

Nexus 플랫폼에서 텔레메트리 구성 및 확인

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[토폴로지](#)

[Nexus-to-Receiver 연결](#)

[배경 정보](#)

[기존 폴링 대 스트리밍 텔레메트리](#)

[실제 활용 사례](#)

[텔레메트리 처리 개요](#)

[텔레메트리 아키텍처 및 용어](#)

[데이터 수집](#)

[데이터 관리 엔진 — DME](#)

[관리 객체 및 관리 정보 트리](#)

[고유 이름 — DN](#)

[센서 경로](#)

[데이터 인코딩](#)

[데이터 전송](#)

[텔레메트리 수신기](#)

[대상 그룹](#)

[센서 그룹](#)

[서브스크립션](#)

[정기 및 이벤트 기반 텔레메트리](#)

[주기적 원격 측정](#)

[이벤트 기반 텔레메트리](#)

[Lab에서 사용되는 수집 동작](#)

[텔레메트리 수신기 준비](#)

[이 실습에서 사용되는 수신기](#)

[NX-OS에서 스트리밍 텔레메트리 구성](#)

[텔레메트리 기능 활성화](#)

[대상 그룹 구성](#)

[센서 그룹 1 구성](#)

[서브스크립션 1 구성](#)

[센서 그룹 2 구성](#)

[서브스크립션 2 구성](#)

[루프백 인터페이스 구성](#)

[센서 그룹 3 구성](#)

[서브스크립션 구성 3](#)

[초기 텔레메트리 컨피그레이션 완료](#)



[주기적인 원격 분석 확인](#)

[텔레메트리 전송 확인](#)

[정기 센서 그룹 확인](#)

[데이터 수집 확인](#)

[수신기에서 정기 데이터 확인](#)

[출력 및 운영 정보 확인](#)

[정기 텔레메트리 확인 요약](#)

[이벤트 기반 텔레메트리 확인](#)

[이벤트 기반 서브스크립션 확인](#)

[이벤트 기반 업데이트 생성](#)

[초기 스냅샷 및 이벤트 메시지 확인](#)

[이벤트 수집 통계 확인](#)

[이벤트 컬렉터 오류 확인](#)

[이벤트 기반 텔레메트리 확인 요약](#)

[기본 문제 해결](#)

[현재 전송 상태 확인](#)

[데이터 수집 카운터 검토](#)

[컬렉션을 건너뛴 이유 확인](#)

[전송 오류 검사](#)

[전송 복구 확인](#)

[컨피그레이션 및 이벤트 컬렉터 오류 확인](#)

[문제 해결 워크플로](#)

[실제 응용 프로그램 예: 시스템 리소스 모니터링](#)

[시스템 리소스 원격 분석 구성](#)

[리소스 경로 레이블 검토](#)

[시스템 리소스 원격 분석 확인](#)

[NX-OS 시스템 리소스와 비교](#)

[수신자의 리소스 데이터 확인](#)

[시간에 따른 리소스 사용률 관찰](#)

[결론](#)

소개

이 문서에서는 Cisco NX-OS를 실행하는 Cisco Nexus 9000 스위치에서 스트리밍 텔레메트리를 구현 및 확인하는 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

Cisco에서는 다음 항목에 대한 기본 지식을 갖춘 것을 권장합니다.

- Cisco NX-OS CLI
- 기본 레이어 3 IP 연결
- 기본 네트워크 모니터링 개념
- 네트워크 텔레메트리 개념에 대한 기본 지식

사용되는 구성 요소

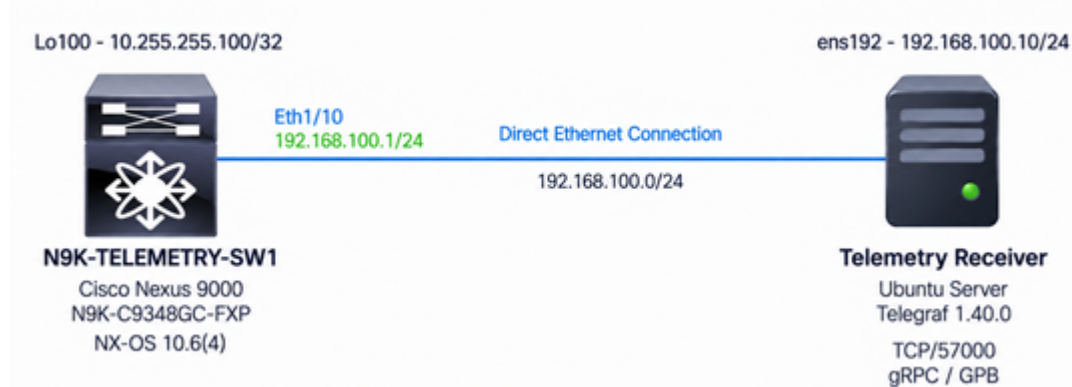
이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

