cognito Sync. Agar kolam identitas Cognito Anda dapat mengakses Amazon S3, Anda harus mengubah peran kolam identitas.

1. Buka Konsol Identity and Access Management dan klik Peran yang di panel sebelah kiri.
2. Ketik nama kolam identitas Anda ke dalam kotak pencarian. Dua peran akan dicantumkan: satu untuk pengguna yang tidak di-autentikasi dan satu untuk pengguna di-autentikasi.

3. Klik peran untuk pengguna yang tidak di-autentikasi (hal ini akan menambahkan unauth pada nama kolam identitas Anda).
4. Klik Buat Kebijakan Peran, pilih Generator Kebijakan, dan kemudian klik Pilih.
5. Pada halaman Edit Izin, masukkan pengaturan yang ditunjukkan dalam gambar berikut, yang mengganti Amazon Resource Name (ARN) dengan punya Anda sendiri. ARN dari bucket S3 terlihat seperti `arn:aws:s3:::examplebucket/*` dan terdiri dari wilayah di mana bucket berada dan nama bucket. Pengaturan yang ditunjukkan di bawah ini akan memberikan kolam identitas Anda izin penuh untuk mengakses semua tindakan untuk bucket yang ditentukan.



1. Klik tombol Tambah Pernyataan dan kemudian klik Langkah Selanjutnya.
2. Penuntun akan menunjukkan konfigurasi yang telah Anda buat. Klik Terapkan Kebijakan.

Untuk informasi lebih lanjut tentang pemberian akses ke S3, lihat Memberikan Akses ke bucket Amazon S3.

Unggah File dari Konsol Tersebut

Untuk mengunggah file uji ke bucket Anda:

1. Di konsol S3, dalam tampilan bucket Anda, klik Unggah.
2. Klik Tambah File dan pilih file uji yang akan diunggah. Untuk tutorial ini, kami akan menganggap Anda mengunggah gambar bernama `myImage.jpg`.
3. Dengan gambar uji yang Anda dipilih, klik Mulai Unggah.

(opsional) Mengkonfigurasi Versi Tanda Tangan untuk Permintaan S3

Setiap interaksi dengan Amazon S3 diautentikasi atau dilakukan secara anonim. AWS menggunakan algoritme Tanda Tangan Versi 4 atau Tanda Tangan Versi 2 untuk meng-autentikasi panggilan ke layanan.

Semua wilayah AWS baru yang dibuat setelah Januari 2014 hanya men-support Tanda Tangan Versi 4. Namun, banyak wilayah lama yang masih men-support permintaan Tanda Tangan Versi 4 dan Tanda Tangan Versi 2.

Jika bucket Anda berada di salah satu wilayah yang tidak mendukung permintaan Signature Version 2 seperti yang tercantum di halaman ini, Anda harus menyetel AWSConfigs S3. UseSignatureVersion4 properti untuk “benar”.

Untuk informasi tentang versi Tanda Tangan AWS, lihat Meng-autentikasi Permintaan (Tanda Tangan AWS Versi 4).

Buat Klien Amazon S3

Untuk menggunakan Amazon S3, pertama-tama kita perlu membuat instance Amazons3Client yang mengambil referensi ke instance Cognito yang Anda buat sebelumnya: AWSCredentials

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

Kelas `AmazonS3Client` ini merupakan titik masuk ke API S3 tingkat tinggi.

Buat Daftar Bucket

Untuk membuat daftar bucket dalam akun AWS, panggil metode `AmazonS3Client.ListBucketsAsync` seperti yang ditunjukkan dalam kode sampel berikut:

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

Daftar Objek

Untuk membuat daftar semua objek dalam bucket, panggil metode `AmazonS3Client.ListObjectsAsync` seperti yang ditunjukkan dalam kode sampel berikut:

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

Mengunduh Objek

Untuk mengunduh objek, buat GetObjectRequest, tentukan nama dan kunci bucket dan teruskan objek tersebut ke panggilan ke Klien. 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 mengambil instance dari GetObjectRequest, callback, dan AsyncOptions instance. Callback harus tipe: `AmazonServiceCallback<GetObjectRequest, GetObjectResponse>`. AsyncOptions Contohnya adalah opsional. Jika ditentukan, maka ia akan menentukan apakah callback akan berjalan pada utama utama.

Meng-unggah Objek

Untuk mengunggah objek, tulis objek Anda ke aliran, buat yang baru, PostObjectRequest dan tentukan kunci, nama bucket, dan data streaming.

