# AWS SNS – iOS (swift)

* 前言

為了在手機中做到簡單的Push Notification功能又不需要寫伺服器，利用Amazon提供的AWS服務可簡單做到，以下將紀錄實現這功能的步驟。

* Apple Push Notification Service入門

APNs前置作業

註冊APNs的iOS程式

APNs的SSL證書

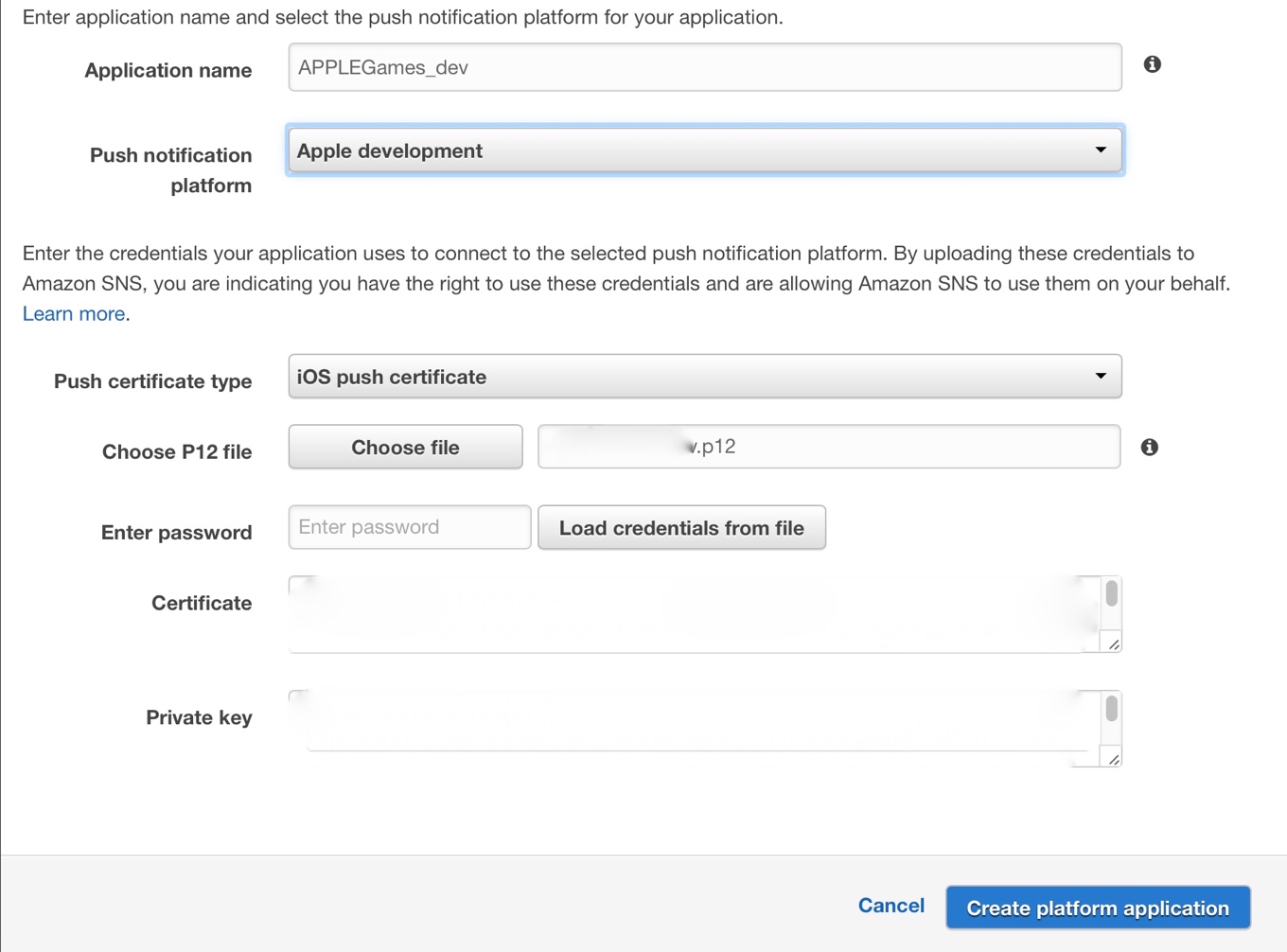
應用程式金鑰

Device Token

有關APNs的前置作業請參閱[Swift 新手指南：為 iOS Apps 添加推送通知（Push Notifications）](http://www.appcoda.com.tw/push-notification-ios/)。

