

在Catalyst 9800上使用Telegraf、ConfusionDB和Grafana配置高级gRPC工作流程

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[配置](#)

[网络图](#)

[配置](#)

[步骤1:准备数据库](#)

[第二步：准备电报](#)

[第三步：确定包含所需度量的遥测订阅](#)

[第四步：在控制器上启用NETCONF](#)

[第五步：在控制器上配置遥测订阅](#)

[第六步：配置Grafana数据源](#)

[步骤 7.创建控制面板](#)

[步骤 8向仪表板添加可视化](#)

[验证](#)

[WLC运行配置](#)

[电话配置](#)

[ConfusionDB配置](#)

[Grafana配置](#)

[故障排除](#)

[WLC一站式反射](#)

[确认网络可达性](#)

[日志记录和调试](#)

[确保指标到达TIG堆栈](#)

[从InclusionDB CLI](#)

[来自电报](#)

[参考](#)

简介

本文档介绍如何部署Telegraf、InfluenceDB和Grafana (TIG)堆栈以及如何将其与Catalyst 9800互连。

先决条件

本文档通过复杂的集成演示Catalyst 9800的编程接口功能。本文档旨在展示如何根据任何需求完全

定制这些功能，以及如何节省日常时间。此处显示的部署依赖于gRPC，并提供遥测配置，以便在任何Telegraf、InfluxDB、Grafana (TIG)可观察堆栈中提供Catalyst 9800的无线数据。

要求

Cisco 建议您了解以下主题：

- Catalyst Wireless 9800配置型号。
- 网络可编程性和数据模型。
- TIG堆栈基础知识。

使用的组件

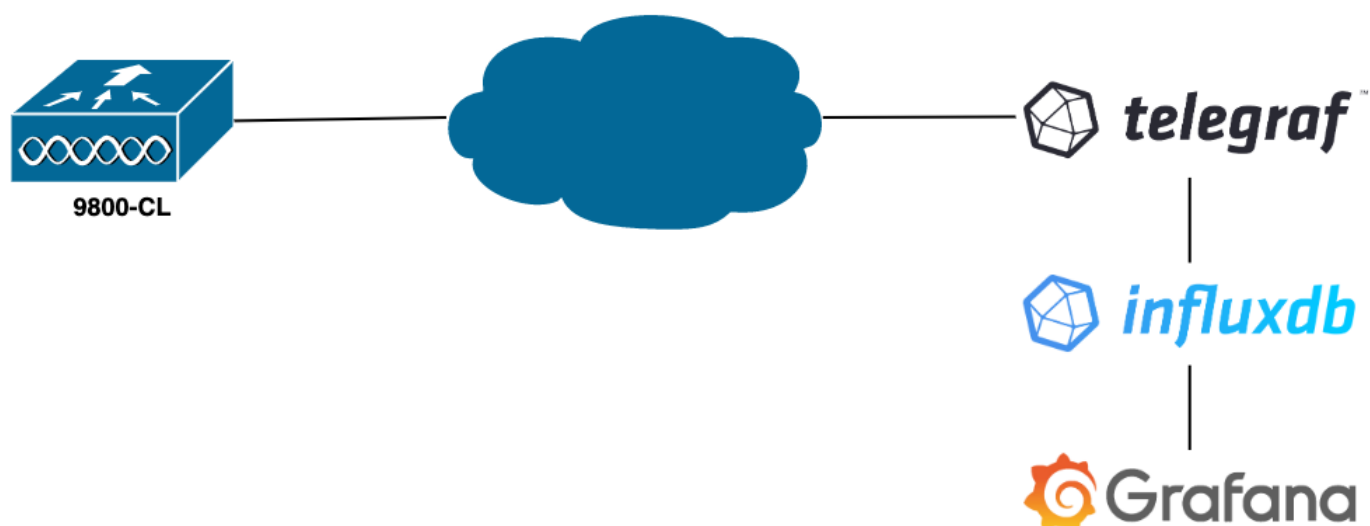
本文档中的信息基于以下软件和硬件版本：

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

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

配置

网络图



配置

在本示例中，使用gRPC拨出，在9800-CL上配置遥测，以便将存储信息的Telegraf应用上的信息推送到InfluxDB数据库中。这里使用了两个设备，

- 托管整个TIG堆栈的Ubuntu服务器。
- Catalyst 9800-CL。

本配置指南并不侧重于这些设备的整个部署，而是侧重于每个应用上为正确发送、接收和提供9800信息所需的配置。

步骤1:准备数据库

在进入配置部分之前，请确保您的Infusion实例运行正常。使用Linux发行版时，可以使用systemctl status 命令轻松完成此操作。

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

例如，Telegraf需要一个数据库来存储度量，还需要一个用户连接到该数据库。可以使用以下命令从InfluxDB CLI轻松创建这些命令：

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

现在创建的数据库可以配置Telegraf以将度量正确存储到其中。

第二步：准备电报

本示例中只有两个值得关注的电话配置有效。可以通过/etc/telegraf/telegraf.conf 配置文件创建这些文件（通常用于Unix上运行的应用程序）。

第一个命令声明Telegraf使用的输出。如上所述，此处使用InfluenceDB，并在telegraf.conf 文件的输出部分进行如下配置：

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

这指示Telegraf进程将其接收到的数据存储于端口8086上运行于同一主机上的InfluxDB中，并使用名为“TELEGRAF”的数据库（以及访问它的凭证telegraf/YOUR_PASSWORD）。

如果第一个声明的内容是输出格式，那么第二个当然是输入格式。要通知Telegraf，它收到的数据来自使用遥测技术的思科设备，您可以使用[cisco telemetry_mdt](#)“输入模块”。要配置此功能，您只需在/etc/telegraf/telegraf.conf 文件中添加以下行：