AWS SDK for Unity menggunakan klien HTTP WWW yang tidak men-support operasi HTTP PUT. Agar dapat meng-unggah objek ke bucket S3 Anda, Anda perlu menggunakan Post Peramban S3, seperti yang ditunjukkan di bawah ini.

```
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 layanan basis data yang cepat, sangat dapat diskalakan, sangat tersedia, hemat biaya, dan bukan basis data relasional. DynamoDB menghilangkan keterbatasan skalabilitas tradisional pada penyimpanan data sekaligus mempertahankan performa latensi rendah dan dapat diprediksi. Untuk informasi selengkapnya tentang DynamoDB, lihat Amazon DynamoDB.

AWS Mobile SDK for Unity menyediakan perpustakaan tingkat tinggi untuk bekerja dengan DynamoDB. Anda juga dapat membuat permintaan langsung terhadap API DynamoDB tingkat rendah, tetapi untuk sebagian besar kasus penggunaan, perpustakaan tingkat tinggi lebih direkomendasikan. AmazonDynamoDBClient Ini adalah bagian yang sangat berguna dari perpustakaan tingkat tinggi. Dengan menggunakan kelas ini, Anda dapat melakukan berbagai operasi buat, baca, perbarui, dan hapus (CRUD) dan menjalankan kueri.

> **Note**
>
> Beberapa sampel dalam dokumen ini mengasumsikan penggunaan variabel kotak teks yang dipanggil ResultText untuk menampilkan keluaran jejak.

Mengintegrasikan Amazon DynamoDB

Untuk menggunakan DynamoDB dalam aplikasi Unity, Anda harus menambahkan Unity SDK ke dalam proyek Anda. Jika Anda belum menambahkannya, unduh SDK for Unity dan ikuti petunjuk di Mengatur AWS Mobile SDK for Unity. Kami merekomendasikan untuk menggunakan Identitas Amazon Cognito untuk memberikan kredensial AWS sementara untuk aplikasi Anda. Kredensial ini memungkinkan aplikasi Anda untuk mengakses layanan AWS dan sumber daya.

Untuk menggunakan DynamoDB dalam aplikasi, Anda harus mengatur izin yang benar. Kebijakan IAM berikut memungkinkan pengguna untuk menghapus, mendapatkan, menempatkan, memindai, dan memperbarui item dalam tabel DynamoDB tertentu, yang diidentifikasi oleh 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"
}]
}
```

Kebijakan ini harus diterapkan pada peran yang ditetapkan ke kolam identitas Cognito, namun Anda harus mengganti nilai **`Resource`** dengan ARN yang benar untuk tabel DynamoDB Anda. Cognito secara otomatis membuat peran untuk kolam identitas baru Anda, dan Anda dapat menerapkan kebijakan tersebut untuk peran ini di konsol IAM.

Anda harus menambahkan atau menghapus tindakan yang diizinkan berdasarkan kebutuhan aplikasi Anda. Untuk mem-pelajari selengkapnya tentang kebijakan IAM, lihat Menggunakan IAM. Untuk mem-pelajari selengkapnya tentang kebijakan khusus DynamoDB, lihat Menggunakan IAM untuk Mengendalikan Akses ke Sumber Daya DynamoDB.

Membuat Tabel DynamoDB

Sekarang karena kita telah mengatur izin dan kredensial, mari kita membuat tabel DynamoDB untuk aplikasi kita. Untuk membuat tabel, buka Konsol DynamoDB dan ikuti langkah-langkah ini:

1. Klik Buat Tabel.
2. Masukkan `Bookstore` sebagai nama tabel baru.
3. Pilih Hash sebagai tipe kunci primer.
4. Pilih Nomor dan masukkan `id` untuk nama atribut hash. Klik Lanjutkan.
5. Klik Lanjutkan lagi untuk melewatkan penambahan indeks.
6. Atur kapasitas baca ke `10` dan kapasitas menulis ke `5`. Klik Lanjutkan.
7. Masukkan email notifikasi dan klik Lanjutkan untuk membuat alarm throughput.
8. Klik Buat. DynamoDB akan membuat basis data Anda.

