

國立臺灣海洋大學

資訊工程學系

碩士學位論文

指導教授：蔡國輝

Azure Application Insights對CKAN之監測
The Monitoring of CKAN by Azure Application
Insights

研究生：呂姝嫻 撰

中華民國 111 年 1 月

Azure Application Insights 對 CKAN 之監測

The Monitoring of CKAN by Azure Application Insights

研 究 生：呂姝嫻

Student：Shu-Xian Lu

指導教授：蔡國輝

Advisor：Kuo-Hui Tsai

國立臺灣海洋大學
資訊工程學系
碩士論文

A Thesis

Submitted to the Department of Computer Science and Engineering
College of Electrical Engineering and Computer Science
National Taiwan Ocean University
In Partial Fulfillment of the Requirements
for the Degree of
Master of Science
in
Computer Science and Engineering

January 2022

Keelung, Taiwan, Republic of China

中華民國 111 年 1 月

摘要

現今許多系統往往在開發上線後難以維護，當系統出現異常時，若沒有強大的分析工具可協助診斷問題，對於 DevOps 人員而言將會是一個嚴重的災難。然而一個由 Microsoft Azure 所開發的監測功能 Azure Application Insights，可以有效解決這些問題，它會自動監測系統效能是否異常，並協助系統持續改善效能以及可用性，可適用於各種平台。

本文使用近期興起的開放資料原始碼入口平台 CKAN 作為監測的對象，在 CKAN 原始碼中加入 Azure Application Insights 的 SDK，以及另外實作一個 Azure Application Insight 的插件去監測 CKAN 的內部運作。

因 CKAN 為用 Python 做為基底所開發出來的平台，所以需要透過 Opencensus 提供的 Azure Application Insights SDK，監測 CKAN 的使用情況、回應時間和出現的例外狀況。並能在系統出現異常時記錄錯誤資訊，並及時發信通知管理人員，利用 Azure Application Insights 做圖像化的監控，幫助開發營運維護人員，能夠更有效率地去運用所監控的資料。

關鍵詞：Azure Application Insights、CKAN、自動監測、監控。

Abstract

Many systems today are often difficult to maintain after development and launch. When the system is abnormal, if there is no powerful analysis tool to assist in diagnosing the problem, it will be a serious disaster for DevOps personnel. The monitoring function, Azure Application Insights, can effectively solve these problems. It will automatically monitor whether the system performance is abnormal and help the system to continuously improve performance and availability. It can be applied to various platforms.

This article uses the recently emerging open source code entry platform CKAN as the monitoring object, and adds the SDK of Azure Application Insights to the CKAN source code to monitor the inner workings of CKAN for the benefit of developers using CKAN.

Since CKAN is a platform developed with Python as the base, it is necessary to use the Azure Application Insights SDK provided by Opencensus to monitor the usage ,the response time, and the exception occurs of CKAN. It able to record error information when the system is abnormal, and send a letter to notify managers immediately, Using Azure Application Insights for visual monitoring can help development, operation and maintenance personnel to use the monitored data more efficiently.

Keywords: Azure Application Insights, CKAN, Automatic monitoring, Monitoring

目次

摘要.....	I
Abstract.....	II
目次.....	III
圖目次.....	IV
表目次.....	V
第一章 序論.....	1
1.1 研究背景.....	1
1.2 研究目的.....	1
1.3 論文架構.....	2
第二章 相關系統介紹.....	3
2.1 CKAN.....	3
2.1.1 使用 CKAN 創建 datasets.....	4
2.1.2 CKAN 架構.....	8
2.2 Azure Application Insights	10
2.2.1 Azure Application Insight 遙測資料模型	11
2.3 Opencensus	13
2.4 相關參考文獻.....	13
第三章 系統實作方法.....	16
3.1 實驗環境.....	16
3.2 系統架構.....	17
3.3 實作方法.....	18
第四章 實驗結果.....	19
4.1 使用者頁面瀏覽.....	19
4.2 資料庫紀錄.....	20
4.2.1 資料庫紀錄應用.....	21
4.3 Datasets 和 Resources 狀態紀錄.....	22
4.4 系統效能監測.....	24
4.5 例外狀況紀錄.....	26
第五章 結論.....	27
參考文獻.....	28

圖目次

圖 1. CKAN 標誌	4
圖 2. CKAN 創建 datasets 頁面 1.....	5
圖 3. CKAN 創建 datasets 頁面 2.....	6
圖 4. datasets 資訊頁面	6
圖 5. resource 資訊頁面	7
圖 6. CKAN 模塊化架構	8
圖 7. Azure Application Insights 應用架構圖	10
圖 8. Azure Application Insights 遙測資料模型	11
圖 9. Leitner 提出的監控系統架構.....	14
圖 10. Leitner 提出的監控事件程式結構.....	14
圖 11. TAB Nguyen 基於角色的雲監控模板抽象模型	15
圖 12. 實驗系統架構圖.....	17
圖 13. 實作 Middleware 插件	18
圖 14. setup.py	18
圖 15. CKAN.ini	18
圖 16. 頁面瀏覽紀錄.....	19
圖 17. 使用者登入狀況.....	19
圖 18. 使用者創建 dataset 和 resource.....	20
圖 19. 資料庫執行紀錄.....	20
圖 20. 所紀錄到的 Postgresql 資料語法	20
圖 21. Postgresql 平均回應時間圖表生成語法	21
圖 22. Postgresql 要求計數分析圖	21
圖 23. Postgresql 平均回應時間	22
圖 24. 剛創建的 resource 詳細資料	22
圖 25. 創建完後的 Insight 回傳資料	23
圖 26. 更改完後的 Insight 回傳資料	23
圖 27. 刪除完後的 Insight 回傳資料	23
圖 28. 系統回應時間.....	24
圖 29. 伺服器要求成功計數.....	25
圖 30. 伺服器要求失敗計數.....	25
圖 31. Trace 資料模型內的 Exception	26

表目次

表 1.	Azure Application Insight API 總表	12
表 2.	Opencensus sdk 對應 Azure Monitor 遙測類型	13
表 3.	CKAN 安裝項目依賴項	16

第一章 序論

1.1 研究背景

隨著時代的變遷，人們的生活越來越離不開網際網路，在生活中不管是要採買物品、影音瀏覽、使用社群網站、線上申辦...等，都需要提供服務的那方提供不同的應用程式在服務器上，如果應用程式一旦發生故障，沒有做及時的處理，對提供服務的那方都會造成不小的財務損失或是相關程序上的錯誤，為了能比一般使用者能及時發現錯誤，所以才有了應用程式效能管理(APM)服務的出現。

應用程式效能管理服務，透過跟應用程式的連結去監控應用程式，並在應用程式運行時，能夠及時回傳我們所想的監控數值，在監控數值達到一定標準時，及時發送事件給開發營運人員，讓營運方能夠預防和即時處理應用程序的故障，以此去將損失降到最小。

1.2 研究目的

當一些應用程式在開發時，會根據他的某些功能，有時會需要使用到所謂的 Open Data，CKAN 在當中就扮演了重要的角色，他時常被政府以及民間相關單位擴展為該組織 Open Data 的入口平台，提供使用者複製連結或者是使用 API，去使用該開放的資料。

但是現在除了 CKAN 使用 Python 的 logging module 會去紀錄訊息外，目前尚未看到有應用程式效能管理服務可以去監控以 CKAN 擴展的應用程式，相對的在開發擴展 CKAN 的過程中，如果可以有效的監控一些重要數值，對使用 CKAN 開發的人員將會更得心應手。

所以在本論文中，希望能透過目前由 Microsoft 所提出的應用程式效能管理服務 Azure Application Insight 去監控 CKAN，Azure Application Insight 可監控由各個不同平台所建構的應用程式，在不同的平台中包括了.NET、Node.js、Python，而 CKAN 正是由 Python 平台所開發的應用程式，在 Azure Application Insight 中去監控由 Python 建置的應用會有比較多的彈性，可讓應用程式開發人員自定義所要監控的數值，協助開發人員能更快速的了解應用程式出現錯誤的問題所在，有益對 CKAN 擴展的開發。

研究的目的是在於要如何有效的將 Azure Application Insight 嵌入於 CKAN 中，且得到有用的監控數據，讓 Azure Application Insight 依據這些數值回傳應用程式的相關報告或是警告。