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

这使主机上运行的Telegraf应用程序(默认端口57000)能够解码从WLC收到的数据。

保存配置后，请务必重新启动Telegraf以将其应用于服务。还要确保服务已正确重新启动：

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

第三步：确定包含所需度量的遥测订阅

如上所述，在思科设备和许多其他设备上，度量是按照YANG模型组织的。可以在[此处](#)找到每个版本的IOS XE（用于9800）的特定Cisco YANG型号，特别是本示例中使用的用于IOS XE Dublin 17.12.03的型号。

在本例中，我们侧重于从使用的9800-CL实例收集CPU利用率指标。通过检查Cisco IOS XE Dublin 17.12.03的YANG型号，可以确定哪个模块包含控制器的CPU使用率，尤其是在最近5秒内。它们是Cisco-IOS-XE-process-cpu-oper模块的一部分，位于cpu-utilization分组（叶5秒）下。

第四步：在控制器上启用NETCONF

gRPC拨出框架依赖于[NETCONF](#)以同样方式工作。因此，必须在9800上启用此功能，可通过运行以下命令来实现：

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

第五步：在控制器上配置遥测订阅

一旦根据YANG模型确定度量的XPaths(*a.k.a*，XML路径语言)，就可以从9800 CLI轻松配置遥测订阅，以便开始将这些遥测流传输到[步骤2](#)中配置的Telegraf实例。这可以通过执行以下命令来完成：

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

在此代码块中，首先定义标识符为101的遥测订阅。订用标识符可以是<0-2147483647>之间的任何数字，只要它与其他订用不重叠。对于此订用，按以下顺序进行配置：

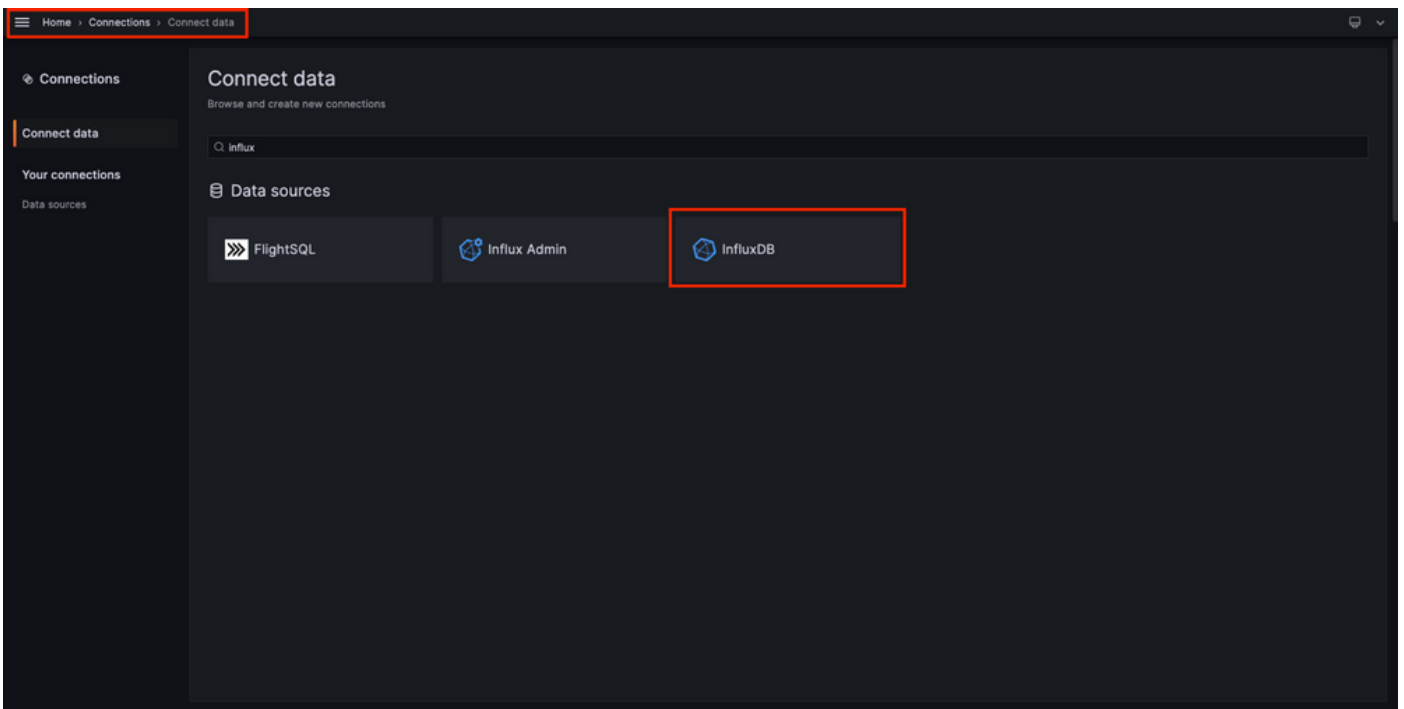
- 使用的编码方法，使用gRPC传输协议时，必须是kvGPB。
- 订用发送的度量的过滤器，是定义我们感兴趣的度量的XPath(若要知道，为 /process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/five-seconds)。
- 控制器用于发送度量的源IP地址。
- 用于传递度量的流类型，在本例中为YANG Push IETF标准。

- 控制器在100^秒内向用户发送数据所使用的频率。在这种情况下，配置为每秒定期发送更新。
- 接收方IP地址和端口号以及用于控制器与用户之间通信的协议。在本示例中，gRPC-TCP用于将度量发送到端口57000上的主机10.48.39.98。

