

Konfigurieren des erweiterten gRPC-Workflows mit Telegraf, InfluxDB und Grafana auf Catalyst 9800

Inhalt

[Einleitung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Konfigurieren](#)

[Netzwerkdiagramm](#)

[Konfigurationen](#)

[Schritt 1: Datenbank vorbereiten](#)

[Schritt 2: Telegraf vorbereiten](#)

[Schritt 3: Telemetrie-Abonnement mit der gewünschten Metrik ermitteln](#)

[Schritt 4: Aktivieren von NETCONF auf dem Controller](#)

[Schritt 5: Konfigurieren des Telemetrie-Abonnements auf dem Controller](#)

[Schritt 6: Grafana-Datenquelle konfigurieren](#)

[Schritt 7: Dashboard erstellen](#)

[Schritt 8: Hinzufügen einer Visualisierung zum Dashboard](#)

[Überprüfung](#)

[Konfiguration des WLC wird ausgeführt](#)

[Telegraf-Konfiguration](#)

[InfluxDB-Konfiguration](#)

[Grafana-Konfiguration](#)

[Fehlerbehebung](#)

[WLC One Stopp Shop Reflex](#)

[Netzwerkerreichbarkeit bestätigen](#)

[Protokollierung und Debuggen](#)

[Erreichbarkeit der Kennzahlen am TIG-Stack](#)

[Von InfluxDB CLI](#)

[Von Telegraf](#)

[Referenzen](#)

Einleitung

In diesem Dokument wird beschrieben, wie der Telegraf-, InfluxDB- und Grafana-Stack (TIG) bereitgestellt und mit dem Catalyst 9800 verbunden wird.

Voraussetzungen

In diesem Dokument werden die programmatischen Schnittstellenkapazitäten des Catalyst 9800 durch eine komplexe Integration beschrieben. Dieses Dokument soll zeigen, wie diese je nach Bedarf vollständig anpassbar sind und tägliche Zeitersparnisse ermöglichen. Die hier gezeigte Bereitstellung stützt sich auf gRPC und präsentiert eine Telemetriekonfiguration, um Wireless-Daten vom Catalyst 9800 in jedem Telegraf-, InfluxDB-, Grafana (TIG) Observability Stack verfügbar zu machen.

Anforderungen

Cisco empfiehlt, dass Sie über Kenntnisse in folgenden Bereichen verfügen:

- Catalyst Wireless 9800-Konfigurationsmodell
- Netzwerkprogrammierbarkeit und Datenmodelle
- TIG-Stapel-Grundlagen.

Verwendete Komponenten

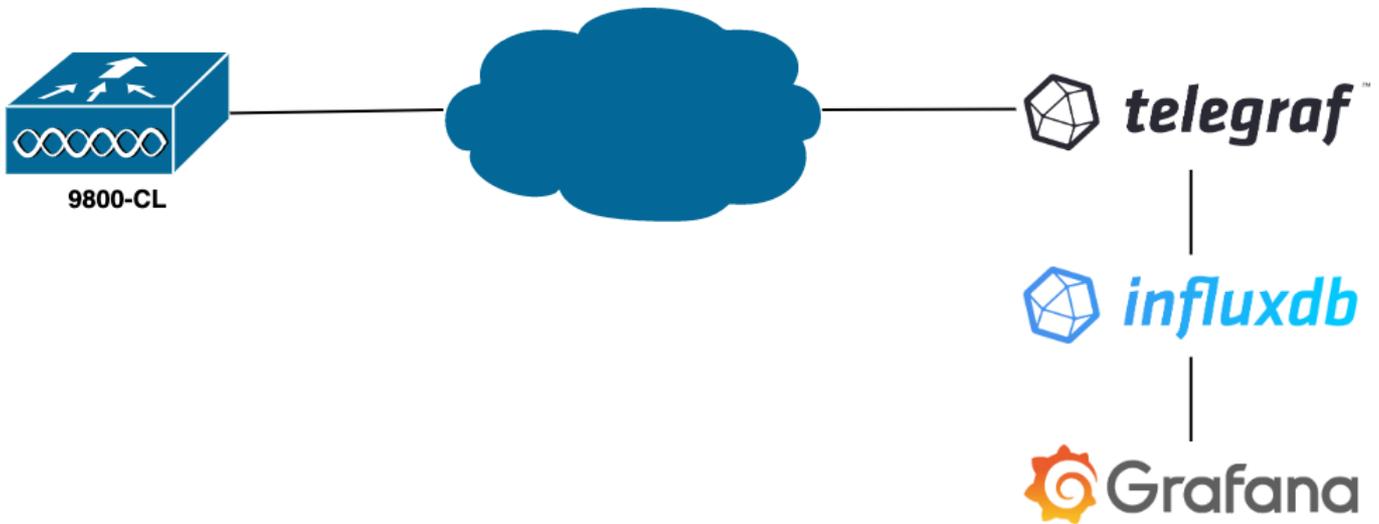
Die Informationen in diesem Dokument basierend auf folgenden Software- und Hardware-Versionen:

- Catalyst 9800-CL (Version 17.12.03)
- Ubuntu (Version 22.04.03)
- InfluxDB (V. 1.06.07).
- Telegraf (V. 1.21.04).
- Grafana (V. 10.2.01).

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle kennen.

Konfigurieren

Netzwerkdiagramm



Konfigurationen

In diesem Beispiel wird die Telemetrie auf einem 9800-CL mit gRPC-Dial-Out konfiguriert, um Informationen einer Telegraf-Anwendung, die sie speichert, in eine InfluxDB-Datenbank zu übertragen. Hier wurden zwei Geräte verwendet,

- Ein Ubuntu-Server, der den gesamten TIG-Stack hostet.
- Ein Catalyst 9800-CL

Der Schwerpunkt dieses Konfigurationsleitfadens liegt nicht auf der vollständigen Bereitstellung dieser Geräte, sondern auf den Konfigurationen, die für jede Anwendung erforderlich sind, damit die 9800-Informationen ordnungsgemäß gesendet, empfangen und dargestellt werden.