1.3 論文架構

本篇論文中涵蓋五個章節，第一章為序論，主要為介紹本文的研究背景以及目的；第二章相關系統介紹，介紹本文所使用到的相關系統以及相關的參考，有 CKAN、Azure Application Insight 以及 Opencensus；第三章系統實作方法，為介紹本文系統建置的環境、架構以及實作方法；第四章實驗觀察，在 Azure Application Insight 中所觀察到的數值以及之間的因果關係；第五章結論，介紹結論以及未來研究的方向。

第二章 相關系統介紹

此章講述了本文所使用到的系統，一開始先介紹 CKAN，因為 CKAN 為要監控的對象，所以會再去了解 CKAN 的核心功能以及他的架構，再來為介紹監測工具 Azure Application Insight，以及去了解它的遙測資料模型，以及會介紹 Opencensus 並概述它提供給 Azure Monitor 的匯出工具，最後提供以及介紹相關的參考文獻。

2.1 CKAN

隨著最近開放數據增加，用於發佈、管理和共享開放數據的平台引起了廣泛的關注，CKAN(Comprehensive Knowledge Archive Network)是一個 Open Data 的入口網站和擴展平台，由 Open Knowledge Foundation 維護[1][2]，幫助管理和發佈數據，也是一個數據分發平台，CKAN 一般被應用在可以收集到大量資料的國家、地方政府、研究機關，或是其他組織，現今在全球的應用有將近 60 多個例子[3]，例如：美國政府入口網站 data.gov、英國政府入口網站 data.gov.uk 和歐洲 open data 入口網站 europeandataportal.eu[4]，CKAN 被那麼廣泛的使用主要有四個原因[5]，首先是因為它為一個開源的平台，再來是它對數據管理功能相較方便許多，第三，是它有與社群相關的各種支持，以及有許多開發人員的參與，最後，它支持數據提供者和使用者的各種功能。

在 CKAN 中，用戶分為數據提供者和數據使用者，這裡的數據使用者指的是在組織內和組織外的任何用戶，只有包含在組織內的數據提供者，才可以將資料上傳[6]，而數據提供者主要是國家和地方政府直接透過 Web 界面和 API 發佈數據，一旦數據提供者將數據發佈後，數據使用者就可以用搜尋功能去查找所需要的資料，並通過 API 從 CKAN Server 導入數據做使用，或是用表格亦或是圖形功能預覽資料，CKAN 還有提供使用者方便的基礎分類系統去搜尋開放數據集，它的基礎分類系統包括組織、組、標籤、格式和許可證。

這裡的資料作為一個 datasets 的單位去發佈，datasets 為許多 resource 組合而成，例如：某個地區的總人口成長總計、某部電影的票房成長數據等，使用者在搜尋時，看到的搜尋結果會是以單一個 datasets 作呈現，而 resource 保存的格式不限，可以是 CVS、Excel、XML、PDF、JPG...等，可在系統內部儲存，也可以將 resource 存為一個網路連結，然而在一個 datasets 中可以包含任意數量的 resource。

除此之外 CKAN 因應開發者需求的不同，利用擴展功能，讓開發者能根據自己的需求擴展 CKAN，它擁有 230 多個註冊為擴展的功能[5]，包括可視化、

文檔預覽、自定義主題、其他存儲、站點鏈接和元數據管理，有代表性的實例包括用於鏈接數據的 DCAT 擴展，還有用於數據收集的 CKAN Harvest 和 DCAT[7] RDF / JSON Harvest，以及儲存 DB 的擴展用於 MongoDB 或 Amazon S3。



圖 1. CKAN 標誌

資料來源:CKAN 官網

2.1.1 使用 CKAN 創建 datasets

要使用 CKAN 上傳 datasets，使用者首先要去做的事情就是在 CKAN 上註冊，便成為組織的成員後，即可開始使用 CKAN 大部分的功能，若使用者沒有加入組織，則在授權與組織相關資料及功能將無法使用[6]，除了在搜尋和查找 datasets 不需要登入以外，CKAN 中所有跟發佈有關的功能都需要使用者登入系統，登入後會依據使用者在組織中的權限，可以使用創建、編輯和刪除 datasets 等功能，以下簡單介紹了 CKAN 創建 datasets 的步驟。

在創建 datasets 時，CKAN 會先要求用戶填入以下訊息[6]:

- Title:添加 datasets 的標題，此標題在 CKAN 中為獨一無二的，所以通常命名上希望能簡短扼要。
- Description:在此處添加對 datasets 的更詳細的描述，包括數據來自何處，以及在使用數據時需要了解的注意事項等訊息。
- Tags:此標籤能幫助用戶找到 datasets，並將其與其他相關 datasets 鏈接。
- License:此欄位包含重要的許可證訊息，以便用戶知道他們可以如何使用該 datasets。
- Organization:如果用戶是任何組織的成員，此下拉選單將使用戶能夠選擇應該擁有該的組織。
- Visibility:決定該 datasets 是否為可公開的，則代表在該站點上的任何用戶都可以搜尋的到，或是要設為組織內的私有 datasets，只允許組織內的成員去做取用。

剩下較為繁瑣的為添加 datasets 的作者、作者電子信箱以及相關自定義的訊息等。

圖 2. CKAN 創建 datasets 頁面 1

在填寫完上面的訊息按下一步後，CKAN 才會開始要用戶添加包含此 datasets 的 resource，可以給 CKAN 一個連結，然後選擇要連結至文件還是連結至 API，或是要將本地端的文件傳送至 CKAN，就選擇上傳文件，在該頁面上還可以選擇添加一些 resource 的相關訊息，完成此步驟後即完成 datasets 的創建。

圖 3. CKAN 創建 datasets 頁面 2

完成 datasets 的創建後，除了能查看 datasets 的資訊頁面外，也可以查看到其內包含的 resource 資訊，其中這裡的 Id 為此 resource 在 CKAN 系統中的識別，Package_id 則為 datasets 在 CKAN 系統中的識別。

全國COVID-19公費疫苗接種院所清單

提供全國COVID-19公費疫苗接種院所清單

Data and Resources

covid19_vac_hosp.csv
院所名稱、縣市、鄉鎮、疫苗種類名稱、醫事機構代碼、地址、電話

Additional Info

Field	Value
State	active
Last Updated	January 7, 2022, 1:24 AM (UTC+08:00)
Created	January 7, 2022, 1:20 AM (UTC+08:00)

圖 4. datasets 資訊頁面

[Datasets](#)[Organizations](#)[Groups](#)[About](#)

Search

[Organizations](#) / [Neogence](#) / [全國COVID-19公費疫苗接種院所清單](#) / covid19_vac_hosp.csv

covid19_vac_hosp.csv

[Manage](#)[Download](#)

URL: http://127.0.0.1:8081/dataset/d2b52d7b-fc83-4130-ae5d-cd3bd839a323/resource/22b1c849-b10e-4933-99a1-2f3b62c64c78/download/covid19_vac_hosp.csv

院所名稱、縣市、鄉鎮、疫苗種類名稱、醫事機構代碼、地址、電話

There are no views created for this resource yet.

