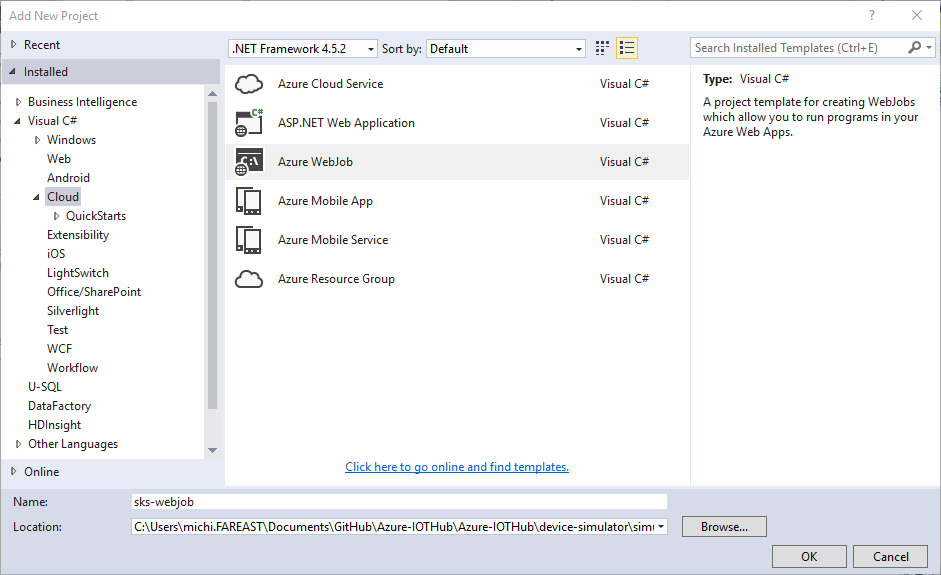
## 環境準備

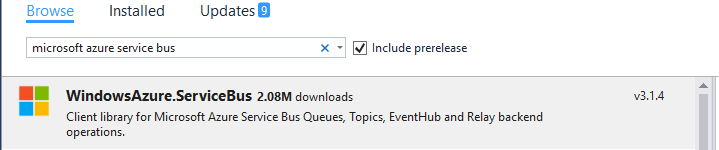
* Visual Studio 2015
* Azure SDK

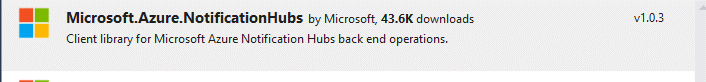
## 步驟

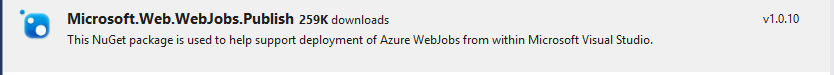
* 已Administrator權限打開Visual Studio 2015，新增一個 Web Job專案



* 以NUGet新增以下參考





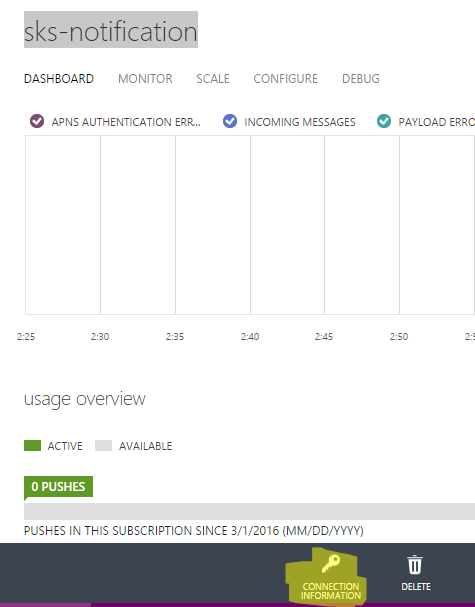


* 打開Functions.cs檔案，加入以下的namespace

using Microsoft.Azure.NotificationHubs;

using Microsoft.ServiceBus.Messaging;

