

在Catalyst 9800上配置高級的gRPC工作流程，使用Telegraf、ConsumeDB和Grafana

目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[設定](#)

[網路圖表](#)

[組態](#)

[步驟 1.準備資料庫](#)

[步驟 2.準備電報](#)

[步驟 3.確定包含所需度量的遙測訂閱](#)

[步驟 4.在控制器上啟用NETCONF](#)

[步驟 5.在控制器上配置遙測訂閱](#)

[步驟 6.配置Grafana資料來源](#)

[步驟 7.建立儀表板](#)

[步驟 8.將視覺化新增至儀表板](#)

[驗證](#)

[WLC運行配置](#)

[Telegraf配置](#)

[GrowthDB配置](#)

[Grafana配置](#)

[疑難排解](#)

[WLC One Stop-Shop反射](#)

[確認網路連線能力](#)

[記錄與除錯](#)

[確保指標到達TIG堆疊](#)

[從HonglumDB CLI](#)

[從Telegraf](#)

[參考資料](#)

簡介

本文檔介紹如何部署Telegraf、ConsumeDB和Grafana (TIG)堆疊以及如何將其與Catalyst 9800互聯。

必要條件

本文檔透過複雜的整合演示了Catalyst 9800的程式設計介面功能。本文檔旨在展示如何根據任何需

求完全定製這些功能，以及如何節省日常時間。此處展示的部署依賴於gRPC，並提供遙測配置，使來自任何Telegraf、ConsumeDB、Grafana (TIG)可觀察堆疊中的Catalyst 9800無線資料可用。

需求

思科建議您瞭解以下主題：

- Catalyst Wireless 9800配置型號。
- 網路可程式設計性和資料模型。
- TIG堆疊基礎知識。

採用元件

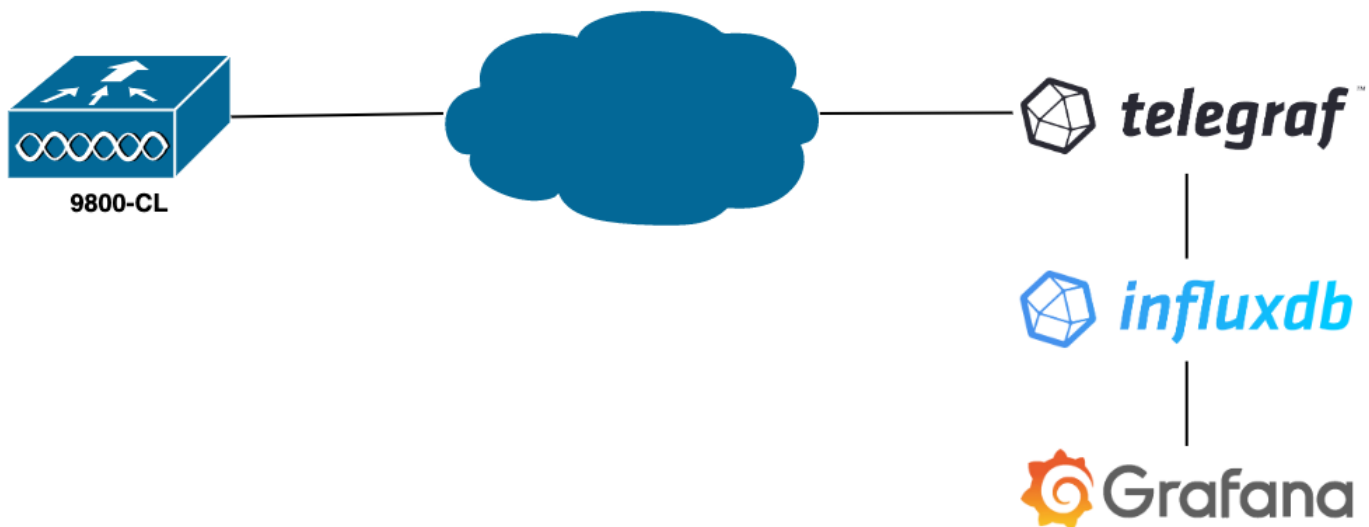
本文中的資訊係根據以下軟體和硬體版本：

- Catalyst 9800-CL (v. 17.12.03)。
- Ubuntu (v. 22.04.03)。
- InfluxDB (v. 1.06.07)。
- Telegraf (v. 1.21.04)。
- 格拉法納(v. 10.02.01)。

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

設定

網路圖表



組態

在本示例中，在9800-CL上使用gRPC撥出來配置遙測，以將儲存資訊的Telegraf應用程式上的資訊推送到ConsumeDB資料庫中。這裡使用了兩個裝置，

- 託管整個TIG堆疊的Ubuntu伺服器。
- Catalyst 9800-CL。

本配置指南並不關注這些裝置的整個部署，而是關注每個應用程式上要正確傳送、接收和顯示9800資訊所需的配置。

步驟 1.準備資料庫

進入配置部分之前，請確保您的Confusion例項運行正常。使用Linux發行版時，可使用systemctl status 命令輕鬆地完成此操作。

```
admin@tig:~$ systemctl status influxd ● influxdb.service - InfluxDB is an open-source, distributed, time series
```

例如，Telegraf需要一個資料庫來儲存測量結果，以及一個使用者來連線至此測量結果。可以使用以下命令，從InfluxDB CLI輕鬆建立這些命令：

```
admin@tig:~$ influx Connected to http://localhost:8086 version 1.8.10 InfluxDB shell version: 1.8.10 >
```

現在已建立資料庫，Telegraf可以設定成將測量結果正確儲存到資料庫中。

步驟 2.準備電報

此示例中只有兩個有趣的Telegraf配置起作用。這些可由/etc/telegraf/telegraf.conf 檔案建立（通常用於Unix上執行的應用程式）。

第一個命令會宣告Telegraf所使用的輸出。如前所述，此處使用ConfusionDB，並在telegraf.conf 檔案的輸出部分中配置如下：

```
##### # OUTPUT PLUGINS # #####
```

這指示Telegraf進程將其接收到的資料儲存在埠8086上運行於同一主機上的ConsumeDB中，並使用名為「TELEGRAF」的資料庫（以及訪問它的憑證telegraf/YOUR_PASSWORD）。

如果第一個宣告的是輸出格式，第二個當然是輸入格式。要通知Telegraf它接收的資料來自使用遙測的思科裝置，您可以使用 [cisco telemetry mdt](#) 輸入模組。要配置此功能，您只需在/etc/telegraf/telegraf.conf 檔案中增加以下行：

```
##### # INPUT PLUGINS # #####
```