Schritt 1: Datenbank vorbereiten

Bevor Sie mit dem Konfigurationsteil beginnen, stellen Sie sicher, dass Ihre Influx-Instanz ordnungsgemäß ausgeführt wird. Wenn Sie eine Linux-Distribution verwenden, ist dies mit dem `systemctl status` Befehl ganz einfach möglich.

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

Damit das Beispiel funktioniert, benötigt Telegraf eine Datenbank, in der die Kennzahlen gespeichert werden, sowie einen Benutzer, der sich mit dieser Datenbank verbindet. Diese können mithilfe der folgenden Befehle auf einfache Weise über die InfluxDB-CLI erstellt werden:

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

Die Datenbank ist nun erstellt, Telegraf kann so konfiguriert werden, dass Metriken darin richtig gespeichert werden.

Schritt 2: Telegraf vorbereiten

Für dieses Beispiel sind nur zwei Telegraf-Konfigurationen interessant. Diese können (wie üblich für unter Unix laufende Anwendungen) aus der `/etc/telegraf/telegraf.conf` Konfigurationsdatei erstellt werden.

Die erste deklariert die von Telegraf verwendete Ausgabe. Wie bereits erwähnt, wird hier InfluxDB verwendet und wie folgt im Ausgabeabschnitt der `telegraf.conf` Datei konfiguriert:

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

Dies weist den Telegraf-Prozess an, die empfangenen Daten in der InfluxDB zu speichern, die auf demselben Host auf Port 8086 ausgeführt wird, und die Datenbank "TELEGRAF" zu verwenden (sowie die Zugangsdaten `telegraf/YOUR_PASSWORD`, um darauf zuzugreifen).

Wenn das erste deklarierte Format das Ausgabeformat war, ist das zweite natürlich das Eingabeformat. Um Telegraf mitzuteilen, dass die empfangenen Daten von einem Cisco Gerät stammen, das Telemetry verwendet, können Sie das [Eingabemodul "cisco telemetry mdt" verwenden](#). Um dies zu konfigurieren, müssen Sie der `/etc/telegraf/telegraf.conf` Datei nur die folgenden Zeilen hinzufügen:

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

Dadurch kann die Telegraf-Anwendung, die auf dem Host (auf dem Standardport 57000) läuft, die vom WLC empfangenen Daten decodieren.

Nachdem Sie die Konfiguration gespeichert haben, starten Sie Telegraf neu, um es auf den Dienst anzuwenden. Stellen Sie außerdem sicher, dass der Dienst ordnungsgemäß neu gestartet wurde:

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

Schritt 3: Telemetry-Abonnement mit der gewünschten Metrik ermitteln

Wie bereits erwähnt, sind die Kennzahlen auf Cisco Geräten wie auf vielen anderen nach dem YANG-Modell organisiert. Die jeweiligen Cisco YANG-Modelle für jede Version von IOS XE (auf dem 9800 verwendet) finden Sie [hier](#), insbesondere die für IOS XE Dublin 17.12.03 in diesem Beispiel.

In diesem Beispiel konzentrieren wir uns auf die Erfassung von CPU-Nutzungsmetriken der verwendeten 9800-CL-Instanz. Durch die Überprüfung des YANG-Modells für Cisco IOS XE Dublin 17.12.03 kann festgestellt werden, welches Modul die CPU-Auslastung des Controllers enthält, insbesondere für die letzten 5 Sekunden. Diese sind Teil des Moduls `Cisco-IOS-XE-process-cpu-oper` unter der CPU-Nutzungsgruppierung (Leaf fünf Sekunden).

Schritt 4: Aktivieren von NETCONF auf dem Controller

Das gRPC-Dial-Out-Framework basiert auf [NETCONF](#), um dasselbe zu erreichen. Daher muss diese Funktion auf dem 9800 aktiviert werden. Dies wird durch die folgenden Befehle erreicht:

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

Schritt 5: Konfigurieren des Telemetrie-Abonnements auf dem Controller

Sobald die [XPath](#)s (*auch bekannt als* XML Paths Language) der Metriken aus dem YANG-Modell ermittelt wurden, kann ein Telemetrie-Abonnement über die 9800-CLI konfiguriert werden, um diese zu der in [Schritt 2](#) konfigurierten Telegraf-Instanz zu streamen. Dazu führen Sie die folgenden Befehle aus:

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

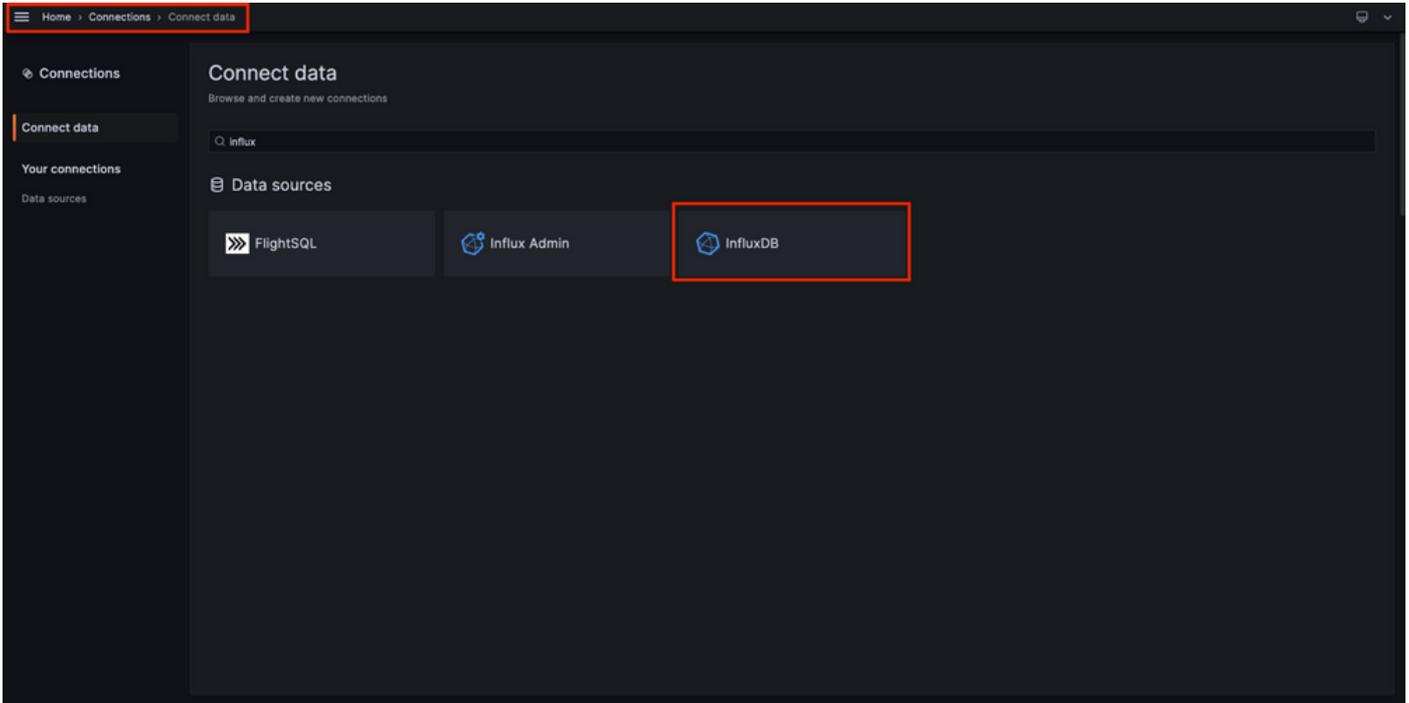
In diesem Codeblock wird zunächst die Telemetrie-Subskription mit der Kennung 101 definiert. Die Abonnement-ID kann eine beliebige Zahl zwischen <0-2147483647> sein, solange sie sich nicht mit einem anderen Abonnement überschneidet. Für dieses Abonnement werden in der folgenden Reihenfolge konfiguriert:

- Die verwendete Kodierungsmethode, die beim Arbeiten mit dem gRPC-Transportprotokoll kvGPB sein muss.
- Der Filter für die Metriken, die vom Abonnement gesendet werden, ist der XPath, der die Metrik definiert, die uns interessiert (zu wissen, /process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/five-seconds).
- Die Quell-IP-Adresse, die vom Controller zum Senden der Kennzahlen verwendet wird.
- Der Stream-Typ, der zur Kommunikation der Metriken verwendet wird, in diesem Fall YANG Push IETF-Standard.
- Die Frequenz, mit der der Controller Daten in 100^{stel} Sekunden an den Teilnehmer sendet. In diesem Fall wurde sie so konfiguriert, dass sie Updates regelmäßig jede Sekunde sendet.
- Die IP-Adresse und die Port-Nummer des Empfängers sowie das Protokoll, das für die Kommunikation zwischen Controller und Teilnehmer verwendet wird. In diesem Beispiel wird gRPC-TCP verwendet, um eine Metrik an den Host 10.48.39.98 an Port 57000 zu senden.

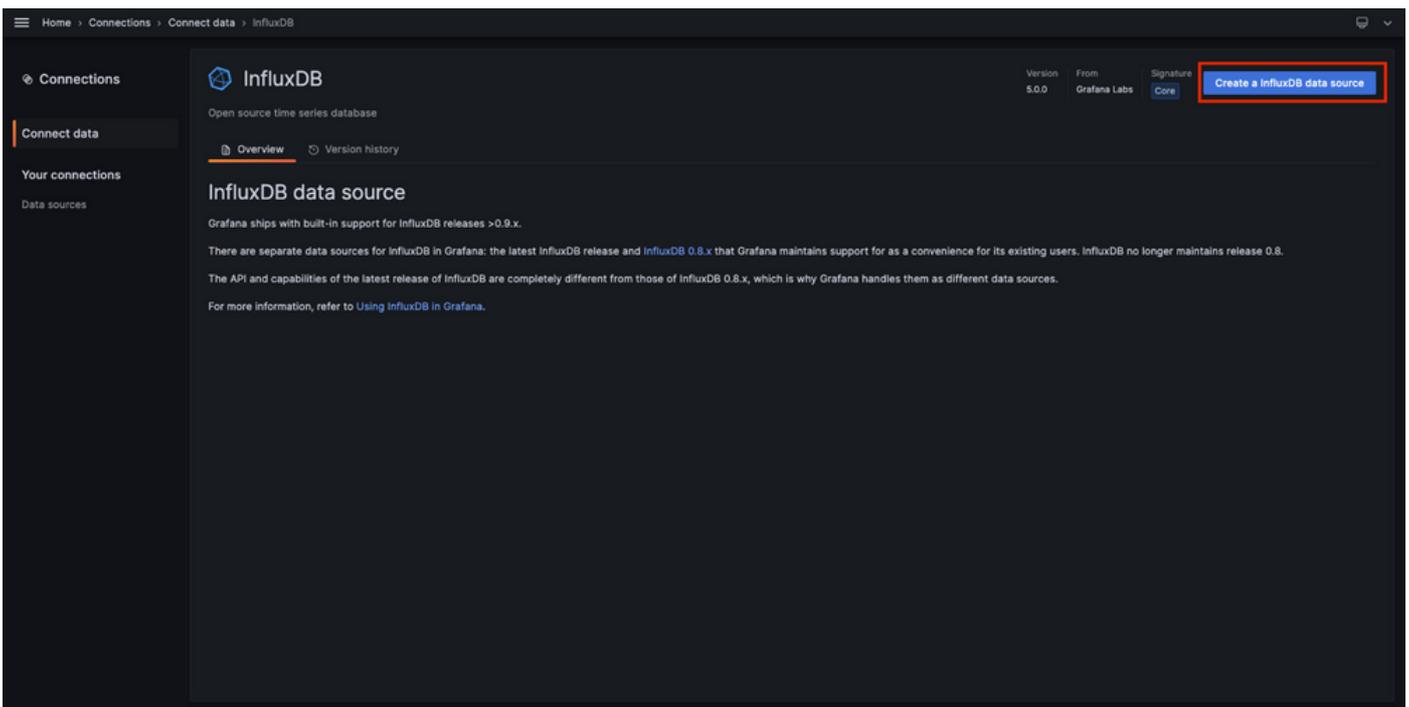
Schritt 6: Grafana-Datenquelle konfigurieren

Nun, da der Controller beginnt, Daten an Telegraf zu senden und diese in der TELEGRAF InfluxDB-Datenbank gespeichert werden, ist es an der Zeit, Grafana so zu konfigurieren, dass es diese Metriken durchsucht.