구성 요소	플랫폼/소프트웨어	버전/값	목적
N9K-TELEMETRY-SW1	N9K-C9348GC-FXP	10.6(4)	텔레메트리 소스
텔레메트리 수신기	Ubuntu 서버	22.04.5	외부 텔레메트리 수신기
텔레메트리 애플리케이션	텔레그라프	1.40.0	Google Protocol Buffers (GPB)-over-gRPC telemetry receiver
수신 포트	TCP	57000	텔레메트리 수신기 포트
수신기 출력 형식	JSON(JavaScript Object Notation)	—	사람이 판독할 수 있는 텔레메트리 출력

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.

토폴로지

이 실습에서는 외부 텔레메트리 수신기로 작동하는 Ubuntu 서버에 직접 연결된 Cisco Nexus 9000 스위치를 사용합니다.



Nexus 스위치와 텔레메트리 수신기는 Ethernet1/10 및 ens192를 통해 직접 연결됩니다.

Nexus-to-Receiver 연결

Ethernet1/10은 Nexus 스위치와 Ubuntu 수신기 간에 레이어 3 연결을 제공합니다.

인터페이스 컨피그레이션은 다음과 같습니다.

```
N9K-TELEMETRY-SW1# show running-config interface ethernet1/10 interface Ethernet1/10 description TELEMETRY-COLLECTOR ip
address 192.168.100.1/24 no shutdown
The Ubuntu receiver uses: Interface: ens192 IP address: 192.168.100.10/24
```

배경 정보

네트워크 모니터링 및 가시성은 네트워크 상태를 파악하고, 비정상적인 동작을 식별하고, 네트워크 문제를 해결하는 데 필수적입니다.

Cisco NX-OS는 CLI, SNMP(Simple Network Management Protocol) 및 Syslog를 포함하여 Nexus 스위치에서 운영 정보를 검색하거나 내보낼 수 있는 몇 가지 메커니즘을 제공합니다. SNMP 폴링 및 CLI 기반 수집과 같은 기존 모니터링 워크플로는 일반적으로 풀(pull) 모델에 의존하며, 이 모델에서는 외부 모니터링 시스템이 네트워크 디바이스에 정보를 주기적으로 요청합니다.

Streaming Telemetry에는 푸시 모델이 도입되어 네트워크 디바이스에서 구성된 서브스크립션을 기반으로 선택한 운영 데이터를 외부 수신기로 전송합니다. 이는 모니터링 시스템이 디바이스를 지속적으로 폴링할 것을 요구하지 않고 네트워크 정보를 수집하기 위한 구조화된 메커니즘을 제공한다.

기존 폴링 대 스트리밍 텔레메트리

전통적인 폴링 모델에서, 모니터링 시스템은 네트워크 디바이스로부터 정보가 검색되는 시점을 결정한다.

예를 들어 NMS(네트워크 관리 시스템)는 60초마다 인터페이스 통계를 요청할 수 있습니다. 이는 디바이스 상태의 정기 스냅샷을 제공합니다. 그러나 모니터링 애플리케이션에서 사용할 수 있는 가시성은 구성된 폴링 간격과 직접 관련이 있습니다.

Nexus 스위치는 스트리밍 텔레메트리를 통해 구성된 서브스크립션에 따라 텔레메트리 수신기로 선택한 정보를 전송합니다.

근본적인 차이는 다음과 같이 요약할 수 있습니다.