* 手動輸入Token

在取得裝置的Device Token後就可以在AWS的控制台來做測試了！首先到[Amazon官網](https://aws.amazon.com/tw/sns/?nc2=h_l3_ms)註冊一個帳號，再點「立即免費使用」登入您的帳號，登入成功就會進入SNS dashboard頁面，在左邊的控制台點選Applications，在這裡我們會用到剛剛取得的Device Token，注意！右上角有一個選擇地區的選項，每個地區的資料都是獨立的。選好地區後回到Applications裡面，點擊「Create platform application」，請在Application name輸入您要的App名稱、Push notification platform選擇您要使用的Notification服務對象類型，如果您申請的APNs服務是開發者模式請選擇「Apple development」。

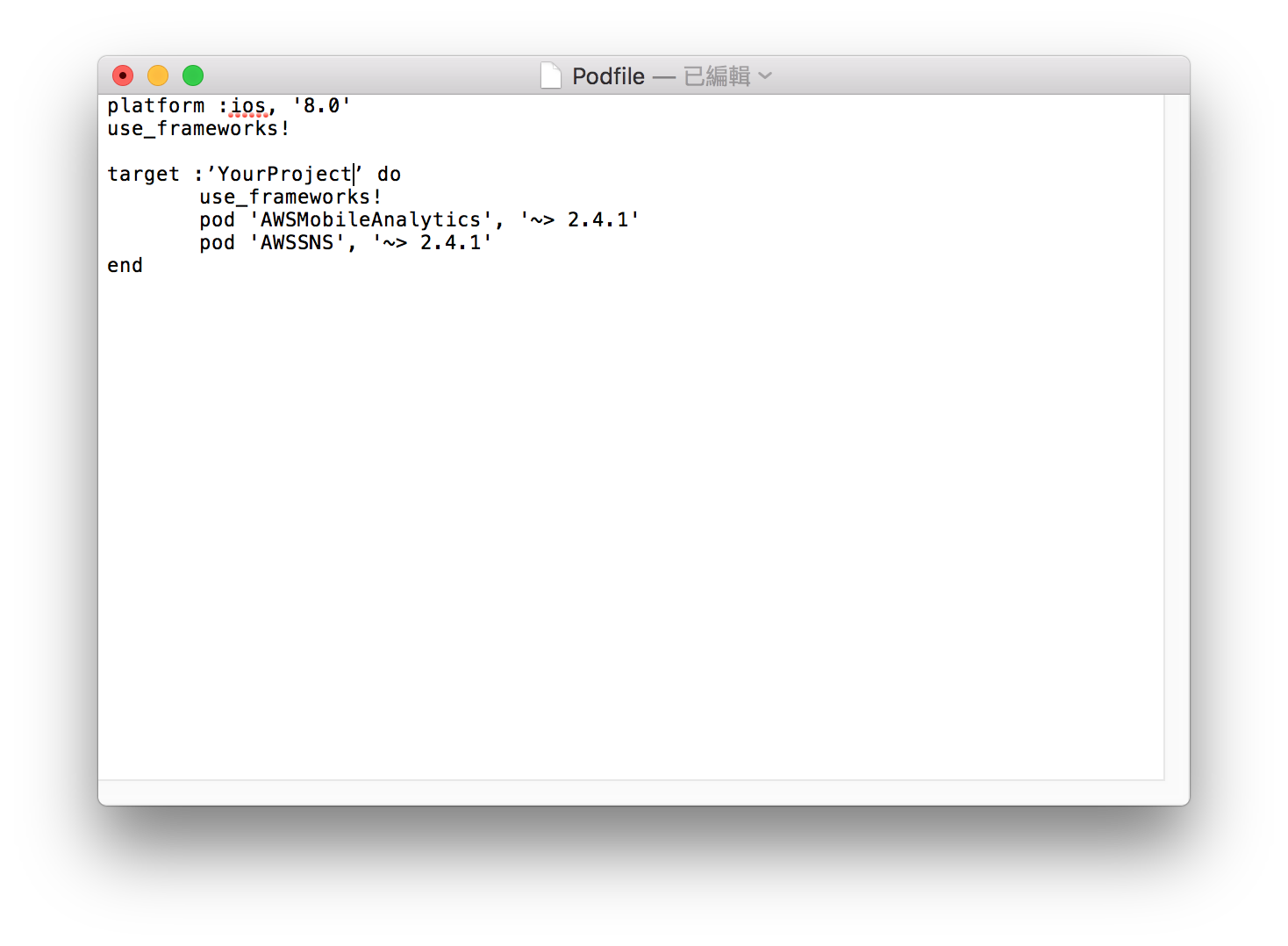
範例

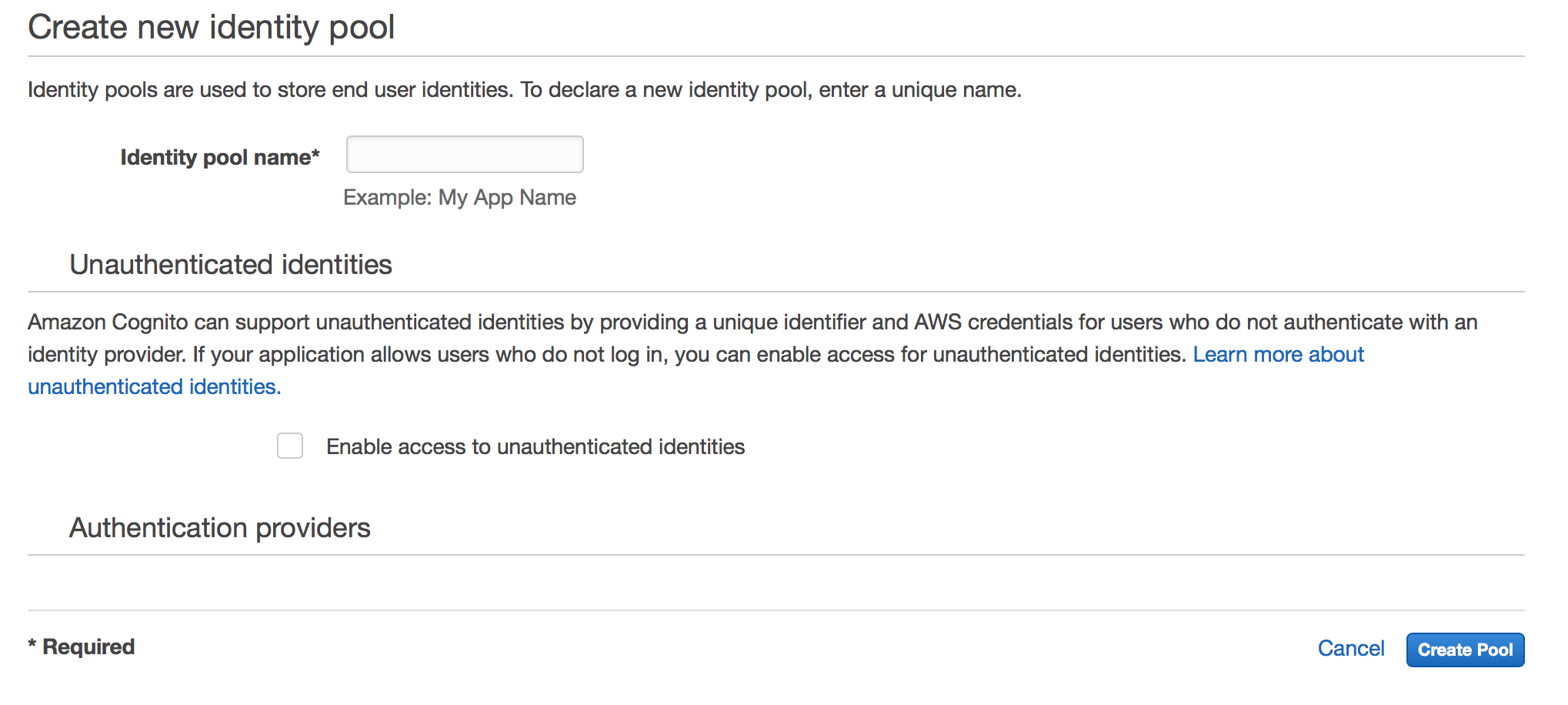
點Load credentials from file就可以看到您的金鑰，再點Create platform application建立程式。