Not seeing the views you were expecting? [Click here for more information.](#)

Resources

covid19_vac_hosp.csv

Social

Twitter

Facebook

Additional Information

Field	Value
Data last updated	January 6, 2022
Metadata last updated	January 6, 2022
Created	January 6, 2022
Format	CSV
License	No License Provided
Id	22b1c849-b10e-4933-99a1-2f3b62c64c78
Mimetype	text/csv
On same domain	True
Package id	d2b52d7b-fc83-4130-ae5d-cd3bd839a323
Size	739 KiB
State	active
Uri type	upload
Hide	

圖 5. resource 資訊頁面

2.1.2 CKAN 架構

了解 CKAN 的架構，有助於我們了解要如何用 Azure Application Insight 去監測他，他是以 Python 為基礎的 Web 應用程式，內部結構採用 Pylons Web Framework，以 PostgreSQL[8]為資料庫，Apache Solr[9]為他的搜尋引擎，Jinja2[10]為視圖模板，以模塊化的架構去理解 CKAN[11]，他能被分為 Controller、Model、View 這些層面。

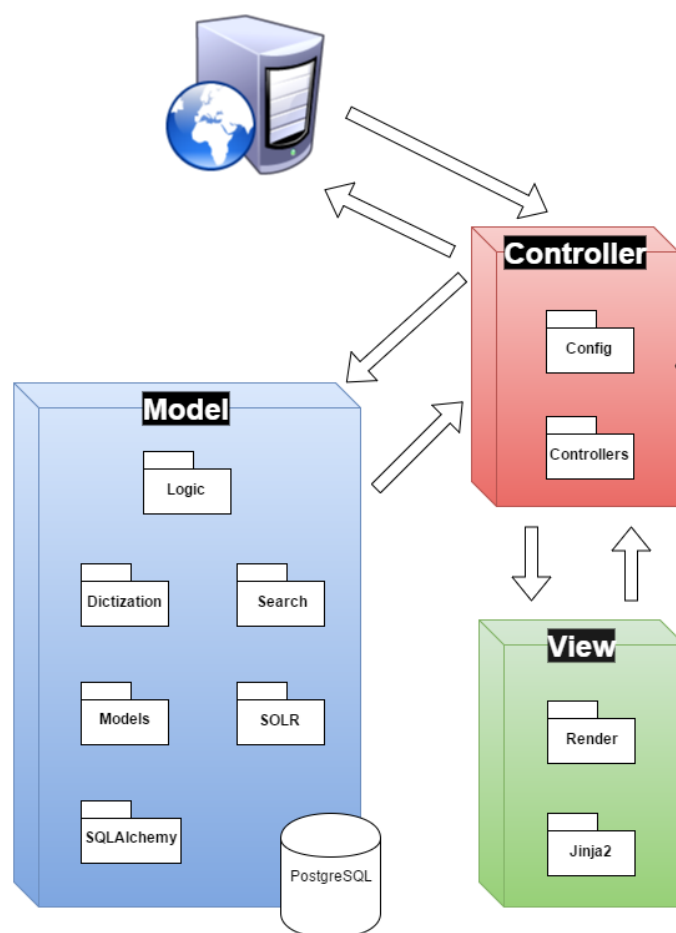


圖 6. CKAN 模塊化架構

資料來源:

<https://delftswa.gitbooks.io/desosa2016/content/CKAN/chapter.html>

1. 控制層(Controller)

當用戶有個 HTTP request 下來時，控制層會將 request 攔截，由 Config 模塊負責將 Middleware 應用於 request，這可以因應 request 的不同去添加其他的功能，並去匹配是否有相應的 Controller，在 request 中儲存要調用的 Controller 訊息，調用時在 Controller 模塊中去使用相應的 class。

2. 模塊層(Model)

此層是三個層中最為複雜的，裡面包含了對 datasets、組織和用戶執行操作的重要模塊，Logic 模塊包含了動作功能、認證功能、後臺任務和業務邏輯，他接受了從 Controller 到訪問數據的呼叫，最後這些數據會被存為 atomic values 至關係數據庫 Postgresql 內，這讓在使用面向物件的編程語言時，難以保留物件存在的概念，所以由 SQLAlchemy[12]去負責物件至關係數據庫的轉換，反之則使用 Dictization 和 Models 模塊去轉換，搜索功能則是通過 Search 模塊，他會去調用執行實際搜索 Apache Solr。

3. 視圖層(View)

由 Render 模塊負責幫來自 Model 層的 request 生成模板，並且導入 Jinja2 模板語言。

2.2 Azure Application Insights

此為 Azure Monitor 中的一項功能，一種應用程式管理效能服務[13][14]，適用於開發以及維護人員，主要功能為協助開發維護人員即時監視應用程式，可以自動監測程式是否異常，以及協助診斷問題，提升其程式的效能和可用性，適用於在各種程式語言上的應用程式。

圖 7 為 Azure Application Insights[15]應用架構，其中左側區域為我們所監控的應用程式，我們需要在該應用程式中安裝 Azure Application Insights 的 SDK，此會檢測我們的應用程式，並且使用檢測金鑰 GUID 註冊後，將遙測資料回傳至 Azure，監測的應用程式並不一定要在 Azure 上執行，也可以是本機中正在開發的應用程式，Azure Application Insights 可監測的不只是 Web 服務的應用程式，也包括了任何的背景服務和網頁中的 JavaScript 等。

當要查看遙測資料時，登入 Azure 的入口網站，並且查找 Azure Application Insights 資源概觀上的 LogAnalysis[16]，LogAnalysis 主要用於查詢和圖像化 Application Insights 中的數據，它使用類似於 SQL 的 Kusto 查詢語言[17]，並且只支持 read-only 的查詢，除了用 LogAnalysis 做分析和查看資料，也可以使用例如 Power BI[18]或是 REST API[19]等方式將資料做分析和匯出。

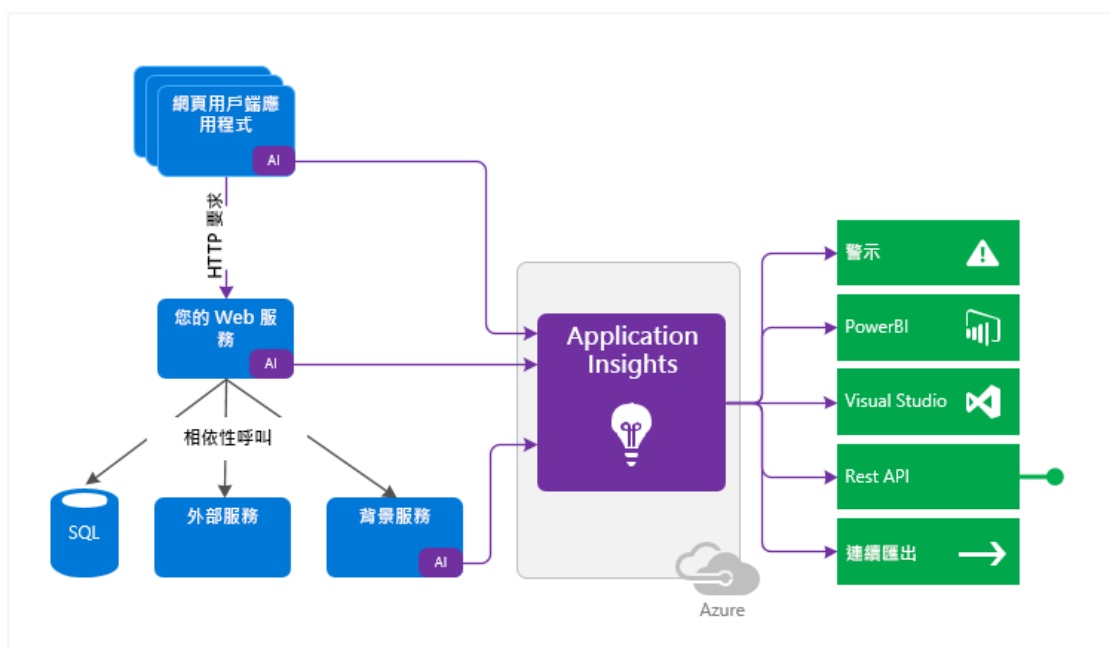


圖 7. Azure Application Insights 應用架構圖

資料來源:Microsoft Azure 監視器文件

2.2.1 Azure Application Insight 遙測資料模型

Azure Application Insight 將遙測數據從應用程式發送到 Azure portal，所回傳的遙測資料經過標準化後，分成六種類型[20]，Request、Exception、Dependency、Trace、Metric、Event，其中 Request、Exception、Dependency，SDK 會透過 Web 應用程式自動收集這三種資料模型，其餘三種則是提供自訂遙測的資料模型，Azure Application Insights 收集的數據模擬了這種典型的應用程序執行模式，如圖 8。