Membuat Klien DynamoDB

Agar aplikasi kita dapat berinteraksi dengan tabel DynamoDB, kita membutuhkan klien. Kita dapat membuat klien DynamodDB default sebagai berikut:

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

```
AmazonDynamoDBClient client = new AmazonDynamoDBClient(credentials);
DynamoDBContext Context = new DynamoDBContext(client);
```

AmazonDynamoDBClient Kelas adalah titik masuk untuk DynamoDB API. Kelas tersebut menyediakan metode instans untuk menciptakan, menjelaskan, memperbarui, dan menghapus tabel, antara operasi lainnya. Konteks menambahkan lapisan lebih lanjut dari abstraksi atas klien dan memungkinkan Anda untuk menggunakan fungsi tambahan seperti Model Persistensi Objek.

Menjelaskan Tabel

Untuk mendapatkan deskripsi tabel DynamoDB kita, kita dapat menggunakan kode berikut:

```
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);
   }
```

Dalam contoh ini, kita membuat klien dan DescribeTableRequest objek, menetapkan nama tabel kita ke **`TableName`**properti, dan kemudian meneruskan objek permintaan ke DescribeTableAsync metode pada AmazonDynamo DBClient objek. DescribeTableAsync juga mengambil delegasi yang akan dipanggil ketika operasi async selesai.

> **Note**
>
> Semua metode async pada delegasi AmazonDynamo DBClient take yang dipanggil saat operasi async selesai.

Menyimpan Objek

Untuk menyimpan objek ke DynamoDB, gunakan SaveAsync <T>metode objek, di mana T adalah jenis objek yang Anda simpan. AmazonDynamo DBClient

Kami memanggil basis data kami “Toko Buku”, dan sesuai dengan tema itu kami akan menerapkan model data yang mencatat atribut terkait buku. Berikut adalah kelas-kelas yang mendefinisikan model data kami.

```
[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; }
    }
```

Tentu saja, untuk aplikasi toko buku nyata kita akan membutuhkan bidang tambahan untuk hal-hal seperti penulis dan harga. Kelas Buku dihiasi dengan atribut [DynamoDBTable], ini mendefinisikan objek tabel database tipe Buku akan ditulis. Kunci untuk setiap instance kelas Buku diidentifikasi menggunakan atribut [Dynamo DBHash Key]. Properti diidentifikasi dengan atribut [DynamoDBProperty], ini menentukan kolom dalam tabel database tempat properti akan ditulis. Dengan model yang ada, kita dapat tulis beberapa metode untuk membuat, mengambil, memperbarui, dan menghapus objek Buku.

Membuat Buku

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

Mengambil Buku

```
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);
}
```