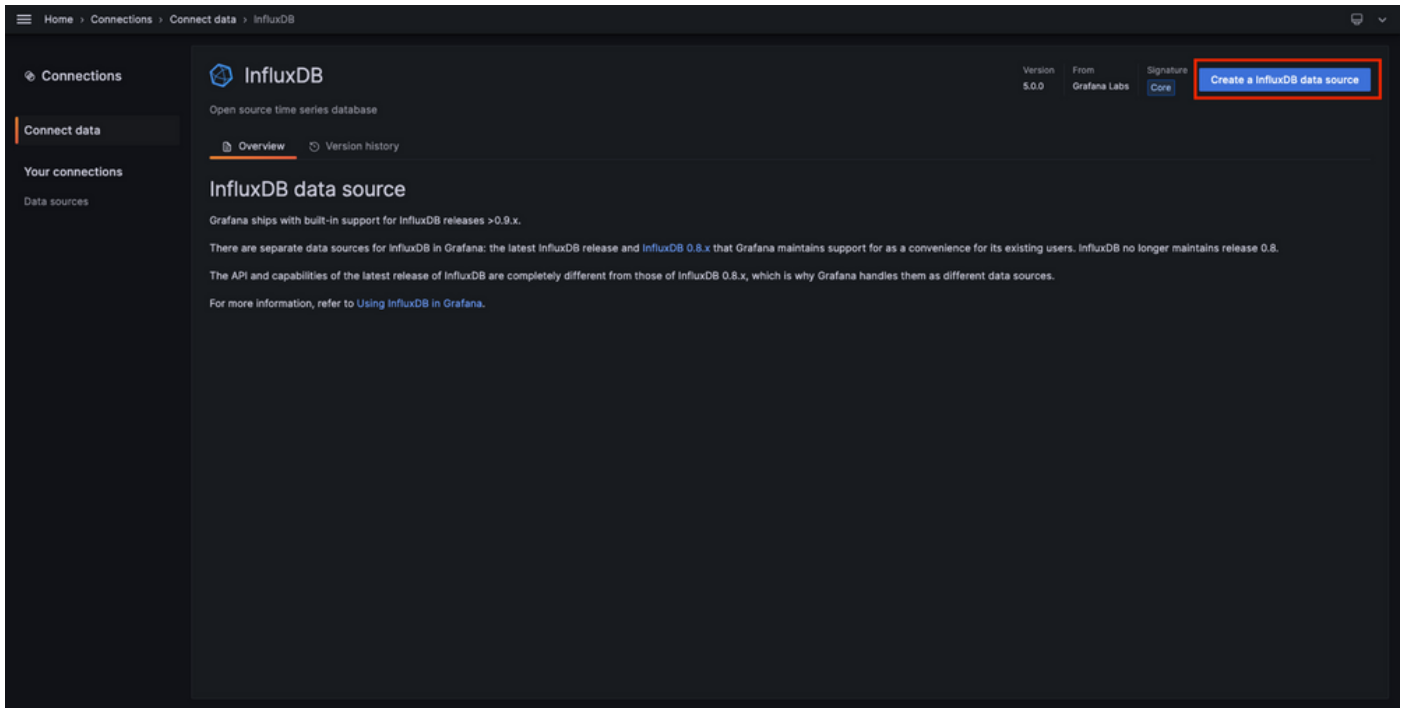
第六步：配置Grafana数据源

现在，控制器开始向Telegraf发送数据，这些数据存储在TELEGRAF InclusionDB数据库中，是时候将Grafana配置为允许其浏览这些指标了。

在Grafana GUI中，导航到Home > Connections > Connect data，然后使用搜索栏查找InfluxDB数据源。

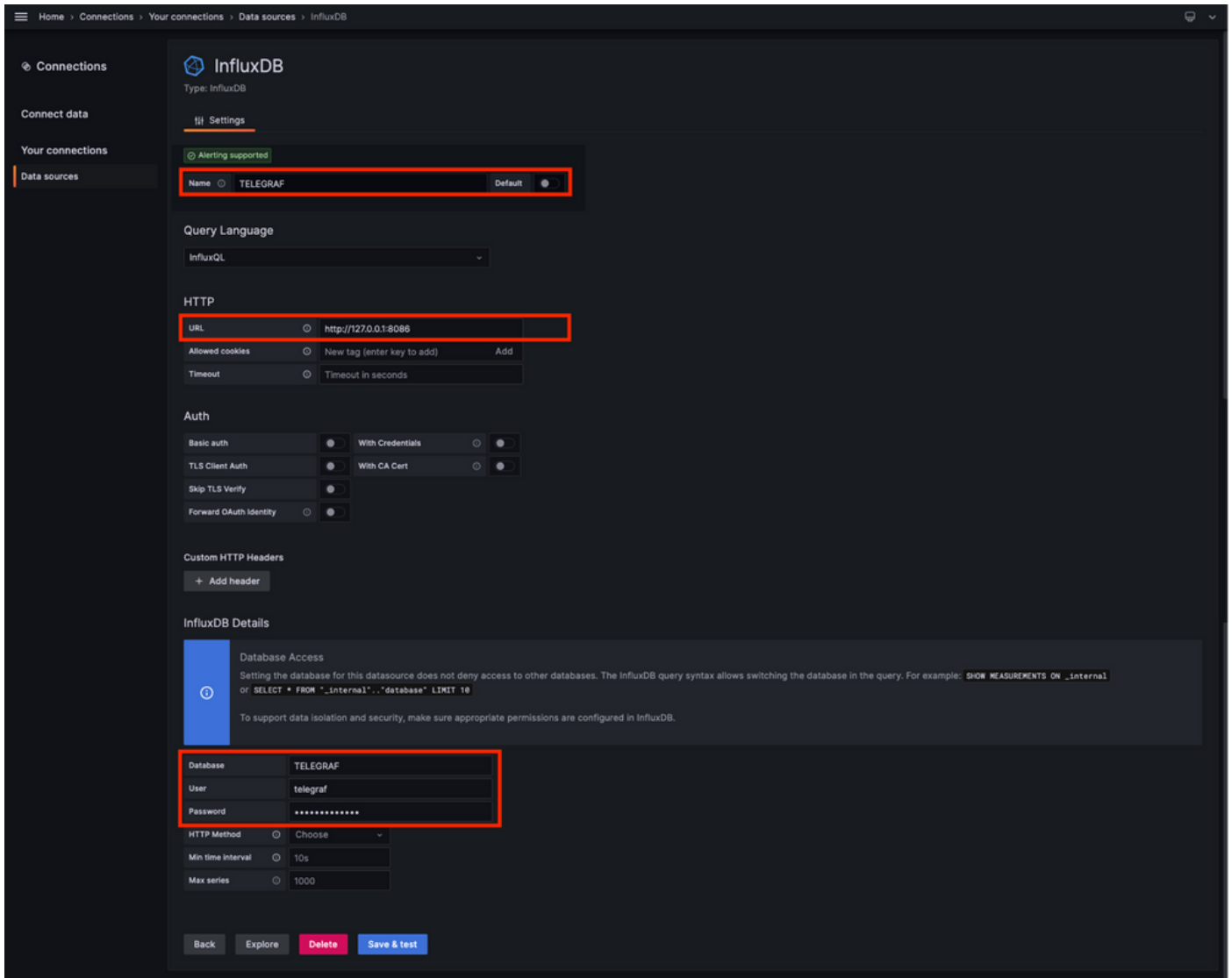


