

Catalyst 9800에서 Telegraf, IntrusionDB 및 Grafana로 고급 gRPC 워크플로 구성

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[구성](#)

[네트워크 다이어그램](#)

[설정](#)

[1단계. 데이터베이스 준비](#)

[2단계. 텔레그래프 준비](#)

[3단계. 원하는 메트릭을 포함하는 텔레메트리 서브스크립션 결정](#)

[4단계. 컨트롤러에서 NETCONF 활성화](#)

[5단계. 컨트롤러에서 텔레메트리 서브스크립션 구성](#)

[6단계. Grafana 데이터 원본 구성](#)

[7단계. 대시보드 만들기](#)

[8단계. 대시보드에 시각화 추가](#)

[다음을 확인합니다.](#)

[WLC 실행 컨피그레이션](#)

[Telegraf 컨피그레이션](#)

[IntrusionDB 컨피그레이션](#)

[Grafana 컨피그레이션](#)

[문제 해결](#)

[WLC 원 스톱 샷 리플렉스](#)

[네트워크 연결성 확인](#)

[로깅 및 디버깅](#)

[메트릭이 TIG 스택에 도달하는지 확인](#)

[IntrusionDB CLI에서](#)

[텔레그래프에서](#)

[참조](#)

소개

이 문서에서는 Telegraf, IntrusionDB 및 TIG(Grafana) 스택을 구축하고 Catalyst 9800과 상호 연결하는 방법에 대해 설명합니다.

사전 요구 사항

이 문서에서는 복잡한 통합을 통해 Catalyst 9800의 프로그래밍 인터페이스 용량을 보여 줍니다. 이

문서에서는 이러한 기능을 필요에 따라 완벽하게 사용자 정의할 수 있고 매일 시간을 절약할 수 있는 방법을 보여 줍니다. 여기에 소개된 구축은 gRPC를 기반으로 하며 텔레메트리 컨피그레이션을 통해 Catalyst 9800의 무선 데이터를 모든 Telegraf, InfluxDB, TIG(Grafana) 관측 가능 스택에서 사용할 수 있도록 합니다.

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- Catalyst Wireless 9800 구성 모델.
- 네트워크 프로그래밍 기능 및 데이터 모델.
- TIG 스택 기본 사항.

사용되는 구성 요소

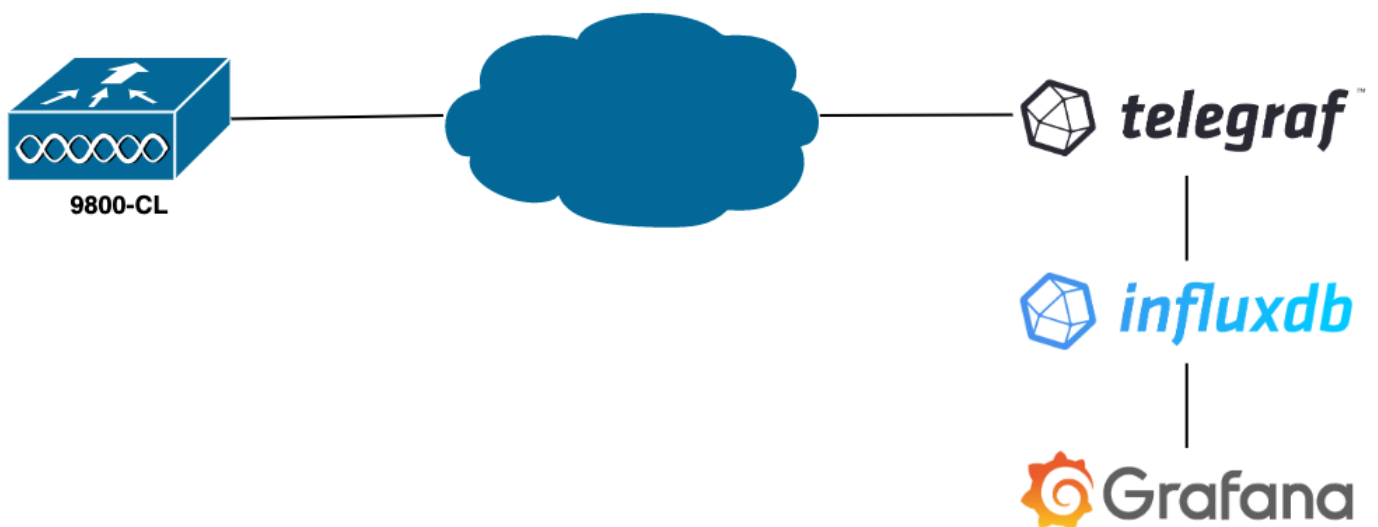
이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

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

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

구성

네트워크 다이어그램



설정

이 예에서는 gRPC 다이얼아웃을 사용하여 9800-CL에서 텔레메트리를 구성하여 텔레그래프 애플리케이션에 대한 정보를 IntrusionDB 데이터베이스에 푸시합니다. 여기에서는 두 개의 디바이스가 사용되었고

- 전체 TIG 스택을 호스팅하는 Ubuntu 서버입니다.
- Catalyst 9800-CL

이 컨피그레이션 가이드는 이러한 디바이스의 전체 구축에 초점을 맞추는 것이 아니라 9800 정보를 적절하게 보내고 받고 제공하기 위해 각 애플리케이션에 필요한 컨피그레이션에 중점을 둡니다.

1단계. 데이터베이스 준비

컨피그레이션 부분으로 들어가기 전에 Intrusion 인스턴스가 제대로 실행되고 있는지 확인하십시오. 이 작업은 Linux 배포판을 `systemctl status` 사용하는 경우 명령을 사용하여 쉽게 수행할 수 있습니다.

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

예를 들어, Telegraf는 메트릭을 저장할 데이터베이스뿐만 아니라 이 메트릭에 연결할 사용자도 필요합니다. 이러한 파일은 IntrusionDB CLI에서 다음 명령을 사용하여 쉽게 생성할 수 있습니다.

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

이제 데이터베이스가 생성되면 메트릭을 올바르게 저장하도록 Telegraf를 구성할 수 있습니다.

2단계. 텔레그래프 준비

이 예에서는 두 가지 Telegraf 컨피그레이션만 가능합니다. 이러한 파일은 `/etc/telegraf/telegraf.conf` 구성 파일에서 만들 수 있습니다 (Unix에서 실행 중인 응용 프로그램의 경우 일반적으로 사용).

첫 번째 명령은 Telegraf에서 사용되는 출력을 선언합니다. 앞에서 설명한 대로 InfluenceDB는 여기에서 사용되며 다음과 같이 파일의 출력 `telegraf.conf` 섹션에서 구성됩니다.

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

이렇게 하면 Telegraf 프로세스가 수신 데이터를 포트 8086의 동일한 호스트에서 실행 중인 IntrusionDB에 저장하고 "TELEGRAF"라는 데이터베이스(및 이 데이터베이스에 액세스하기 위한 자격 증명 `telegraf/YOUR_PASSWORD`)를 사용하도록 지시합니다.