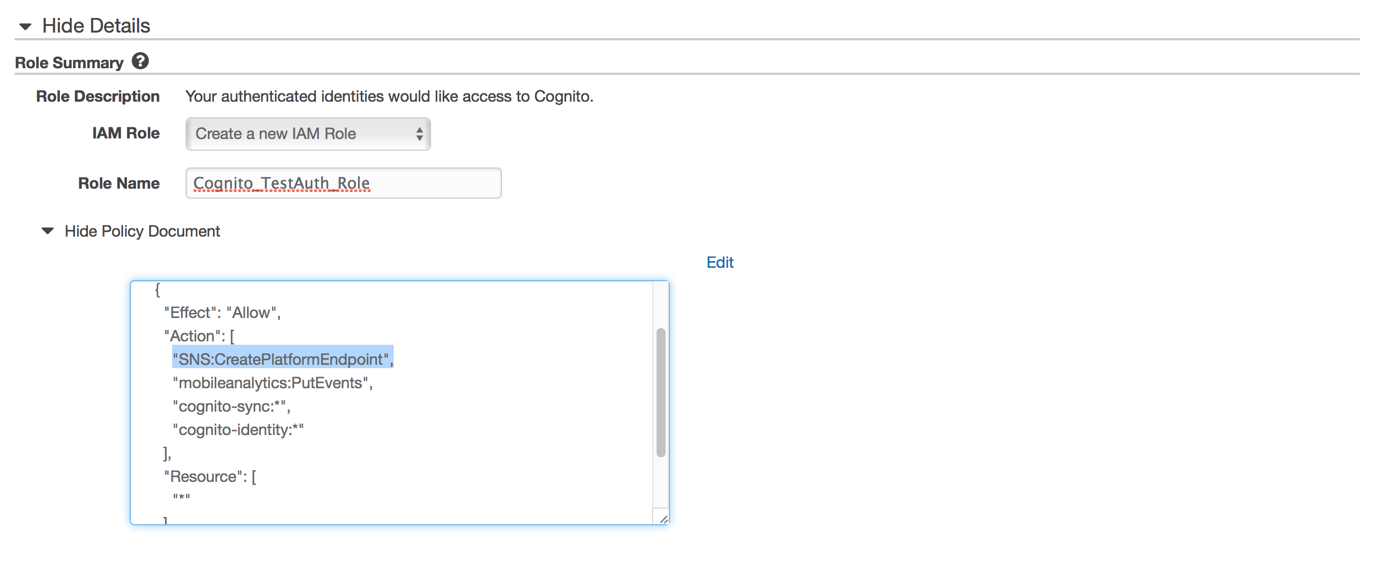
建立成功後可以看到您的程式出現，點選您的程式再點Create platform endpoint會跳出另一個視窗，把您剛剛取得的Token輸入進去再點Add endpoint，建立成功後點進您的程式中就能看到剛剛用Token建立的一個endpoint，再來只要利用Publish message功能就可以輕鬆送出通知到您的手機。

* 利用API建立Endpoint

前一段我們用手動輸入Token的方式成功送出通知，但在現實運作中必須讓每一個Device都可以自動到AWS SNS上取得專屬的Endpoint才行，以下我們將實現這個方法，利用AWS提供的CreatePlatformEndpoint方法([AWS提供的範例](https://github.com/awslabs/aws-sdk-ios-samples/tree/master/SNS-MobileAnalytics-Sample/Swift/))。

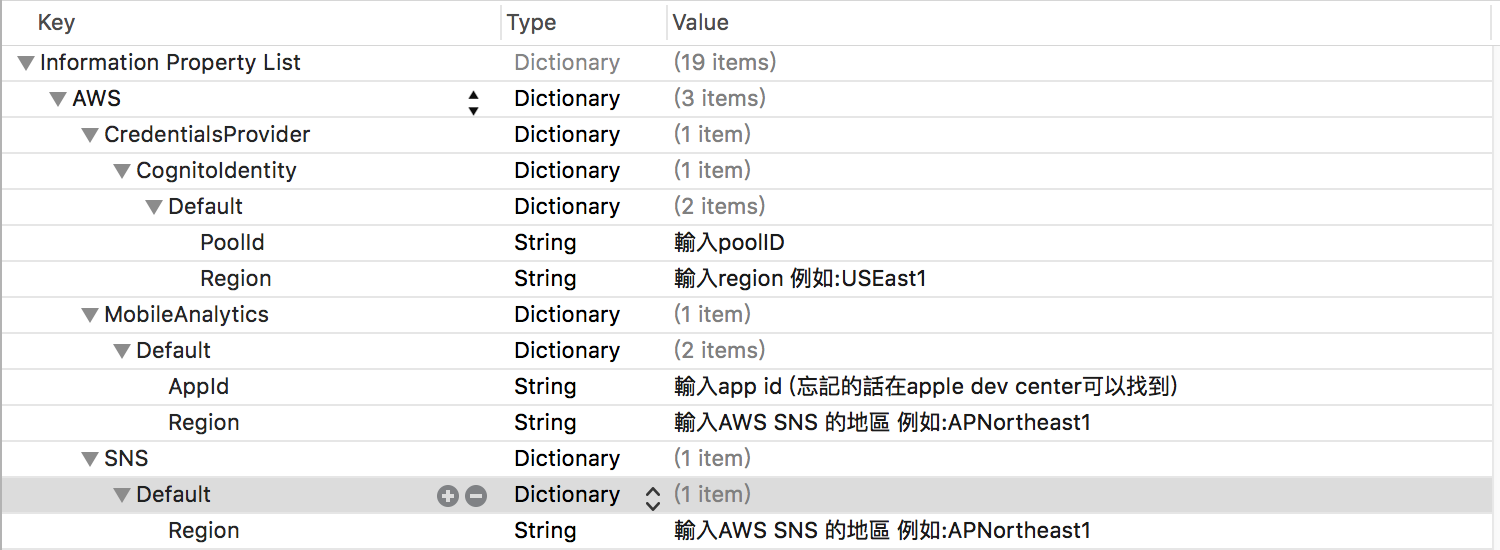
根據官方的範例說明，首先要用cocoapods安裝'AWSMobileAnalytics',‘AWSCognito’和 'AWSSNS'