选择此数据源类型并使用“Create a InfluxDB data source”按钮连接Grafana和在[步骤1](#)中创建的TELEGRAPH数据库。



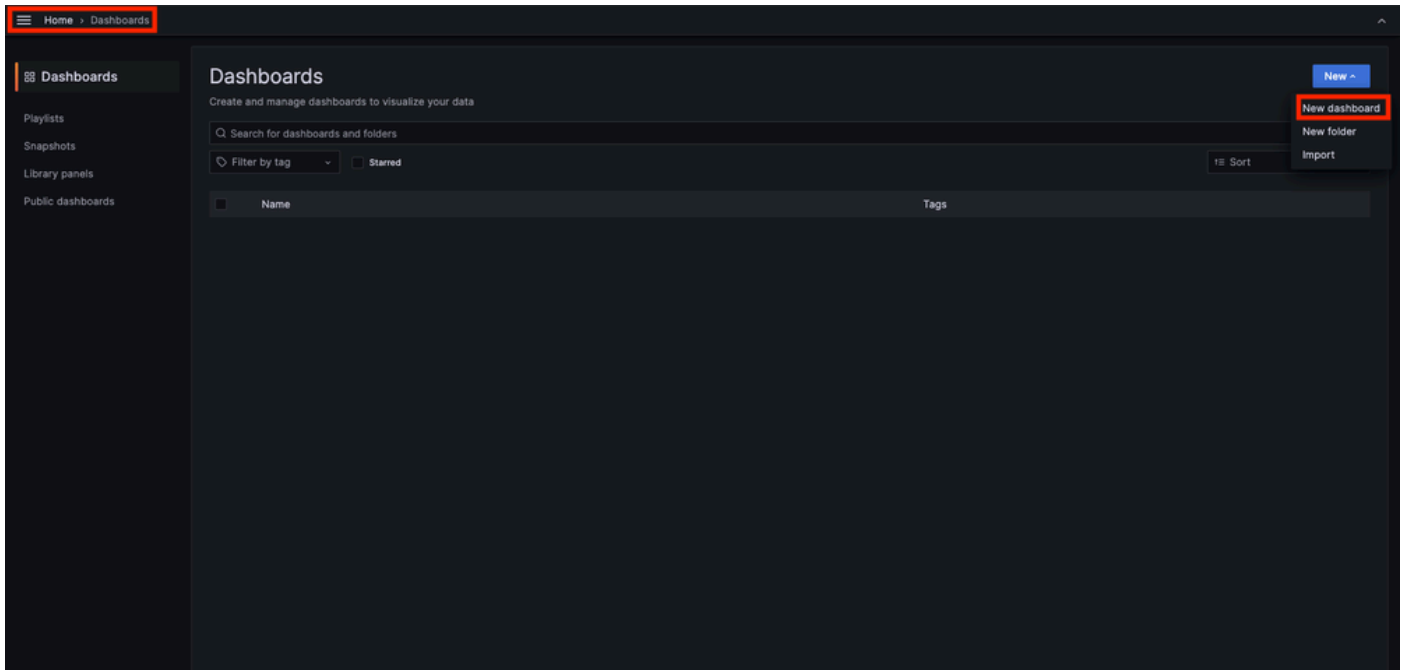
填写出现在屏幕上的表单，尤其是提供：

- 数据源的名称。
- 所用InclusionDB实例的URL。
- 使用的数据库名称（在本例中为“TELEGRAF”）。
- 用户定义的访问它的凭据（在本例中为telegraf/YOUR_PASSWORD）。