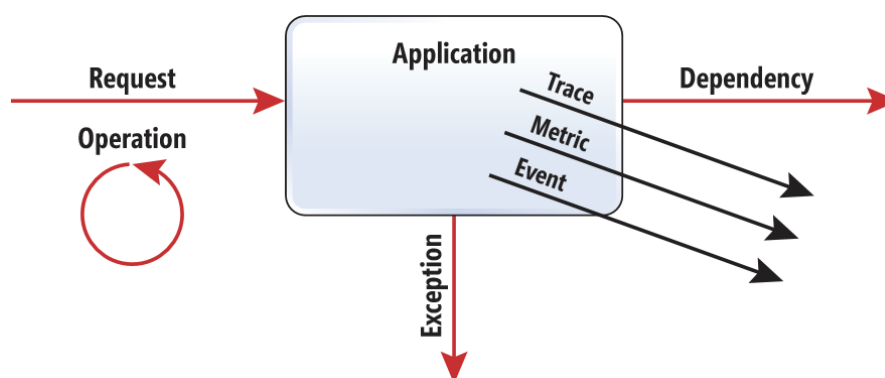


圖 8. Azure Application Insights 遙測資料模型

資料來源:Microsoft Azure 監視器文件

在自動收集的資料模型中：

- Request:為記錄該監測中的應用程式所有接收的要求，代表著外部要求對應用程式所觸發的執行順序，在回傳的記錄中，每一個所執行過的要求，都包含它的唯一識別值 ID，以及包含了所有查詢字串參數要求的 url，和 name 為要處理這項要求所產生的程式碼路徑，source 指的為要求的來源，其中裡面包括了呼叫端的檢測金鑰或是 IP，以及 duration 該欄為必要欄位，還有其他的欄位為 Response code、Success、Custom properties、Custom measurements。
- Dependency:代表外部元件和應用程式的互動狀況，如應用程式和資料庫之間所收集的資料就為該資料模型，在該模型的欄位中 name 為使用此相依性呼叫所起始之命令的名稱，type 表示該相依性類型，target 表示該應用程式的外部元件名稱，還有其他的欄位為 duration、Result code、Success、Custom properties、Custom measurements。
- Exception:表示應用程式在被監視期間發生的已經處理或是還沒處理的異常，其中包含的欄位有 Problem Id 表示在代碼中發生異常的位置，Severity level、Exception details、Custom properties、Custom measurements。

在自訂遙測的資料模型中：

- Trace:它提供我們以自己熟悉的檢測架構去實作診斷，欄位有 Message、Severity level 和 Custom properties。
- Event:通常用於擷取使用者與應用程式的互動，有助於分析使用者模式，欄位有 name 代表自訂的事件名稱，還有 Custom properties 通常用於擴展遙測的資料，例如，擴展一個為訂單編號的欄位、以及 Custom measurements。
- Metric:則用於定期的報告檢測。

在這些遙測的資料模型中，則需要在監測的應用程式中依監測所需要的類型插入 API[21]，再使用 SDK 傳送自訂的遙測資料，例如自訂的事件以及數值至 Azure 中，以下表 1 為 API 的類型與用法。

Method	Used for
TrackPageView	Pages, screens, blades, or forms.
TrackEvent	User actions and other events. Used to track user behavior or to monitor performance.
GetMetric	Zero and multi-dimensional metrics, centrally configured aggregation, C# only.
TrackMetric	Performance measurements such as queue lengths not related to specific events.
TrackException	Logging exceptions for diagnosis. Trace where they occur in relation to other events and examine stack traces.
TrackRequest	Logging the frequency and duration of server requests for performance analysis.
TrackTrace	Resource Diagnostic log messages. You can also capture third-party logs.
TrackDependency	Logging the duration and frequency of calls to external components that your app depends on.

表 1. Azure Application Insight API 總表

資料來源:Microsoft Azure 監視器文件

2.3 Opencensus

針對 Azure 對 Python 應用程式的監測，是透過 Opencensus Python SDK，Opencensus[22]是一組開放原始碼的程式庫，協助收集遙測數據以及分散式追蹤，再將收集的資料傳至後端，這裡的後端除了 Azure Monitor 以外，還有 Datedog、Instana、Jaeger...等，我們透過 Opencensus 來整合 Azure Monitor 對 Python 應用程式的分散式追蹤，以及其他數據的收集。

Opencensus 的 SDK 提供給 Azure Monitor 三種匯出工具，將不同類型的遙測資料傳送至 Azure Application Insight，分別為 Log、Metrics、Tracing，這三種分別對應著不同類型的遙測資料模型，對應內容請查看表 2[23]，在其中 Opencensus 沒辦法自動傳送例外處理遙測，必須透過 Log 來轉送，這部分需要使用者自定義追蹤以及例外狀況。

Opencensus	Azure Monitor 的遙測類型	說明
Log	Trace Exception Event	記錄遙測、例外狀況遙測、使用者自訂義事件遙測
Metrics	Metrics performanceCounters	使用者自訂義計量、效能計數
Tracing	Requests Dependency	傳入要求、傳出要求、相依性

表 2. Opencensus sdk 對應 Azure Monitor 遙測類型

資料來源:Microsoft Azure 監視器文件

2.4 相關參考文獻

在 Leitner 的論文[24]中提出了一種基於事件的雲應用監控架構，再將這架構與 CEP[25]的概念結合，在他們的監控系統中，如圖 9 所示，各種相關的元件都可以發送事件，這些事件會去明確的指示當前雲應用的狀態，並使用 CEP 技術將產生的事件與其他監控事件做相關聯，將其處理為更高級別的資訊，再去定義相關的監控指標，而系統中所產生的事件主要會在三個層次中有相互的關聯，分別為在 host、resource 和 metric 中，並將監控事件的程式結構分為二種，為預定義和自定義，有 15 個重要預定義的事件，如圖 10 所示，該架構允許應用程式開發人員指定和監控高級應用程式性能指標，而不像是 CPU 的利用率、內存消耗和網路頻寬，這些則是在工業中比較注重的指標，雖然他們跟監控的應用程式有相關，但是沒有辦法像自定義的高級應用程式性能指標可以反應應用程式的實際效能。

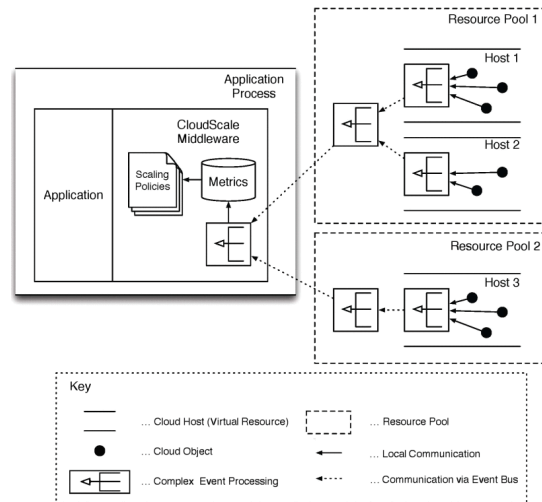


圖 9. Leitner 提出的監控系統架構

資料來源: <https://ieeexplore.ieee.org/abstract/document/6449437>

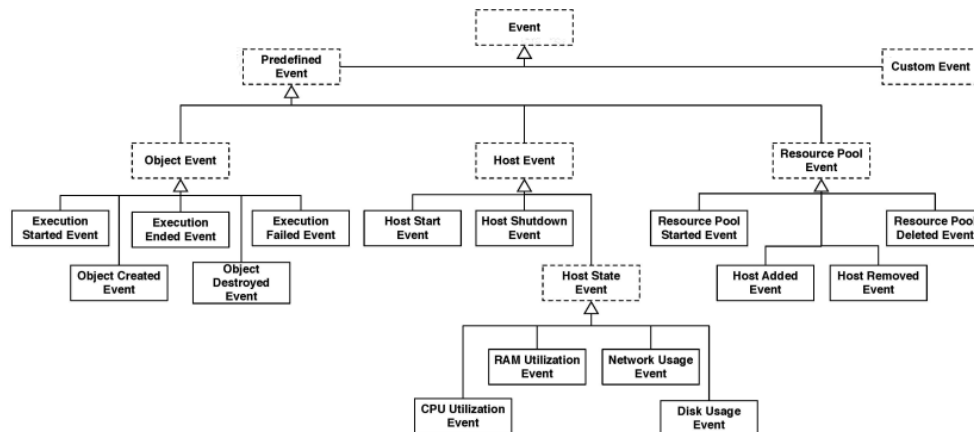


圖 10. Leitner 提出的監控事件程式結構

資料來源: <https://ieeexplore.ieee.org/abstract/document/6449437>

而在 Siebenhaar 的論文[26]中，設計出了一個基於角色的模板去擴展基於 model 的監控，如圖 11 所示，因應參與系統的角色不同，需要的監控需求也會有所不同，所以將需要考慮到參與者的角色去做監控，在雲監控中有三種角色，分別為最終用戶、開發人員和基礎設施營運商，先依照不同的角色去創建模板後，將模板儲存於 database 中，當對應的角色使用系統時，查找其模板給該用戶使用，該模板的設計分為三個部分，第一為 Cloud Endpoint 其包含雲服務的監控層和監控單元的部屬相關訊息，第二為 Monitoring Variable 裡面包含可直接測量的簡單變量和需要多個簡單變量去做組合計算的複合變量，這些變量都由用戶自己去指定，最後的部分 Monitoring Policy 則是去定義監控規則以及後續的操作，TAB Nguyen 在最後將該角色模板與用於雲監控的代理去做結合，達到在雲監控中依照角色去決定監控的數據流量，去減少需要傳輸的數據量，以達到降低監控的複雜性。

