當.Net Core遇到AWS Lambda

Jamis Liao

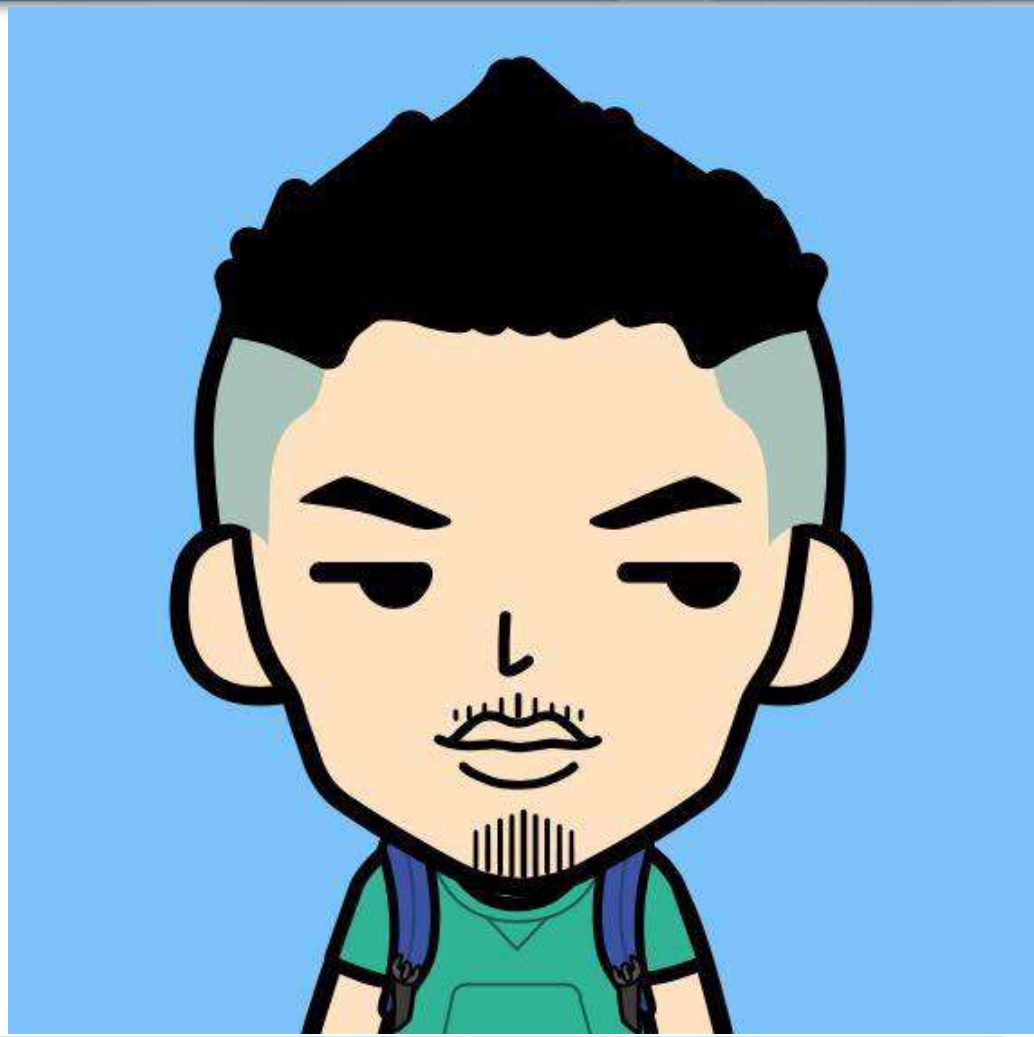


Microsoft® Most Valuable Professional

About Me

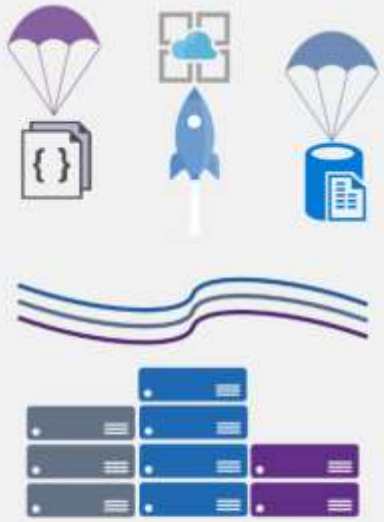
- Jamis Liao
- 微軟最有價值專家

- Tag
  - ASP.Net
  - MVC
  - C#
  - DevOps
  - 軟體開發流程
  - 系統架構
  - Cloud Architecture
- Blog
  - 天橋下的說書人