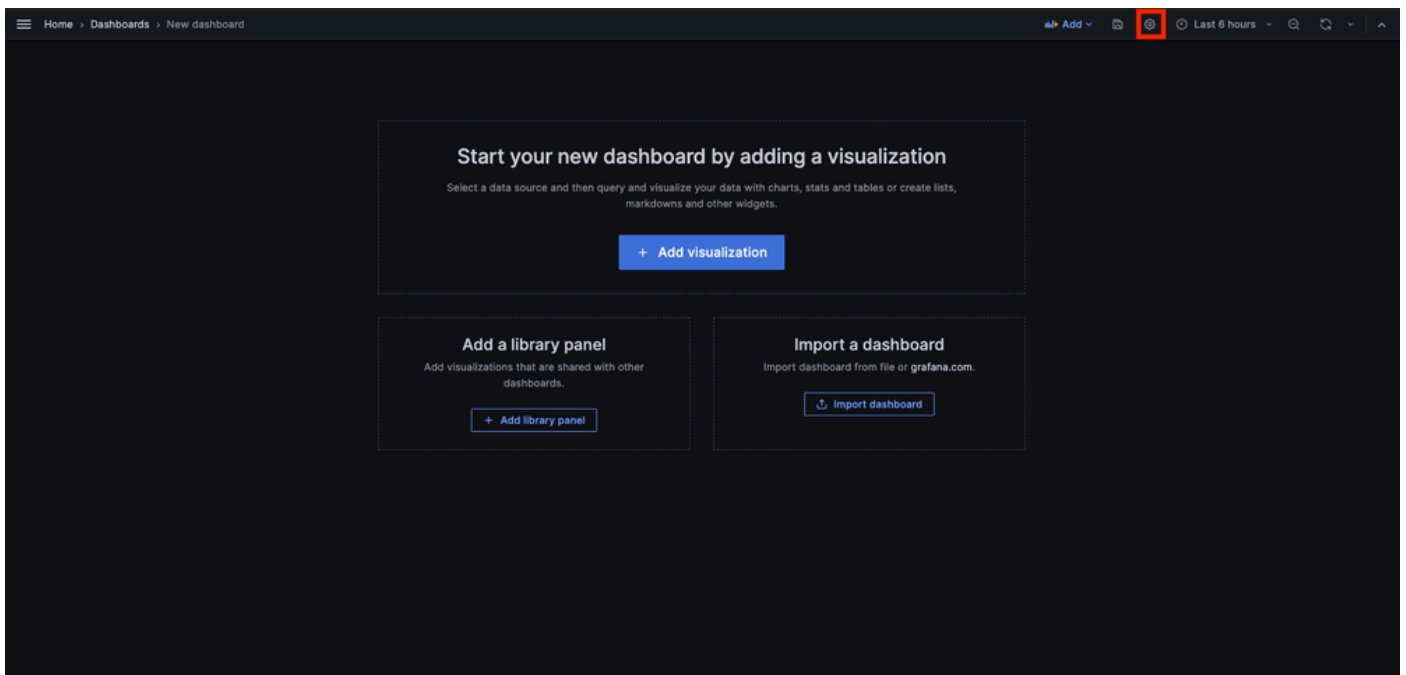


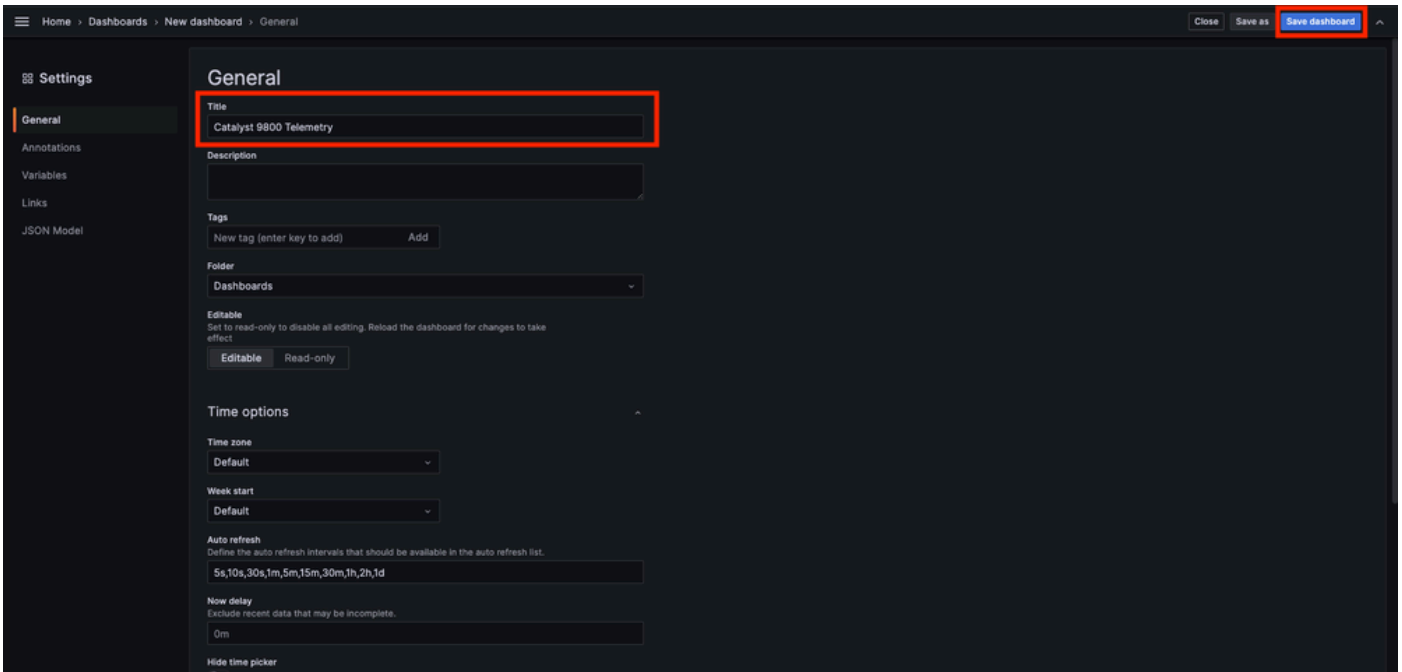
步骤 7. 创建控制面板

Grafana 可视化内容被组织到控制面板中。要创建包含 Catalyst 9800 指标可视化的控制面板，请导航到主页 > 控制面板，然后使用“新建控制面板”按钮