在終端機輸入pod install，安裝好接著打開您的xcode專案，建立Objective-C bridging header file([不會的參考這裡](http://www.learnswiftonline.com/getting-started/adding-swift-bridging-header/))，完成後打開瀏覽器到[Amazon Cognito console](https://console.aws.amazon.com/cognito/home?region=us-east-1)，點擊Manage Federated Identities，接著在左上角點Create new identity pool可以看到下圖的頁面。

輸入您要的pool name後請注意Enable access to unauthenticated identities要打勾，點右下角的Create Pool進行權限編輯，由於我們要用到SNS的CreatePlatformEndpoint所以應如下圖在authenticated和unauthenticated都加上"SNS:CreatePlatformEndpoint"，如果這邊沒有編輯好也可以點開右上角您的使用者帳號的小三角，選Security Credentials再點進左邊的Roles編輯剛剛建立的Role，在Permissions裡可以用搜尋的方式找到不同權限來使用，這邊我們可以加入AmazonSNSFullAccess來快速達到我們的目的。

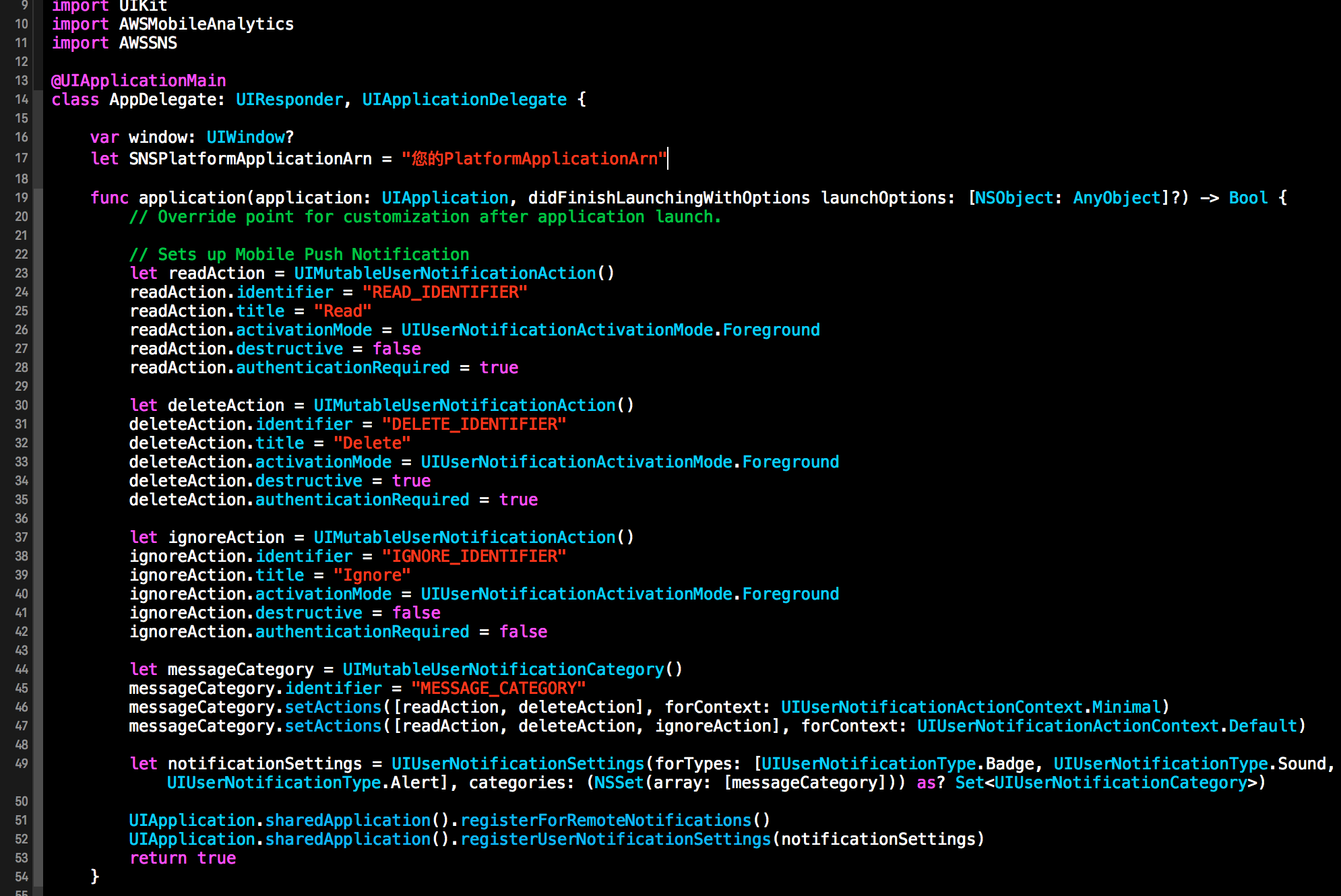
點選右下角的Allow完成這段工作，您可以在Sample code中看到您的identityPoolId:"us-east-1:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxx"以及regionType:.USEast1，記住這兩個東西待會我們都會用到。