선언된 첫 번째 것이 출력 형식이었다면 두 번째 것은 물론 입력 형식입니다. Telegraf에서 수신하는 데이터가 텔레메트리를 사용하는 Cisco 디바이스에서 온다는 것을 알려주기 위해 [cisco telemetry mdt](#) 입력 모듈을 사용할 수 있습니다. 이 설정을 구성하려면 파일에 다음 행만 추가하면 `/etc/telegraf/telegraf.conf` 됩니다.

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

이렇게 하면 호스트에서 실행 중인 Telegraf 애플리케이션(기본 포트 57000)이 WLC에서 수신되는 데이터를 디코딩할 수 있습니다.

컨피그레이션이 저장되면 Telegraf를 다시 시작하여 서비스에 적용해야 합니다. 또한 서비스가 올바르게 다시 시작되었는지 확인하십시오.

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

3단계. 원하는 메트릭을 포함하는 텔레메트리 서브스크립션 결정

앞서 언급했듯이 다른 많은 Cisco 디바이스에서와 마찬가지로 메트릭은 YANG 모델에 따라 구성됩니다. IOS XE(9800에서 사용)의 각 버전에 대한 특정 Cisco YANG 모델은 [여기](#)에서 확인할 수 있습니다. 특히 이 예에서 사용된 IOS XE Dublin 17.12.03에 대한 모델은 여기에 나와 있습니다.

이 예에서는 사용된 9800-CL 인스턴스에서 CPU 사용률 메트릭을 수집하는 데 중점을 둡니다. Cisco IOS XE Dublin 17.12.03용 YANG 모델을 검사하여 어떤 모듈에 컨트롤러의 CPU 사용률이 포함되어 있는지, 특히 최근 5초 동안 확인할 수 있습니다. 이는 Cisco-IOS-XE-process-cpu-oper 모듈의 일부로서 cpu-utilization 그룹화(리프 5초) 아래에 있습니다.

4단계. 컨트롤러에서 NETCONF 활성화

gRPC 다이얼아웃 프레임워크는 NETCONF를 사용하여 동일하게 작동합니다. 따라서 9800에서 이 기능을 활성화해야 하며, 이 작업은 다음 명령을 실행하여 수행됩니다.

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

5단계. 컨트롤러에서 텔레메트리 서브스크립션 구성

YANG 모델에서 결정된 메트릭의 XPath(즉, XML Paths Language)가 되면 9800 CLI에서 텔레메트리 서브스크립션을 쉽게 구성하여 [2단계](#)에서 구성한 텔레그래프 인스턴스로 스트리밍을 시작할 수 있습니다. 이 작업은 다음 명령을 실행하여 수행합니다.

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

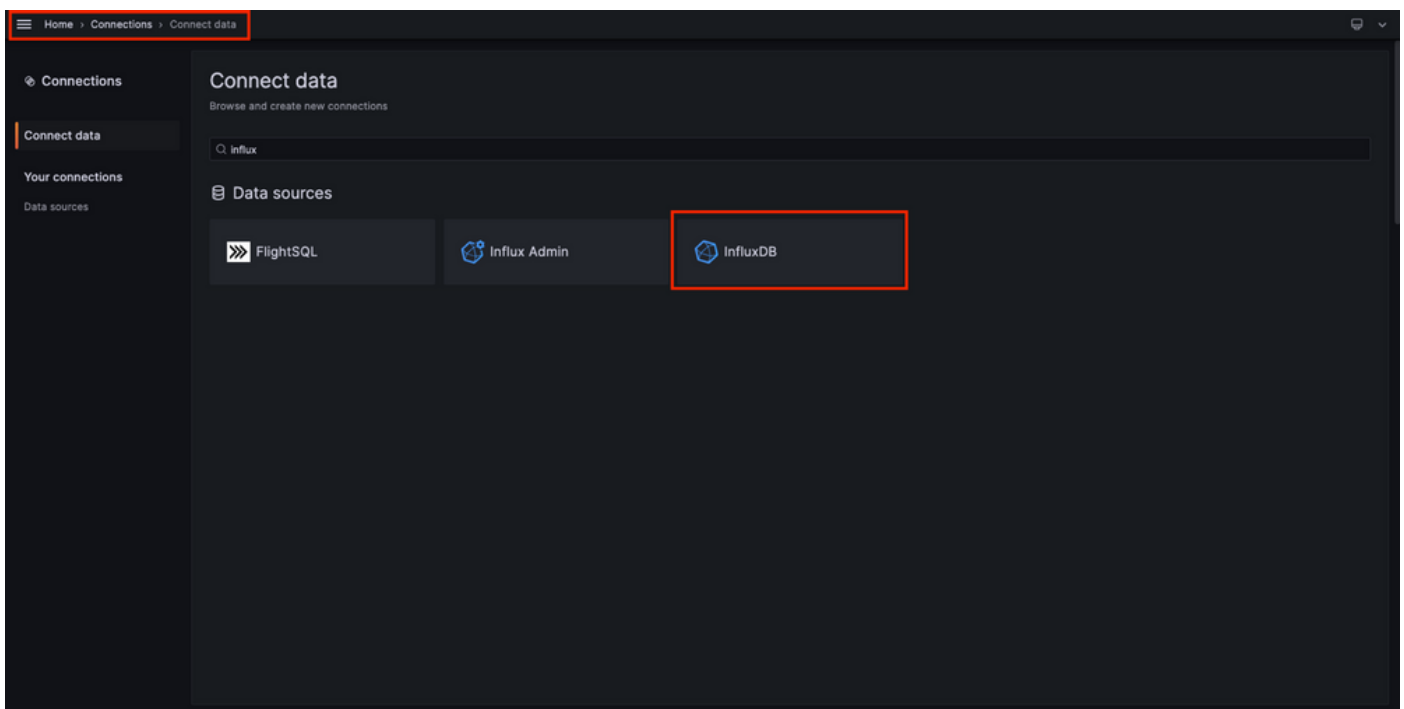
이 코드 블록에서, 먼저 식별자 101을 갖는 텔레메트리 서브스크립션이 정의된다. 구독 식별자는 다른 구독과 겹치지 않는 한 <0-2147483647> 사이의 숫자일 수 있습니다. 이 서브스크립션은 다음 순서로 구성됩니다.

- 사용된 인코딩 방법. gRPC 전송 프로토콜과 작업할 때 kvGPB여야 합니다.
- /process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/five-seconds
- 서브스크립션에서 보낸 메트릭에 대한 필터로, XPath는 관심 있는 메트릭을 정의합니다(알아야 함).
 - 컨트롤러가 메트릭을 전송하는 데 사용하는 소스 IP 주소입니다.
 - 메트릭을 통신하는 데 사용되는 스트림 유형입니다(이 경우 YANG Push IETF 표준).
 - 컨트롤러가 100초 내에 데이터를 가입자에게 보내는 데 사용되는 빈도입니다. 이 경우 1초마다 주기적으로 업데이트를 전송하도록 구성되었습니다.
 - 컨트롤러와 가입자 간의 통신에 사용되는 프로토콜 및 수신기 IP 주소와 포트 번호입니다. 이 예에서 gRPC-TCP는 포트 57000의 호스트 10.48.39.98에 메트릭을 전송하는 데 사용됩니다.