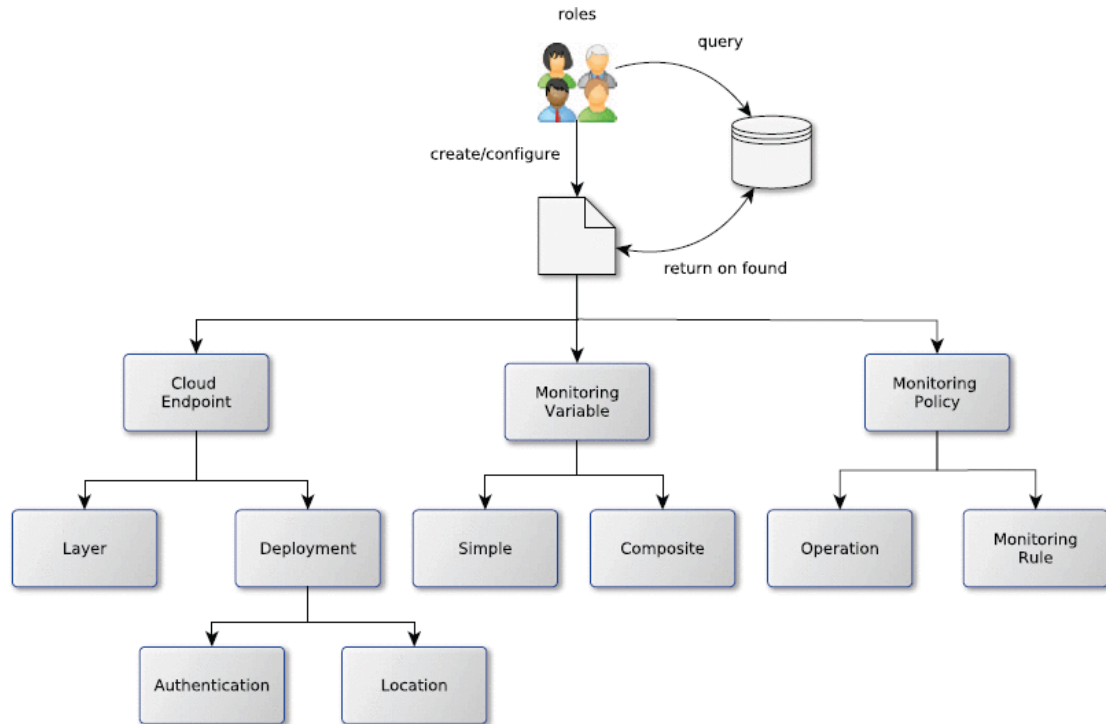


圖 11. TAB Nguyen 基於角色的雲監控模板抽象模型

資料來源: <https://ieeexplore.ieee.org/abstract/document/7027500>

在以上這兩篇論文中，Azure 也包含了 CEP 的技術，Azure 的 CEP 別名為 Stream Analytics[27](串流分析作業)，Azure Application Insight 可將資料匯出至 Azure storage 後，再使用 Stream Analytics 做資料的過濾，再過濾時就能夠以使用者的角色去過濾，去達到各個角色的監控需求。

第三章 系統實作方法

在此章中為介紹實驗時的環境，介紹整個監控系統的架構，最後介紹實作的方法。

3.1 實驗環境

依照 CKAN 文件，使用 source 安裝 CKAN 版本 2.9.2 於虛擬機，以及額外安裝 Opencensus Azure Monitor 匯出工具，並使用 Nginx 做部屬 CKAN 網站，虛擬機的作業系統為 Ubuntu20.04，安裝 CKAN 時需要安裝的依賴項目介紹以及版本為以下表 3[3]內容：

Package	Description
Python	The Python programming language, v3.6 or newer (or v2.7)
PostgreSQL	The PostgreSQL database system, v9.5 or newer
libpq	The C programmer's interface to PostgreSQL
pip	A tool for installing and managing Python packages
python3-venv	The Python3 virtual environment builder (or for Python 2 use 'virtualenv' instead)
Git	A distributed version control system
Apache Solr	A search platform
Jetty	An HTTP server (used for Solr).
OpenJDK JDK	The Java Development Kit (used by Jetty)
Redis	An in-memory data structure store
Flask	a micro web framework written in Python,v1.1.2

表 3. CKAN 安裝項目依賴項

資料來源:CKAN 技術文件

3.2 系統架構

系統架構如圖 12，系統用 Nginx 部屬，但 CKAN 為以 Python 作為基底開發的系統，所以今天當一個用戶向 CKAN 發送一個 request 時，因為 Nginx 沒辦法實現 WSGI[28]的規範，所以來自用戶的請求會先被封裝成 WSGI 的形式，再由 uWSGI server 轉送給 WSGI Middleware 之後再轉發給 CKAN，WSGI 為 Python 標準的網頁伺服器介面協定，如果要用 Azure Application Insights 去遙測 request 的話，則需要設置一個屬於 Azure Application Insights 的 Middleware,讓這個 Middleware 去監測所有來自用戶的 request，之後上傳至 Azure Application Insight

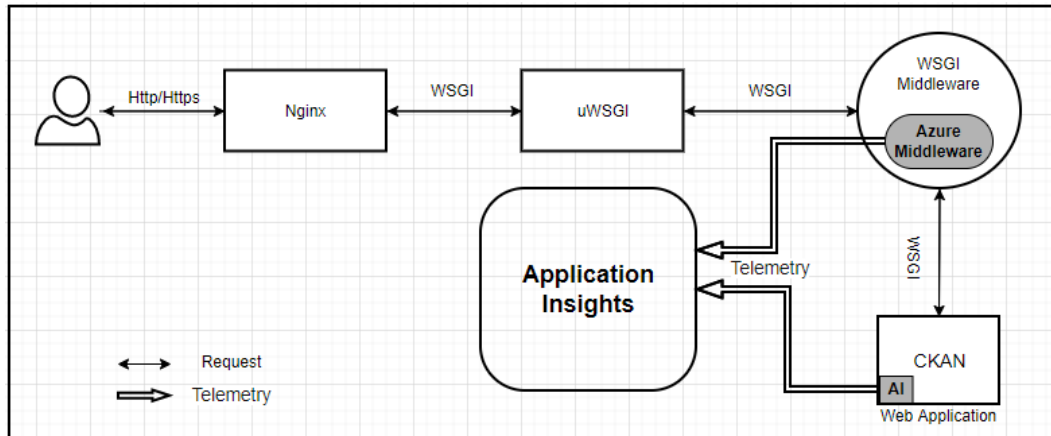


圖 12. 實驗系統架構圖

3.3 實作方法

CKAN 提供開發人員自訂擴展客製化系統，所以我們將 Azure Application Insights 以新增擴展的方式至 CKAN 系統中，以要遙測自動收集的資料模型 Request 和 Dependency 為例子，先依照 CKAN 擴展指南創建一個新的擴展後[29]，在該擴展中新增一個實作 Middleware 的插件類別，並在該插件檔中引用 Opencensus 所提供的匯出工具 Tracing 以及註冊 Azure Application Insights 提供的 GUID，如圖 13，最後於 CKAN 的 setup.py 以及 CKAN.ini 中各在不同的位置加入所設置的插件名稱，這裡設置的插件名稱為 Azuremiddleware，如圖 14 以及圖 15，在運行 CKAN 時，就可以在 Azure Application Insight 中看見即時回傳過來的 Request 和 Dependency 資料。

```
24
25 class MyPlugin(plugins.SingletonPlugin):
26     plugins.implements(plugins.IMiddleware, inherit=True)
27
28     def make_middleware(self, app, config):
29         app = FlaskMiddleware(
30             app, exporter=AzureExporter(connection_string="InstrumentationKey=995ed9d0-56
31             sampler=ProbabilitySampler(rate=1.0))
32         return app
```