Navigieren Sie in der Grafana-GUI zu *Home > Connections > Connect data* und suchen Sie mithilfe der Suchleiste nach der Datenquelle InfluxDB.



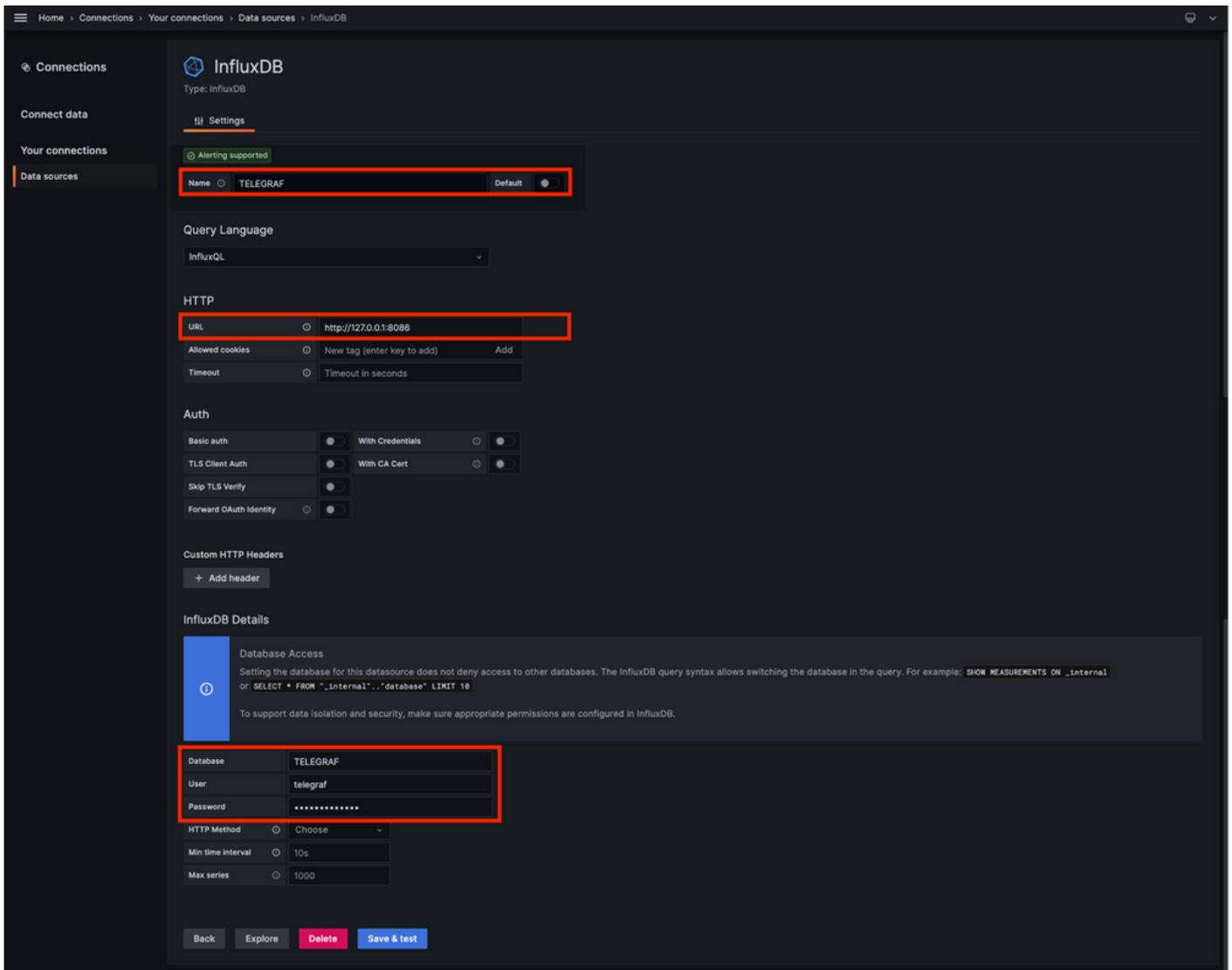
Wählen Sie diesen Datenquellentyp aus und verwenden Sie die Schaltfläche "Eine InfluxDB-Datenquelle erstellen", um Grafana mit der in [Schritt 1](#) erstellten TELEGRAPH-Datenbank zu verbinden.



Füllen Sie das Formular aus, das auf dem Bildschirm angezeigt wird, und stellen Sie insbesondere Folgendes bereit:

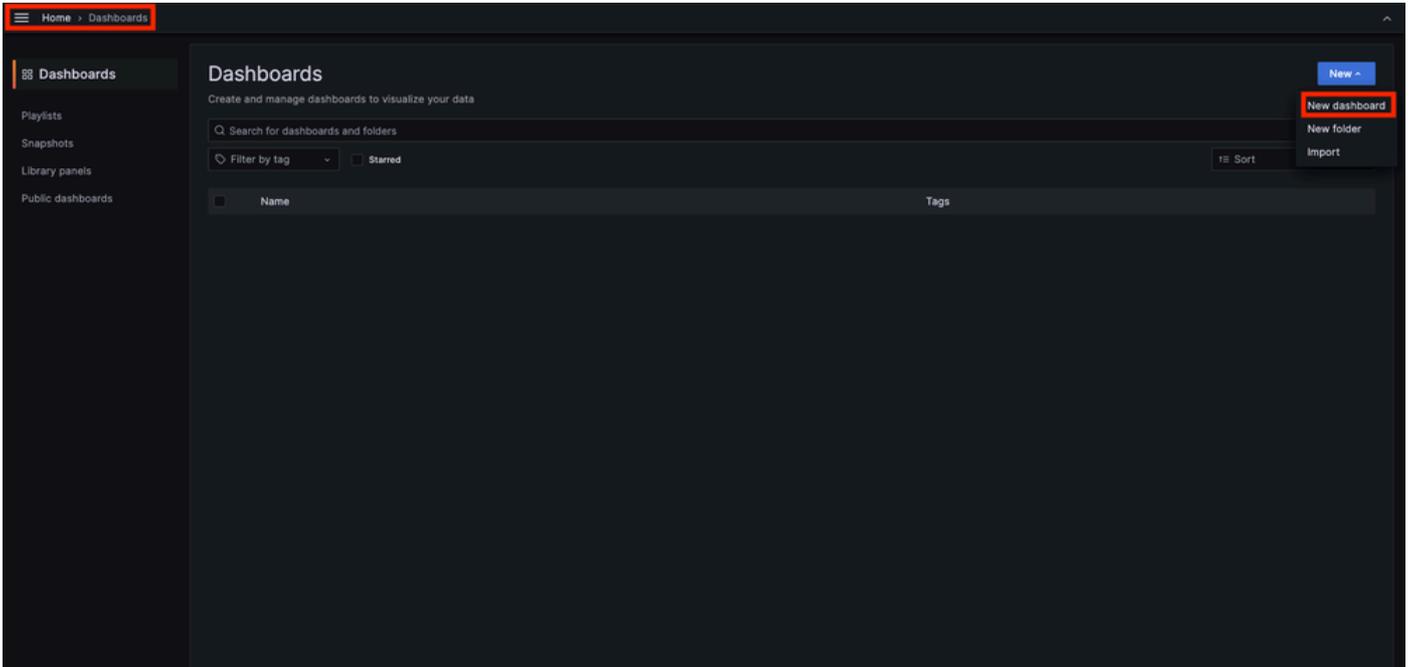
- Ein Name für die Datenquelle.
- Die URL der verwendeten InfluxDB-Instanz.
- Der verwendete Datenbankname (in diesem Beispiel TELEGRAF).
- Die Anmeldeinformationen des Benutzers, der für den Zugriff darauf definiert wurde (in diesem Beispiel

telegraf/YOUR_PASSWORD).