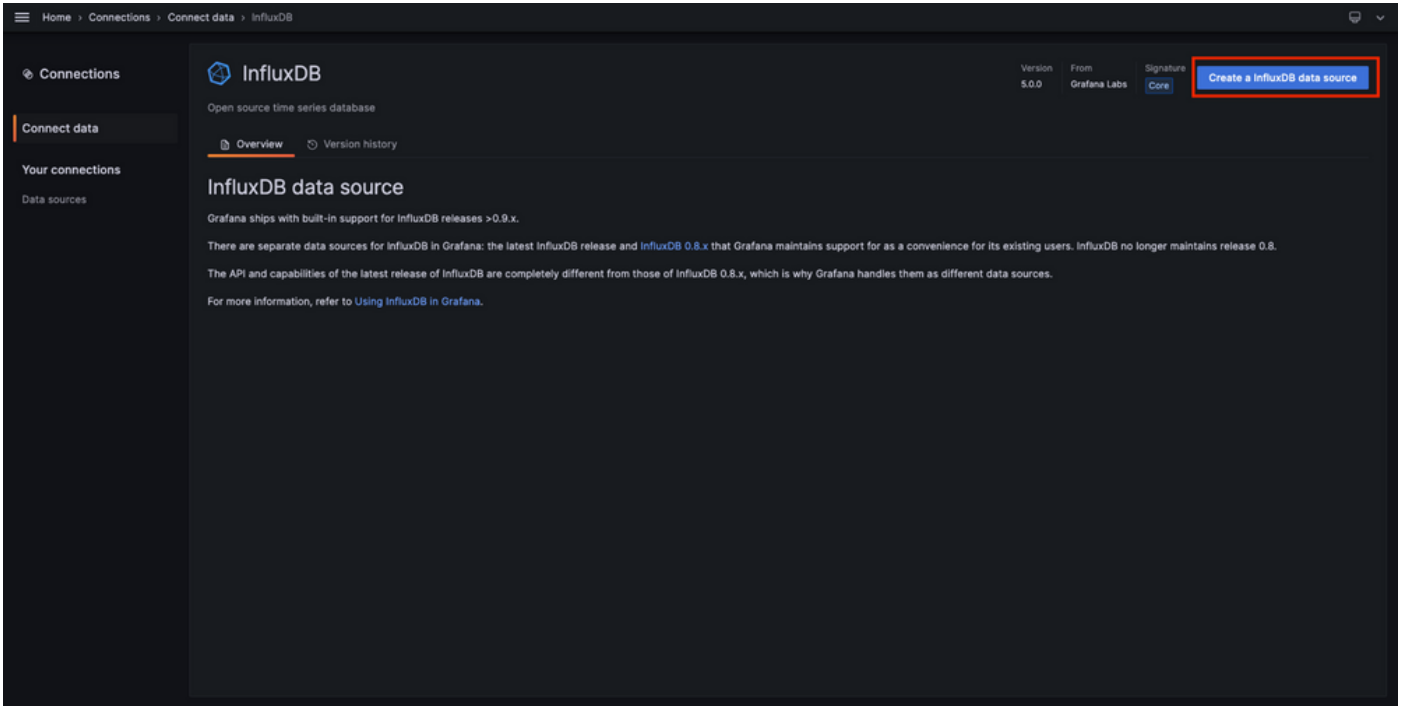
6단계. Grafana 데이터 원본 구성

이제 컨트롤러가 Telegraf에 데이터를 전송하기 시작하고 이러한 데이터가 TELEGRAF InfluxDB 데이터베이스에 저장되므로 Grafana에서 이러한 메트릭을 찾아보도록 구성해야 합니다.

Grafana GUI에서 *Home(홈) > Connections(연결) > Connect data(데이터 연결)*로 이동하고 검색 막대를 사용하여 InfluxDB 데이터 소스를 찾습니다.