这将打开已创建的新控制面板。单击齿轮图标可访问操控板参数并更改其名称。在本例中，使用“Catalyst 9800遥测”。执行此操作后，请使用“保存控制面板”按钮保存控制面板。

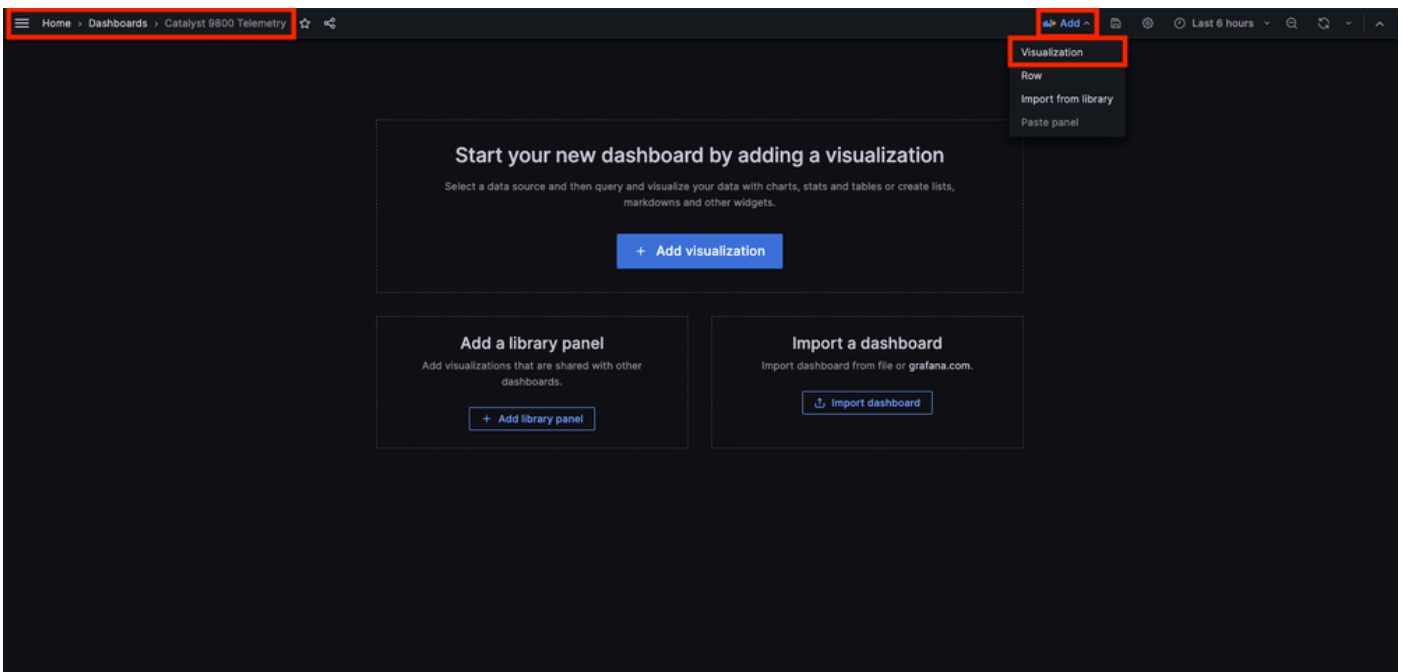




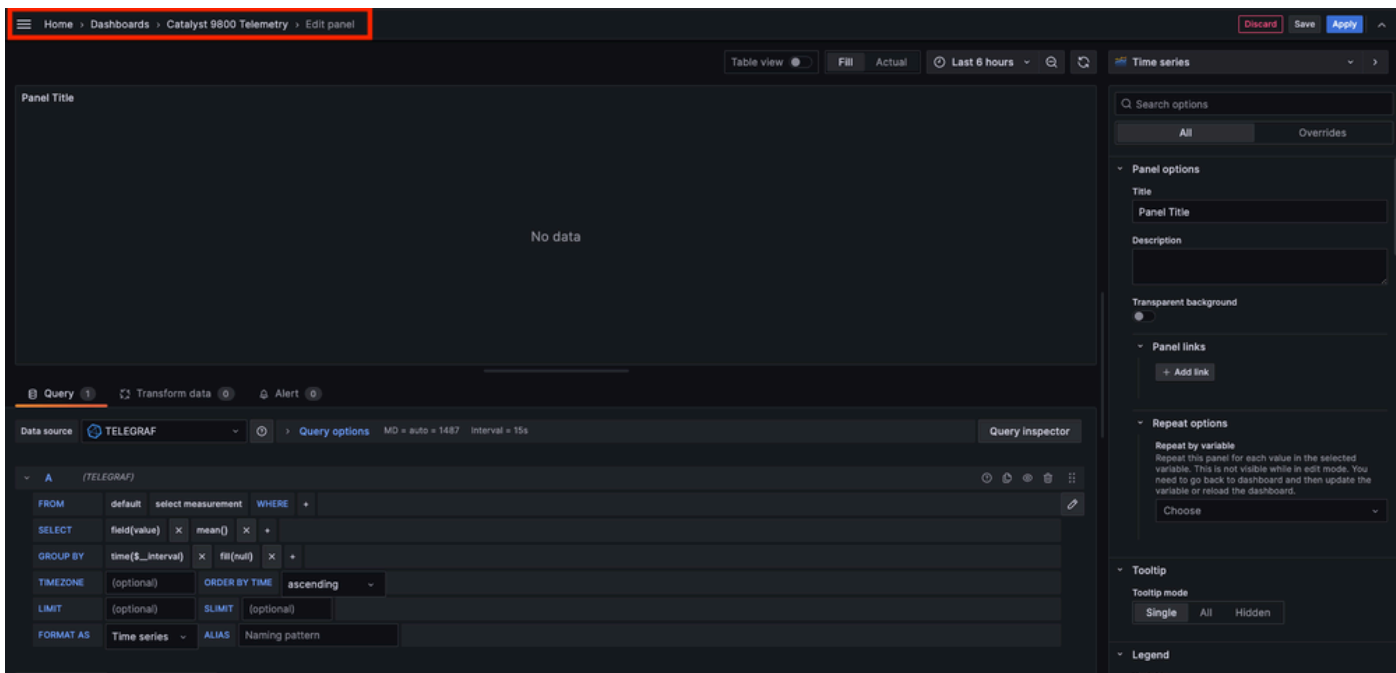
步骤 8向仪表板添加可视化

现在，数据可以正确发送、接收和存储，而且Grafana可以访问该存储位置，因此，是时候为他们创建可视化界面了。

从任意Grafana控制面板中，使用“添加”按钮，并从显示的菜单中选择“可视化”以创建指标的可视化。

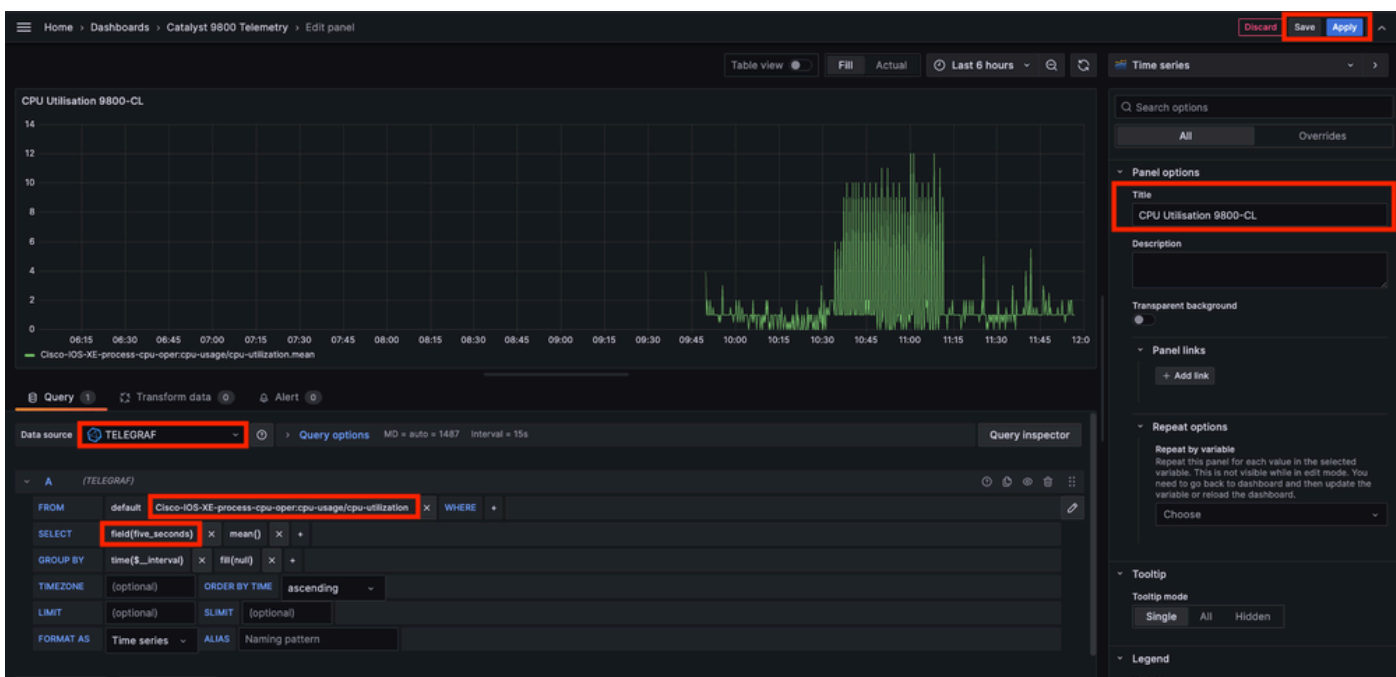


这将打开已创建可视化的Edit面板：

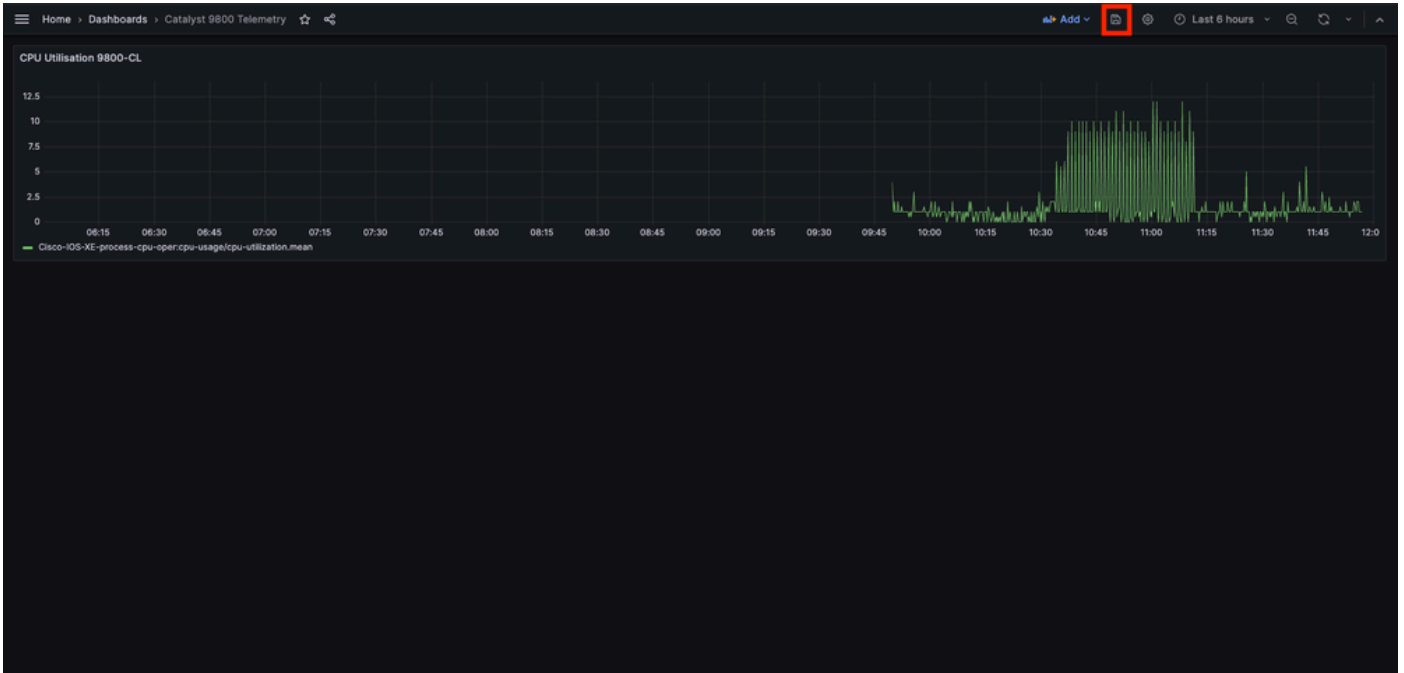


在此面板中选择

- 在步骤6中创建的数据源的名称，在本示例中为TELEGRAF。
- 包含要可视化的数据的度量（架构），在本例中为“Cisco-IOS-XE-process-cpu-oper : cpu-usage/cpu-utilization”。
- 表示要可视化的度量的数据库字段，在本例中为“five_seconds”。
- 此示例中可视化内容的标题“CPU Utilization 9800-CL”。



一旦按下上一个图中的“保存/应用”按钮，显示Catalyst 9800控制器CPU使用率随时间的可视化信息就会被添加到控制面板中。使用软盘图标按钮可保存对控制面板所做的更改。



验证

WLC运行配置

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

电话配置

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

ConfusionDB配置

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

Grafana配置

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