* 新增以下的Function；其中<hub name>為先前建立的notification hub名稱；Connection String為Full Access的Connection String，可以在以下找到



* 在Functions.cs新增namespace引用

using Microsoft.Azure.WebJobs;

using Microsoft.Azure.NotificationHubs;

using Newtonsoft.Json;

* 新增以下程式碼

private static string nhConnectionString = "<notification hub connection string>";

private static string hubName = "<notification hub name>";

// This function will get triggered/executed when a new message is written

// on an Azure Queue called queue.

private static string sbQueueAdminConnectinString = "<service bus queue level connection string>";

private static string sbQueueName = "<queue name>";

public static void ProcessQueueMessage([QueueTrigger("queue")] string message, TextWriter log)

{

log.WriteLine(message);

}

private static async Task SendNotificationAsync(string msg)

{

await SendWindowsNotificationAsync(msg);

await SendAndroidNotificationAsync(msg);

}

private static async Task SendWindowsNotificationAsync(string msg)

{

NotificationHubClient hub = NotificationHubClient

.CreateClientFromConnectionString(nhConnectionString, hubName);

var toast = $"<toast launch=\"launch\_arguments\"><visual><binding template=\"ToastText01\"><text id=\"1\">{msg}</text></binding></visual></toast>";

await hub.SendWindowsNativeNotificationAsync(toast);

}

private static async Task SendAndroidNotificationAsync(string msg)

{

NotificationHubClient hub = NotificationHubClient

.CreateClientFromConnectionString(nhConnectionString, hubName);

Newtonsoft.Json.Linq.JObject o = JsonConvert.DeserializeObject(msg) as Newtonsoft.Json.Linq.JObject;

var toast = "{data:{message:'{device} alert at {time}'}}".Replace("{device}", (string)o["deviceid"]).Replace("{time}", (string)o["time"]);

var results = await hub.SendGcmNativeNotificationAsync(toast);

}

* 新增以下程式碼

public static async Task LookupNotification()

{

QueueClient qc = QueueClient.CreateFromConnectionString(

sbQueueAdminConnectinString,

sbQueueName);

OnMessageOptions options = new OnMessageOptions();

options.AutoComplete = false;

options.AutoRenewTimeout = TimeSpan.FromMinutes(1);

// Callback to handle received messages.

qc.OnMessage((message) =>

{

try

{

// Process message from event processor host.

if (message.Properties.Keys.Contains("message-source") && (string)message.Properties["message-source"] == "evh")

{

var o = message.GetBody<System.IO.Stream>();

using (var r = new StreamReader(o))

{

var msg = r.ReadToEnd();

Console.WriteLine("Body: " + msg);

Console.WriteLine("MessageID: " + message.MessageId);

SendNotificationAsync(msg);

// Remove message from queue.

message.Complete();

}

}

else

{

// Process message from stream analytics.

var msg = message.GetBody<string>();

Console.WriteLine("Body: " + msg);

Console.WriteLine("MessageID: " + message.MessageId);

SendNotificationAsync(msg);

// Remove message from queue.

message.Complete();

}

}

catch (Exception exp)

{

// Indicates a problem, unlock message in queue.

Console.WriteLine("EXCEPTION:" + exp.Message);

if(exp.InnerException != null)

{

Console.WriteLine("INNER:" + exp.Message);

}

if(exp.StackTrace != null)

{

Console.WriteLine($"Stack:{exp.StackTrace}");

}

message.Abandon();

//message.Complete();

}

}, options);

* 修改Main()如下：

static void Main()

{

var host = new JobHost();

Functions.LookupNotification();