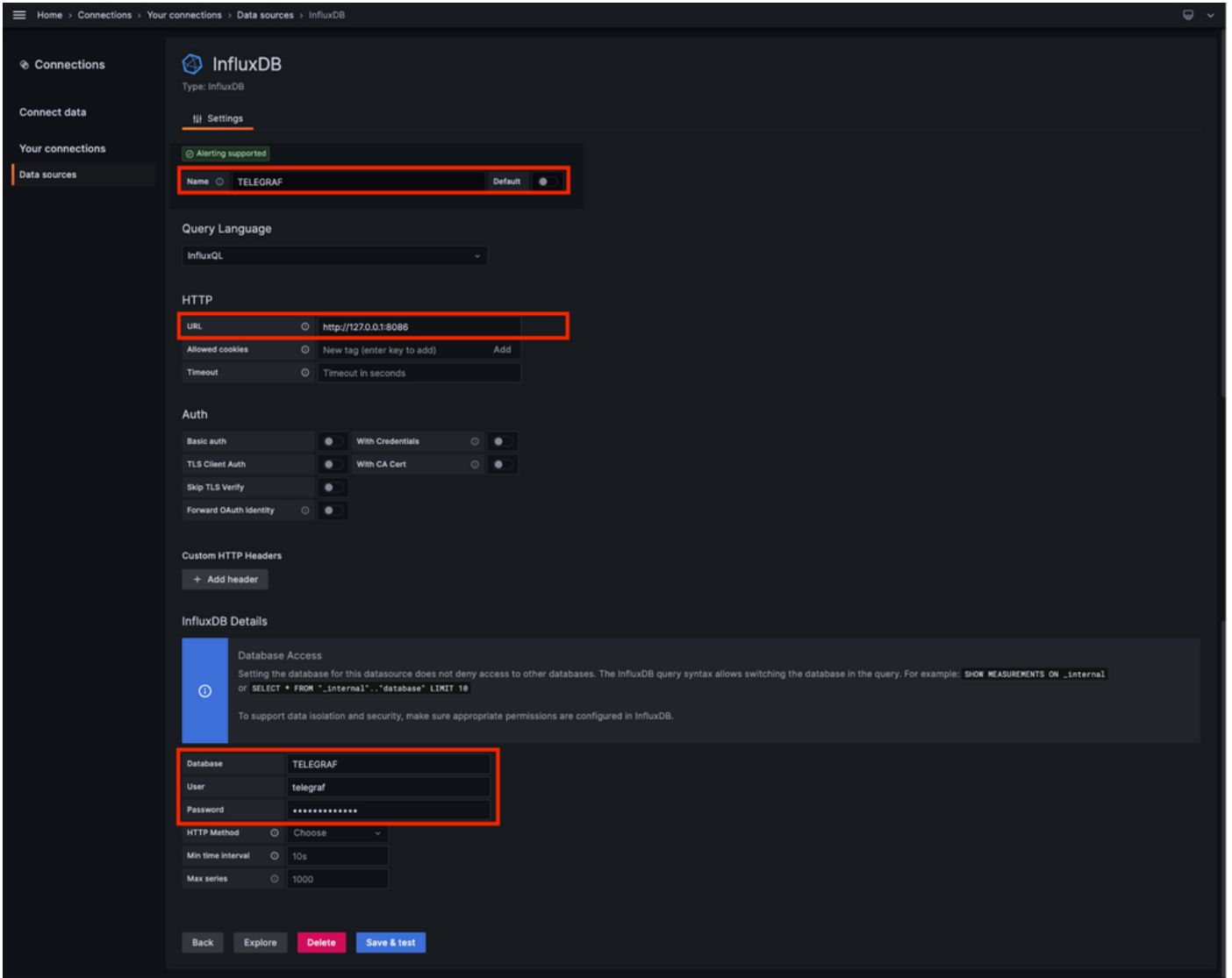


이 데이터 소스 유형을 선택하고 "InfluxDB 데이터 소스 생성" 단추를 사용하여 [1단계](#)에서 생성한 TELEGRAPH 데이터베이스와 Grafana를 연결합니다.



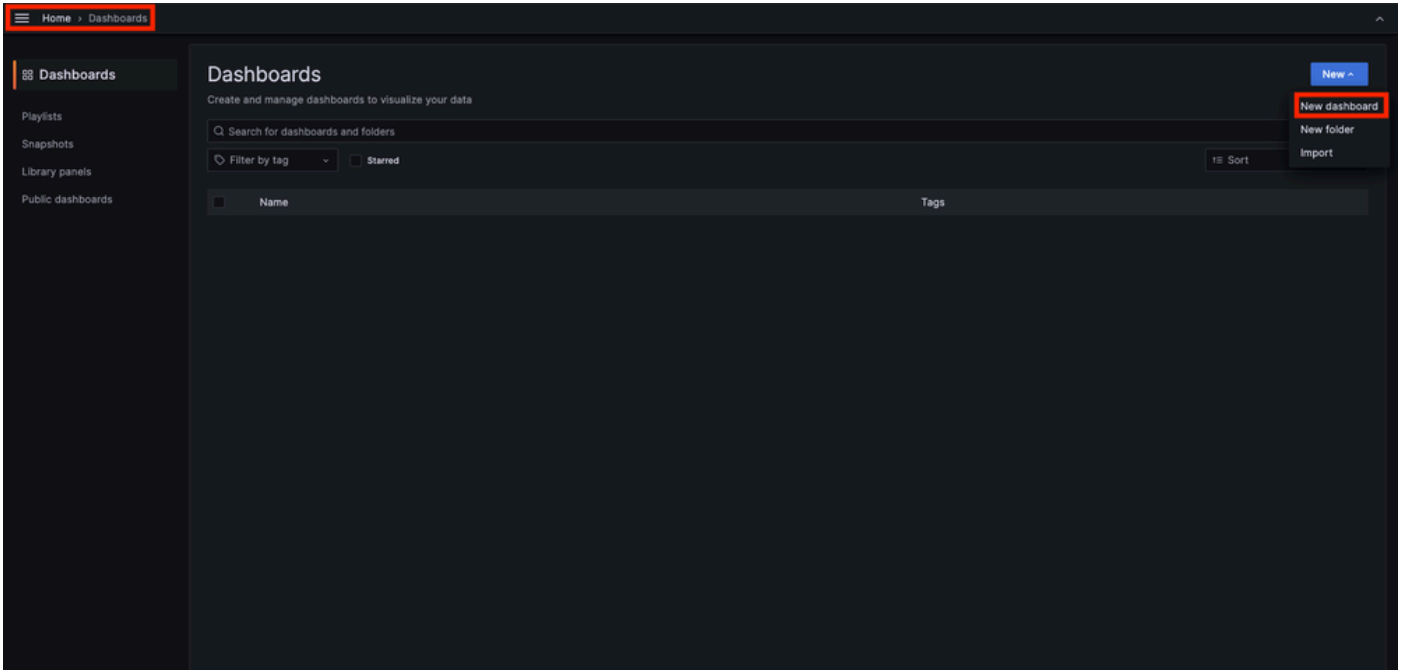
화면에 나타나는 양식을 작성하며, 특히 다음을 제공합니다.

- 데이터 소스의 이름입니다.
- 사용된 InfluxDB 인스턴스의 URL입니다.
- 사용되는 데이터베이스 이름입니다(이 예에서는 "TELEGRAF").
- 액세스하기 위해 정의된 사용자의 자격 증명(이 예에서는 telegraf/YOUR_PASSWORD).