什麼是Serverless





Servers are
fully-abstracted



Scaling is event-driven
not resource-driven



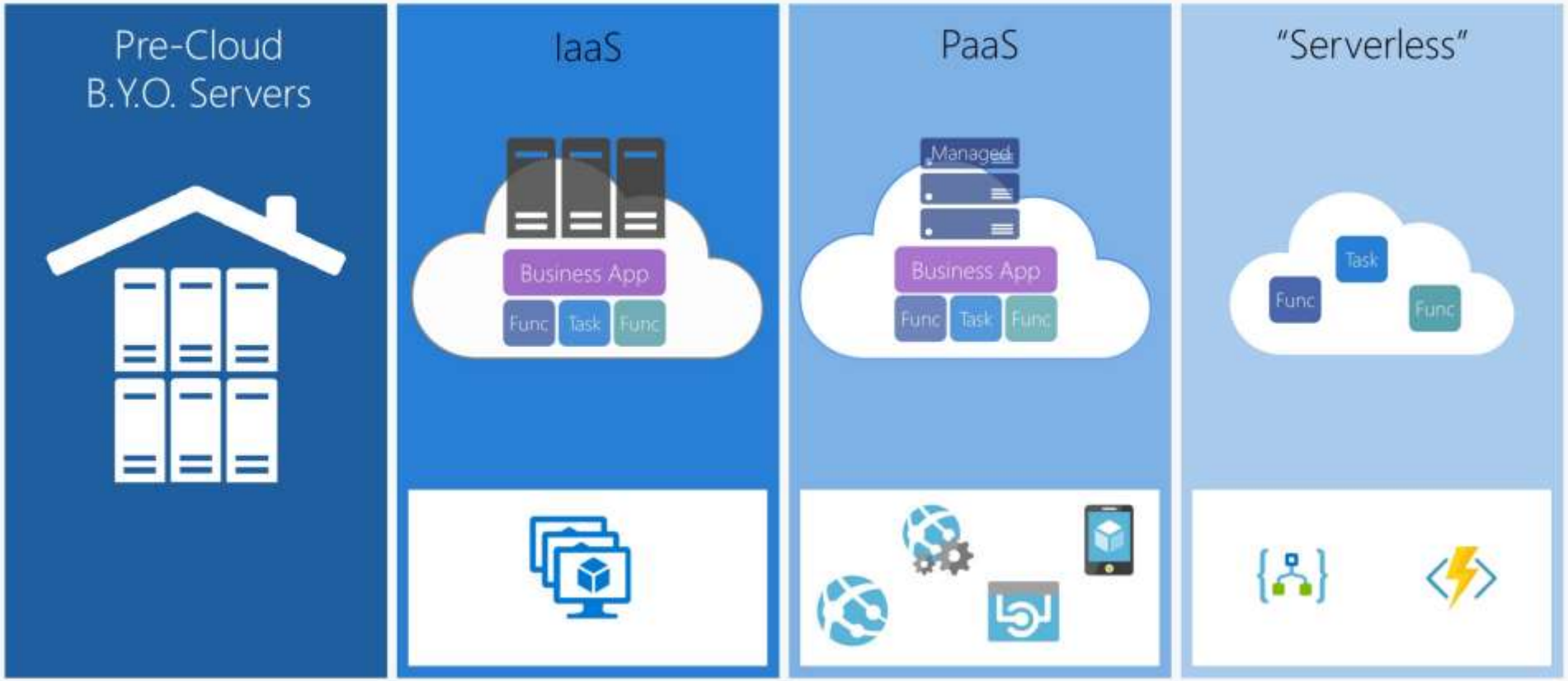
Pay only
for what you use

Source: https://goo.gl/kMecqi

什麼是Serverless

- 架構模式
- 無伺服器的維護，AutoScaling的負擔

Different for IaaS PaaS

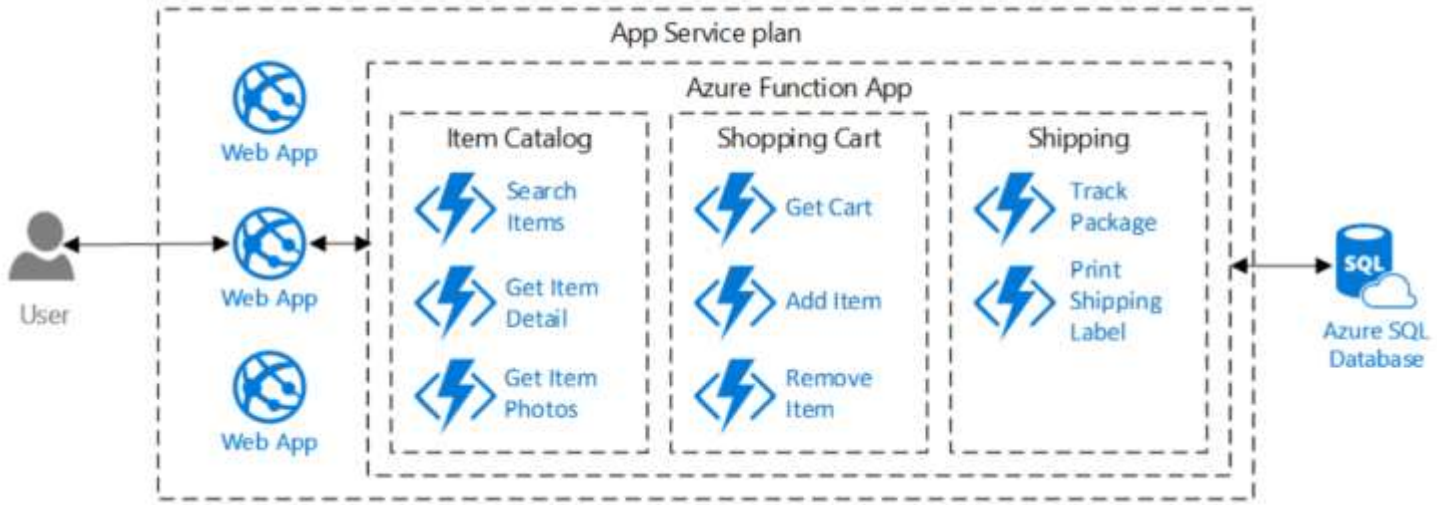


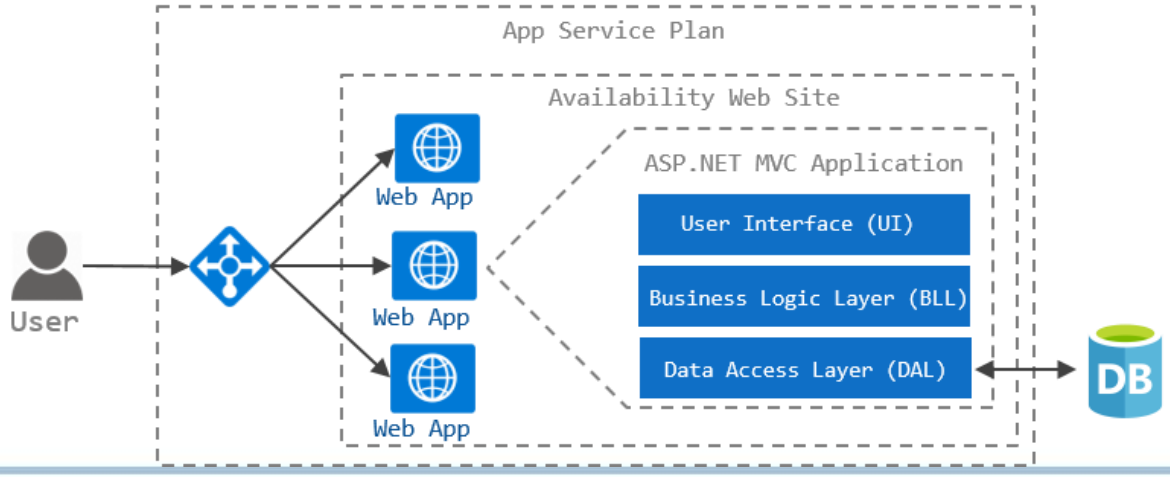
Source: https://goo.gl/kMecqi

Funtion as a Service

- Functions as a Service / FaaS
  - AWS Lambda
  - Azure Function
  - Cloud Function

Funtion as a Service





Serverless的好處



Do less
Manage less



Ease of scale



Pay per
execution

Source: https://goo.gl/kMecqi

AWS Lambda

- AWS在2014年推出的Lambda無伺服器運算服務
- 2017年開始支援.NET Core

到底是少維護了什麼.....

伺服器

作業系統

容量佈建

自動擴展

安全性修補程式部署

監控和記錄程式碼

以上通通不用你管

到底是少維護了什麼.....

伺服器

作業系統

容量佈建

自動擴展

安全性修補程式部署

監控和記錄程式碼

AWS Lambda

- Event-driven
- 容易整合AWS自家其他服務
  - S3
  - CloudWatch
  - DynamoDB

今天會用到

- CloudWatch
  - Schedule
- API Gateway

講講Lambda support的.NET Core

- AWS Lambda對.NET Core的版本支援如下
  - .NET Core: 1.0.5
  - .NET standard: 1.6
  - EF Core : 1.0.3

不講你會Try到死的事

- 最新的套件不一定能用，就算是有支援.NET Core

開始講講跟開發Lambda有關的事吧.....

你必須先安裝幫你產專案的工具

Lambda template安裝

- Lambda template CLI Tool
  - dotnet new -i Amazon.Lambda.Templates::*
  - dotnet new -all

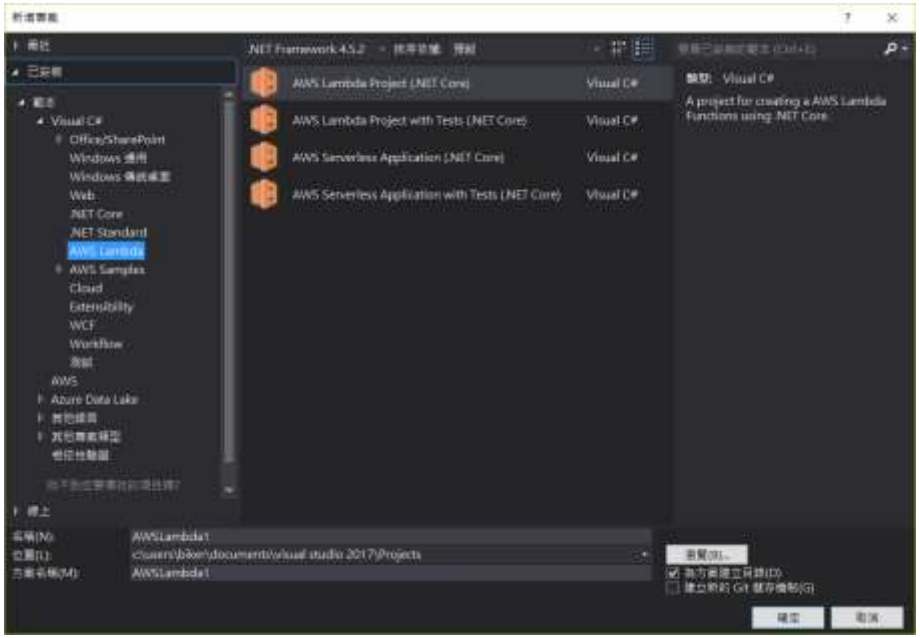
你還可以這樣產Lambda project

- Yeoman
  - npm install –g yo generator-aws-lambda-dotnet
  - yo aws-lambda-dotnet

https://github.com/aws/aws-lambda-dotnet

其實還有一個更簡單的方法

- AWS Toolkit for Visual Studio 2017
  - https://goo.gl/U5xrer





麻煩的.csproj

- .NET Core 1.1 
  - .csproj

- .NET Core 1.1 
  - Project.json

但是……產出來專案都是Project.json

- Lambda template CLI Tool
- Yeoman

migration



從 project.json 移轉至 csproj

您可以使用下列其中一種方法從 *project.json* 移轉至 *.csproj*：

- Visual Studio 2017
- dotnet migrate 命令列工具

這兩種方法使用相同的基礎引擎來移轉專案，因此結果會相同。 在大部分情況下，您只需要使用這兩種方式之一即可將 *project.json* 移轉至 *csproj*，而不需要進一步手動編輯專案檔。 所產生的 *.csproj* 檔案將會以包含目錄的相同名稱命名。