// The following code ensures that the WebJob will be running continuously

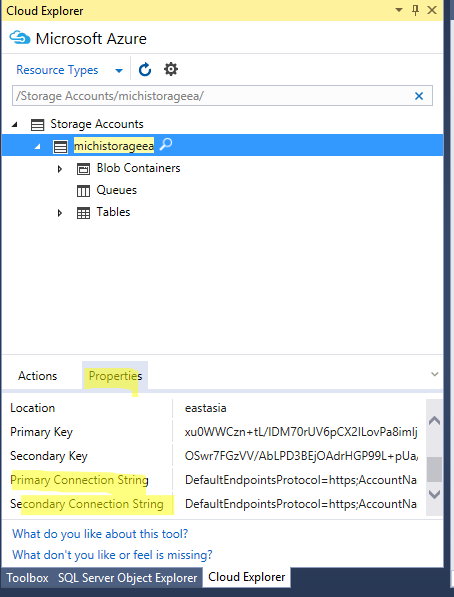
host.RunAndBlock();

}

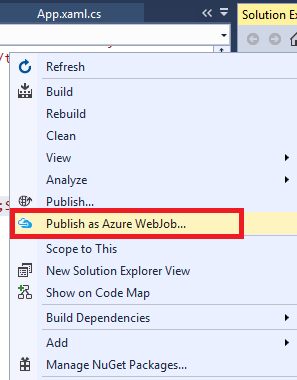
* 打開App.Config，修改以下兩個Connection String；此兩個Storage為提供Web Job Log以及程式使用；你可以在Visual Studio的Cloud Explorer中取得Connection String

<add name="AzureWebJobsDashboard" connectionString="" />

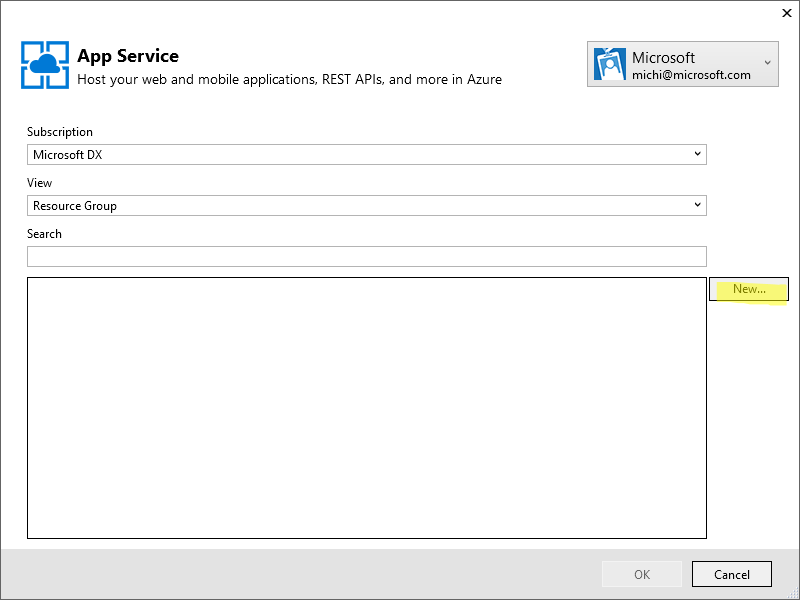
<add name="AzureWebJobsStorage" connectionString="" />