這使得在主機上運行的Telegraf應用程式(在預設埠57000)能夠對來自WLC的接收資料進行解碼。

儲存配置後，請確保重新啟動Telegraf以將其應用於服務。還要確保服務已正確重新啟動：

```
admin@tig:~$ sudo systemctl restart telegraf admin@tig:~$ systemctl status telegraf.service • telegraf.s
```

步驟 3. 確定包含所需度量的遙測訂閱

如上所述，在思科裝置上和其他許多裝置上，度量是按照YANG模型組織的。可以在[此處](#)找到每個版本的IOS XE (用於9800)的特定思科YANG型號，特別是本示例中使用的用於都柏林IOS XE 17.12.03的型號。

在本例中，我們側重於從使用的9800-CL例項收集CPU利用率指標。透過檢查Cisco IOS XE都柏林17.12.03的YANG型號，可以確定哪個模組包含控制器的CPU使用率，尤其是最近5秒內的使用率。這些是Cisco-IOS-XE-process-cpu-oper模組的一部分，位於cpu-utilization分組 (枝葉5秒)下。

步驟 4. 在控制器上啟用NETCONF

gRPC撥出架構依賴[NETCONF](#)來正常運行。因此，必須在9800上啟用此功能，可透過運行以下命令實現此功能：

```
WLC(config)#netconf ssh WLC(config)#netconf-yang
```

步驟 5. 在控制器上配置遙測訂閱

根據YANG模型確定的度量的[XPath](#)(*a.k.a*，XML路徑語言)之後，可以從9800 CLI輕鬆配置遙測訂閱，以便開始將這些遙測訂閱流式傳輸到[步驟2](#)中配置的Telegraf例項。這可以透過執行以下命令來完成：

```
WLC(config)#telemetry ietf subscription 101 WLC(config-mdt-subs)#encoding encode-kvgpb WLC(config-mdt-s
```

在此代碼塊中，首先定義識別符號為101的遙測訂閱。訂閱識別符號可以是<0-2147483647>之間的任何數字，只要它不與其他訂閱重疊。此訂閱已按以下順序進行配置：

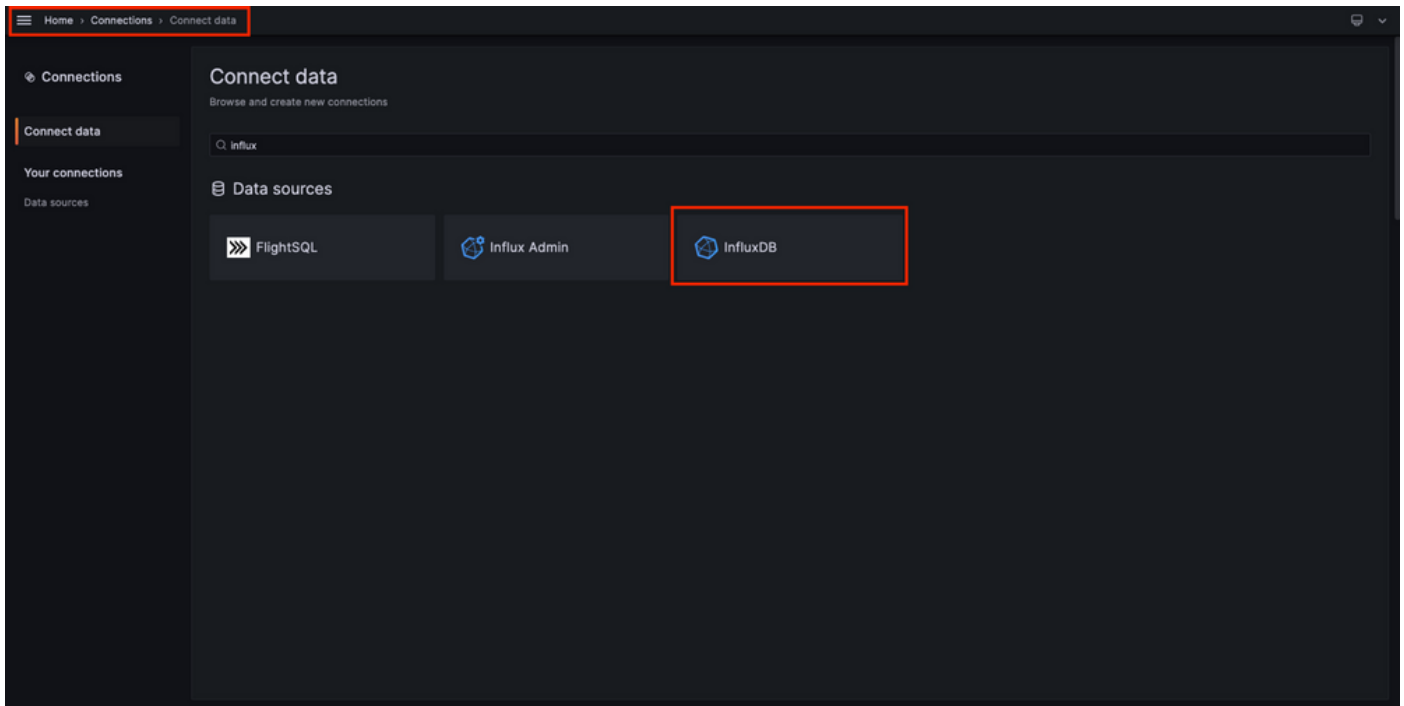
- 使用的編碼方法，在使用gRPC傳輸協定時，必須是kvGPB。
- 訂閱所傳送度量的過濾器，是定義我們感興趣度量的XPath(若要知道，請使用 /process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/five-seconds)。
- 控制器用於傳送度量的源IP地址。
- 用於傳遞度量的流型別，在本例中為YANG Push IETF標準。
- 控制器用來在100^秒內傳送資料給訂戶的頻率。在本例中，它被配置為每秒定期傳送更新。