故障排除

WLC一站式反射

从WLC方面来看，首先要检验的是与编程接口相关的进程已启动并正在运行。

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

对于NETCONF (由gRPC拨出)，这些命令也可以帮助检查进程的状态。

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

检查进程状态后，另一个重要检查是Catalyst 9800和Telegraf接收器之间的遥测连接状态。可使用“show telemetry connection all”命令查看该报告。

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

如果WLC和接收器之间的遥测连接处于启用状态，则还可以使用show telemetry ietf subscription all brief 命令确保配置的订用有效。

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

此命令的详细版本 show telemetry ietf subscription all detail提供有关预订的更多信息，可帮助指出其配置中的问题。

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

确认网络可达性

Catalyst 9800控制器将gRPC数据发送到为每个遥测订阅配置的接收器端口。

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

要验证WLC与接收器之间在此已配置端口上的网络连接，可以使用多种工具。

从WLC中，可以对已配置的接收器IP/端口(此处为10.48.39.98 : 57000)使用telnet来验证此接收器是否打开以及是否可从控制器本身访问。如果流量未被阻塞，端口必须在输出中显示为打开：

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

或者，您可以使用[Nmap](#)从任何主机确保接收方在已配置的端口上正确暴露。

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

日志记录和调试

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

确保指标到达TIG堆栈

从**IncusionDB CLI**

与其他数据库系统一样，InfluenceDB也随附一个CLI，可用于检查Telegraf是否正确接收并存储在定义的数据库中。InfluenceDB将度量值（称为点）组织成测量值，这些测量值本身被组织成系列。此处介绍的一些基本命令可用于验证InfluenceDB端的数据方案，并确保数据到达此应用程序。

首先，您可以检查序列、测量及其结构（键）是否正确生成。它们由Telegraf和InfluenceDB根据使用的RPC结构自动生成。



注意：当然，此结构可以从Telegraf和InfluxDB配置完全自定义。但是，这已超出本配置指南的范围。

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

一旦数据结构明确（整数、字符串、布尔值……），就可以根据特定字段获取在这些测量上存储的数据点的数量。

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

如果特定字段的点数和最后一次出现的时间戳增加，则表示TIG堆栈正确接收和存储WLC发送的数据是个好兆头。

来自电报

要验证电报接收器确实从控制器获取了一些指标并检查其格式，您可以将电报指标重定向到主机上的输出文件。在设备互连故障排除方面，这非常方便。为此，只需使用Telegraf的[“file”输出插件](#)(可通过 `/etc/telegraf/telegraf.conf`进行配置)即可。

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

参考

[硬件规模调整指南](#)

[Grafana要求](#)

关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。