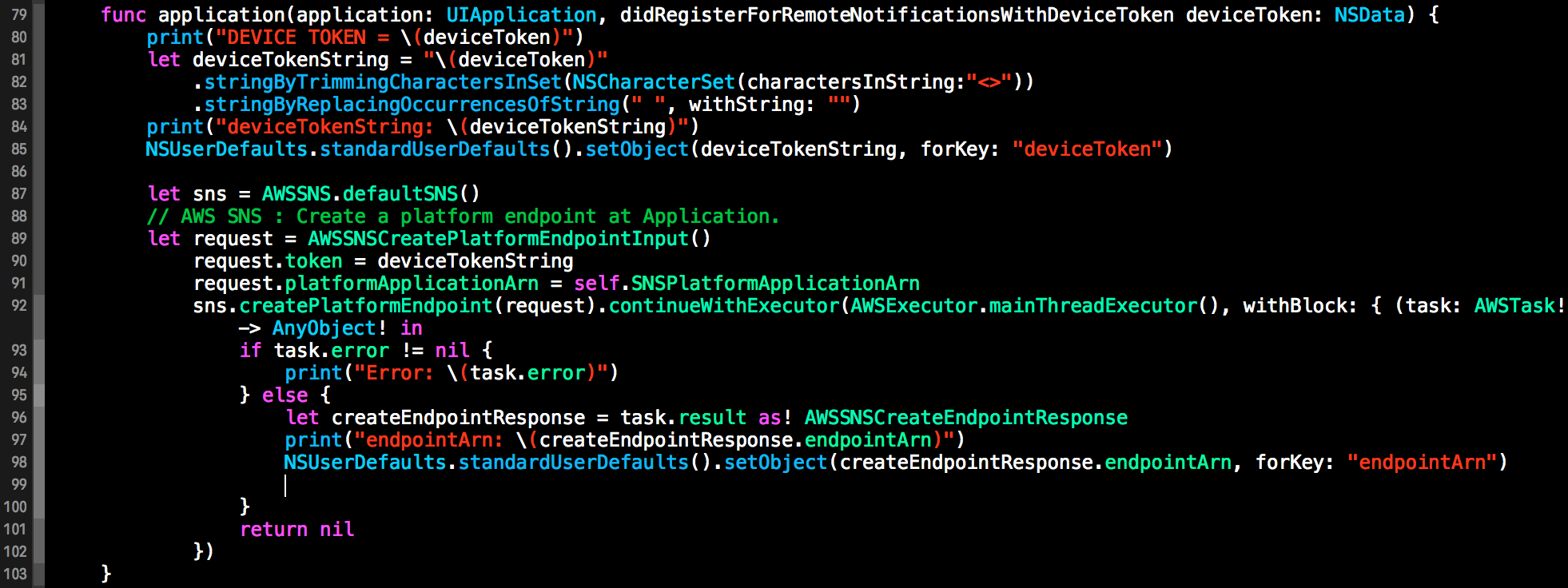
接下來回到xcode專案，在info.plist中新增AWS，如下圖



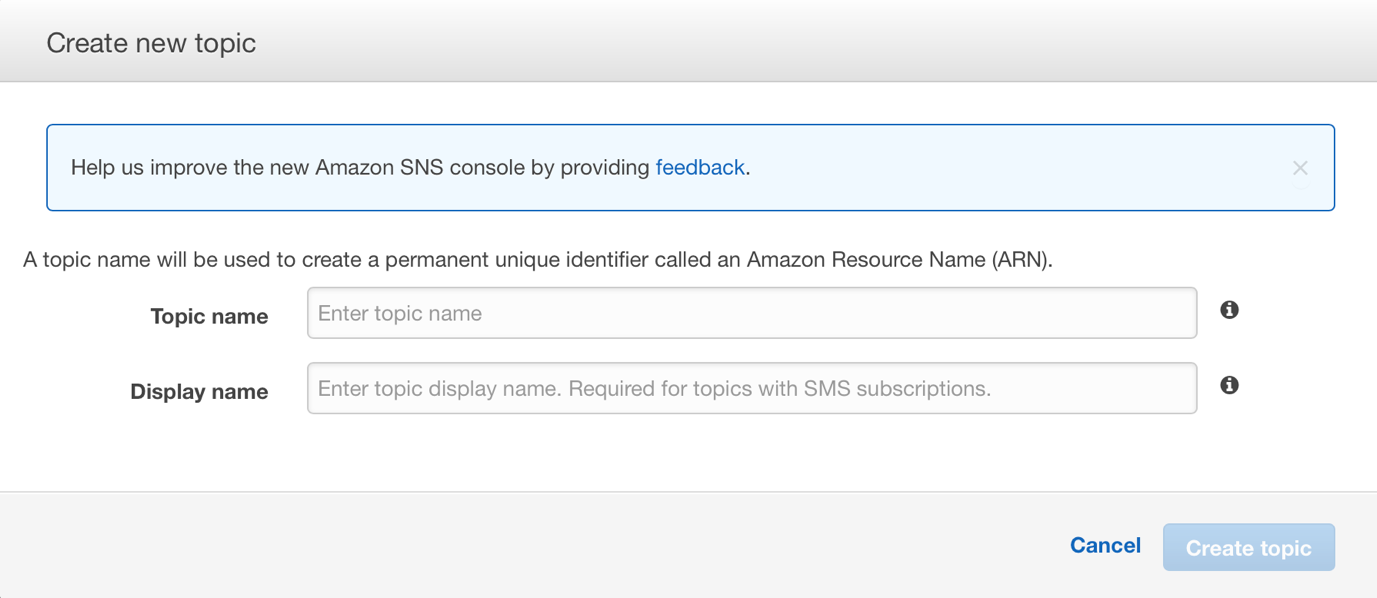
再來就要進入寫程式的階段啦！

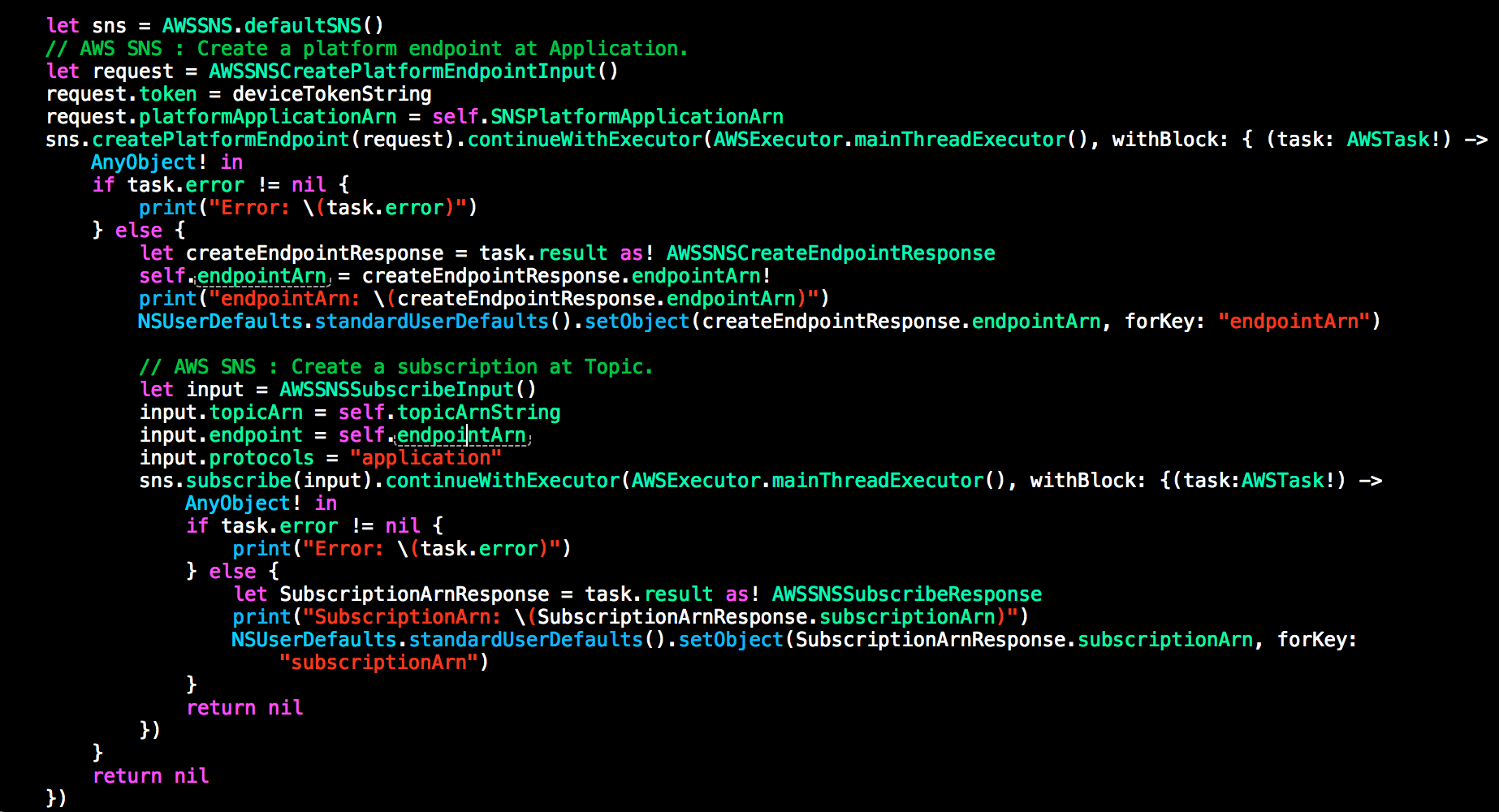
需要程式範例可到github下載[aws-sdk-ios-samples](https://github.com/awslabs/aws-sdk-ios-samples.git)。

以下是我的程式範例 全都寫在AppDelegate.swift





以上這些都設定好後App就可以自動到AWS SNS上註冊一個Endpoint，也可以推播訊息到您的裝置，但做到這邊後測試發現一次只能推播一台裝置，很明顯的這不符合我們的需求，因此我們必須在SNS控制台新增一個Topic，接下來要讓裝置可以訂閱我們的Topic，再從控制台用Topic發送訊息到所有訂閱它的裝置。方法也很簡單，首先Create new topic，