- 接收器IP地址和埠號，以及用於在控制器和使用者之間通訊的協定。在本示例中，gRPC-TCP用於將度量傳送到埠57000上的主機10.48.39.98。

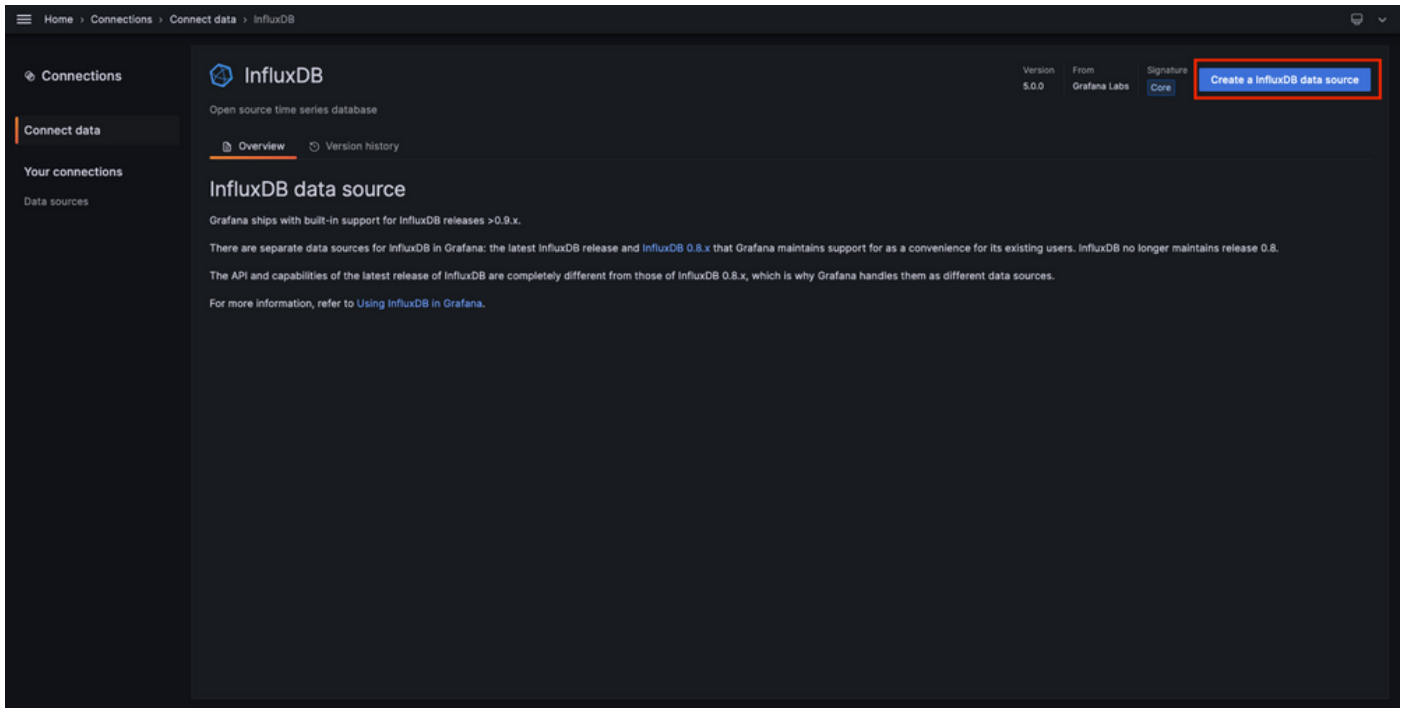
步驟 6. 配置Grafana資料來源

現在，控制器開始向Telegraf傳送資料，這些資料儲存在TELEGRAF HonglumDB資料庫中，現在應該配置Grafana使其瀏覽這些指標。

在您的Grafana GUI中，導航到`Home > Connections > Connect data`，然後使用搜尋欄查詢InfluxDB資料來源。