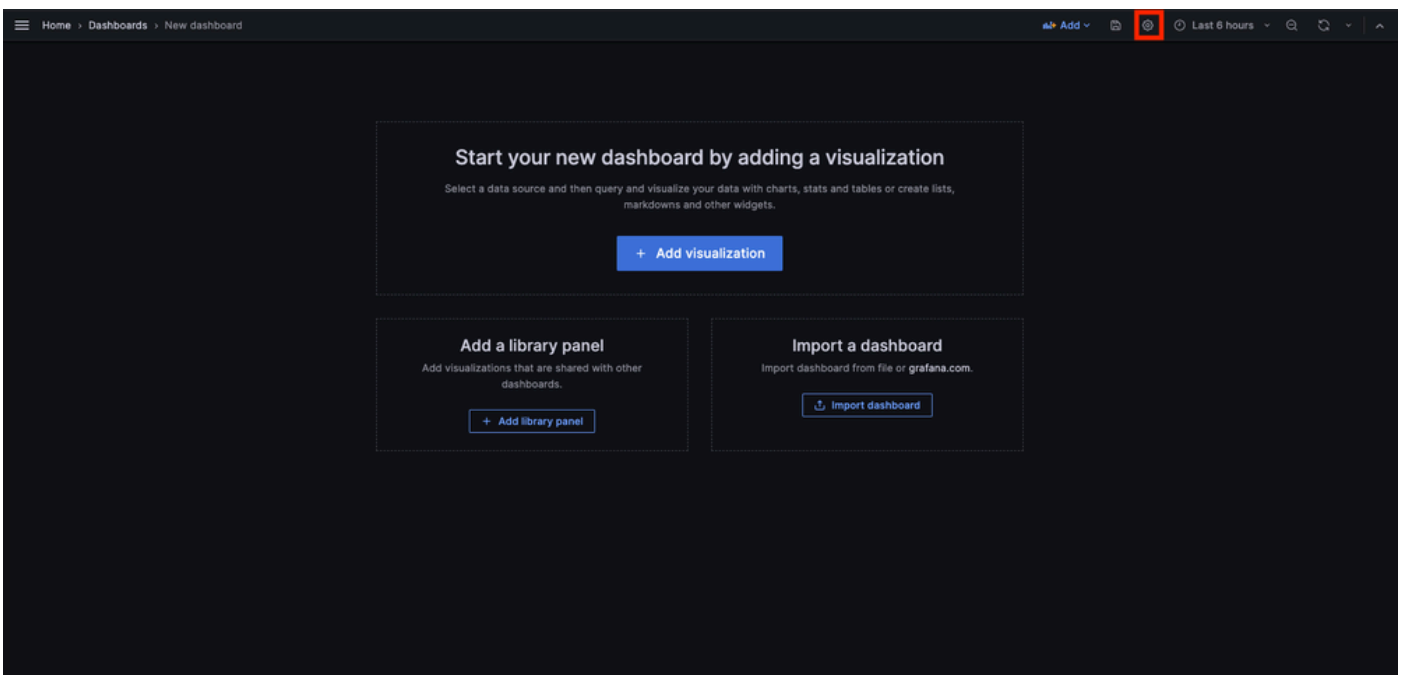


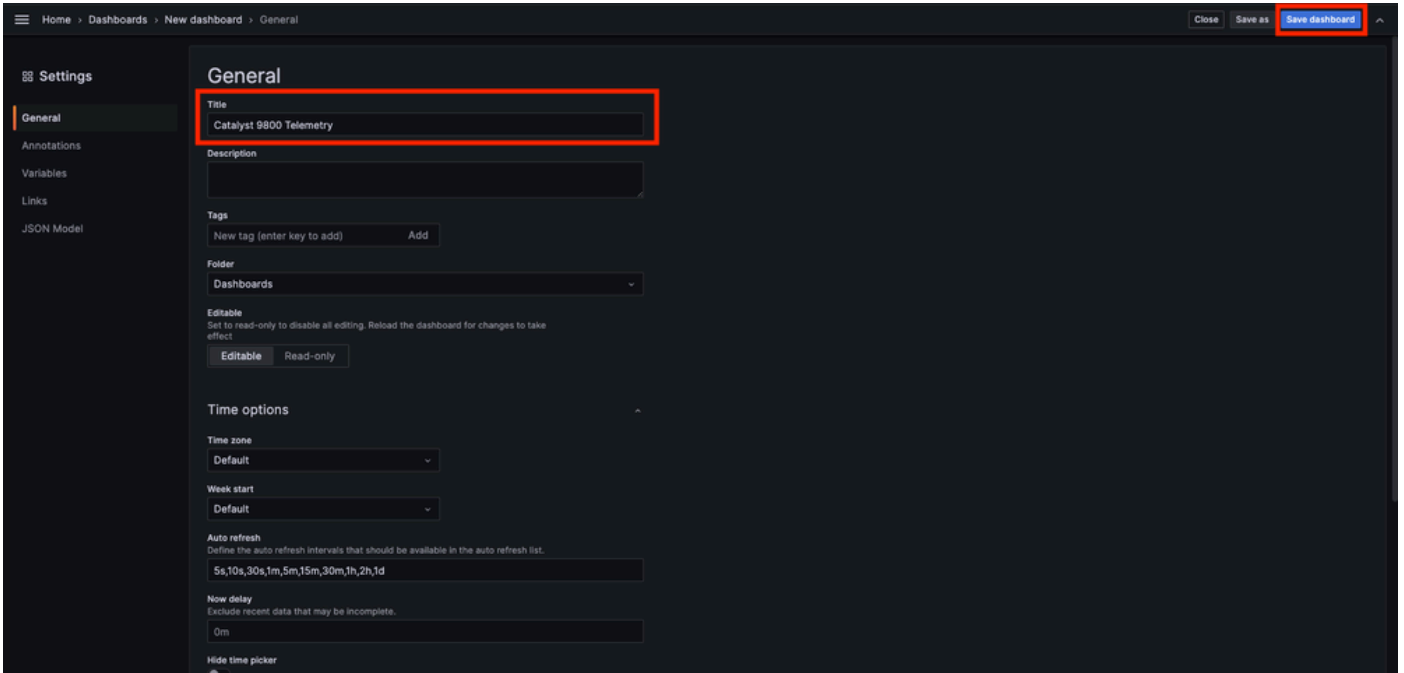
7단계. 대시보드 만들기

Grafana 시각화는 대시보드로 구성됩니다. Catalyst 9800 메트릭스 시각화가 포함된 대시보드를 생성하려면 *Home(홈)* > *Dashboards(대시보드)*로 이동하여 "New dashboard(새 대시보드)" 버튼을 사용합니다



그러면 생성된 새 대시보드가 열립니다. 기어 아이콘을 클릭하여 대시보드 매개변수에 액세스하고 이름을 변경합니다. 이 예에서는 "Catalyst 9800 Telemetry"가 사용됩니다. 이 작업을 수행한 후에는 "Save dashboard(대시보드 저장)" 버튼을 사용하여 대시보드를 저장합니다.