圖 13. 實作 Middleware 插件

```
entry_points='''
    [ckan.plugins]
    iauthfunctions=ckanext.iauthfunctions.plugin:IauthfunctionsPlugin
    Azuremiddleware=ckanext.iauthfunctions.plugin:MyPlugin
    [babel.extractors]
    ckan = ckan.lib.extract:extract_ckan
'''
```

圖 14. setup.py

```
8 ckan.plugins = stats text_view image_view recline_view iauthfunctions
9 Azuremiddleware datastore
```

圖 15. CKAN.ini

除了傳送自動收集的遙測以外，在自訂遙測這方面，使用匯出工具 Log 和註冊 GUID 插入在 CKAN 的 source code 中，以要觀察的方向在不同程式區塊內做插入，並將特定要回傳的內容新增至資料模型內的 Custom properties 欄位。

第四章 實驗結果

在此章節中，因應 Opencensus 提供給 Azure Monitor 的匯出工具，我們使用 Log 和 Tracing。用 Tracing 收集自動監測的資料模型，而用 Log 去收集要自訂的資料模型，使用這兩個工具，可以監測到 CKAN 的使用者頁面瀏覽紀錄、資料庫執行紀錄、dataset 和 resource 的紀錄、系統的效能監測和例外狀況。

4.1 使用者頁面瀏覽

在 Tracing 的自動收集下，我們可以在 Azure Application Insight 的 Request 資料模型中，觀察到使用者對 CKAN 所傳入的要求，在 Azure Application Insight 的 LogAnalysis 中查詢，可以查詢到使用者的頁面瀏覽記錄如圖 16，可以得知的資訊分別為時間戳、該作業的識別 ID、name 為處理要求所產生的路徑、url、要求是否成功、result code、duration，圖 16 中第一筆資料，在 2022/1/8 的時候，可以看到有一個使用者瀏覽了 CKAN 的 About 頁面，此要求為成功的，resultCode 為 200，duration 為 179。另外在使用者登入以及創建 dataset 和 resource 時，這裡使用 Log 去自訂遙測資料，能從 Trace 資料模型察看結果，在圖 17 顯示了使用者登入的時間以及帳號，在圖 18 中，message 顯示了使用者對 dataset 或是 resource 的動作，以及執行該動作的 package_id 和 id。

timestamp [當地時間]...	id	name	url	success	resultCode	duration
2022/1/8 下午6:17:58.018	c76c020263acf...	GET /about	http://127.0.0.1/about	True	200	179
2022/1/8 下午6:17:57.551	b1443cb5011e...	GET /	http://127.0.0.1/	True	200	455
2022/1/8 下午6:17:57.030	8332e63e20a3...	GET /	http://127.0.0.1/	True	200	503
2022/1/8 下午6:17:56.786	0c87054394c1...	GET /	http://127.0.0.1/	True	200	238
2022/1/8 下午6:17:56.559	910f4fba22b85...	GET /	http://127.0.0.1/	True	200	217
2022/1/8 下午6:17:56.323	bcd1a96afade...	GET /	http://127.0.0.1/	True	200	229

圖 16. 頁面瀏覽紀錄

timestamp [當地時間]...	message	user	itemType
2022/1/7 下午10:24:56.773	login	client03	trace
2022/1/7 下午10:24:55.765	login	client02	trace
2022/1/7 下午10:24:54.767	login	client01	trace
2022/1/7 下午9:56:24.859	login	rabby	trace

圖 17. 使用者登入狀況

timestamp [當地時間]...	message	id	package_id	user
2022/1/7 下午10:43:38.162	create resource	07937c7a-aeb0-4479-bd4d-78fe72...	f00b5e0f-aac9-4804-ac5b-e29f11d...	rabby
2022/1/7 下午10:42:49.130	create datasets		f00b5e0f-aac9-4804-ac5b-e29f11d...	rabby

圖 18. 使用者創建 dataset 和 resource

4.2 資料庫紀錄

資料庫屬於 Dependency 資料模型的範疇，透過 Tracing 自動收集，當使用者在 CKAN 中，不管是搜尋資料、上傳資料還是下載資料，都會使用到 CKAN 的資料庫 Postgresql，在 LogAnalysis 的資料中可以查詢到 CKAN 呼叫 Postgresql 的紀錄，在圖 19 中，name 的顯示為 postgresql.query，代表 CKAN 呼叫 Postgresql 的命令為要它做 query，而在圖 20 用戶自訂義的欄位 custom Dimensions 中，可以觀察到 Postgresql 所執行的 query 語法。

timestamp [當地時間]...	id	target	type	name	success	resultCode	duration	performanceBucket
> 31/3/2021 上午11:28:01.029	2bc8640e828c1...	postgresql.qu...	Other	postgresql.qu...	True	0	0	<250ms
> 31/3/2021 上午11:28:01.041	ca8d157af0f46c...	postgresql.qu...	Other	postgresql.qu...	True	0	1	<250ms
> 31/3/2021 上午11:28:01.044	3f56dcaac8c02...	postgresql.qu...	Other	postgresql.qu...	True	0	5	<250ms
> 31/3/2021 上午11:28:01.051	98c6cd94d915c...	postgresql.qu...	Other	postgresql.qu...	True	0	0	<250ms
> 31/3/2021 上午11:28:01.055	2cb4a91e6f635...	postgresql.qu...	Other	postgresql.qu...	True	0	4	<250ms
> 31/3/2021 上午11:28:01.060	afb68a0bffdc70...	postgresql.qu...	Other	postgresql.qu...	True	0	5	<250ms
> 31/3/2021 上午11:28:01.084	26eadf6474b5b...	postgresql.qu...	Other	postgresql.qu...	True	0	1	<250ms
> 31/3/2021 上午11:28:01.337	4c76a2364f675...	postgresql.qu...	Other	postgresql.qu...	True	0	1	<250ms

圖 19. 資料庫執行紀錄

timestamp [UTC]	2021-03-31T03:28:01.044Z
id	3f56dcaac8c0235a
target	postgresql.query
type	Other
name	postgresql.query
success	True
resultCode	0
duration	5
performanceBucket	<250ms
itemType	dependency
customDimensions	<pre>{ "postgresql.cursor.method.name": "execute", "postgresql.query": "SELECT \"group\".id AS group_id, \"group\".name AS group_name \nFROM \"group\" \nWHERE \"group\".state = \"(state_1)s AND \"group\".is_organization = true AND \"group\".type = \"(type_1)s ORDER BY \"group\".title ASC \nLIMIT \"(param_1)s" }</pre>

圖 20. 所紀錄到的 Postgresql 資料語法

4.2.1 資料庫紀錄應用

除了單純的資料查看，還可以利用這些資料去做其他的分析，並繪製圖表，在圖 22 以及圖 23，這裡看到的是資料庫在 2021/05/05 至 2021/05/06 這期間，資料庫被 CKAN 所呼叫的計數，以及該期間的平均回應時間，Azure Application Insight 的 LogAnalysis 可以透過撰寫 Kusto 資料庫語法，生成要分析的資料內容，如圖 21。

```
1 // 平均 相依性 的持續時間 (依 目標)
2 let start=datetime("2021-05-05T09:09:00.000Z");
3 let end=datetime("2021-05-06T21:09:00.000Z");
4 let timeGrain=5m;
5 let dataset=dependencies
6   // 其他可以在此套用的篩選
7   | where timestamp > start and timestamp < end
8   | where client_Type != "Browser"
9 ;// 計算所有 相依性 之 平均 相依性 的持續時間
10 dataset
11 | summarize avg(duration) by bin(timestamp, timeGrain)
12 | extend dependencies='Overall'
13 // 將結果轉譯成圖表
14 | render timechart
```