일반 폴링	스트리밍 텔레메트리
클라이언트가 요청을 시작합니다.	네트워크 디바이스가 데이터를 전송합니다.
끌어오기 모델	푸시 모델
데이터는 폴링 간격에 따라 검색됩니다.	텔레메트리 서브스크립션에 따라 데이터가 전송됩니다.
일반적으로 정기적인 스냅샷을 제공합니다.	정기 및 이벤트 기반 수집을 지원합니다.
모니터링 시스템은 디바이스에서 정보를 요청합니다.	디바이스는 선택한 정보를 수신기로 스트리밍합니다.

스트리밍 텔레메트리가 반드시 기존의 모니터링 메커니즘을 대체하는 것은 아닙니다. CLI, SNMP, Syslog 및 텔레메트리를 함께 사용할 수 있으며 서로 다른 운영 목적을 제공합니다.

가장 큰 차이점은 데이터 수집 모델입니다.

실제 활용 사례

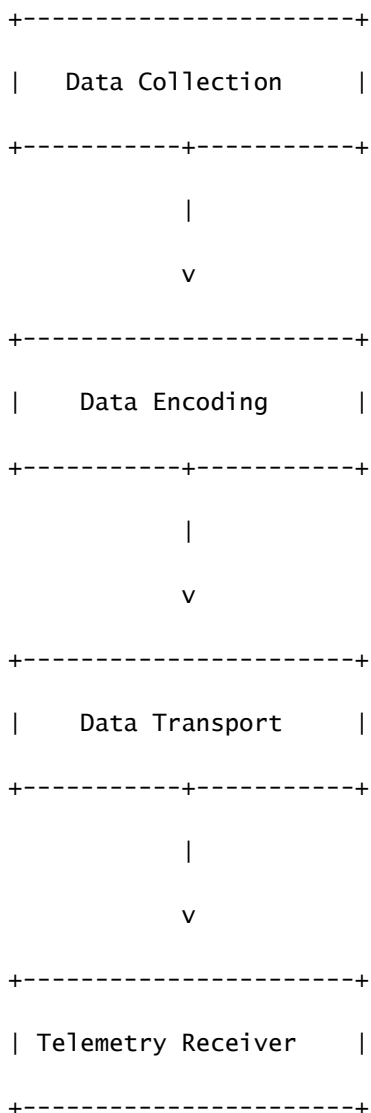
프로덕션 환경에서는 네트워크 및 시스템 상태에 대한 지속적인 가시성을 제공하기 위해 스트리밍 텔레메트리를 일반적으로 사용합니다.

일반적인 활용 사례로는 인터페이스 카운터 및 상태 변경 모니터링, CPU 및 메모리 사용률과 같은 시스템 리소스, VXLAN(Virtual Extensible LAN) 피어 및 BGP(Border Gateway Protocol) 피어 상태와 같은 데이터 센터 패브릭 정보가 있습니다. 이벤트 기반 텔레메트리를 사용하여 변경 사항에 따른 운영 상태 변경 사항을 보고할 수도 있습니다.

내보낸 텔레메트리 데이터는 대시보드, 경고, 기록 분석 및 문제 해결을 위한 외부 모니터링 및 관찰 가능 플랫폼에서 사용할 수 있습니다.

텔레메트리 처리 개요

상위 레벨에서 NX-OS의 스트리밍 텔레메트리는 4가지 주요 단계로 이해할 수 있습니다.



이 단계에서는 다음과 같은 4가지 기본 질문에 답합니다.

- 데이터 수집: 어떤 정보가 수집됩니까?
- 데이터 인코딩: 정보는 어떻게 표현됩니까?
- 데이터 전송: 텔레메트리 데이터는 어떻게 전달됩니까?
- 텔레메트리 수신기: 텔레메트리 데이터는 어디로 전송됩니까?

텔레메트리 아키텍처 및 용어

데이터 수집

첫 번째 단계에서는 스위치에서 수집해야 하는 정보를 결정합니다.

이 문서의 예에서는 DME(Data Management Engine)에서 텔레메트리 데이터를 수집합니다.

데이터 관리 엔진 — DME

Data Management Engine은 NX-OS 내에서 구성 및 운영 정보의 구조화된 표시를 유지 관리합니다 .

DME는 디바이스 정보를 CLI 텍스트로만 표시하는 대신 계층 경로를 통해 액세스할 수 있는 관리 객체로 정보를 구성합니다.