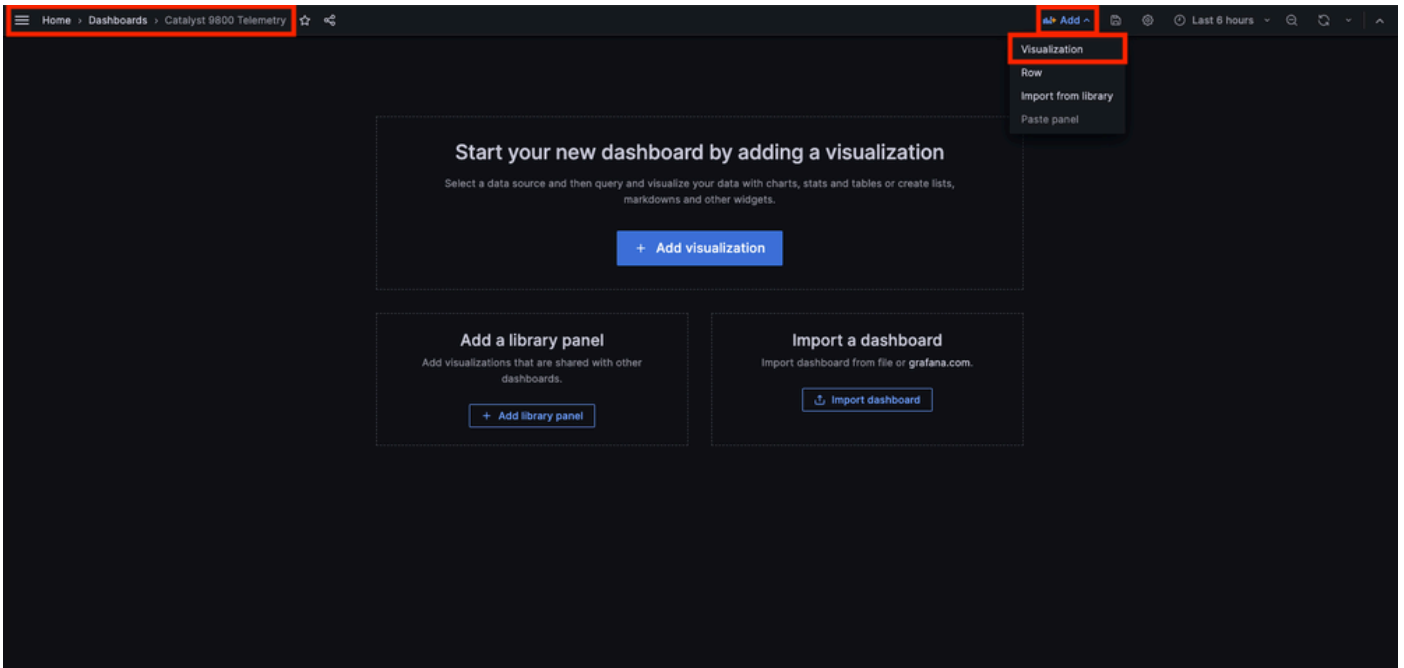




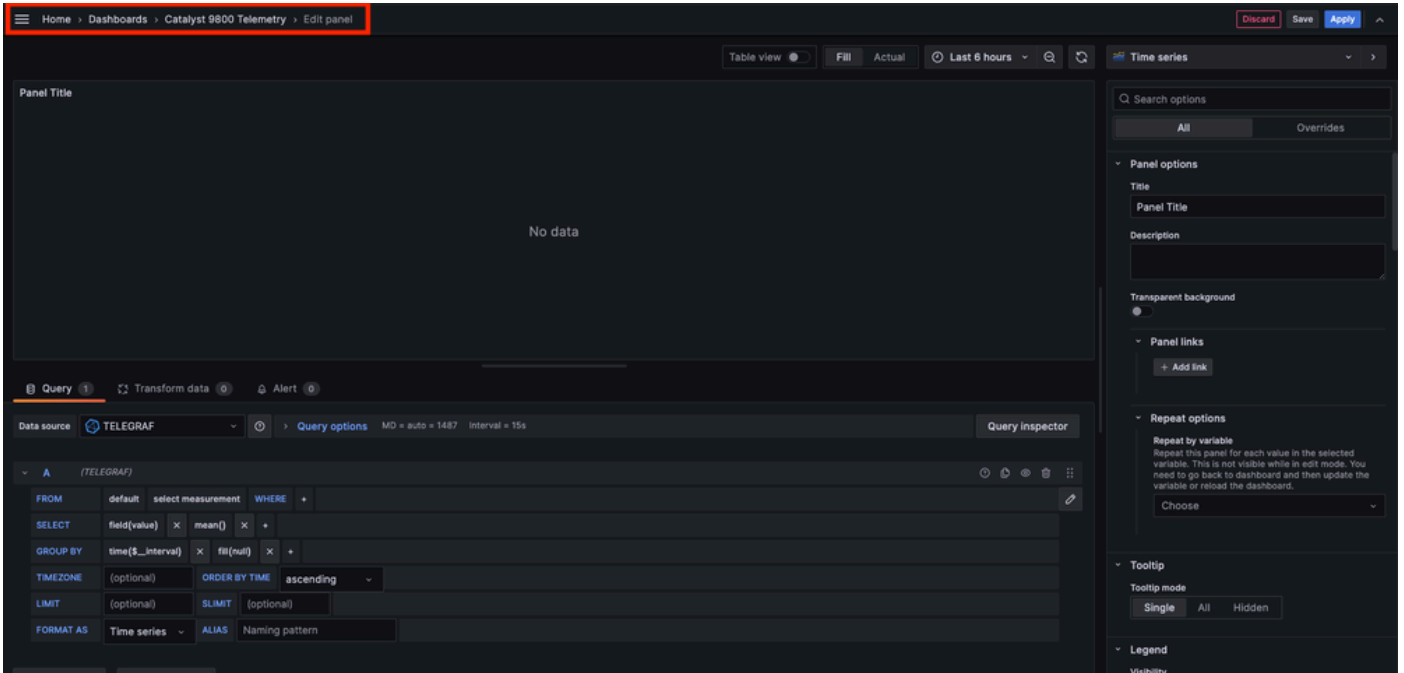
8단계. 대시보드에 시각화 추가

이제 데이터가 제대로 전송, 수신 및 저장되었고 Grafana가 이 저장 위치에 액세스할 수 있게 되었으므로 데이터를 시각화할 차례입니다.

Grafana 대시보드에서 "Add(추가)" 버튼을 사용하고 표시되는 메뉴에서 "Visualization(시각화)"을 선택하여 메트릭의 시각화를 생성합니다.