圖 21. Postgresql 平均回應時間圖表生成語法

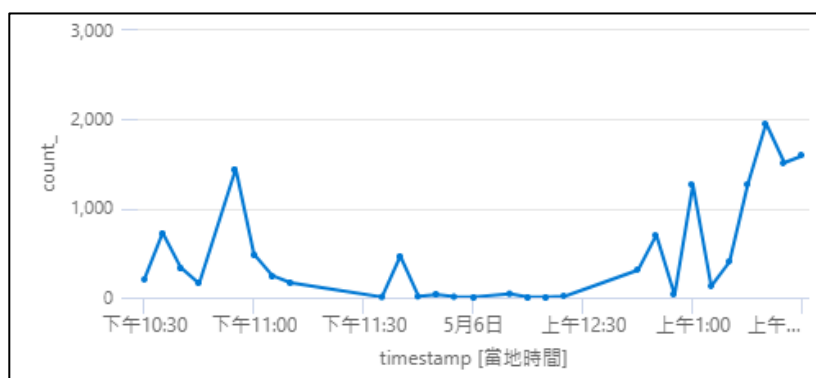


圖 22. Postgresql 要求計數分析圖

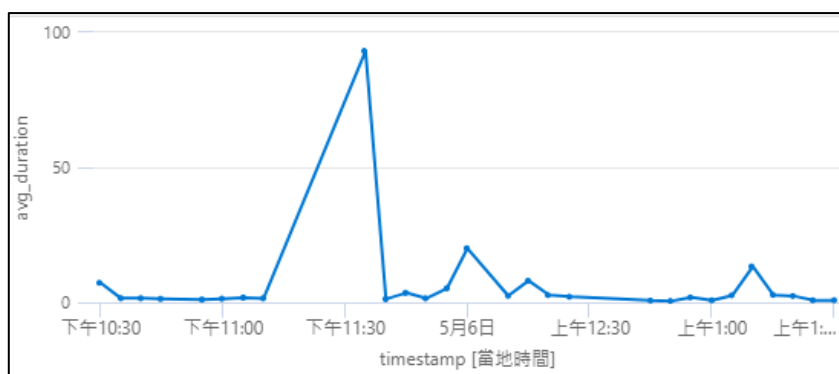


圖 23. Postgresql 平均回應時間

4.3 Datasets和Resources狀態紀錄

這裡使用 Log 去自訂遙測資料，用戶進行創建、更改和刪除 datasets 和 resources 時，可以在 Trace 的資料模型內觀察的到。

先以創建為例，使用者在 CKAN UI 介面上創建完 dataset 和 resource 後，在頁面上能看到詳細資訊如圖 24，再來去 Azure Application Insight 內查看如圖 25，可以看到有兩筆資料，分別為 create dataset 和 create resource，這兩筆分別記載了剛創建 dataset 的 package_id 還有 resource 所屬的 package_id 和他自己的 id，以及創建他們的使用者帳戶。

Additional Information

Field	Value
Data last updated	January 7, 2022
Metadata last updated	January 7, 2022
Created	January 7, 2022
Format	JSON
License	Open Data Commons Open Database License (ODbL)
Id	e7133e48-837e-440a-bf2e-e733dca13bb1
Package id	883fc08d-786d-4e10-9887-47934774d1b3
State	active

圖 24. 剛創建的 resource 詳細資料

timestamp [當地時間]...	message	id	package_id	user
2022/1/8 上午1:16:43.725	create resource	e7133e48-837e-440a-bf2e-e733dca13bb1	883fc08d-786d-4e10-9887-47934774d1b3	rabby
2022/1/8 上午1:15:51.985	create datasets		883fc08d-786d-4e10-9887-47934774d1b3	rabby

圖 25. 創建完後的 Insight 回傳資料

再來為更改 dataset 和 resource，以剛剛前面新增的資料為例，將他們更新後，也可以在 Azure Application Insight 上找到該筆資料，如圖 26，並分別顯示他們的 package_id、id 和做這動作的使用者。

timestamp [當地時間]...	message	id	package_id	user
2022/1/8 上午1:39:21.136	update resource	e7133e48-837e-440a-bf2e-e733dca13bb1	883fc08d-786d-4e10-9887-47934774d1b3	rabby
2022/1/8 上午1:39:21.093	update dataset		883fc08d-786d-4e10-9887-47934774d1b3	rabby

圖 26. 更改完後的 Insight 回傳資料

最後刪除的動作，以刪除前面新增的 dataset 和 resource 後觀察，在 Azure Application Insight 上的 id 如圖 27，和已刪除 resource 的 id 相互吻合，該 resource 所屬的 datasets 的 package_id 也會紀錄，再來刪除 datasets 時，也能記錄的到 package_id，以及做這動作的使用者。

timestamp [當地時間]...	message	id	package_id	user
2022/1/8 上午1:48:27.347	delete datasets		883fc08d-786d-4e10-9887-47934774d1b3	rabby
2022/1/8 上午1:48:10.248	delete resources	e7133e48-837e-440a-bf2e-e733dca13bb1	883fc08d-786d-4e10-9887-47934774d1b3	rabby

圖 27. 刪除完後的 Insight 回傳資料

在這些觀察中，每個要執行動作的資料都在 Azure Application Insight 內 觀察的資料吻合，這些代表著 Azure Application Insight 在 CKAN 內所傳送的遙測資料為正確的。

4.4 系統效能監測

透過在 Middleware 所回傳 Request 資料模型，根據這些資料 Azure Application Insight 將它們以欄位內的 duration 去製作圖表，分析出在不同的時間點上，伺服器的平均回應時間，以圖 28 來看，伺服器在 4 月 7 日時，平均回應時間來到 1.5 秒多，在生成這圖 28 的時間區間為 2021/03/28 至 2021/04/28，而在這時間區段，伺服器的平均回應時間為 316.81 毫秒，在圖中的虛線為沒有收集到 Request 資料所至，因為目前所使用的 CKAN 尚未發佈給公眾使用，所以呈現的資料相對的會比較零散。

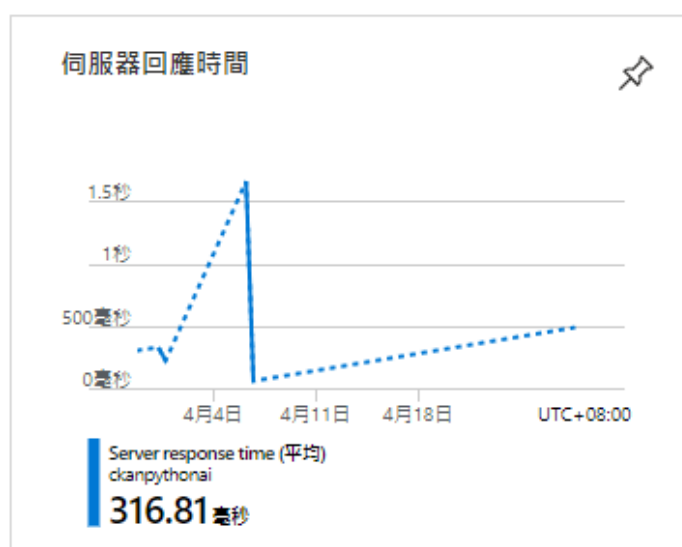


圖 28. 系統回應時間

在圖 29 中，以圖表去呈現伺服器所處理的要求計數，依照在 Request 資料模型內的筆數去做分析，圖中在 3 月 29 日時的要求計數來到了將近 150 筆資料，而在生成該圖表的時間區間 2021/03/28 至 2021/04/28 內，伺服器的要求總計數為 261 筆。



圖 29. 伺服器要求成功計數

圖 30 失敗要求計數為使用欄位 `success` 去分析，當 `success` 等於 `false` 時紀錄，圖表中在 3 月 29 日時，失敗計數為 3 個，在該時間區段 2021/03/28 至 2021/04/28 總失敗計數為 5 個。