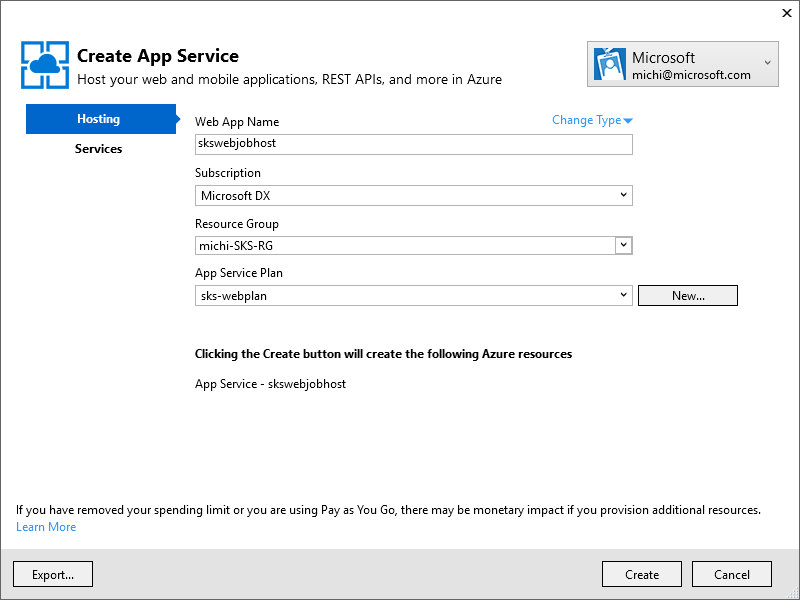
* 發布此Web Job



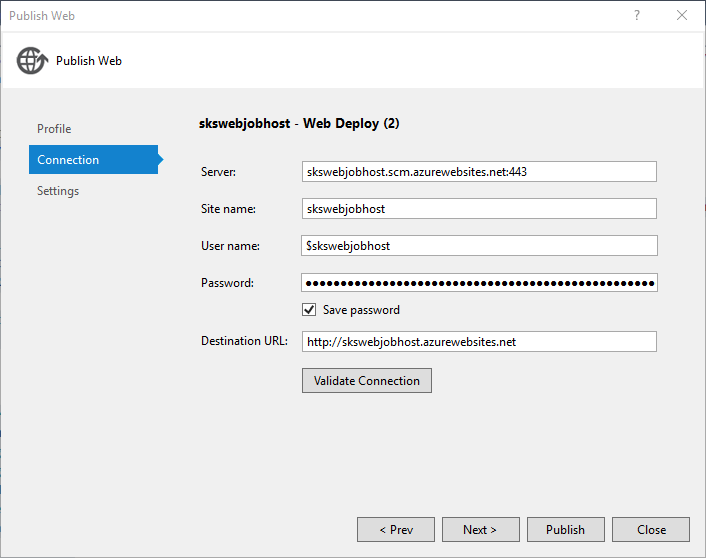
* 選擇新增一個Web Job



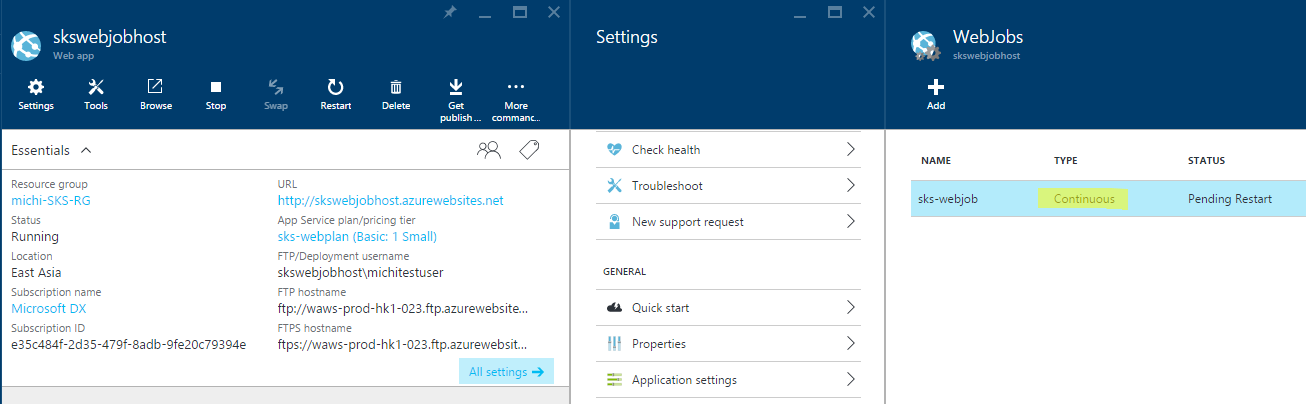
* 輸入基本資料



* 按下Publish發布



* 回到Portal可以看到剛剛的Web Job已經建立



* <https://azure.microsoft.com/zh-tw/documentation/articles/websites-dotnet-webjobs-sdk-service-bus/>