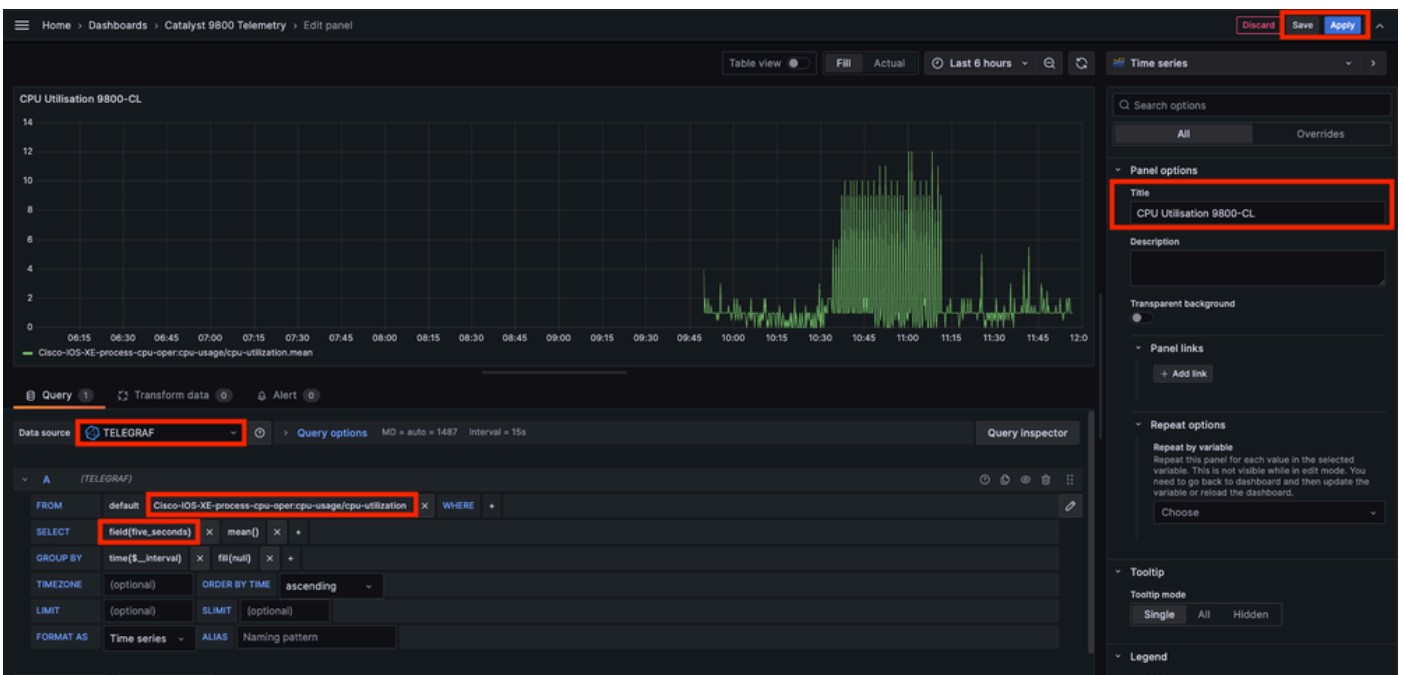


이렇게 하면 생성된 시각 형상의 [편집] 패널이 열립니다.

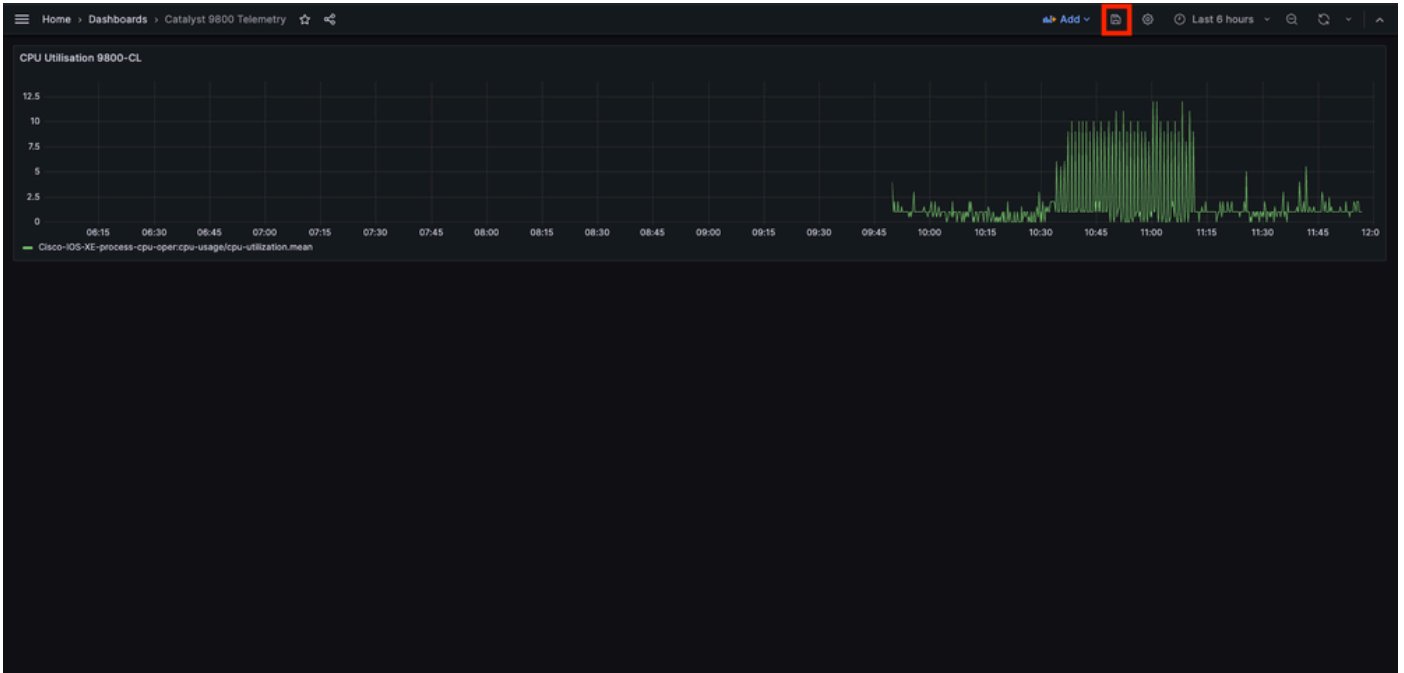


이 패널에서

- [6단계](#), 이 예에서는 TELEGRAF에서 생성한 데이터 소스의 이름입니다.
- 이 예에서 시각화할 데이터, "Cisco-IOS-XE-process-cpu-oper:cpu-usage/cpu-utilization"이 포함된 측정(스키마)
- 이 예에서 시각화할 메트릭을 나타내는 데이터베이스의 필드, "five_seconds"입니다.
- 이 예에서 시각화의 제목, "CPU Utilization 9800-CL"입니다.



이전 그림의 "Save/Apply(저장/적용)" 버튼을 누르면 시간 경과에 따른 Catalyst 9800 컨트롤러의 CPU 사용량을 보여주는 시각화가 대시보드에 추가됩니다. 대시보드의 변경 사항은 플로피 디스크 아이콘 버튼을 사용하여 저장할 수 있습니다.



다음을 확인합니다.

WLC 실행 컨피그레이션

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

Telegraf 컨피그레이션

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

IntrusionDB 컨피그레이션

```
### Welcome to the InfluxDB configuration file. reporting-enabled = false [meta] dir = "/var/lib/influxdb" data_dir = "/var/lib/influxdb/data" log_file = "/var/log/influxdb/meta.log" log_level = "info" [http] bind_address = "0.0.0.0" bind_port = 8086 [http_auth] auth_basic = "InfluxDB: foo" auth_basic_user_file = "/etc/passwd" [http_listener] [server] http_addr = "0.0.0.0" http_port = 8086 [http_listener] [server] http_addr = "0.0.0.0" http_port = 8086 [http_listener] [server] http_addr = "0.0.0.0" http_port = 8086
```

Grafana 컨피그레이션

```
##### Server ##### [server] http_addr = "0.0.0.0" http_port = 3000
```

문제 해결

WLC 원 스톱 샵 리플렉스

WLC 측에서 가장 먼저 확인해야 할 것은 프로그래밍 인터페이스와 관련된 프로세스가 실행 중이라는 것입니다.

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

NETCONF(gRPC 다이얼아웃에서 사용)의 경우 이 명령은 프로세스의 상태를 확인하는 데에도 도움이 됩니다.

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

프로세스 상태를 확인했다면 또 다른 중요한 확인은 Catalyst 9800과 Telegraf 수신기 간의 텔레메트리 연결 상태입니다. "show telemetry connection all" 명령을 사용하여 볼 수 있습니다.