예를 들면 다음과 같습니다.

```
sys/intf/phys-[eth1/10]
```

Ethernet1/10과 연결된 DME 객체를 식별합니다.

이 Lab에서 사용되는 기타 DME 경로는 다음과 같습니다.

```
sys/intf/phys-[eth1/10]/dbgIfIn  
sys/intf/phys-[eth1/10]/dbgIfOut  
sys/intf/phys-[eth1/10]/phys  
sys/intf/lb-[l0100]
```

각 경로는 DME를 통해 이용 가능한 정보의 다른 객체 또는 부분을 나타낸다. 이는 실습 컨피그레이션 전체에서 이미 사용된 것과 동일한 경로입니다.

관리 객체 및 관리 정보 트리

DME 데이터베이스는 MO(Managed Object)로 구성됩니다.

Managed Object는 다음과 같이 NX-OS 관리 모델 내의 엔터티를 나타냅니다.

- 물리적 인터페이스

- 루프백 인터페이스
- 인터페이스 입력 통계
- 인터페이스 출력 통계

관리 객체는 MIT(Management Information Tree)에서 계층적으로 구성됩니다.

이 실습에서 사용되는 객체의 단순화 표현은 다음과 같습니다.

```
sys
|
+-- intf
|
+-- phys-[eth1/10]
| |
| +-- dbgIfIn
| +-- dbgIfOut
| +-- phys
|
+-- lb-[1o100]
```

이 계층 구조는 텔레메트리 센서 경로가 DME 내의 특정 정보를 식별하는 방법을 이해하는 데 유용합니다.

고유 이름 — DN

각 관리 객체는 DN(Distinguished Name)에 의해 고유하게 식별될 수 있습니다.

DN은 DME 트리의 루트에서 대상 객체까지의 계층 경로를 나타냅니다.

For example: sys/intf/lb-[1o100]

Loopback100 관리 개체를 식별합니다.

마찬가지로:

sys/intf/phys-[eth1/10]/dbgIfIn

Ethernet1/10과 연결된 입력 통계 객체를 식별합니다.

DN을 DME 계층 구조 내에 있는 객체의 완전한 주소로 간주하는 간단한 방법은 DN을 이해하는 것입니다.

센서 경로

센서 경로는 NX-OS가 텔레메트리 서브스크립션을 위해 모니터링해야 하는 정보를 식별합니다.

초기 Lab 예에서는 DME DN이 센서 경로로 사용됩니다. 이후의 실제 예에서는 사전 정의된 리소스 경로 레이블을 보여 줍니다.

예를 들면 다음과 같습니다.

```
sensor-group 2
path sys/intf/phys-[eth1/10]/dbgIfIn
path sys/intf/phys-[eth1/10]/dbgIfOut
path sys/intf/phys-[eth1/10]/phys
```

이러한 경로는 Ethernet1/10에 대한 입력 통계, 출력 통계 및 운영 정보를 제공합니다.

데이터 인코딩

NX-OS가 요청된 정보를 수집한 후 데이터를 전송하려면 먼저 데이터를 인코딩해야 합니다.

이 Lab에서 사용되는 인코딩은 Google Protocol Buffers(GPB)입니다.

대상 컨피그레이션에서는 다음을 지정합니다.

```
ip address 192.168.100.10 port 57000 protocol grpc encoding GPB
```

GPB는 수집된 텔레메트리 정보가 메시지에 표시되는 방식을 정의합니다.

인코딩과 전송은 별도의 기능입니다. GPB는 데이터 표현을 정의하는 반면, 전송 메커니즘은 메시지 전달 방식을 결정합니다.

데이터 전송

이 Lab에서 사용되는 전송 프로토콜은 gRPC입니다.

Nexus 스위치는 GPB로 인코딩된 텔레메트리 데이터를 다음과 같이 전송합니다.

192.168.100.10:57000

gRPC 사용.

따라서

GPB는 텔레메트리 정보가 인코딩되는 방식을 정의합니다.

gRPC는 텔레메트리 메시지를 수신자에게 전달하는 데 사용되는 전송 메커니즘을 제공합니다.