Memperbarui Buku

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

Menghapus Buku

```
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

Dengan menggunakan Amazon Simple Notification Service (SNS) dan Unity SDK, Anda dapat tulis aplikasi iOS dan Android yang dapat menerima notifikasi push seluler. Untuk informasi tentang SNS, lihat Amazon Simple Notification Service.

Topik ini akan memandu Anda dalam mengonfigurasi aplikasi sampel AWS SDK for Unity SNSExample, .unity, untuk menerima notifikasi push seluler melalui Amazon SNS.

Anda dapat membuat aplikasi iOS dan Android menggunakan SNSExample sampel.unity. Langkah-langkah konfigurasi berbeda antara iOS dan Android, silakan baca bagian yang sesuai di bawah ini untuk platform yang Anda targetkan.

Prasyarat

Prasyarat berikut diperlukan untuk menggunakan solusi ini.

Mengatur Izin untuk SNS

Ketika Anda membuat Kolam Identitas Cognito, dihasilkan dua IAM role:

- Cognito/_ <Identity-Pool-Name>Auth_ DefaultRole - peran IAM default untuk pengguna yang diautentikasi
- Cognito/_ <Identity-Pool-Name>Unauth_ DefaultRole - peran IAM default untuk pengguna yang tidak diautentikasi

Anda harus menambahkan izin untuk mengakses layanan Amazon SNS untuk peran-peran tersebut. Untuk melakukannya:

1. Buka Konsol IAM dan pilih IAM role yang akan dikonfigurasi.
2. Klik Lampirkan Kebijakan, pilih kebijakan SNSFull Akses Amazon dan klik Lampirkan Kebijakan.

> **Note**
>
> Menggunakan Amazon SNSFull Access tidak disarankan di lingkungan produksi, kami menggunakannya di sini untuk memungkinkan Anda bangun dan berjalan dengan cepat.

Untuk informasi selengkapnya tentang bagaimana menentukan izin untuk IAM role, lihat Gambaran Umum Izin IAM Role.

Prasyarat iOS

- Keanggotaan dalam Program Developer iOS Apple
- Buat identitas penandatanganan
- Membuat profil penyediaan yang dikonfigurasi untuk notifikasi push

Anda akan perlu untuk menjalankan aplikasi Anda pada perangkat fisik untuk menerima notifikasi push. Untuk menjalankan aplikasi Anda di perangkat, Anda harus memiliki keanggotaan di Keanggotaan Program Developer iOS Apple. Setelah Anda memiliki keanggotaan, Anda dapat menggunakan Xcode untuk membuat identitas penandatanganan. Untuk informasi selengkapnya, lihat dokumentasi Quick Start Distribusi Aplikasi. Selanjutnya Anda memerlukan profil penyediaan yang dikonfigurasi untuk notifikasi push, untuk informasi lebih lanjut, lihat dokumentasi Mengonfigurasi Notifikasi Push.

Prasyarat Android

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

Mengonfigurasi Aplikasi Sampel Unity untuk iOS

Buka editor Unity dan buat proyek baru. Impor paket AWS SDK for Unity dengan memilih Assets/Import Package/Custom Package dan aws-unity-sdk-sns memilih -2.0.0.1.unitypackage. Pastikan semua item dalam dialog Mengimpor Paket dipilih dan klik Mengimpor.

Konfigurasi Unity

Lakukan langkah-langkah berikut untuk mengonfigurasi proyek Unity:

1. Di panel Project, navigasikan ke Assets AWSSDK//examples dan buka SNSExample scene.

2. Di panel Hierarki, pilih. SNSExample
3. Di panel Inspector, tentukan ID Kolam Identitas Cognito Anda.
4. Perhatikan ada kotak teks untuk ARN Aplikasi Platform iOS, Anda akan menghasilkan informasi itu nanti.
5. Pilih File/Membangun Pengaturan, di dialog Membangun Pengaturan, klik tombol Tambah Saat Ini di bawah kotak daftar Adegan dalam Membangun untuk menambahkan adegan saat ini untuk membangun.
6. Di bawah Platform pilih iOS dan klik tombol Pengaturan pemutar..., di Panel Inspector dari editor Unity, klik ikon iPhone dan gulir ke bawah hingga ke bagian Identifikasi dan tentukan Pengidentifikasi Paket.

Konfigurasi iOS

Lakukan langkah-langkah berikut untuk mengonfigurasi sampel untuk mengonfigurasi pengaturan khusus iOS:

1. Di peramban web, buka situs web Pusat Anggota Developer Apple, klik Sertifikat, Pengidentifikasi & Profil.
2. Klik Pengidentifikasi di bawah Aplikasi iOS, lalu klik tombol plus yang ada di sudut kanan atas halaman web untuk menambahkan ID Aplikasi iOS baru, lalu masukkan deskripsi ID Aplikasi.
3. Gulir ke bawah hingga ke bagian Tambah Akhiran ID dan pilih ID Aplikasi Eksplisit dan masukkan pengidentifikasi paket Anda.
4. Gulir ke bawah hingga ke bagian Layanan Aplikasi dan pilih bagian Notifikasi push.
5. Klik tombol Lanjutkan.
6. Klik tombol Kirim.
7. Klik tombol Selesai.
8. Pilih ID Aplikasi yang baru saja Anda buat dan kemudian klik tombol Edit.
9. Gulir ke bawah hingga ke bagian Notifikasi Push.
10. Klik tombol Buat Sertifikat di bawah Sertifikat SSL Pengembangan.
11. Ikuti petunjuk untuk membuat Permintaan Penandatanganan Sertifikat (CSR), unggah permintaan, dan unduh sertifikat SSL yang akan digunakan untuk berkomunikasi dengan Layanan Notifikasi Apple (APNS).