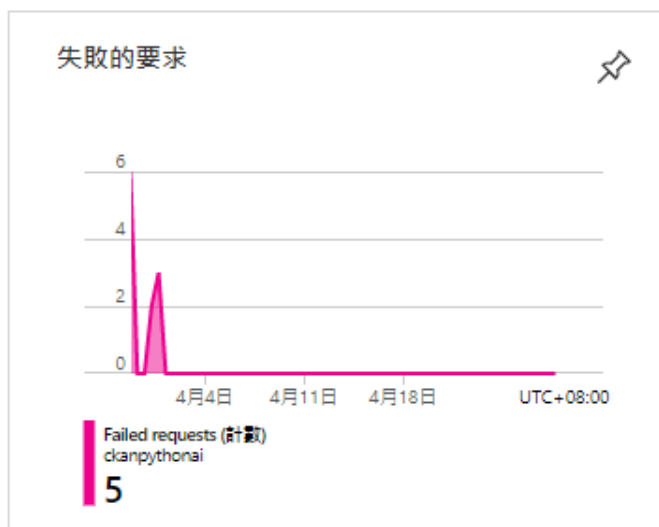


圖 30. 伺服器要求失敗計數

4.5 例外狀況紀錄

因 Opencensus 無法自動監測例外狀況，只能使用 Log 去傳送自訂遙測資料，這裡回傳的資料會是用 Trace 資料模型，而不是用 Exception 資料模型，因此在 CKAN 的程式內需要明確的去定義例外狀況，這樣才能在例外狀況觸發時以 Trace 資料模型回傳資料，在圖 31 的第一筆資料中，回傳了當 CKAN 的刪除 dataset 功能出現例外狀況時的顯示。

timestamp [當地時間]...	message	itemType
2022/1/15 下午5:21:47.299	delete datasets error	trace
2022/1/15 上午4:32:46.774	delete resources	trace
2022/1/15 上午4:32:46.773	update dataset	trace

圖 31. Trace 資料模型內的 Exception

以上講述了在 Azure Application Insight 的監控下，可以在 CKAN 觀察到的數據，這裡的數據主要都以 Opencensus 的兩個匯出工具 Tracing 和 Log 去收集。Tracing 負責在我們自己加入的 CKAN 擴展插件 Azuremiddleware 中，去收集 Request 和 Dependency 資料模型，並且能透過這兩個資料模型去查看使用者的頁面瀏覽和資料庫的執行紀錄。還可以透過 Kusto 語法，將資料以不同的需求去分析，就像是本文的系統效能監測和資料庫紀錄的應用。Log 為要傳送自訂的遙測，在創建、更改、刪除 dataset 和 resource 的程式碼以及要明確定義的例外狀況中插入 Log 匯出工具，並且定義要回傳的值。

第五章 結論

我們透過將 Opencensus 提供給 Azure Monitor 的匯出工具 Log 和 Tracing，以作為插件的方式加入至 CKAN 的 middleware，讓它將使用者要求回傳至 Azure Application Insight，以及依照 CKAN 功能的不同，直接將它嵌入至 CKAN 的 source code。透過這兩種方式，可以在 CKAN 內監測到使用者的頁面瀏覽、資料庫執行紀錄、dataset 和 resource 紀錄、系統的效能監控和例外狀況紀錄。CKAN 在 Azure Application Insight 的監控下，除了讓應用程式開發人員得到自定義的監控數值以及協助開發人員能更快速的了解應用程式出現的錯誤，也可以透過這些數據，分析使用者的行為模式，或是了解目前資料庫的效能，透過這些基本的數值監控，在未來的發展下，這些數據使用 Azure Application Insight 分析後，能夠成為開發人員以及維護人員可以運用的資料。

比較可惜的是，雖然透過 Azure Application Insight 可在 CKAN 上監測到一些重要的數據及資料，但因為 CKAN 為一個龐大的開源平台，在監控例外狀況時使用 Opencensus 的 Log 去傳送，需要去明確定義例外狀況何時觸發，反而因此去增加了維運監測 CKAN 的難度，這些是在將來希望能夠改善的地方，並期望在未來能利用 Azure Application Insight 將 CKAN 做全面的監控，也期望在未來若能完整的監控 CKAN，能夠去評估該監控系統，是否會因為 CKAN 的 source code 有所改動，而影響了 CKAN 本身的效能。

參考文獻

- [1] O. Katarzyna and Jaroslaw(2015), "Open Data collection using mobile phones based on CKAN platform" , Federated Conference on Computer Science and Information Systems , 2015, Vol. 5, pp. 1191-1196
- [2] Open Knowledge Foundation (OKFN), why open data, Available from <https://okfn.org/opendata/why-open-data/>
- [3] CKAN Documentation, Available from : <http://docs.CKAN.org>
- [4] F. Kirstein et al., Linked data in the European data portal: A comprehensive platform for applying DCAT-AP, in Proc. Int. Conf. Electron. Gov. (Tronto, Italy), July 2019, pp. 192–204.
- [5] Dasol Kim, Myeong-Seon Gil, Minh Chau Nguyen, Heesun Won, Yang-Sae Moon(2021), "Comprehensive Knowledge Archive Network harvester improvement for efficient open-data collection and management" , ETRI Journal , 2021, Volume 43 , Issue5 , pp. 835-855
- [6] CKAN User guide, Available from : <http://docs.CKAN.org/en/2.9/user-guide.html>
- [7] F. Maali, J. Erickson, and P. Archer, Data catalog vocabulary (DCAT), W3C Recommendation, Jan. 2014.
- [8] B. Momjian, PostgreSQL: Introduction and concepts, vol. 192, Addison-Wesley, Boston, MA, USA, 2001.
- [9] D. Smiley et al., Apache solr enterprise search server, Packt, Birmingham, UK, 2015.
- [10] Jinja2 documentation, available at <http://jinja.palletsprojects.com/en/2.10.x/>.
- [11] Andy Chiu , Boyang Tang , Jihong Ju , Bo Wang(2016) , "CKAN: The open source data portal", Delft Students on Software Architecture : DESOSA 2016, Available from: <https://delftswa.gitbooks.io/desosa2016/content/CKAN/chapter.html>
- [12] R. Copeland, Essential SQLAlchemy, O'Reilly Media, Sebastopol, CA, USA, 2008.
- [13] Chilberto J., Zaal S., Aroraa G., Price E.(2020) , "Application Insights.In Cloud Debugging and Profiling in Microsoft Azure", Apress Berkeley CA, Available from : https://doi.org/10.1007/978-1-4842-5437-0_4
- [14] Kurniawan A., Lau W. (2019), "Monitoring Azure Functions with Application Insights. In: Practical Azure Functions", Apress Berkeley CA, Available from : https://doi.org/10.1007/978-1-4842-5067-9_9
- [15] Application Insights overview , Available from : <https://docs.microsoft.com/en-us/azure/azure-monitor/app/app-insights-overview>

- [16] Overview of Log Analytics in Azure Monitor , Available from : <https://docs.microsoft.com/en-us/azure/azure-monitor/logs/log-analytics-overview>
- [17] Kusto Query Language (KQL) overview, Available from : <https://docs.microsoft.com/en-us/azure/data-explorer/kusto/query/>
- [18] FERRARI, Alberto; RUSSO, Marco. Introducing Microsoft Power BI. Microsoft Press, 2016.
- [19] Azure REST API reference, Available from : <https://docs.microsoft.com/en-us/rest/api/azure/>
- [20] Application Insights telemetry data model , Available from: <https://docs.microsoft.com/zh-tw/azure/azure-monitor/app/data-model>
- [21] Application Insights API for custom events and metrics , Available from: <https://docs.microsoft.com/zh-tw/azure/azure-monitor/app/api-custom-events-metrics>
- [22] Opencensus Documentation, Available from : <https://opencensus.io/introduction/>
- [23] Set up Azure Monitor for your Python application. Available from : <https://docs.microsoft.com/en-us/azure/azure-monitor/app/opencensus-python>
- [24] LEITNER, Philipp, et al. Application-level performance monitoring of cloud services based on the complex event processing paradigm. In: 2012 Fifth IEEE International Conference on Service-Oriented Computing and Applications (SOCA). IEEE, 2012. p. 1-8.
- [25] ROBINS, D. Complex event processing. In: Second International Workshop on Education Technology and Computer Science. Wuhan. 2010. p. 1-10.
- [26] NGUYEN, The An Binh, et al. Role-based templates for cloud monitoring. In: 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing. IEEE, 2014. p. 242-250.
- [27] Azure Stream Analytics documentation, Available from : <https://docs.microsoft.com/en-us/azure/stream-analytics>
- [28] Flask 教學系列 , Available from : <https://www.maxlist.xyz/2020/05/06/flask-wsgi-nginx/>
- [29] CKAN Writing extensions tutorial , Available from: <https://docs.CKAN.org/en/latest/extensions/tutorial.html>