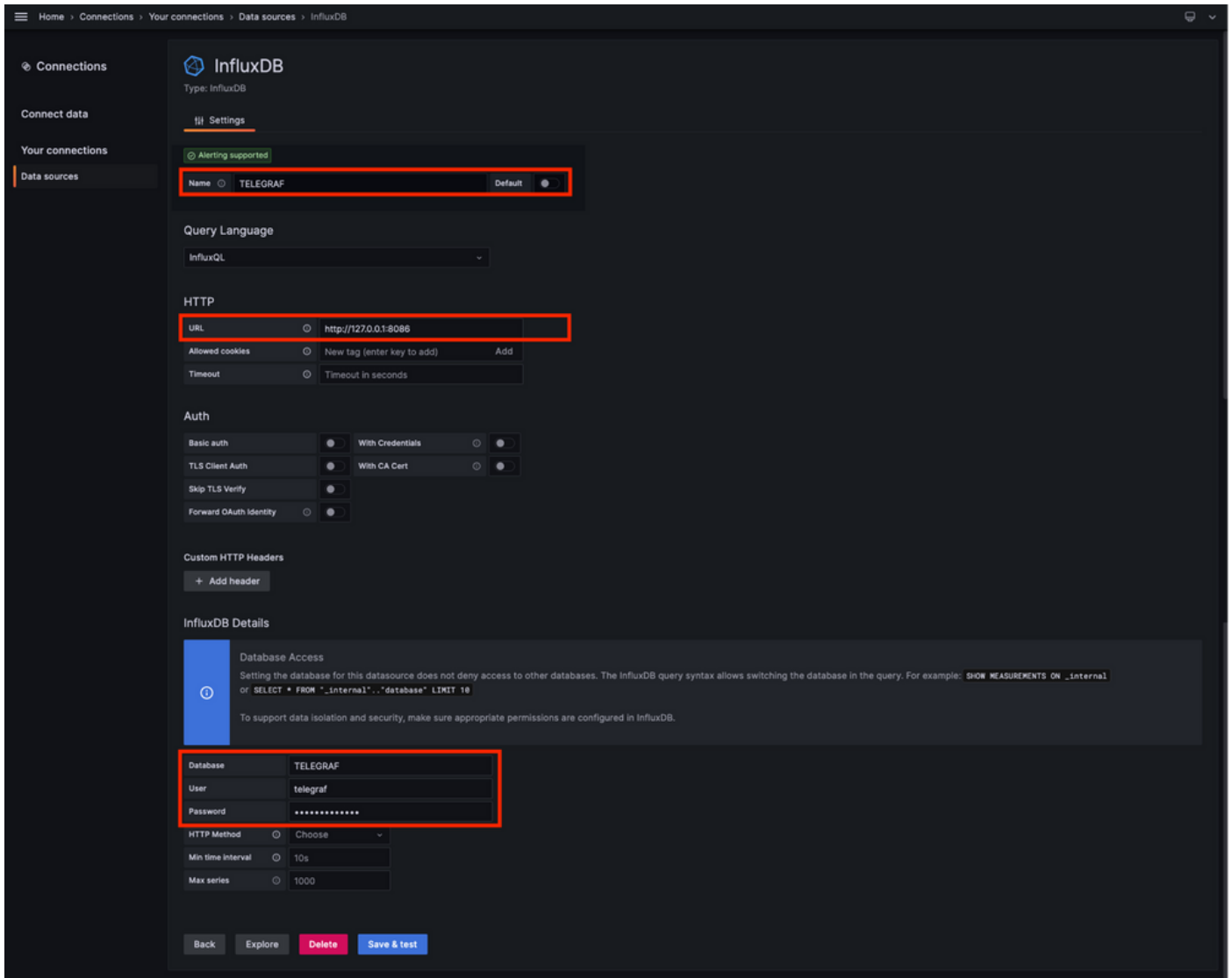


選擇此資料來源型別，然後使用「建立ConsumeDB資料來源」按鈕連線Grafana和在[步驟1](#)中建立的TELEGRAPH資料庫。



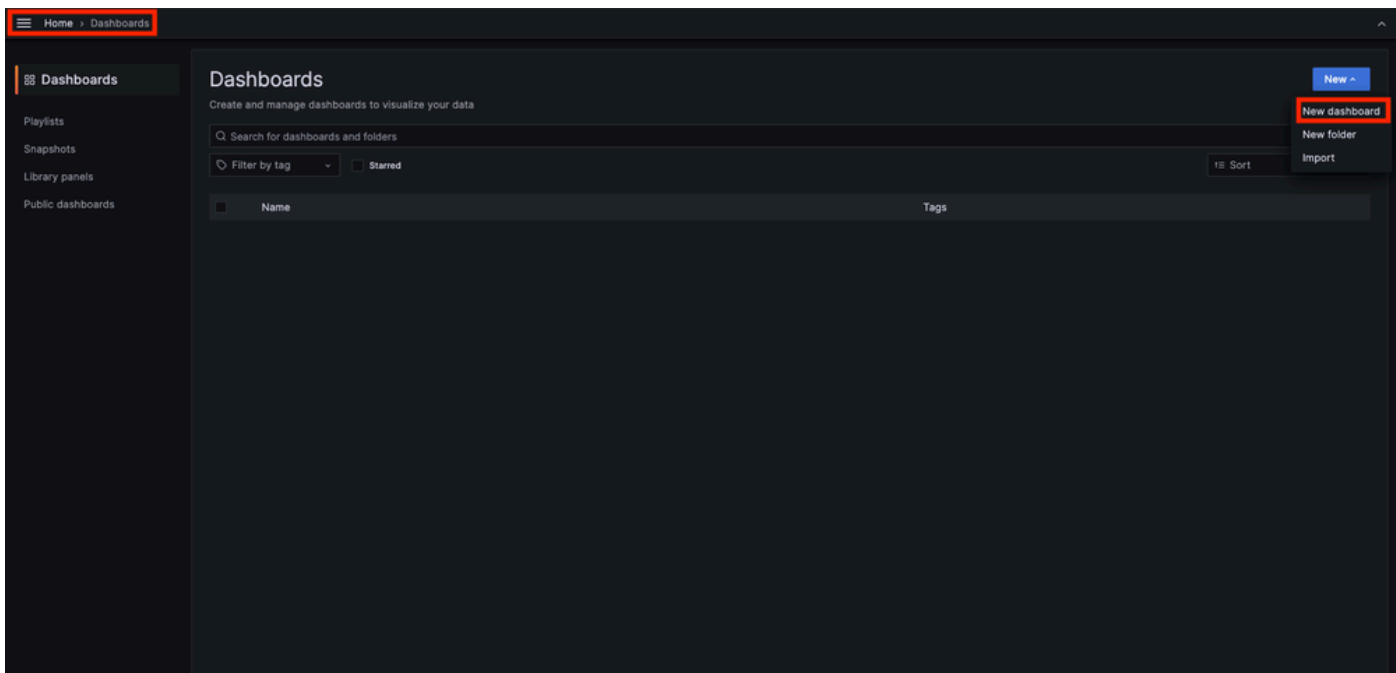
填入螢幕上顯示的表單，特別是提供：

- 資料來源的名稱。
- 所使用的InfluxDB例項的URL。
- 使用的資料庫名稱（在本範例中為「TELEGRAF」）。
- 使用者定義的認證以存取它（在此範例中為telegraf/YOUR_PASSWORD）。