建立後進入您的Topic中在Other topic actions中點Edit topic policy將Allow these users to subscribe to this topic調成Everyone然後點Update policy，這樣就開放了權限。回到xcode我們加幾段code來實現自動訂閱。

利用AWSSNSCreatePlatformEndpointInput()回傳的endpointArn和剛剛建立Topic時得到的topicArn以及您要使用的服務類型protocols ”application”，我們用AWSSNSSubscribeInput()就能實現訂閱Topic的功能。Command+R跑一次專案，當您的App成功啟動後會看到下面出現許多Log，如果沒有看到AWS回傳的Error理論上您就已經成功Create endpoint並訂閱Topic，回到SNS控制台您應該會在您的Topic裡看到訂閱它的裝置，再點上面的Publish to topic就可以送出推播訊息給所有訂閱該Topic的裝置囉！關於更多AWS可使用的**Amazon Simple Notification Service API**請參考[這裡](http://docs.aws.amazon.com/zh_cn/sns/latest/api/Welcome.html)。

* 後記

我在嘗試的過程中卡了很久，網路上又找不到太非英文的資料…好不容易才看懂AWS傳來的Error，原來是因為沒設定好Role的權限還有info.plist檔中的Region寫錯，怕下次忘記要怎麼做所以寫個筆記。

* 在SNS控制台用JSON推播範例

推播到APNS開發者跟正式版

{

"default": "Hello",

"APNS": "{\"aps\":{\"alert\": \"Hello\",\"sound\":\"default\",\"badge\":1} }",

"APNS\_SANDBOX":"{\"aps\":{\"alert\":\"Hello\",\"sound\":\"default\",\"badge\":1}}"

}