스트리밍 원격 분석에 사용되는 gRPC 전송을 gNMI(gRPC Network Management Interface) 및 gNOI(gRPC Network Operations Interface) 등의 서비스에 사용되는 NX-OS gRPC 에이전트와 혼동해서는 안 됩니다.

텔레메트리 수신기

Telemetry Receiver는 텔레메트리 스트림을 수신하고 처리하는 외부 시스템 또는 애플리케이션입니다.

이 Lab에서는 Telegraf를 실행하는 Ubuntu 서버가 수신기로 사용됩니다. 수신기는 TCP 포트 57000에서 수신 대기하며 Nexus 스위치에 의해 생성된 GPB-over-gRPC 텔레메트리 스트림을 수락합니다.

Lab에서 사용되는 수신기 구현은 Telemetry Receiver 준비 섹션의 뒷부분에서 설명합니다.

대상 그룹

대상 그룹은 텔레메트리 데이터를 전송해야 하는 위치와 전송 방법을 정의합니다.

이 실습에서는 다음을 사용합니다.

```
destination-group 1
ip address 192.168.100.10 port 57000 protocol gRPC encoding GPB
use-vrf default
```

이는 텔레메트리 전달에 사용되는 수신기 주소, 목적지 포트, 전송 프로토콜, 인코딩 및 VRF(Virtual Routing and Forwarding) 인스턴스를 정의합니다.

센서 그룹

센서 그룹은 모니터링해야 하는 정보를 정의합니다.

예를 들면 다음과 같습니다.

```
sensor-group 2
path sys/intf/phys-[eth1/10]/dbgIfIn
path sys/intf/phys-[eth1/10]/dbgIfOut
path sys/intf/phys-[eth1/10]/phys
```

이 예에서 센서 그룹 2는 Ethernet1/10 입력 통계, 출력 통계 및 작동 인터페이스 정보를 모니터링합니다. dbgIfIn 경로는 입력 인터페이스 통계를 제공하고, dbgIfOut은 출력 인터페이스 통계를 제공하며, phy는 인터페이스에 대한 운영 정보를 제공합니다.

센서 그룹은 여러 관련 센서 경로를 포함할 수 있으므로 다음과 같은 질문에 답합니다.

어떤 데이터가 수집되니까?

서브스크립션

서브스크립션은 센서 그룹을 목적지 그룹과 연결하고 수집 동작을 정의합니다.

예를 들면 다음과 같습니다.

```
subscription 2
dst-grp 1
snsr-grp 2 sample-interval 60000
```

이 예에서는 다음을 수행합니다.

- 센서 그룹 2는 수집되는 정보를 결정합니다.
- 샘플 간격이 수집 동작을 결정합니다.
- 대상 그룹 1은 정보가 전송되는 위치와 방법을 결정합니다.

따라서 서브스크립션은 다음과 같은 기본 텔레메트리 구성 요소를 연결합니다.

센서 그룹 + 수집 동작 + 대상 그룹

정기 및 이벤트 기반 텔레메트리

Cisco NX-OS Streaming Telemetry는 DME 기반 서브스크립션에 대해 정기 및 이벤트 기반 수집을 지원합니다.

수집 동작은 서브스크립션 내의 센서 그룹과 연결된 샘플 간격에 의해 제어됩니다.

주기적 원격 측정

주기적인 원격 측정을 통해 NX-OS는 구성된 간격으로 모니터링되는 정보를 수집하고 전송합니다.

샘플링 간격은 밀리초 단위로 지정됩니다.

For example: `snsr-grp 2 sample-interval 60000`

60초 수집 간격을 구성합니다.

정기적인 텔레메트리는 지속적으로 변경되고 일반적으로 다음과 같이 시간이 지남에 따라 분석되는 정보에 유용합니다.

- 인터페이스 패킷 카운터
- 바이트 카운터
- 오류 및 폐기
- 운영 통계
- 리소스 사용률

이 Lab에서 서브스크립션 1 및 서브스크립션 2는 정기적인 수집을 사용합니다.

서브스크립션 1은 10초마다 Ethernet1/10 인터페이스 객체를 수집하는 반면, 서브스크립션 2는 60초마다 인터페이스 통계 및 운영 정보를 수집합니다.