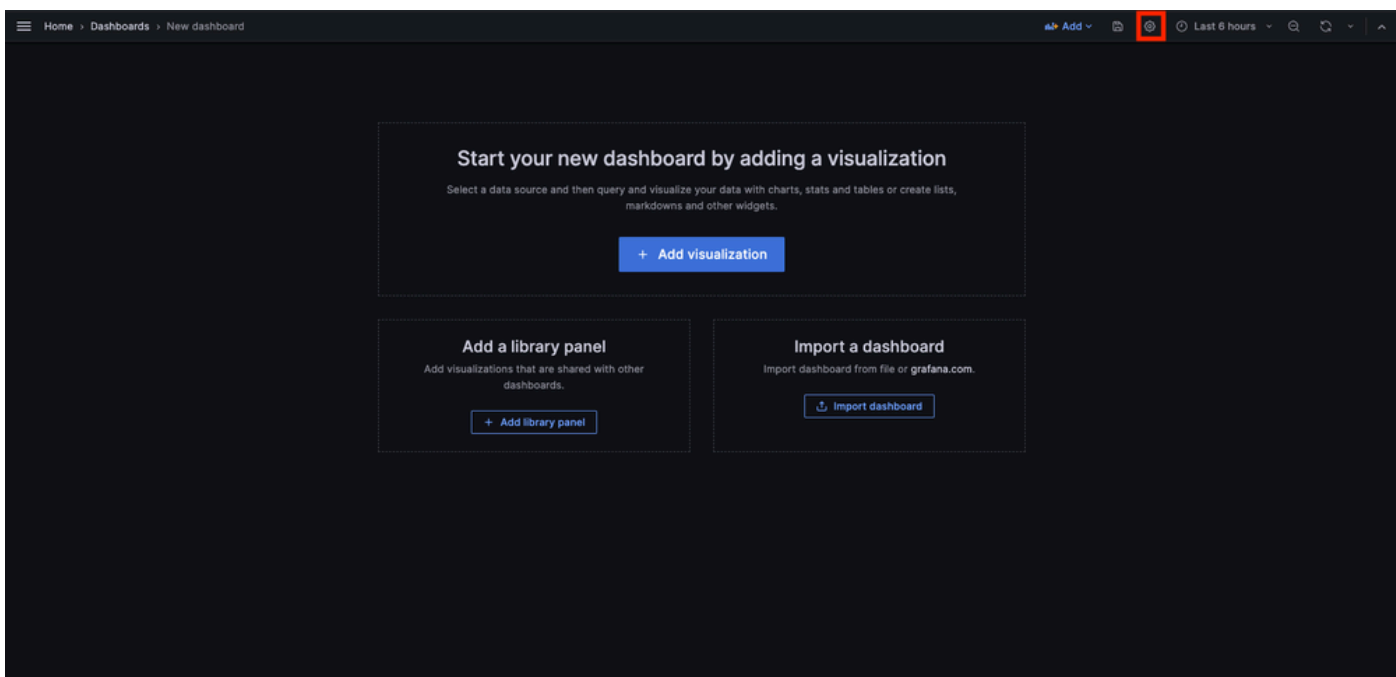


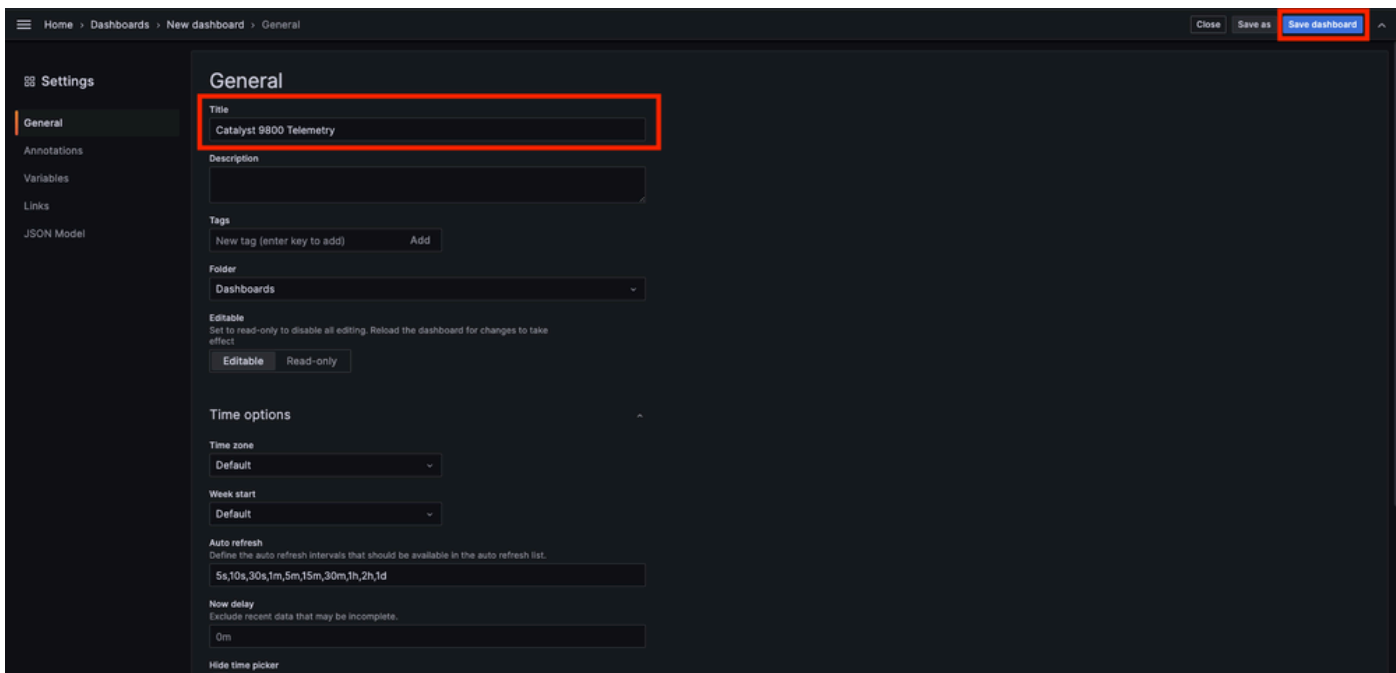
步驟 7. 建立儀表板

Grafana 視覺化內容被組織到控制台中。要建立包含 Catalyst 9800 度量視覺化的儀表板，請導航到首頁 > 儀表板，然後使用「新建儀表板」按鈕



這樣會開啟新建的圖示板。按一下齒輪圖示以存取圖示板引數並變更其名稱。本例使用「Catalyst 9800遙測」。執行此操作後，請使用「儲存控制台」按鈕儲存控制台。