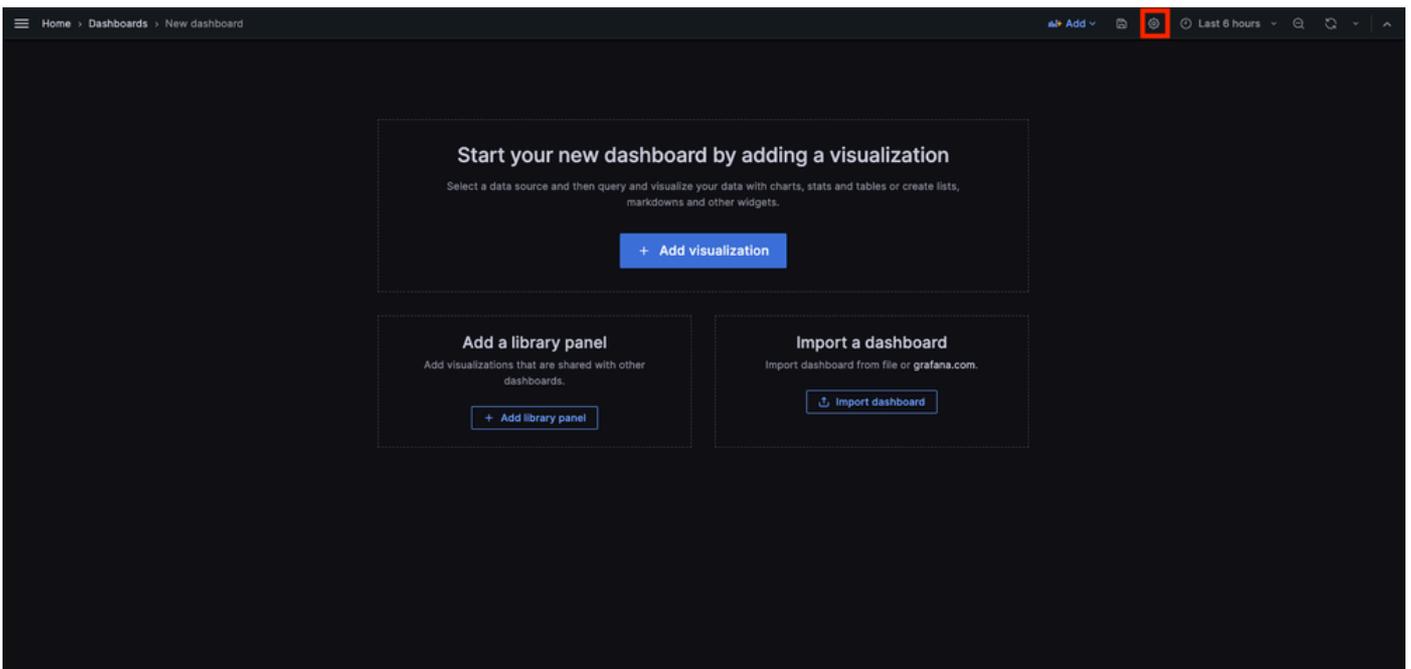


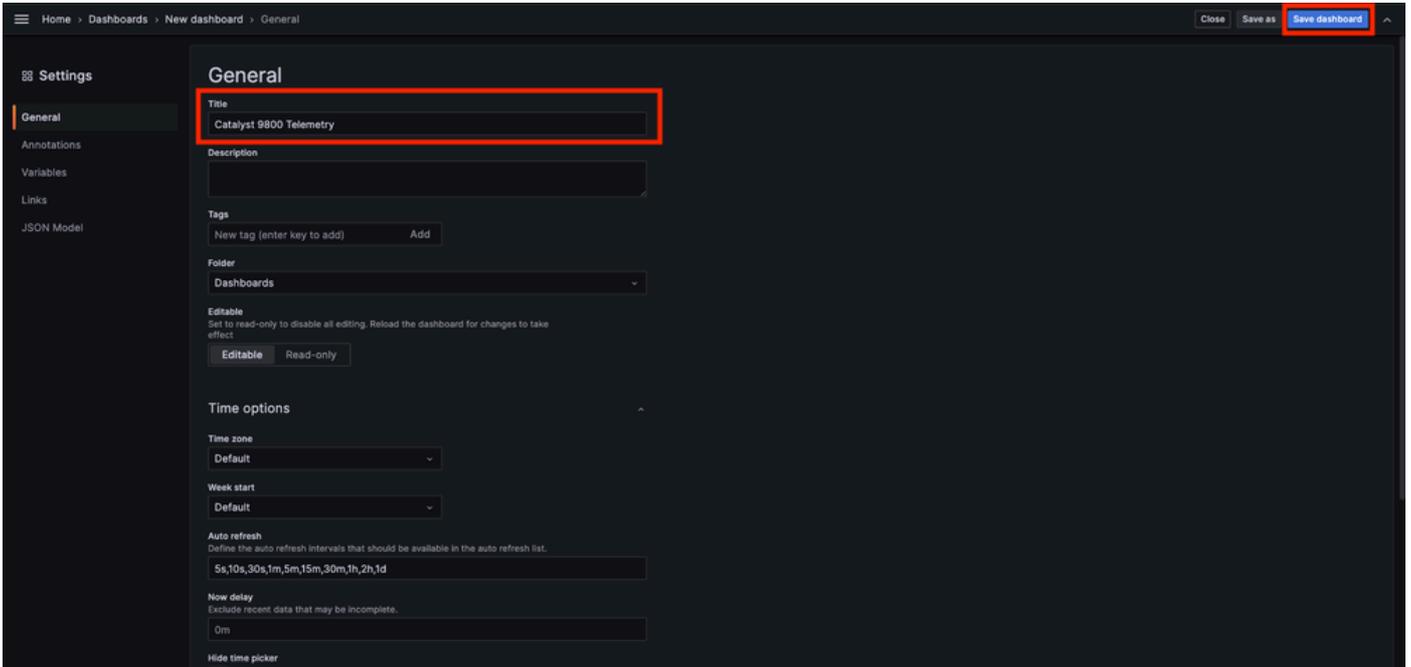
Schritt 7. Dashboard erstellen

Grafana-Visualisierungen sind in *Dashboards* organisiert. Um ein Dashboard mit den Metriken des Catalyst 9800 zu erstellen, navigieren Sie zu *Home > Dashboards*, und drücken Sie auf "New Dashboard" (Neues Dashboard).



Dadurch wird das neu erstellte Dashboard geöffnet. Klicken Sie auf die Zahnradsymbole, um auf den Dashboard-Parameter zuzugreifen und dessen Namen zu ändern. Im Beispiel wird "Catalyst 9800 Telemetry" verwendet. Anschließend können Sie Ihr Dashboard mit der Schaltfläche "Dashboard speichern" speichern.