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

WLC와 수신기 간에 텔레메트리 연결이 설정된 경우 명령을 사용하여 구성된 서브스크립션이 유효한지 확인할 수도 show telemetry ietf subscription all brief 있습니다.

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

이 명령의 세부 버전에서는 서브스크립션에 show telemetry ietf subscription all detail대한 자세한 정보를 제공하며 컨피그레이션에서 문제를 지적하는 데 도움이 될 수 있습니다.

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

네트워크 연결성 확인

Catalyst 9800 컨트롤러는 각 텔레메트리 서브스크립션에 대해 구성된 수신기 포트로 gRPC 데이터를 전송합니다.

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

이 구성된 포트에서 WLC와 수신기 간의 네트워크 연결을 확인하기 위해 몇 가지 도구를 사용할 수 있습니다.

WLC에서 구성된 수신기 IP/포트(여기서는 10.48.39.98:57000)에서 텔넷을 사용하여 이 디바이스가 열려 있고 컨트롤러 자체에서 연결 가능한지 확인할 수 있습니다. 트래픽이 차단되고 있지 않으면 포트가 출력에 open으로 표시되어야 합니다.

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

또는 모든 호스트에서 [Nmap](#)을 사용하여 수신기가 구성된 포트에서 올바르게 표시되도록 할 수 있습니다.

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

로깅 및 디버깅

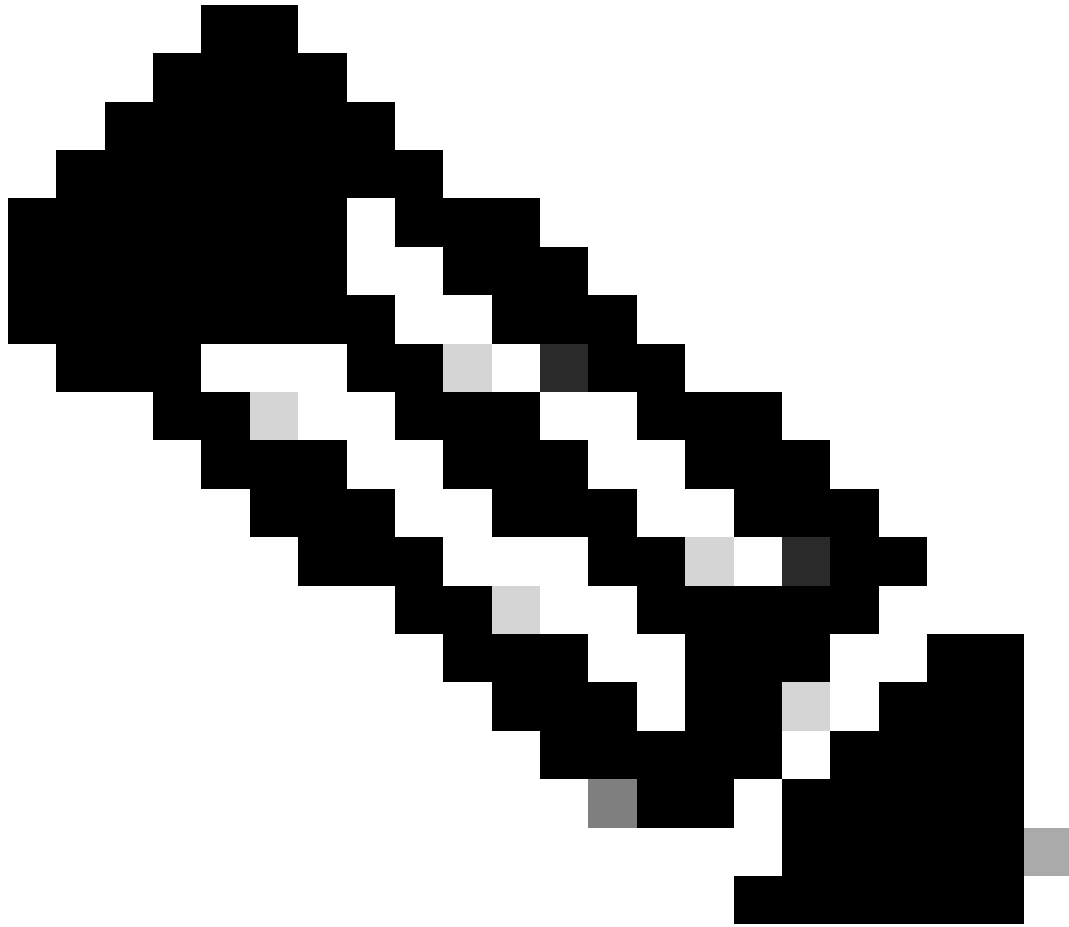
```
2024/05/23 14:40:36.566486156 {pubd_R0-0}{2}: [mdt-ctrl] [30214]: (note): **** Event Entry: Configured
```

메트릭이 TIG 스택에 도달하는지 확인

IntrusionDB CLI에서

다른 데이터베이스 시스템과 마찬가지로 IntrusionDB에는 Telegraf에서 올바르게 수신한 메트릭을 확인하고 정의된 데이터베이스에 저장하는 데 사용할 수 있는 CLI가 포함되어 있습니다. IntrusionDB는 점이라고 하는 메트릭을 시리즈로 구성된 측정 단위로 구성합니다. 여기에 나와 있는 몇 가지 기본 명령을 사용하여 IntrusionDB 측의 데이터 체계를 확인하고 데이터가 이 애플리케이션에 도착하는지 확인할 수 있습니다.

먼저 계열, 측정 및 그 구조(키)가 제대로 생성되었는지 확인할 수 있다. 이는 사용된 RPC의 구조를 기반으로 Telegraf 및 IntrusionDB에 의해 자동으로 생성됩니다.



참고: 물론 이 구조는 Telegraf 및 InfluxDB 구성에서 완벽하게 사용자 정의할 수 있습니다. 그러나 이는 이 컨피그레이션 가이드의 범위를 벗어나는 것입니다.

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

데이터 구조가 명확해지면(정수, 문자열, 부울, ...), 특정 필드에 따라 이러한 측정값에 저장되는 데이터 포인트의 수를 가져올 수 있습니다.

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

특정 필드의 포인트 수와 마지막 발생의 타임스탬프가 증가하면 TIG 스택이 WLC에서 전송한 데이터를 올바르게 수신하여 저장하는 것이 좋습니다.

텔레그라프에서

Telegraf 수신기가 실제로 컨트롤러에서 일부 메트릭을 가져오고 해당 형식을 확인하는지 확인하기 위해 Telegraf 메트릭을 호스트의 출력 파일로 리디렉션할 수 있습니다. 이는 장치 상호 연결 문제 해결에 있어 매우 편리할 수 있습니다. 이를 실현하려면 Telegraf에서 구성 가능한 ["file" 출력 플러그인](#)을 사용하면 됩니다 /etc/telegraf/telegraf.conf.

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

참조

[하드웨어 크기 조정 지침](#)

[Grafana 요구 사항](#)

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.