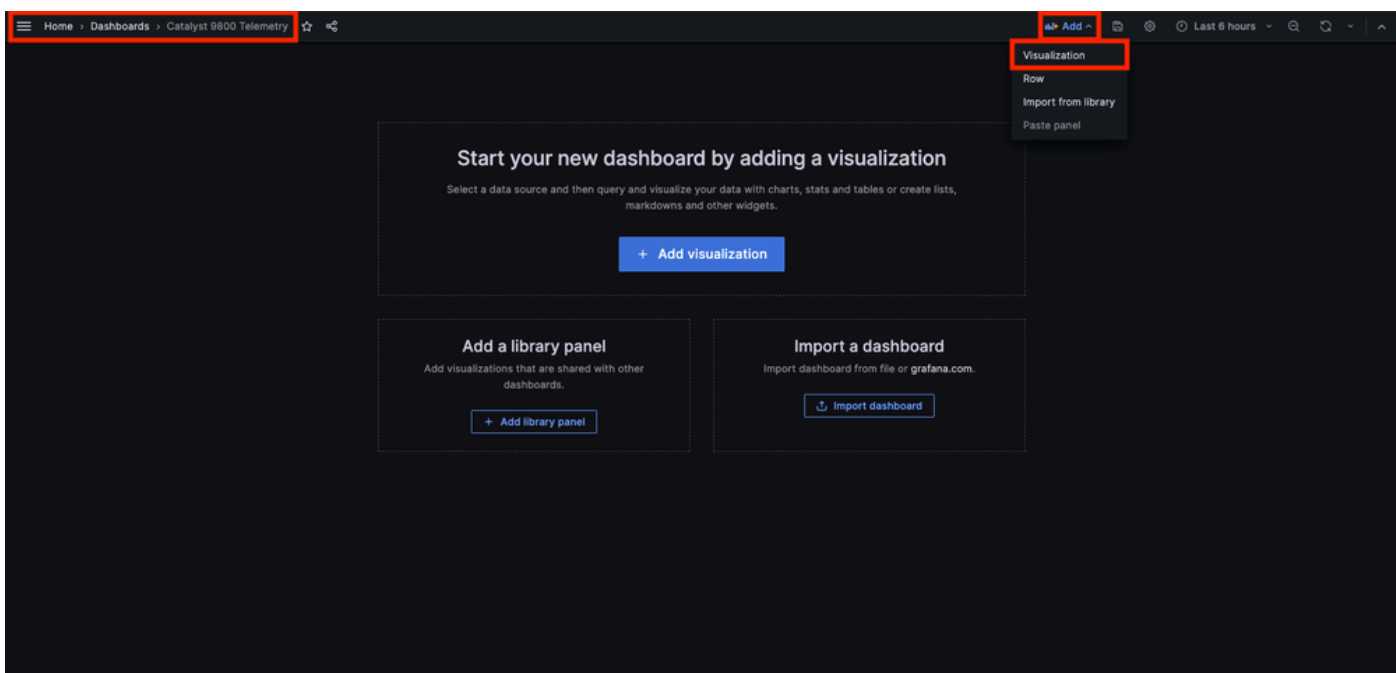




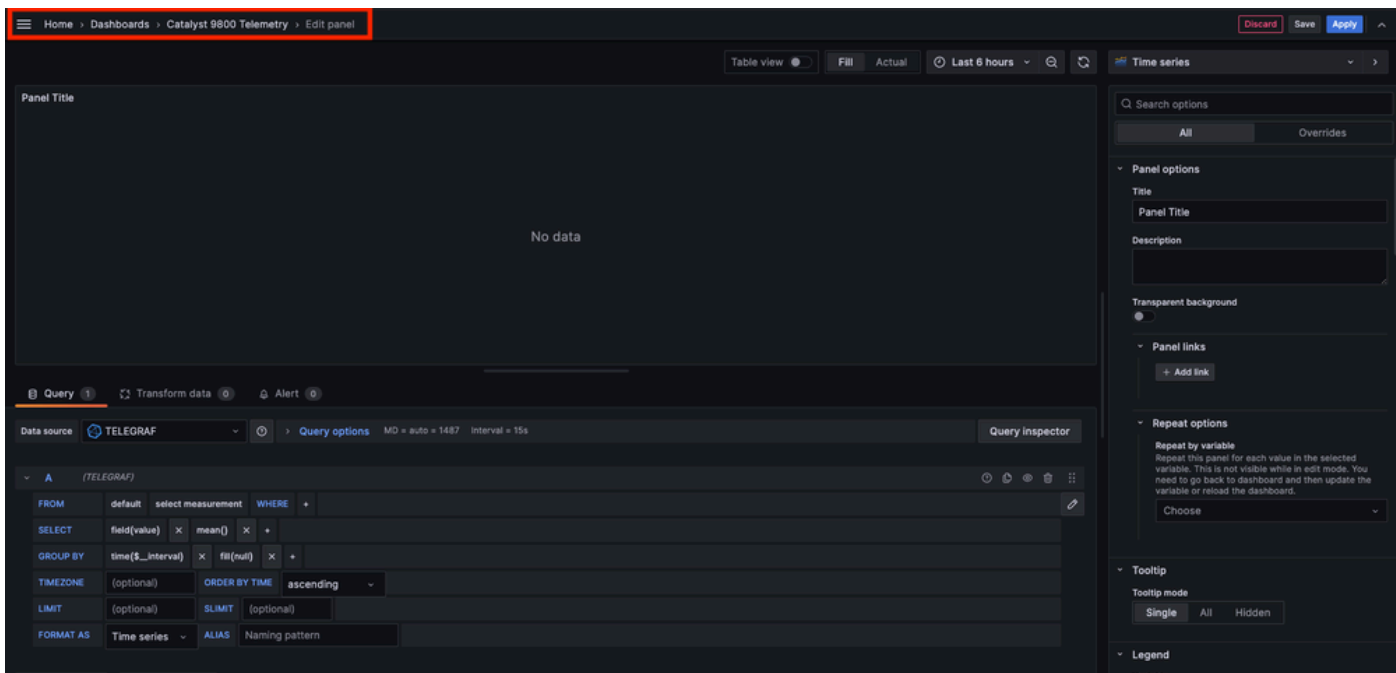
步驟 8. 將視覺化新增至儀表板

既然資料已經正確傳送、接收和儲存，而且Grafana可以訪問這個儲存位置，那麼現在是時候為他們建立視覺化了。

從任何Grafana儀表板中，使用「新增」按鈕，並從顯示的功能表中選取「視覺化」，建立測量結果的視覺化。

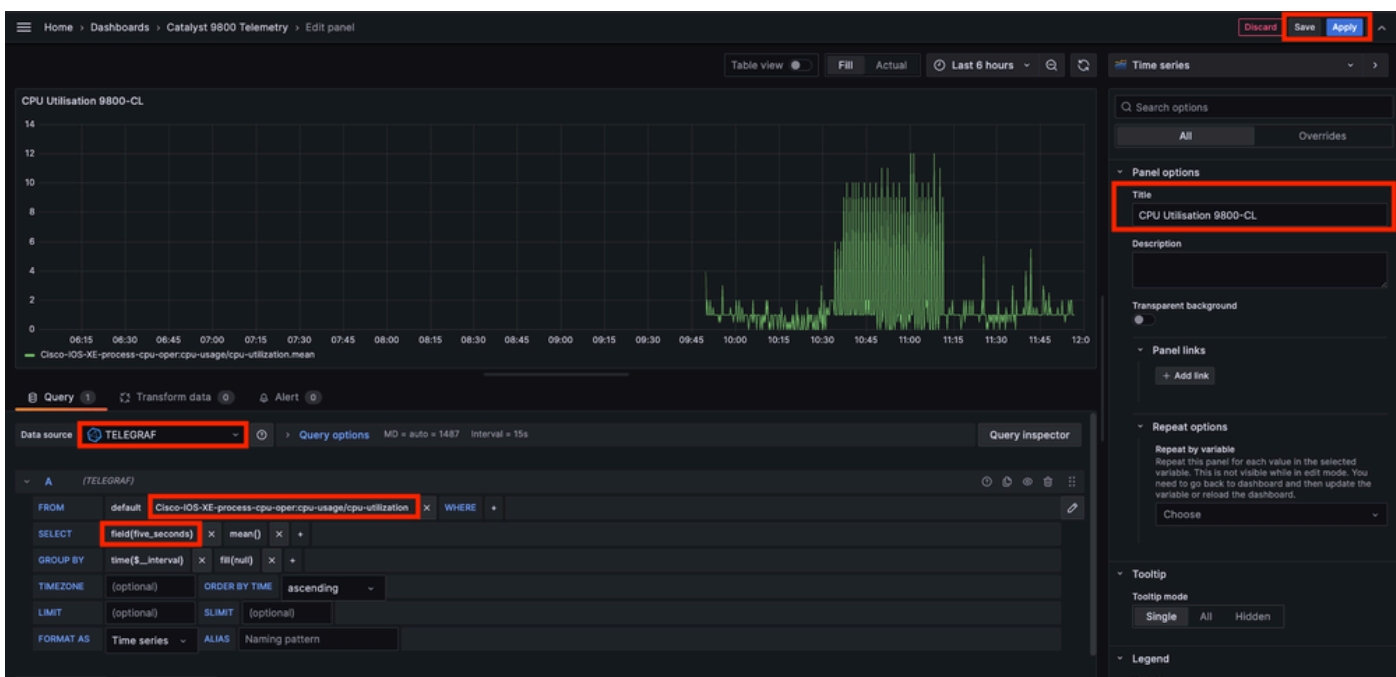


這將打開所建立的視覺化的編輯面板：

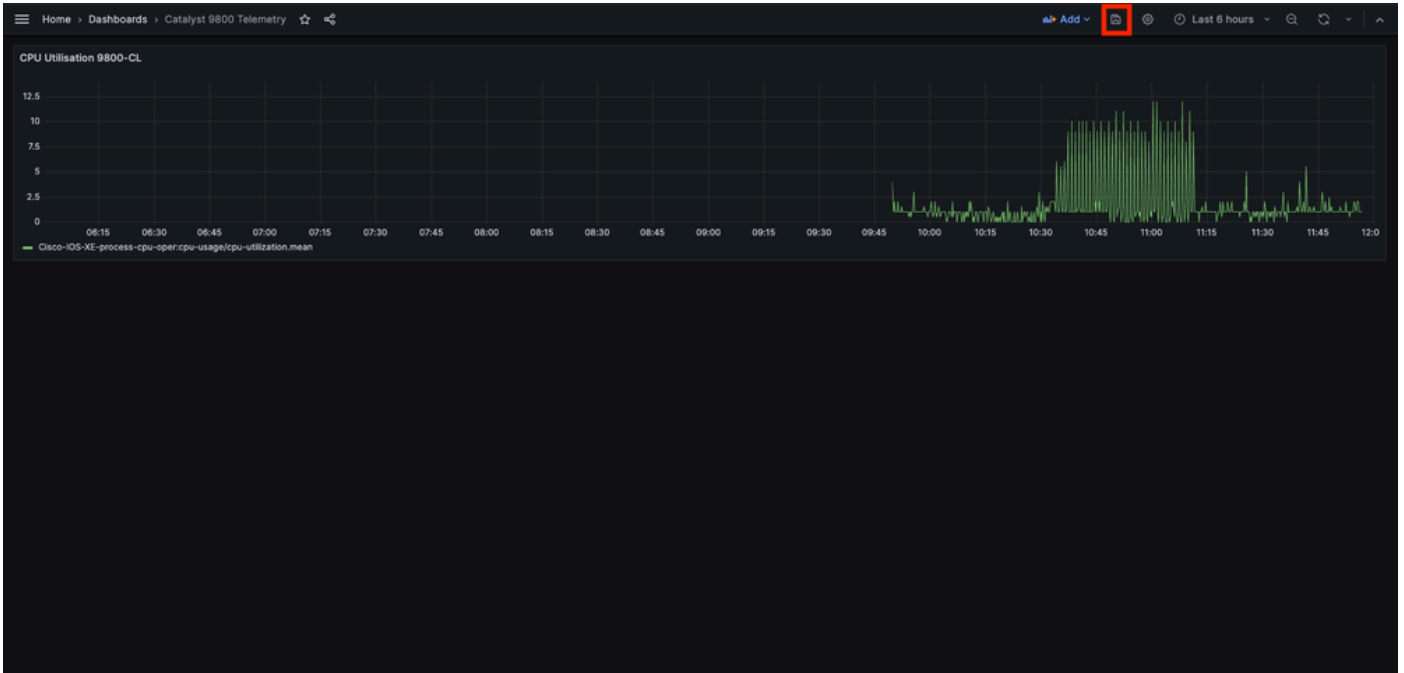


在此面板中，選取

- 您在步驟6中建立的資料來源的名稱，在本示例中為TELEGRAF。
- 包含要視覺化資料的度量（架構），在本例中為「Cisco-IOS-XE-process-cpu-oper:cpu-usage/cpu-utilization」。
- 代表您要視覺化之測量結果的資料庫欄位，在本範例中為「five_seconds」。
- 此範例中視覺化功能的標題「CPU使用率9800-CL」。



按下上一個圖中的「儲存/應用」按鈕後，顯示Catalyst 9800控制器隨時間變化的CPU使用情況的視覺化內容將增加到控制台中。使用磁片圖示按鈕可以儲存對控制台所做的更改。



驗證

WLC運行配置

```
Building configuration... Current configuration : 112215 bytes !! Last configuration change at 14:28:3
```

Telegraf配置

```
# Configuration for telegraf agent [agent] metric_buffer_limit = 10000 collection_jitter = "0s" debug =
```

GrowthDB配置

```
### Welcome to the InfluxDB configuration file. reporting-enabled = false [meta] dir = "/var/lib/influx
```

Grafana配置

```
##### Server ##### [server] http_addr = 1
```

疑難排解

WLC One Stop-Shop反射

從WLC端，首先要檢驗的是與程式設計介面相關的進程是否已啟動並正在運行。

```
#show platform software yang-management process confd : Running nesd : Running syncfd : Running ncssh
```

對於NETCONF (由gRPC撥出) ，這些命令還可以幫助檢查進程的狀態。

```
WLC#show netconf-yang status netconf-yang: enabled netconf-yang candidate-datastore: disabled netconf-y
```

檢查進程狀態後，另一個重要檢查是Catalyst 9800和電信接收器之間的遙測連線狀態。可使用「show telemetry connection all」命令檢視它。

```
WLC#show telemetry connection all Telemetry connections Index Peer Address Port VRF Source Address Stat
```

如果WLC和接收器之間的遙測連線已啟動，您還可以使用show telemetry ietf subscription all brief 命令確保配置的訂閱有效。

```
WLC#show telemetry ietf subscription all brief ID Type State State Description 101 Configured Valid Sub
```