Visual Studio 2017

當您開啟 *.xproj* 檔或參考 *.xproj* 檔的方案檔時，[單向升級] 對話方塊會隨即出現。 此對話方塊會顯示要移轉的專案。 如果您開啟方案檔，則會列出方案檔中指定的所有專案。 檢閱要移轉的專案清單，然後選取 [確定]。

所以建議使用

- AWS T

```
<Project Sdk="Microsoft.NET.Sdk">

  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>netstandard1.6</TargetFramework>
    <TargetFramework>netcoreapp1.0</TargetFramework>
  </PropertyGroup>

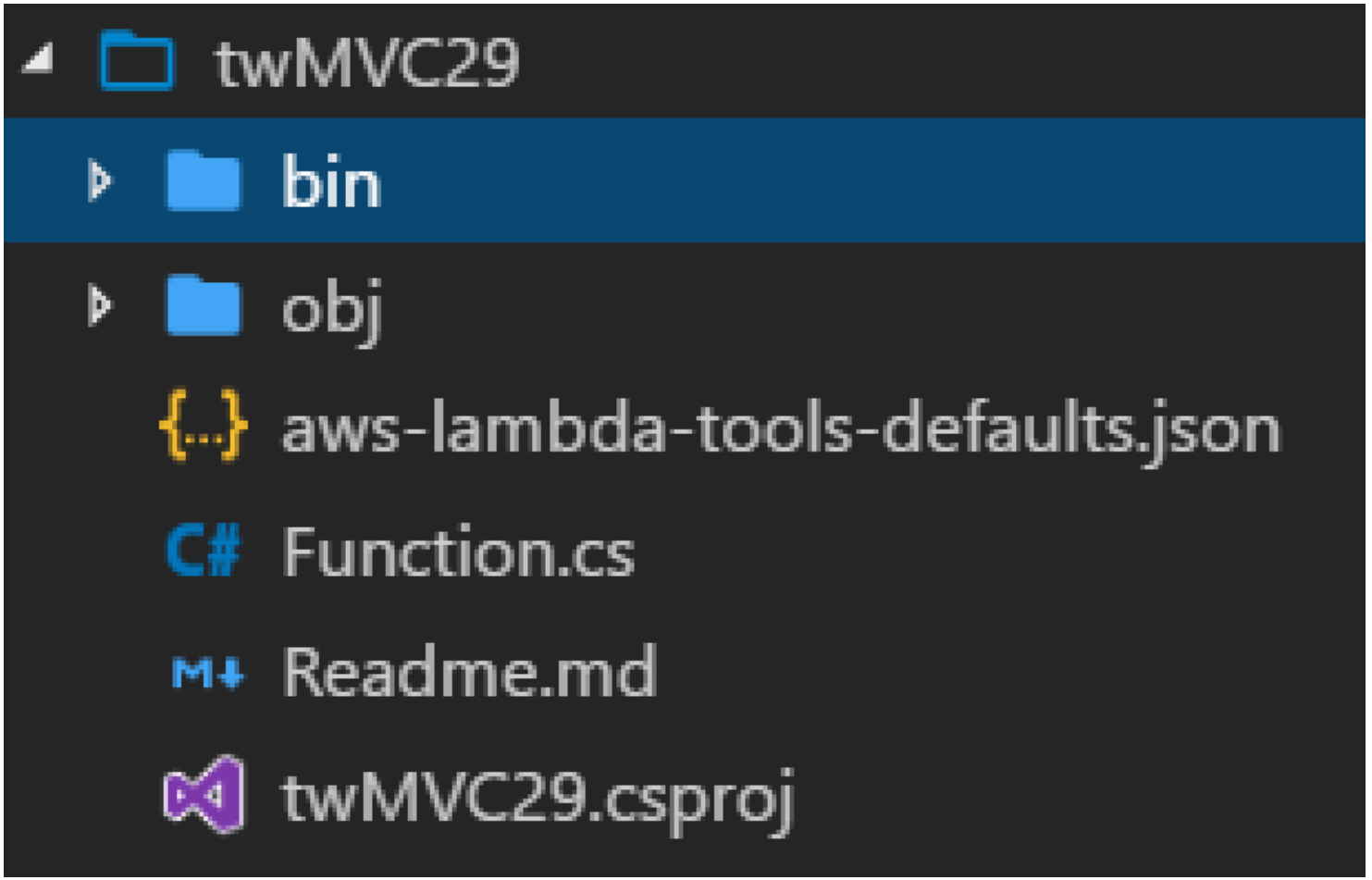
  <ItemGroup>
    <PackageReference Include="Amazon.Lambda.Core" Version="1.0.0" />
    <PackageReference Include="Amazon.Lambda.Serialization.Json" Version="1.1.0" />
    <PackageReference Include="Amazon.Lambda.APIGatewayEvents" Version="1.1.0" />
  </ItemGroup>

  <ItemGroup>
    <DotNetCliToolReference Include="Amazon.Lambda.Tools" Version="1.5.0" />
  </ItemGroup>

   <ItemGroup>
    <ProjectReference Include="..\twMVC29.Model\twMVC29.Model.csproj" />
  </ItemGroup>

</Project>
```

專案結構長這樣


twMVC29
bin
obj
aws-lambda-tools-defaults.json
Function.cs
Readme.md
twMVC29.csproj

產出來的程式碼長這樣

```
using Amazon.Lambda.Core;

// Assembly attribute to enable the Lambda function's JSON input to be converted into a .NET class.
[assembly: LambdaSerializer(typeof(Amazon.Lambda.Serialization.Json.JsonSerializer))]

namespace AWSLambda1
{
    0 references
    public class Function
    {

        /// <summary>
        /// A simple function that takes a string and does a ToUpper
        /// </summary>
        /// <param name="input"></param>
        /// <param name="context"></param>
        /// <returns></returns>
        0 references
        public string FunctionHandler(string input, ILambdaContext context)
        {
            return input?.ToUpper();
        }
    }
}
```

開發你必須要注意的

- 指定Target framework
  - netstandard 1.6
  - netcoreapp 1.0
- 看懂csproj的Tag
  - PackageReference
  - DotNetCliToolReference
  - ProjectReference

開發你必須要注意的

- 指定Target framework
  - netstandard 1.6
  - netcoreapp 1.0
- 看懂csproj的Tag
  - PackageReference
  - DotNetCliToolReference
  - ProjectReference

- 序列化套件的問題
- 不要自己裝Json.net

- 序列化套件的問題
- 不

```
using Amazon.Lambda.Core;

// Assembly attribute to enable the Lambda function's JSON input to be converted into a .NET class.
[assembly: LambdaSerializer(typeof(Amazon.Lambda.Serialization.Json.JsonSerializer))]

namespace AWSLambda1
{
    0 references
    public class Function
    {

        /// <summary>
        /// A simple function that takes a string and does a ToUpper
        /// </summary>
        /// <param name="input"></param>
        /// <param name="context"></param>
        /// <returns></returns>
        0 references
        public string FunctionHandler(string input, ILambdaContext context)
        {
            return input?.ToUpper();
        }
    }
}
```

開發 & Debug小技巧

- 變成Console程式
  - OutputType: Exe
- 一樣可以使用Multi project

aws-lambda-tools-defaults.json

- profile
- region
- function-memory-size
- function-timeout
- function-handler
  - {DLL Name}::{Class namespace}:{Method name}



demo

AWS Lambda

How to deploy

- Lambda Tool
  - lambda deploy-function
  - lambda invoke-function
  - lambda delete-function
  - lambda update-function
  - package + web
- Lambda help

Deploy完了……那要怎麼用呢?