12. Kembali ke halaman web Sertifikat, Pengidentifikasi & Profil, klik Semua di bawah Profil Penyediaan.

13. Klik tombol plus yang ada di sudut kanan atas untuk menambahkan profil penyediaan baru.
14. Pilih Pengembangan Aplikasi iOS dan klik tombol Lanjutkan.
15. Pilih ID Aplikasi Anda dan klik tombol Lanjutkan.
16. Pilih sertifikat developer Anda dan klik tombol Lanjutkan.
17. Pilih perangkat Anda dan klik tombol Lanjutkan.
18. Masukkan nama profil dan klik tombol Buat.
19. Unduh dan klik dua kali file penyediaan untuk memasang profil penyediaan.

Anda mungkin perlu me-refresh Profil Penyediaan di Xcode setelah menambahkan yang baru. Dalam Xcode:

1. Pilih item menu Xcode/Preferensi.
2. Pilih tab Akun, pilih ID Apple dan klik Tampilkan Detail.
3. Klik tombol refresh yang ada di sudut kiri bawah dialog untuk menyegarkan profil penyediaan dan pastikan profil baru Anda yang ditampilkan.

Konfigurasi SNS

1. Jalankan aplikasi KeyChain akses, pilih Sertifikat Saya di sisi kiri bawah layar, klik kanan sertifikat SSL yang Anda buat untuk terhubung ke APNS dan pilih Ekspor, Anda akan diminta untuk menentukan nama untuk file dan kata sandi untuk melindungi sertifikat. Sertifikat tersebut akan disimpan dalam file P12.
2. Dalam peramban web, buka Konsol SNS dan klik Aplikasi yang ada di sisi kiri layar.
3. Klik Buat aplikasi platform untuk membuat aplikasi platform SNS baru.
4. Masukkan Nama aplikasi.
5. Pilih Layanan Notifikasi Push Apple Sandbox (APNS_SANDBOX) untuk Platform notifikasi push.
6. Klik Pilih File dan pilih file P12 yang Anda buat saat Anda mengekspor sertifikat SSL.
7. Masukkan kata sandi yang Anda tentukan saat Anda mengekspor sertifikat SSL dan klik Memuat Kredensial dari File.
8. Klik Buat aplikasi platform.
9. Pilih Aplikasi Platform yang baru saja Anda buat dan salin ARN Aplikasi-nya.
10. Kembali ke proyek Anda di Unity Editor, pilih SNSExampledi panel Hierarchy, di panel Inspector dan tempel ARN Aplikasi Platform ke dalam kotak teks berlabel iOS Platform Application ARN.

11. Pilih File/Membangun Pengaturan dan klik tombol Membangun, hal ini akan membuat sebuah proyek Xcode.

Menggunakan Xcode

1. Buka proyek Xcode, dan pilih proyek di Navigator Proyek.
2. Verifikasi apakah pengidentifikasi paket diatur dengan benar
3. Verifikasi apakah Akun Developer Apple Anda ditentukan di Tim - hal ini diperlukan agar Profil Penyediaan Anda dapat berlaku.
4. Membangun proyek dan menjalankannya pada perangkat Anda.
5. Ketuk tombol Daftar untuk Notifikasi, ketuk OK untuk mengizinkan notifikasi, aplikasi akan menampilkan token perangkat Anda

Di Konsol SNS, klik Aplikasi, pilih aplikasi platform anda dan klik Buat Titik Akhir Platform, dan masukkan token perangkat yang ditampilkan oleh aplikasi.

Pada titik ini, aplikasi Anda, APNS, dan NSN telah sepenuhnya dikonfigurasi. Anda dapat memilih aplikasi platform pilihan Anda, memilih titik akhir pilihan Anda, dan klik Publikasikan ke titik akhir untuk mengirim notifikasi push ke perangkat Anda.

Sampel Unity (iOS)

Sampel membuat AWSCredentials instance Cognito untuk menghasilkan kredenal lingkup terbatas sementara yang memungkinkan aplikasi memanggil layanan AWS. Ini juga menciptakan contoh AmazonSimpleNotificationServiceClient untuk berkomunikasi dengan SNS. Aplikasi ini menampilkan dua tombol berlabel Daftar untuk Notifikasi dan Batalkan Daftar.

Saat tombol Daftar untuk Notifikasi diketuk, metode `RegisterDevice()` dipanggil. `RegisterDevice()` memanggil `UnityEngine.iOS.NotificationServices.RegisterForNotifications`, yang menentukan jenis notifikasi (pemberitahuan, suara, atau lencana) yang akan digunakan. Hal ini juga membuat panggilan tak serempak ke APNS untuk mendapatkan token perangkat. Karena tidak ada callback yang ditentukan, `CheckForDeviceToken` dipanggil berulang kali (hingga 10 kali) untuk memeriksa token perangkat.

Ketika token diambil, `AmazonSimpleNotificationServiceClient.CreatePlatformEndpointAsync()` dipanggil untuk membuat titik akhir untuk aplikasi platform SNS.

Sampel tersebut sekarang telah dikonfigurasi untuk menerima notifikasi push. Anda dapat menjelajah ke Konsol SNS, klik Aplikasi yang ada di sisi kiri halaman, pilih aplikasi platform Anda, pilih titik akhir, dan klik Publikasikan ke titik akhir. Pilih titik akhir yang akan digunakan dan klik Publikasikan ke Titik Akhir. Ketik pesan teks di kotak teks dan klik Publikasikan pesan untuk mempublikasikan pesan.

Mengonfigurasi Aplikasi Sampel Unity untuk Android

Buka editor Unity dan buat proyek baru. Impor paket AWS SDK for Unity dengan memilih Assets/ Import Package/Custom Package dan aws-unity-sdk-sns memilih -2.0.0.1.unitypackage. Pastikan semua item dalam dialog Mengimpor Paket dipilih dan klik Mengimpor.