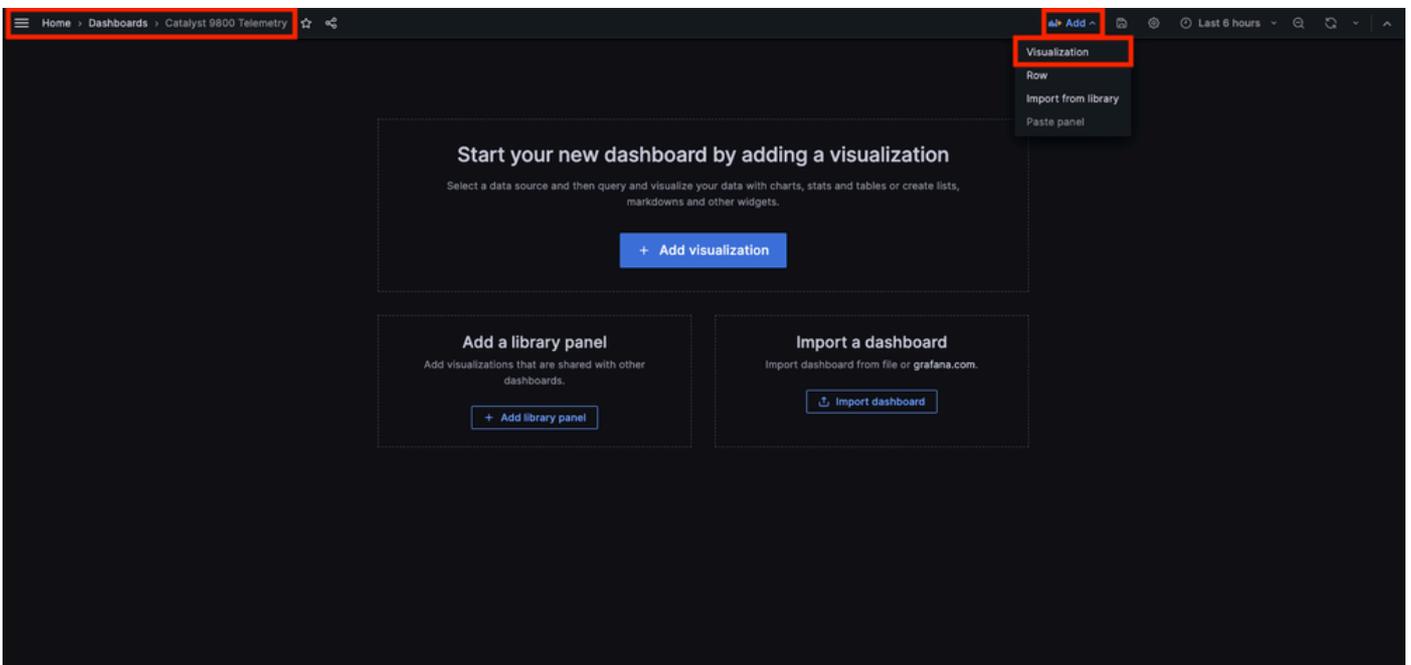




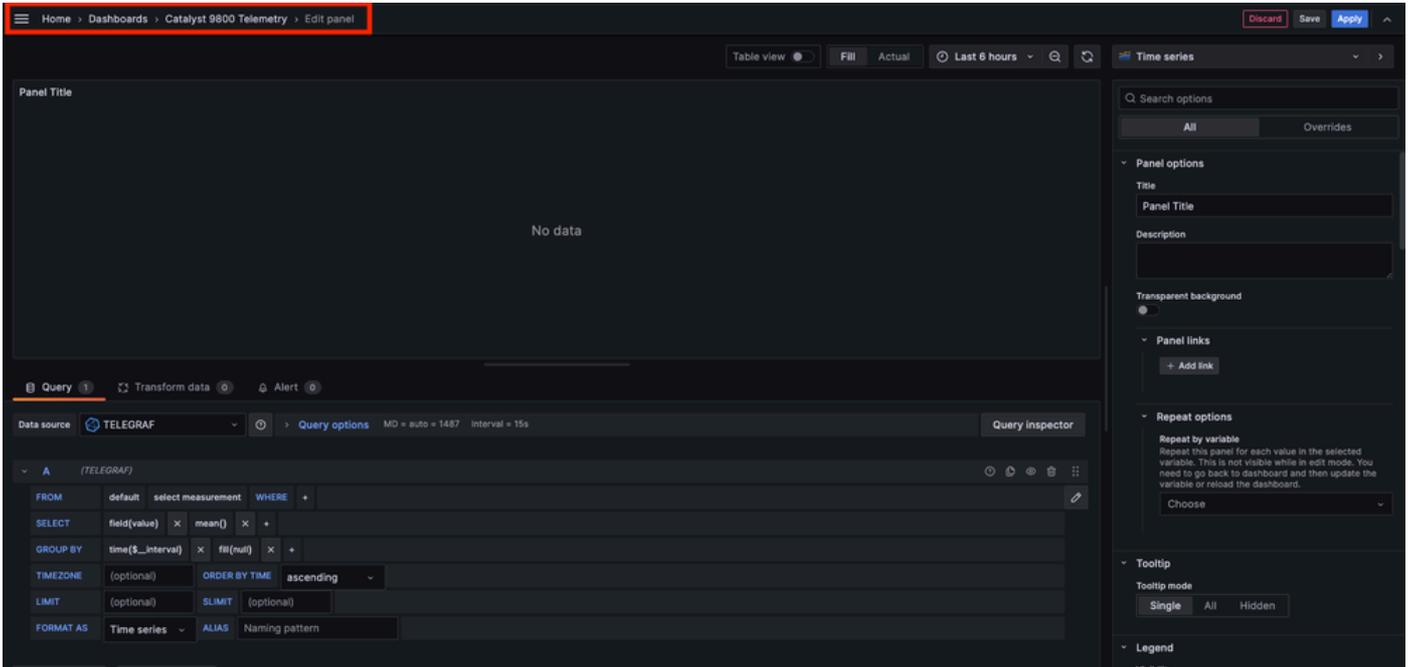
Schritt 8: Hinzufügen einer Visualisierung zum Dashboard

Jetzt, da die Daten korrekt gesendet, empfangen und gespeichert werden und Grafana Zugriff auf diesen Speicherort hat, ist es an der Zeit, eine Visualisierung für sie zu erstellen.

Verwenden Sie in jedem Grafana Dashboard die Schaltfläche "Hinzufügen", und wählen Sie im angezeigten Menü "Visualisierung" aus, um eine Visualisierung Ihrer Metriken zu erstellen.

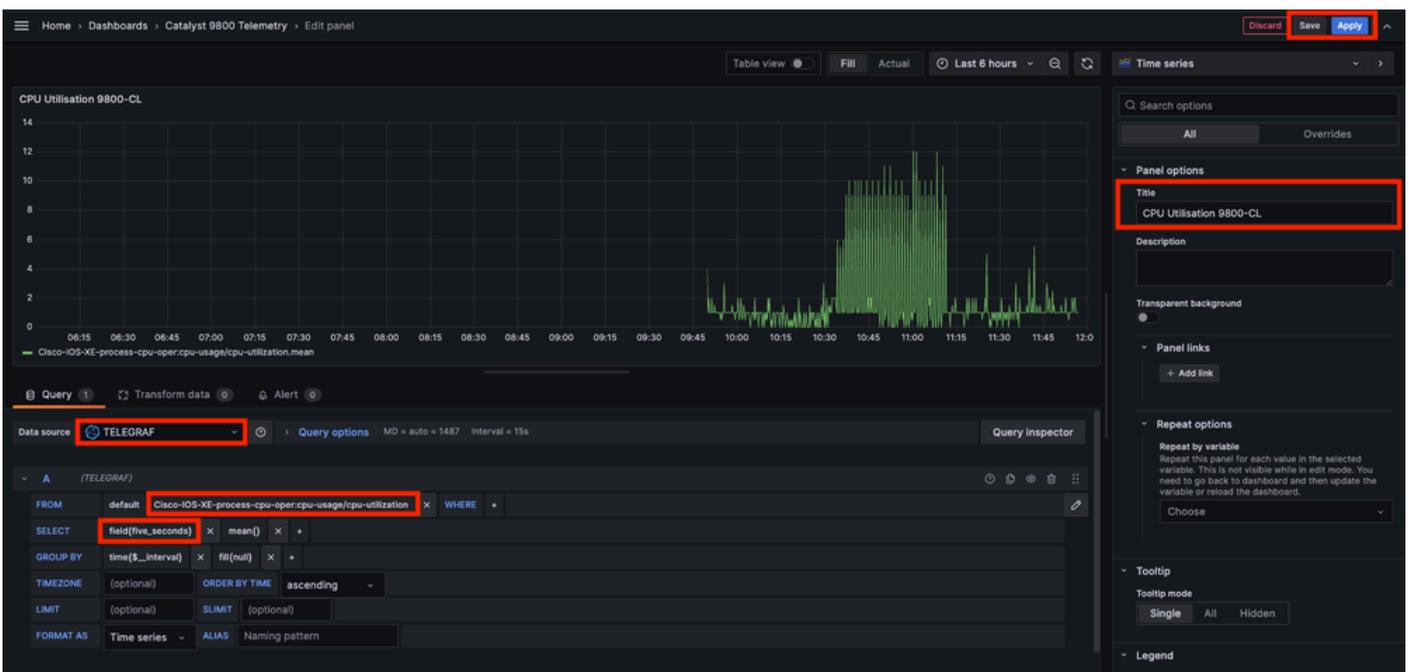


Daraufhin wird der *Bearbeitungsbereich* der erstellten Visualisierung geöffnet:



Wählen Sie in diesem Bereich

- Der Name der Datenquelle, die Sie in [Schritt 6](#) erstellt haben, in diesem Beispiel TELEGRAF.
- Die Messung (Schema), die die Daten enthält, die Sie visualisieren möchten, "Cisco-IOS-XE-process-cpu-oper:cpu-usage/cpu-usage" in diesem Beispiel.
- Das Feld aus der Datenbank, das die Metriken darstellt, die Sie visualisieren möchten, in diesem Beispiel "five_seconds".
- Der Titel der Visualisierung, "CPU Utilisation 9800-CL" in diesem Beispiel.



Sobald Sie die Schaltfläche "Save/Apply" (Speichern/Anwenden) aus der vorherigen Abbildung gedrückt haben, wird die Visualisierung der CPU-Nutzung des Catalyst 9800 Controllers im Laufe der Zeit zum Dashboard hinzugefügt. Die am Dashboard vorgenommenen Änderungen

können mithilfe der Diskette gespeichert werden.



Überprüfung

Konfiguration des WLC wird ausgeführt

```
Building configuration... Current configuration : 112215 bytes ! ! Last configuration change at 14:28:30
```

Telegraf-Konfiguration

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

InfluxDB-Konfiguration

```
### Welcome to the InfluxDB configuration file. reporting-enabled = false [meta] dir = "/var/lib/influxdb2"
```

Grafana-Konfiguration

```
##### Server ##### [server] http_addr = 10.10.10.10
```

Fehlerbehebung

WLC One Stopp Shop Reflex

Was den WLC angeht, ist das erste, was überprüft werden muss, dass die mit den Programmierschnittstellen verbundenen Prozesse in Gang sind.

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

Bei NETCONF (für gRPC-Dial-Out) kann dieser Befehl auch dabei helfen, den Status des Prozesses zu überprüfen.

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

Eine weitere wichtige Prüfung nach Überprüfung des Prozessstatus ist der Status der Telemetrie-Verbindung zwischen dem Catalyst 9800 und dem Telegraf-Empfänger. Sie kann mit dem Befehl "show telemetry connection all" (Telemetriedaten anzeigen) angezeigt werden.

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

Wenn die Telemetrieverbindung zwischen dem WLC und dem Empfänger besteht, kann mithilfe des show telemetry ietf subscription all brief Befehls auch sichergestellt werden, dass die konfigurierten Abonnements gültig sind.

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

Die detaillierte Version dieses Befehls enthält show telemetry ietf subscription all detail weitere Informationen zu Abonnements und kann beim Hinweisen auf ein Problem in der Konfiguration helfen.

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

Netzwerkerreichbarkeit bestätigen

Der Catalyst 9800-Controller sendet gRPC-Daten an den für jedes Telemetrie-Abonnement konfigurierten Empfängerport.

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

Zur Überprüfung der Netzwerkverbindung zwischen dem WLC und dem Empfänger an diesem konfigurierten Port stehen verschiedene Tools zur Verfügung.

Vom WLC aus kann über Telnet auf dem konfigurierten IP/Port des Empfängers (hier 10.48.39.98:57000) überprüft werden, ob dieser offen ist und vom Controller selbst erreicht werden kann. Wenn der Datenverkehr nicht blockiert wird, muss der Port in der Ausgabe als offen angezeigt werden:

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

Alternativ kann [Nmap](#) von einem beliebigen Host verwendet werden, um sicherzustellen, dass der Empfänger am konfigurierten Port richtig angezeigt wird.

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

Protokollierung und Debuggen

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

Erreichbarkeit der Kennzahlen am TIG-Stack

Von InfluxDB CLI

Wie jedes andere Datenbanksystem verfügt auch InfluxDB über eine Kommandozeile, mit der sich überprüfen lässt, ob Metriken von Telegraf korrekt empfangen und in der definierten Datenbank gespeichert werden. InfluxDB organisiert Metriken, so genannte Punkte, in Messungen, die selbst als Serie organisiert sind. Einige der hier vorgestellten Grundbefehle können verwendet werden, um das Datenschema auf InfluxDB-Seite zu überprüfen und sicherzustellen, dass die Daten diese Anwendung erreichen.

Zunächst können Sie überprüfen, ob die Reihen, Messungen und ihre Struktur (Schlüssel) korrekt generiert werden. Diese werden automatisch von Telegraf und InfluxDB generiert, basierend auf der Struktur des verwendeten RPC.



Hinweis: Diese Struktur kann natürlich vollständig über die Konfigurationen Telegraf und InfluxDB angepasst werden. Dies geht jedoch über den Rahmen dieses Konfigurationsleitfadens hinaus.

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

Sobald die Datenstruktur geklärt ist (Integer, String, boolean, ...), kann man die Anzahl der Datenpunkte, die auf diesen Messungen gespeichert werden, basierend auf einem bestimmten Feld abrufen.

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

Wenn die Anzahl der Punkte für ein bestimmtes Feld und der Zeitstempel für den letzten Vorfall ansteigen, ist es ein gutes Zeichen, dass der TIG-Stapel die vom WLC gesendeten Daten korrekt empfängt und speichert.

Von Telegraf

Um zu überprüfen, ob der Telegraf-Empfänger tatsächlich Metriken vom Controller erhält und deren Format überprüft, können Sie die Telegraf-Metriken in eine Ausgabedatei auf dem Host umleiten. Dies kann sehr praktisch sein, wenn es um die Fehlerbehebung bei Geräteverbindungen geht. Um dies zu erreichen, benutzen Sie einfach [das "Datei"-Ausgabemodul](#) von Telegraf, konfigurierbar von der `/etc/telegraf/telegraf.conf`.

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

Referenzen

[Richtlinien zur Hardware-Bedarfsbestimmung](#)

[Grafana-Anforderungen](#)

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.