- invoke-function
  - --pyload from file



demo

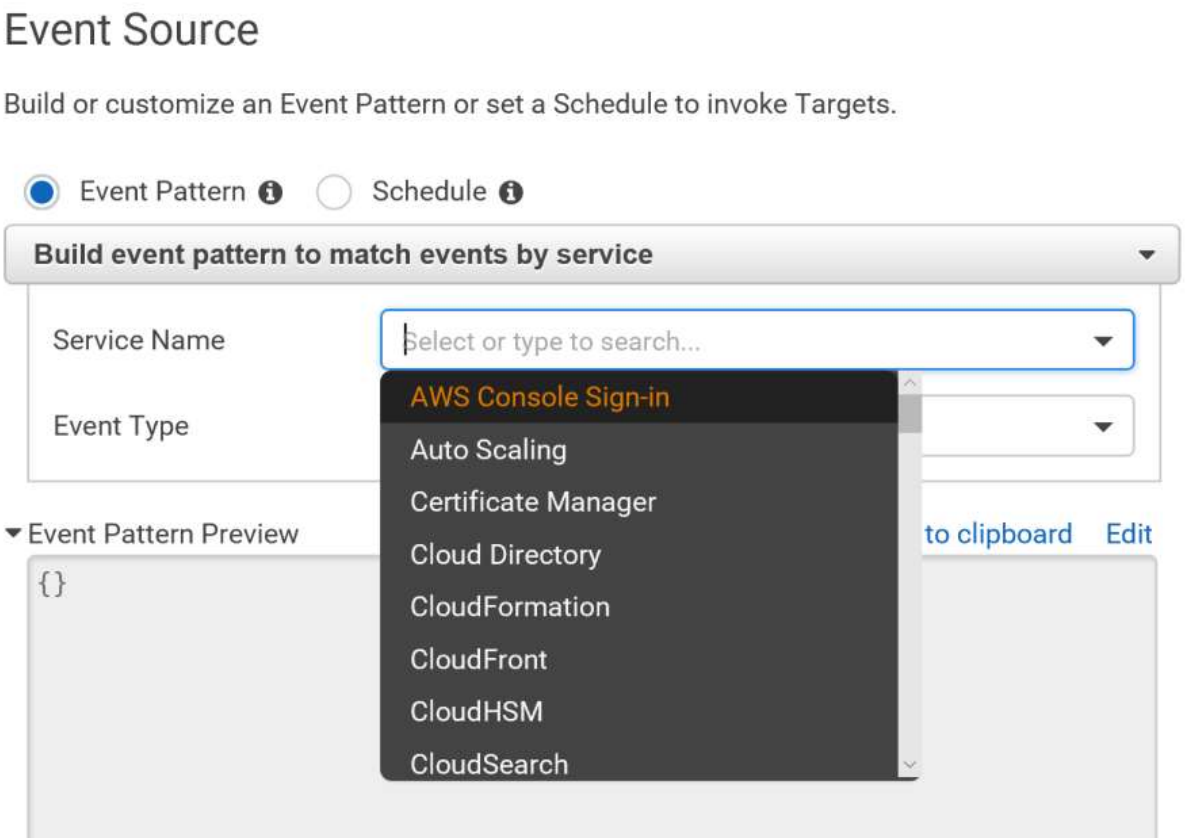
Lambda tool

CloudWatch

- Schedule
  - Event觸發
    - AutoScaling
    - S3
    - ……
  - 定時觸發
    - 可自行定義輸入參數