Konfigurasi Unity

Lakukan langkah-langkah berikut untuk mengonfigurasi proyek Unity:

1. Di panel Project, navigasikan ke Assets AWSSDK//examples dan buka SNSExample scene.
2. Di panel Hierarki, pilih. SNSExample
3. Di panel Inspector, tentukan ID Kolam Identitas Cognito Anda.
4. Perhatikan ada kotak teks untuk ARN Aplikasi Platform Android dan ID Proyek Konsol Google, Anda akan menghasilkan informasi itu nanti.
5. Pilih File/Membangun Pengaturan, di dialog Membangun Pengaturan, klik tombol Tambah Saat Ini di bawah kotak daftar Adegan dalam Membangun untuk menambahkan adegan saat ini untuk membangun.
6. Di bawah Platform pilih Android dan klik tombol Pengaturan pemutar..., di Panel Inspector dari editor Unity, klik ikon Android dan gulir ke bawah hingga ke bagian Identifikasi dan tentukan Pengidentifikasi Paket.
7. Salin android-support-v 4.jar dan google-play-services .jar ke direktori Assets/Plugins/Android di panel Project.

Untuk informasi selengkapnya tentang tempat menemukan android-support-v 4.jar, lihat Penyiapan Android Support Library. Untuk informasi selengkapnya tentang cara google-play-services menemukan.jar, lihat Pengaturan Google APIs untuk Android.

Konfigurasi Android

Pertama, tambahkan proyek Google API baru:

1. Di peramban web, buka Konsol Developer Google, klik Buat Proyek.
2. Di kotak Proyek baru, masukkan nama proyek, catat nomor proyek (Anda akan membutuhkannya nanti) dan klik Buat.

Selanjutnya, aktifkan layanan Google Cloud Messaging (GCM) untuk proyek Anda:

1. Di Konsol Developer Google, proyek baru Anda seharusnya sudah dipilih, jika tidak, pilih di drop down di bagian atas halaman.
2. Pilih APIs & auth dari bilah samping di sisi kiri halaman.
3. Di kotak pencarian, ketik “Google Cloud Messaging untuk Android” dan klik tautan Google Cloud Messaging untuk Android di bawah ini.
4. Klik Aktifkan API.

Akhirnya dapatkan Kunci API:

1. Di Google Developers Console, pilih APIs & auth > Credentials.
2. Di bawah Akses API Publik, klik Buat kunci baru.
3. Pada dialog Buat kunci baru, klik Kunci server.
4. Dalam dialog yang dihasilkan, klik Buat dan salin kunci API yang ditampilkan.

Anda akan menggunakan kunci API untuk melakukan autentikasi nanti.

Konfigurasi SNS

1. Dalam peramban web, buka Konsol SNS dan klik Aplikasi yang ada di sisi kiri layar.
2. Klik Buat aplikasi platform untuk membuat aplikasi platform SNS baru.
3. Masukkan Nama Aplikasi
4. Pilih Google Cloud Messaging (GCM) untuk Platform notifikasi push
5. Tempelkan kunci API ke dalam kotak teks berlabel Kunci API.
6. Klik Buat aplikasi platform

7. Pilih Aplikasi Platform yang baru saja Anda buat dan salin ARN Aplikasi-nya.
8. Kembali ke project Anda di Unity Editor, pilih SNSExampledi panel Hierarchy, di panel Inspector dan tempel ARN Aplikasi Platform ke dalam kotak teks berlabel Android Platform Application ARN dan nomor project Anda ke dalam kotak teks berlabel Google Console Project ID.
9. Connect perangkat Android Anda ke komputer Anda, pilih File/Membangun Pengaturan dan klik Membangun dan Jalankan.

Sampel Unity (Android)

Sampel membuat AWSCredentials instance Cognito untuk menghasilkan kredenal lingkup terbatas sementara yang memungkinkan aplikasi memanggil layanan AWS. Ini juga menciptakan contoh AmazonSimpleNotificationServiceClient untuk berkomunikasi dengan SNS.

Aplikasi ini menampilkan dua tombol berlabel Daftar untuk Notifikasi dan Batalkan Daftar.
Saat tombol Daftar untuk Notifikasi diketuk, metode `RegisterDevice()` ini dipanggil.
`RegisterDevice()` memanggil `GCM.Register`, yang mendaftarkan aplikasi dengan GCM. GCM adalah kelas yang didefinisikan dalam kode contoh. Ia membuat panggilan tak serempak untuk mendaftarkan aplikasi dengan GCM.

Ketika callback dipanggil,
`AmazonSimpleNotificationServiceClient.CreatePlatformEndpointAsync` dipanggil untuk membuat titik akhir platform untuk menerima pesan SNS.

Sampel tersebut sekarang telah dikonfigurasi untuk menerima notifikasi push. Anda dapat menjelajah ke Konsol SNS, klik Aplikasi yang ada di sisi kiri halaman, pilih aplikasi platform Anda, pilih titik akhir, dan klik Publikasikan ke titik akhir. Pilih titik akhir yang akan digunakan dan klik Publikasikan ke Titik Akhir. Ketik pesan teks di kotak teks dan klik Publikasikan pesan untuk mempublikasikan pesan.

AWS Lambda

AWS Lambda adalah layanan komputasi yang menjalankan kode Anda sebagai respons atas permintaan atau peristiwa dan secara otomatis mengelola sumber daya komputasi untuk Anda, sehingga membangun aplikasi yang merespons informasi baru dengan cepat menjadi mudah. Fungsi AWS Lambda dapat dipanggil langsung dari aplikasi mobile, IoT, dan Web dan mengirimkan respon kembali secara serentak, sehingga mudah untuk membuat backend yang dapat diskalakan, aman, dan sangat tersedia untuk aplikasi seluler Anda tanpa perlu penyediaan atau mengelola infrastruktur.