此命令的詳細版本 show telemetry ietf subscription all detail提供有關訂閱的更多資訊，有助於指出其配置中的問題。

```
WLC#show telemetry ietf subscription all detail Telemetry subscription detail: Subscription ID: 101 Typ
```

確認網路連線能力

Catalyst 9800控制器將gRPC資料傳送到為每個遙測訂閱配置的接收器埠。

```
WLC#show run | include receiver ip address receiver ip address 10.48.39.98 57000 protocol grpc-tcp
```

要驗證WLC和此已配置埠上的接收器之間的網路連線，可使用多種工具。

從WLC中，可以在已設定的接收器IP/連線埠(這裡為10.48.39.98 : 57000)上使用telnet來驗證此接收器是否已開啟以及是否可從控制器本身連線。如果沒有封鎖流量，則連線埠必須在輸出中顯示為開啟：

```
WLC#telnet 10.48.39.98 57000 Trying 10.48.39.98, 57000 ... Open <-----
```

或者，您可以使用來自任何主機的[Nmap](#)來確保接收器已在配置的埠上正確曝光。

```
$ sudo nmap -sU -p 57000 10.48.39.98 Starting Nmap 7.95 ( https://nmap.org ) at 2024-05-17 13:12 CEST N
```

記錄與除錯

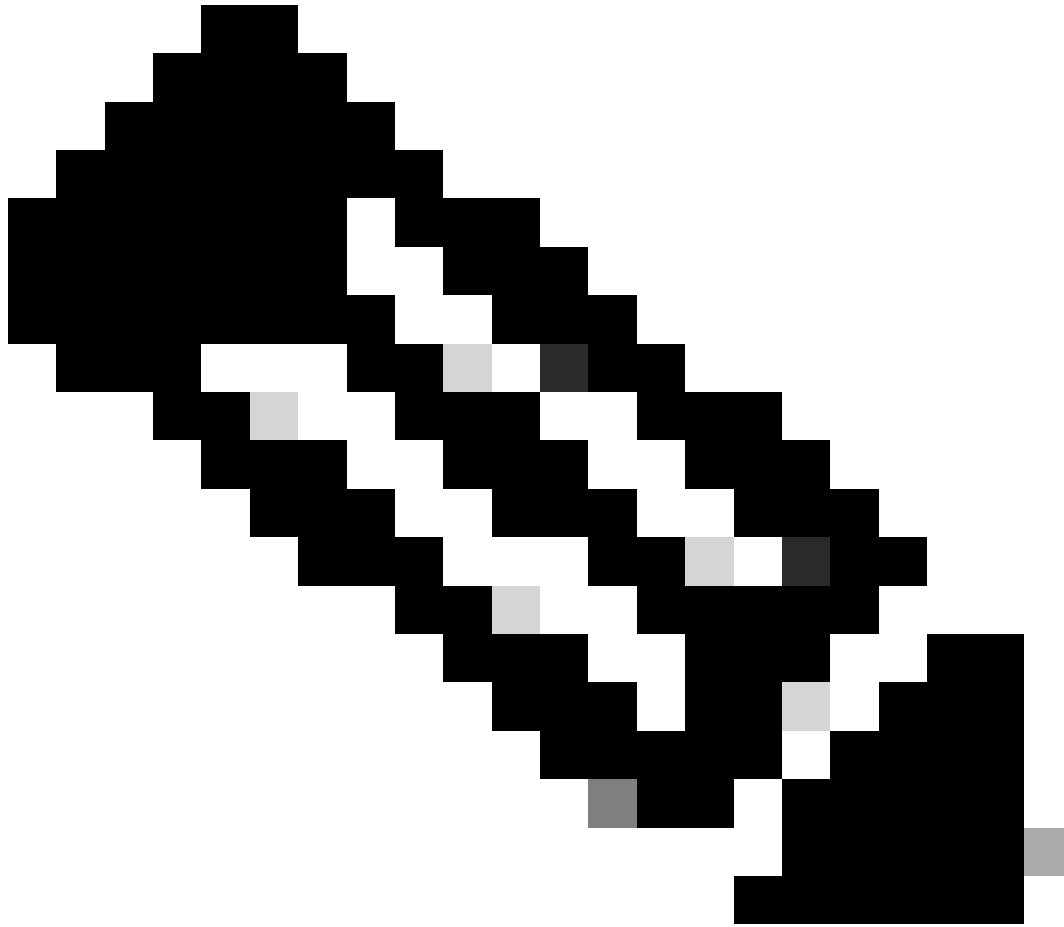
```
2024/05/23 14:40:36.566486156 {pubd_R0-0}{2}: [mdt-ctrl] [30214]: (note): **** Event Entry: Configured
```

確保指標到達TIG堆疊

從HonglumDB CLI

與其他資料庫系統一樣，ConfusionDB隨附一個CLI，可用於檢查Telegraf是否正確接收度量並儲存在定義的資料庫中。ConfusionDB將指標（稱為點）組織為度量，這些度量本身被組織為序列。此處介紹的一些基本命令可用於驗證ConsumeDB端的資料方案並確保資料到達此應用程式。

首先，您可以檢查序列、測量及其結構（鍵）是否正確產生。它們由Telegraf和ConsumeDB根據使用的RPC結構自動生成。



注意：當然，此結構可以從Telegraf和InfluxDB配置完全定製。但是，這超出了本配置指南的範圍。

```
$ influx Connected to http://localhost:8086 version 1.6.7~rc0 InfluxDB shell version: 1.6.7~rc0 > USE T
```

一旦資料結構明確（整數、字串、布林值、...），就可以根據特定欄位獲取在這些測量上儲存的資料點的數量。

```
# Get the number of points from "Cisco-IOS-XE-process-cpu-oper:cpu-usage/cpu-utilization" for the field
```

如果特定欄位的點數和上次出現的時間戳增加，則TIG堆疊正確接收和儲存WLC傳送的資料是很好的跡象。

從Telegraf

要驗證Telegraf接收方確實從控制器獲取了一些度量並檢查其格式，您可以將Telegraf度量重定向到主機上的輸出檔案。在裝置互連故障排除方面，這非常方便。為此，只需使用Telegraf的[「file」輸出外掛](#)(可透過 `/etc/telegraf/telegraf.conf` 進行配置)。

```
# Send telegraf metrics to file(s) [[outputs.file]] # ## Files to write to, "stdout" is a specially han
```

參考資料

[硬體大小調整準則](#)

[格拉法納要求](#)

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。