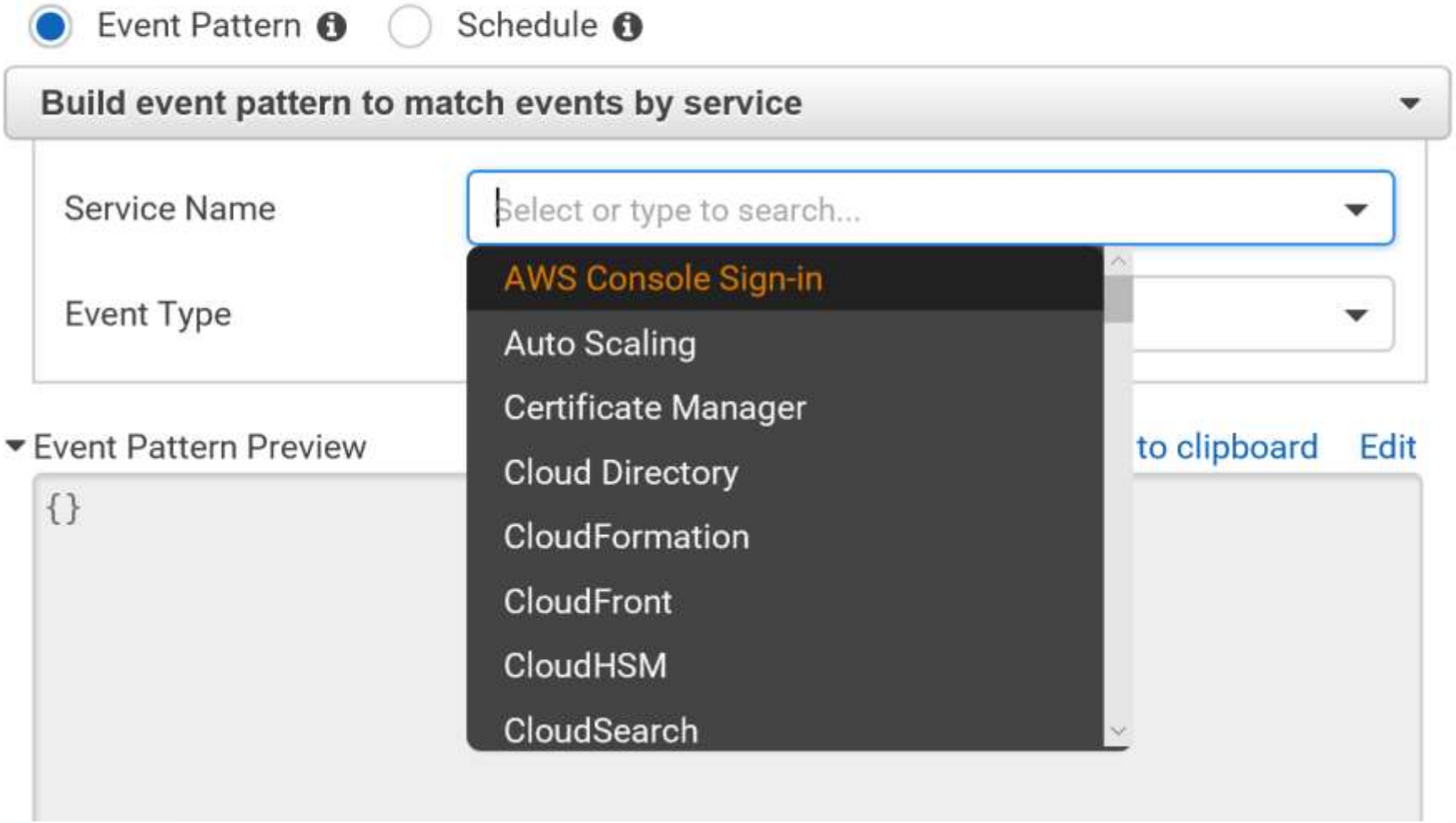
Trigger

- Event



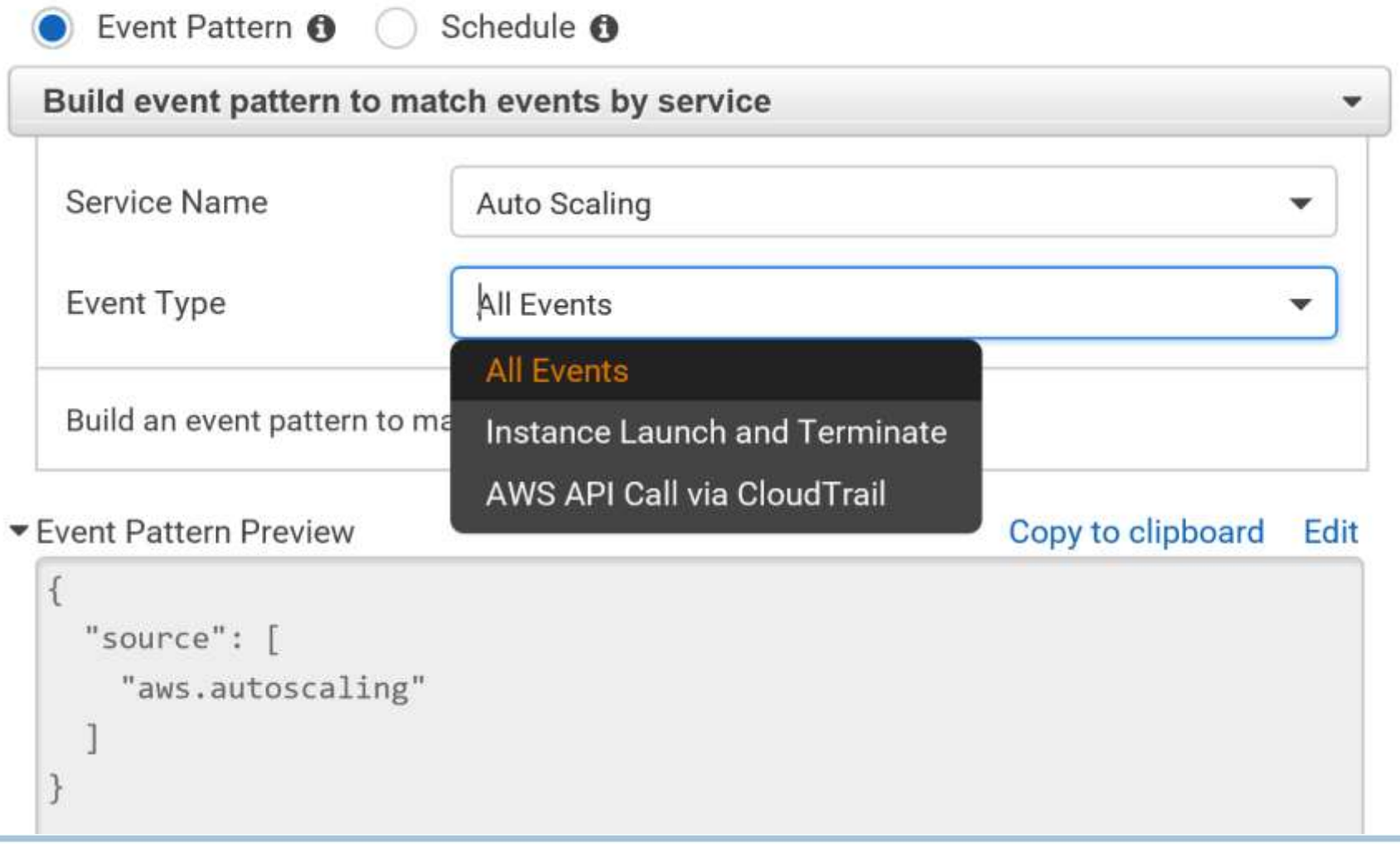
Event Source

Build or customize an Event Pattern or set a Schedule to invoke Targets.



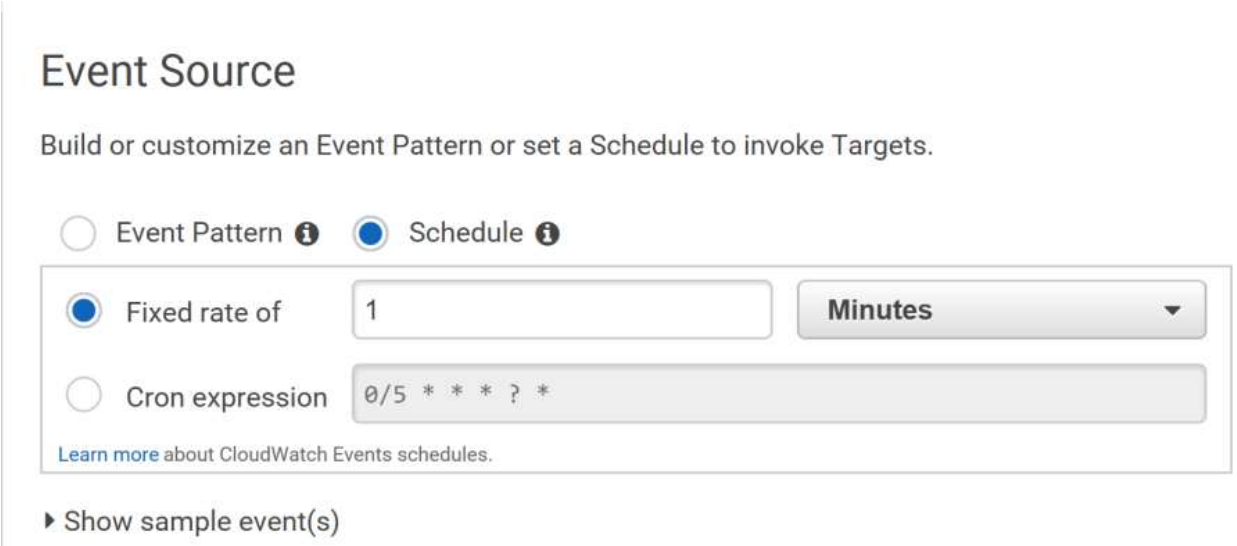
Event Source

Build or customize an Event Pattern or set a Schedule to invoke Targets.