AWS Lambda dapat menjalankan fungsi Lambda Anda sebagai respons atas salah satu dari berikut:

- Peristiwa, seperti pembaruan diskrit (misalnya, peristiwa yang dibuat objek di Amazon S3 atau CloudWatch peringatan), atau pembaruan streaming (misalnya, aliran klik situs web atau output dari perangkat yang terhubung).
- Masukan JSON atau perintah HTTPS dari aplikasi kustom Anda.

AWS Lambda menjalankan kode Anda hanya saat diperlukan dan menskalakan secara otomatis, dari beberapa permintaan per hari hingga ribuan per detik. Dengan kemampuan ini, Anda dapat menggunakan Lambda untuk dengan mudah membangun pemicu layanan AWS seperti Amazon S3 dan Amazon DynamoDB, melakukan proses streaming data yang tersimpan di Amazon Kinesis, atau membuat back-end Anda sendiri yang beroperasi pada skala, performa, dan keamanan AWS.

Untuk mem-pelajari selengkapnya tentang cara AWS Lambda bekerja, lihat AWS Lambda: Cara Kerjanya.

Izin

Ada dua jenis izin yang terkait dengan fungsi Lambda:

- Izin eksekusi — Izin yang diperlukan fungsi Lambda Anda untuk mengakses sumber daya AWS lainnya di akun Anda. Anda memberikan izin ini dengan membuat IAM role, yang dikenal sebagai peran eksekusi.
- Izin invokasi — Izin yang dibutuhkan sumber peristiwa untuk berkomunikasi dengan fungsi Lambda Anda. Tergantung pada model invokasi-nya (model push atau pull), Anda dapat memberikan izin ini baik menggunakan peran eksekusi atau kebijakan sumber daya (kebijakan akses yang di-associate dengan fungsi Lambda Anda).

Penyiapan Proyek

Mengatur Izin untuk AWS Lambda

1. Buka Konsol AWS IAM.
2. Lampirkan kebijakan kustom ini untuk peran Anda, yang memungkinkan aplikasi Anda untuk membuat panggilan ke AWS Lambda.

JSON

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

Buat Peran Eksekusi Baru

Peran ini berlaku untuk fungsi Lambda yang akan Anda buat pada langkah berikutnya dan menentukan sumber daya AWS yang dapat diakses oleh fungsi tersebut.

1. Buka Konsol AWS IAM.
2. Klik Peran.
3. Klik Buat Peran Baru.
4. Ikuti petunjuk yang ditampilkan di layar untuk menentukan pilihan layanan dan kebijakan terkait yang perlu diakses oleh fungsi Lambda Anda. Misalnya, jika Anda ingin fungsi Lambda Anda membuat bucket S3, maka kebijakan Anda akan membutuhkan akses tulis ke S3.
5. Klik Buat Peran.

Membuat Fungsi di AWS Lambda

1. Membuka Konsol AWS Lambda.
2. Klik Buat Fungsi Lambda.
3. Klik Lewati untuk melewati membuat cetak biru.
4. Konfigurasikan fungsi Anda sendiri di layar berikutnya. Masukkan nama fungsi, deskripsi, dan pilih waktu aktif Anda. Ikuti petunjuk yang ada di layar berdasarkan waktu aktif yang Anda pilih. Tentukan izin eksekusi Anda dengan menetapkan peran eksekusi yang baru dibuat untuk fungsi Anda.
5. Setelah selesai, klik Selanjutnya.
6. Klik Buat Fungsi.

Buat Klien Lambda

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

Buat Objek Permintaan

Buat objek permintaan untuk menentukan jenis invokasi dan nama fungsi:

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

Melakukan Invokasi Fungsi Lambda

Panggil invokasi, yang memberikan objek permintaan:

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

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

Pemecahan masalah

Karena keterbatasan kelas Unity.WWW yang digunakan oleh AWS SDK for Unity, pesan kesalahan detail tidak diberikan ketika terjadi masalah saat memanggil layanan AWS. Topik ini menjelaskan beberapa gagasan untuk menyelesaikan masalah tersebut.

Pastikan IAM role Memiliki Izin yang Diperlukan

Ketika memanggil layanan AWS, pastikan aplikasi Anda menggunakan identitas dari kolam identitas Cognito. Setiap identitas dalam kolam tersebut di-associate dengan peran IAM (Identity and Access Management). Peran tersebut memiliki satu file kebijakan atau lebih yang terkait dengannya yang menentukan sumber daya AWS yang ditetapkan pengguna untuk peran yang memiliki akses. Secara default dua peran dibuat, satu untuk pengguna yang di-autentikasi, dan satu untuk pengguna yang tidak di-autentikasi. Anda perlu memodifikasi file kebijakan yang ada atau meng-associate file kebijakan baru dengan izin yang diperlukan oleh aplikasi Anda. Jika aplikasi Anda memungkinkan pengguna baik yang di-autentikasi maupun yang tidak di-autentikasi, kedua peran harus diberikan izin untuk mengakses sumber daya AWS yang dibutuhkan aplikasi Anda.

File kebijakan berikut menunjukkan cara memberikan akses ke bucket S3:

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

File kebijakan berikut menunjukkan cara untuk memberikan akses ke basis data 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"
}]
}
```