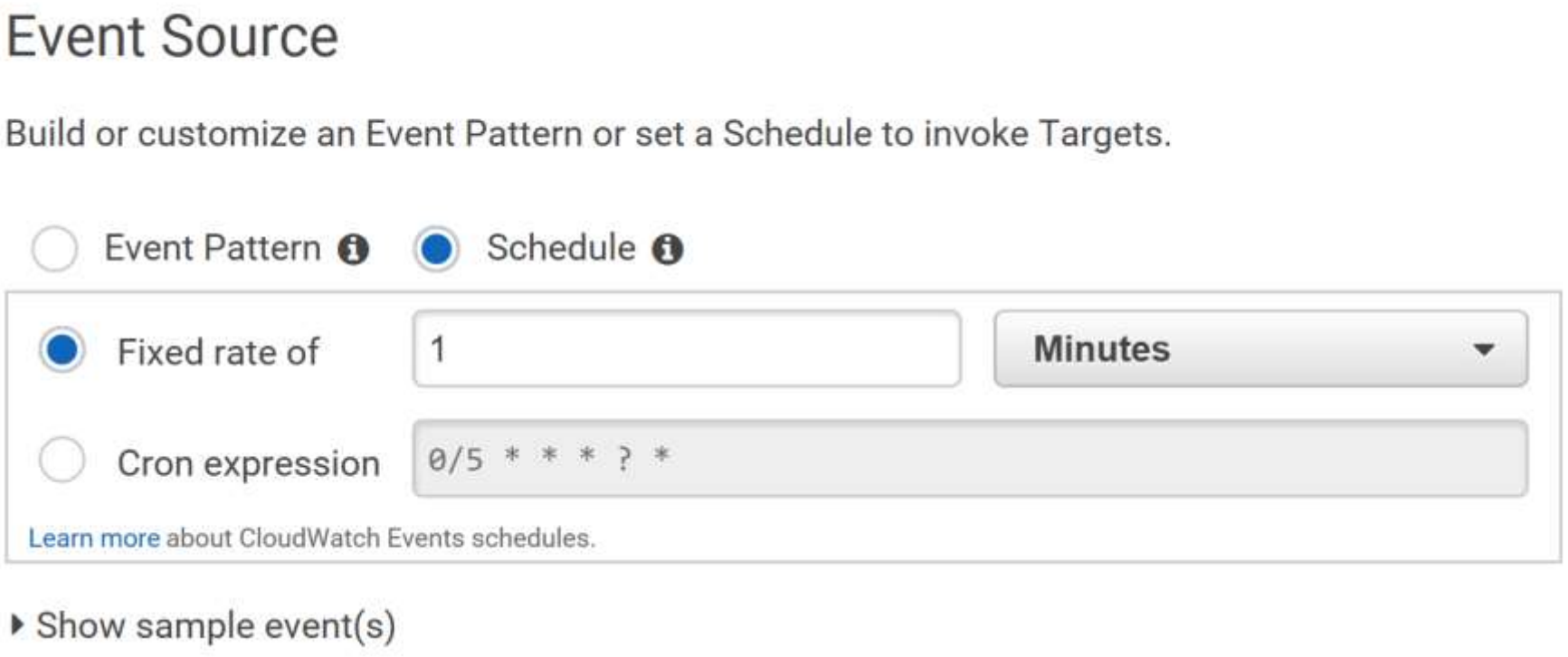


Schedule

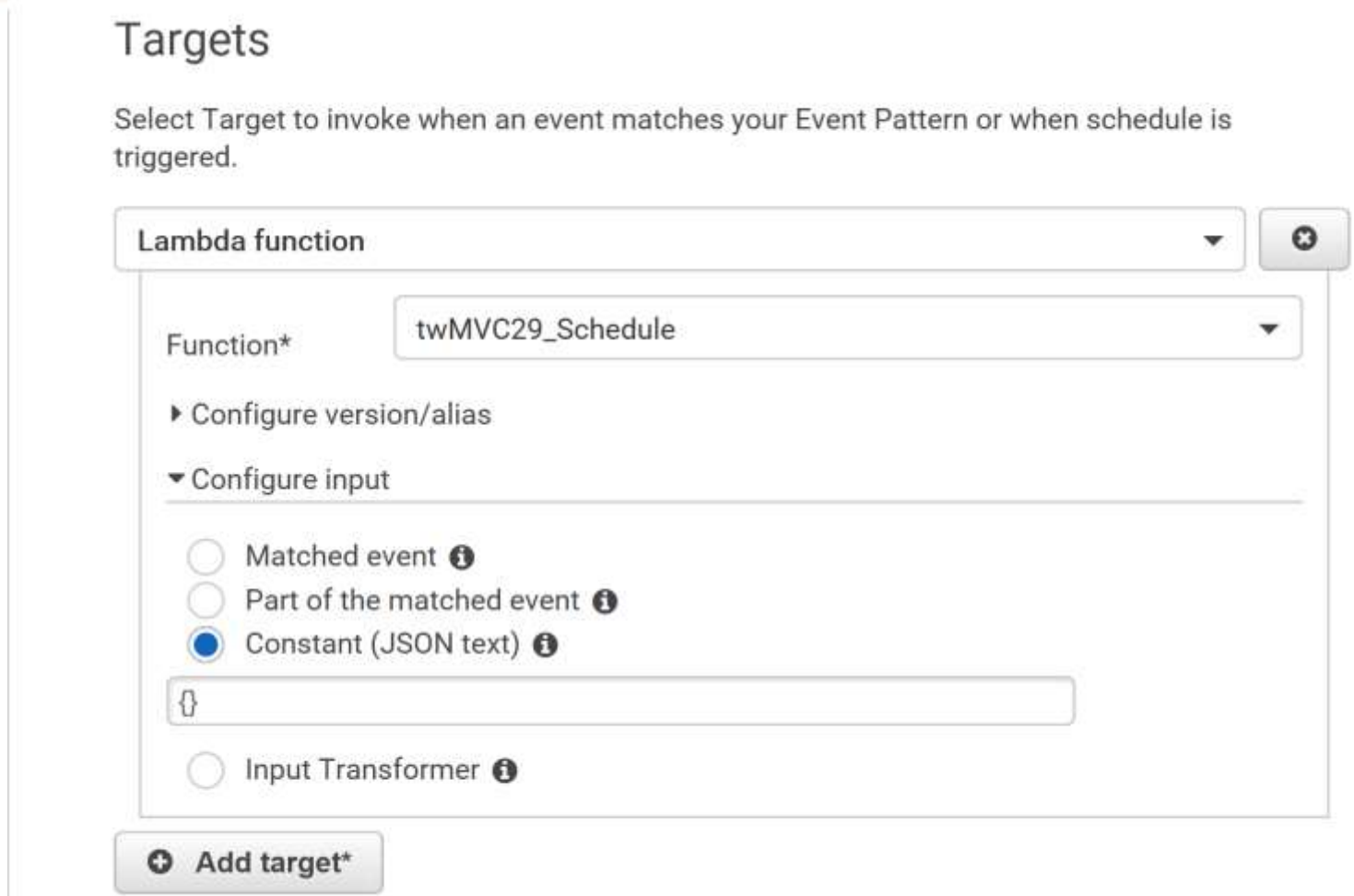
- Fixed
- Cron



Schedule

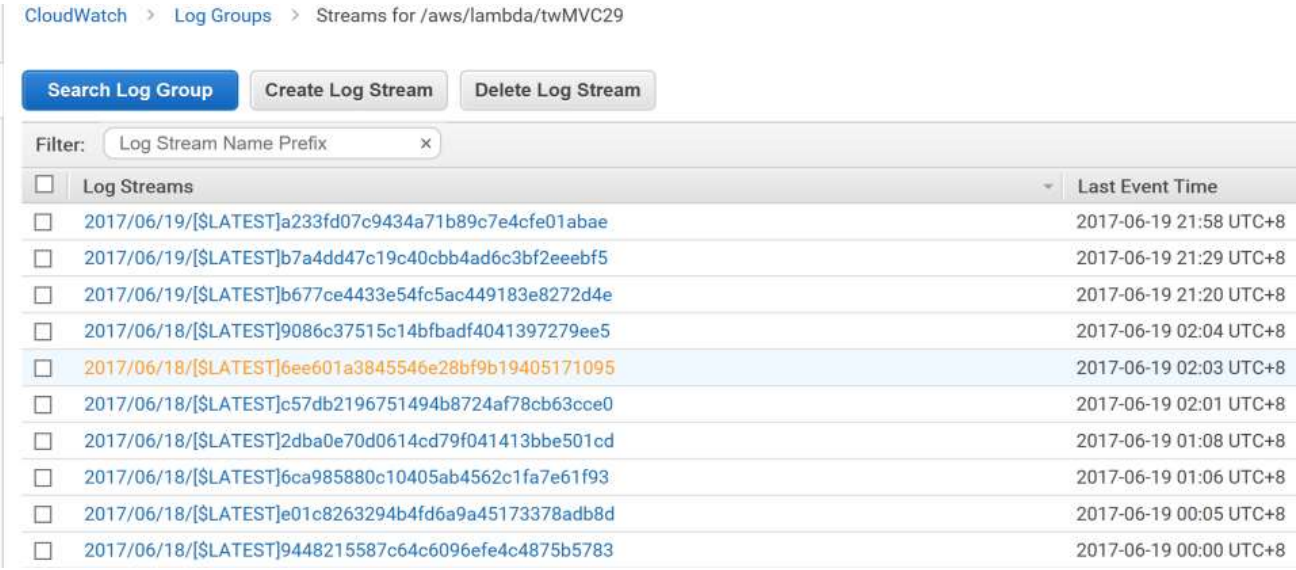

Event Source
Build or customize an Event Pattern or set a Schedule to invoke Targets.
Event Pattern
Schedule
Fixed rate of
1
Minutes
Cron expression
0/5 * * * ? *
Learn more about CloudWatch Events schedules.
Show sample event(s)