Untuk informasi lebih lanjut tentang menentukan kebijakan, lihat Kebijakan IAM.

Menggunakan HTTP Proxy Debugger

Jika layanan AWS yang dipanggil aplikasi Anda memiliki titik akhir HTTP atau HTTPS, Anda dapat menggunakan debugger HTTP/HTTPS proxy untuk melihat permintaan dan merespons untuk mendapatkan lebih banyak wawasan tentang apa yang terjadi. Ada sejumlah HTTP proxy debugger yang tersedia seperti:

- Charles - sebuah web debugging proxy untuk OSX
- Fiddler - sebuah web debugging proxyfidd untuk Windows

> **Important**
>
> Ada sebuah masalah yang diketahui dengan Penyedia Kredensial Cognito saat menjalankan Charles web debugging proxy yang mencegah penyedia kredensial bekerja dengan semestinya.

Charles dan Fiddler, keduanya memerlukan beberapa konfigurasi untuk dapat melihat lalu lintas SSL terenkripsi, silakan baca dokumentasi untuk alat-alat ini untuk mendapatkan informasi lebih lanjut. Jika Anda menggunakan proxy debugging web yang tidak dapat dikonfigurasi untuk menampilkan lalu lintas terenkripsi, buka file aws_endpoints_json (terletak di) AWSUnity SDK/AWSCore/Resources dan atur tag HTTP untuk layanan AWS yang perlu Anda debug ke true