輸入參數


Targets
Select Target to invoke when an event matches your Event Pattern or when schedule is triggered.
Lambda function
Function*
twMVC29_Schedule
Configure version/alias
Configure input
Matched event
Part of the matched event
Constant (JSON text)
{}
Input Transformer
Add target*

Lambda Log

- CloudWatch



Log

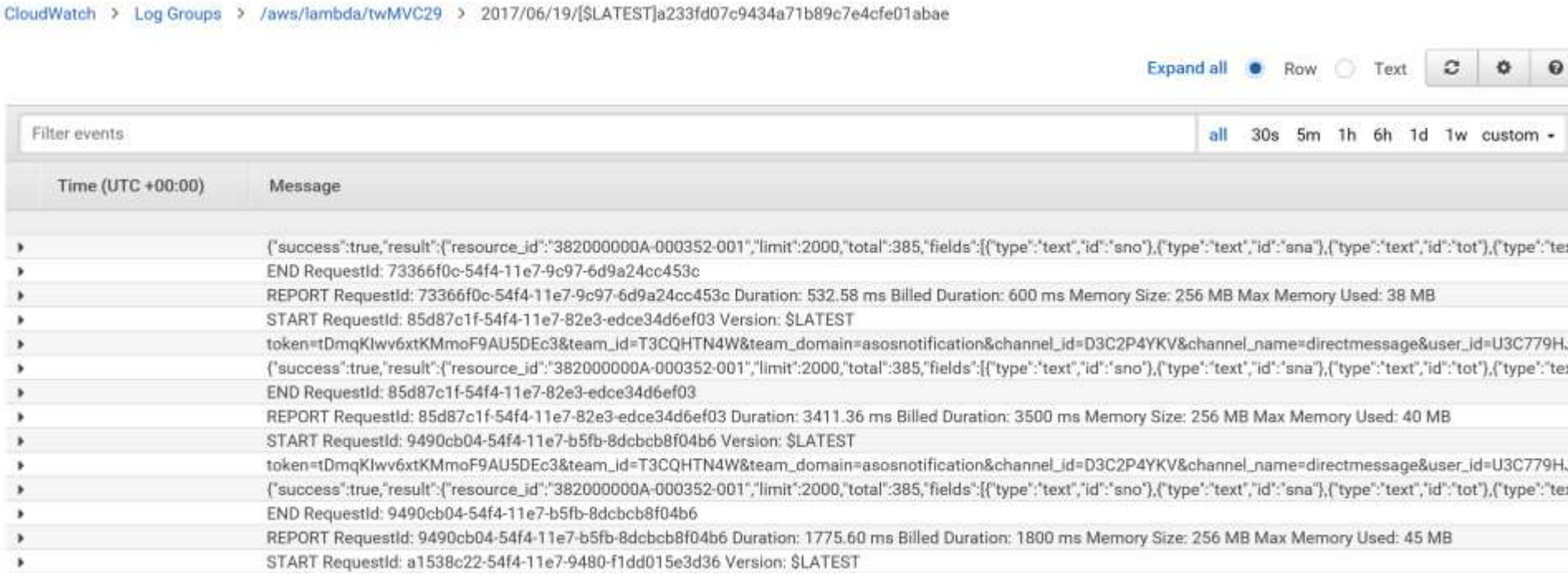
CloudWatch > Log Groups > Streams for /aws/lambda/twMVC29

Search Log Group | Create Log Stream | Delete Log Stream

Filter: Log Stream Name Prefix

Log Streams	Last Event Time
2017/06/19/[$LATEST]a233fd07c9434a71b89c7e4cfe01abae	2017-06-19 21:58 UTC+8
2017/06/19/[$LATEST]b7a4dd47c19c40cbb4ad6c3bf2eeebf5	2017-06-19 21:29 UTC+8
2017/06/19/[$LATEST]b677ce4433e54fc5ac449183e8272d4e	2017-06-19 21:20 UTC+8
2017/06/18/[$LATEST]9086c37515c14bfbadf4041397279ee5	2017-06-19 02:04 UTC+8
2017/06/18/[$LATEST]6ee601a3845546e28bf9b19405171095	2017-06-19 02:03 UTC+8
2017/06/18/[$LATEST]c57db2196751494b8724af78cb63cce0	2017-06-19 02:01 UTC+8
2017/06/18/[$LATEST]2dba0e70d0614cd79f041413bbe501cd	2017-06-19 01:08 UTC+8
2017/06/18/[$LATEST]6ca985880c10405ab4562c1fa7e61f93	2017-06-19 01:06 UTC+8
2017/06/18/[$LATEST]e01c8263294b4fd6a9a45173378adb8d	2017-06-19 00:05 UTC+8
2017/06/18/[$LATEST]9448215587c64c6096efe4c4875b5783	2017-06-19 00:00 UTC+8

Log


CloudWatch > Log Groups > /aws/lambda/twMVC29 > 2017/06/19/[$LATEST]a233fd07c9434a71b89c7e4cfe01abae
Expand all
Row
Text
Filter events
all 30s 5m 1h 6h 1d 1w custom
Time (UTC +00:00)
Message
{"success":true,"result":{"resource_id":"382000000A-000352-001","limit":2000,"total":385,"fields":[{"type":"text","id":"sno"},{"type":"text","id":"sna"},{"type":"text","id":"tot"},{"type":"te
END RequestId: 73366f0c-54f4-11e7-9c97-6d9a24cc453c
REPORT RequestId: 73366f0c-54f4-11e7-9c97-6d9a24cc453c Duration: 532.58 ms Billed Duration: 600 ms Memory Size: 256 MB Max Memory Used: 38 MB
START RequestId: 85d87c1f-54f4-11e7-82e3-edce34d6ef03 Version: $LATEST
token=tDmqKlwv6xtKMmoF9AU5DEc3&team_id=T3CQHTN4W&team_domain=asosnotification&channel_id=D3C2P4YKV&channel_name=directmessage&user_id=U3C779H
{"success":true,"result":{"resource_id":"382000000A-000352-001","limit":2000,"total":385,"fields":[{"type":"text","id":"sno"},{"type":"text","id":"sna"},{"type":"text","id":"tot"},{"type":"te
END RequestId: 85d87c1f-54f4-11e7-82e3-edce34d6ef03
REPORT RequestId: 85d87c1f-54f4-11e7-82e3-edce34d6ef03 Duration: 3411.36 ms Billed Duration: 3500 ms Memory Size: 256 MB Max Memory Used: 40 MB
START RequestId: 9490cb04-54f4-11e7-b5fb-8dcbcb8f04b6 Version: $LATEST
token=tDmqKlwv6xtKMmoF9AU5DEc3&team_id=T3CQHTN4W&team_domain=asosnotification&channel_id=D3C2P4YKV&channel_name=directmessage&user_id=U3C779H
{"success":true,"result":{"resource_id":"382000000A-000352-001","limit":2000,"total":385,"fields":[{"type":"text","id":"sno"},{"type":"text","id":"sna"},{"type":"text","id":"tot"},{"type":"te
END RequestId: 9490cb04-54f4-11e7-b5fb-8dcbcb8f04b6
REPORT RequestId: 9490cb04-54f4-11e7-b5fb-8dcbcb8f04b6 Duration: 1775.60 ms Billed Duration: 1800 ms Memory Size: 256 MB Max Memory Used: 45 MB
START RequestId: a1538c22-54f4-11e7-9480-f1dd015e3d36 Version: $LATEST

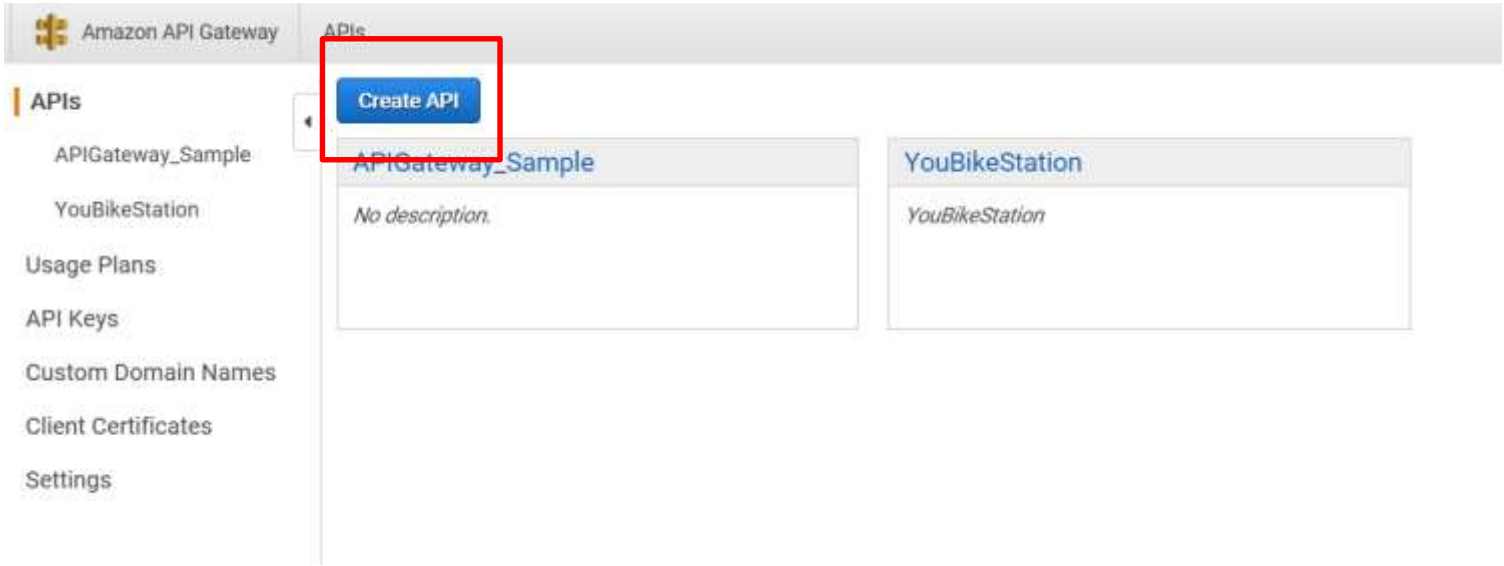


demo

CloudWatch setting

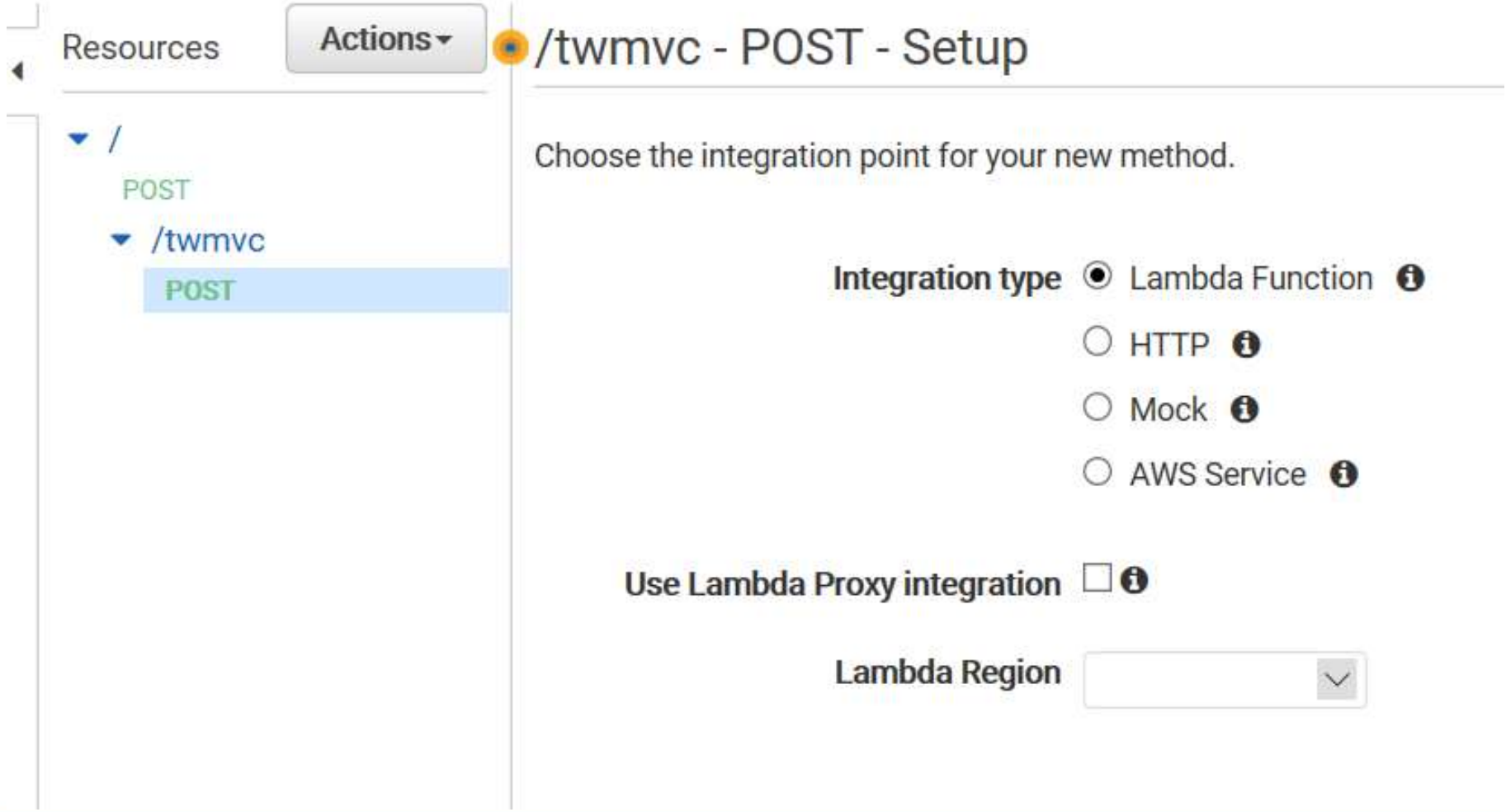
幫Lambda加個URL吧

- API Gateway
  - API



幫Lambda加個URL吧

- API Gateway
  - API
    - Resource
      - Method


Resources
Actions
/twmvc - POST - Setup
/
POST
/twmvc
POST
Choose the integration point for your new method.
Integration type
Lambda Function
HTTP
Mock
AWS Service
Use Lambda Proxy integration
Lambda Region



demo

APIGateway setting

Blog 是記錄知識的最佳平台



https://dotblogs.com.tw

Jetbrains 重構必備工具



ReSharper

https://www.jetbrains.com/resharper/

OzCode 偵錯的魔法師



業界師資、實戰教學



https://skilltree.my

謝謝各位

https://mvc.tw

- 本投影片所包含的商標與文字皆屬原著作者所有。
- 本投影片使用的圖片皆從網路搜尋。
- 本著作係採用姓名標示-非商業性-相同方式分享 3.0 台灣授權。閱讀本授權條款，請到 http://creativecommons.org/licenses/by-nc-sa/3.0/tw/，或寫信至Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.