이벤트 기반 텔레메트리

이벤트 기반 텔레메트리를 사용하는 경우 NX-OS는 반복 수집 타이머를 사용하지 않습니다.

DME 기반 텔레메트리의 경우 이벤트 기반 동작이 다음으로 구성됩니다.

```
sample-interval 0
```

모니터링되는 객체가 변경되면 텔레메트리는 해당 변경과 관련된 업데이트를 생성할 수 있습니다.

이 수집 방법은 다음과 같은 정보에 유용합니다.

- 구성 변경
- 관리 상태 변경
- 객체 생성 또는 삭제
- 모니터링되는 DME 객체 아래의 기타 변경 사항

이 Lab에서 서브스크립션 3은 Loopback100 DME 객체를 모니터링합니다.

```
sys/intf/lb-[1o100]:  
snsr-grp 3 sample-interval 0
```

따라서 모니터링되는 루프백100 객체에 대한 변경 사항은 이 문서의 뒷부분에서 이벤트 기반 텔레메트리를 시연하는 데 사용됩니다.

Lab에서 사용되는 수집 동작

초기 실행 컨피그레이션에서 사용되는 3가지 서브스크립션은 다음과 같이 요약할 수 있습니다.

서브스크립션	모니터링되는 정보	샘플 간격	수집 동작
1	Ethernet1/10 인터페이스 개체	10000밀리초	정기
2	Ethernet1/10 통계 및 운영 상태	60000밀리초	정기
3	Loopback100 개체	0	이벤트 기반

주요 차이점은 텔레메트리 데이터를 생성하는 데 사용되는 트리거입니다.

주기적 원격 측정	이벤트 기반 텔레메트리
구성된 타이머를 사용합니다.	정기 타이머를 사용하지 않습니다.
sample-interval > 0	샘플 간격 0
반복 샘플을 생성합니다.	모니터링되는 개체가 변경될 때 업데이트를 생성합니다.
일반적으로 카운터 및 통계에 사용됩니다.	컨피그레이션 또는 상태 변경에 일반적으로 사용됩니다.

컨피그레이션 및 확인 섹션에서는 위에서 정의한 서브스크립션을 사용하여 두 수집 동작을 모두 보여줍니다.

텔레메트리 수신기 준비

스트리밍 텔레메트리를 사용하려면 Nexus 스위치에서 보내는 텔레메트리 데이터를 수락하고 처리할 수 있는 외부 수신기가 필요합니다.

이 문서에 사용된 GPB-over-gRPC 전송의 경우 수신자는 다음을 수행할 수 있어야 합니다.

- gRPC 연결 수신
- 구성된 대상 포트에서 수신 대기
- GPB 텔레메트리 메시지 디코딩
- 수신한 데이터의 처리 또는 저장

텔레메트리 수신기의 구현은 이 문서에 설명된 NX-OS 텔레메트리 컨피그레이션과 무관합니다.



참고: 타사 텔레메트리 수신기 소프트웨어의 설치, 구성, 운영 및 문제 해결은 이 문서의 범위에 포함되지 않습니다. 컨피그레이션 및 지원 정보는 수신기 공급업체가 제공한 설명서를 참조하십시오.

이 실습에서 사용되는 수신기

실습에서는 데모용으로 Telegraf를 실행하는 외부 Ubuntu 서버를 텔레메트리 수신기로 사용합니다

수신기 매개변수는 다음과 같습니다.

매개변수	가치
수신자 주소	192.168.100.10
전송	gRPC
수신 포트	TCP/57000
인코딩	GPB

이 문서에서는 수신측 소프트웨어 구성에 대해 다루지 않습니다.



참고: 이 문서에 표시된 수신기 출력 예는 각 확인 단계와 관련된 필드를 강조 표시하도록 필터링됩니다.

Nexus 스위치에서 스트리밍 텔레메트리를 구성하기 전에 다음을 확인합니다.

- Nexus 스위치와 수신기 간에 IP 연결이 존재합니다.
- 수신기 응용 프로그램이 실행 중입니다.
- 구성된 대상 포트가 수신 대기하고 있습니다.
- 수신기는 구성된 전송 프로토콜을 지원합니다.
- 수신기는 구성된 텔레메트리 인코딩을 지원합니다.

이 실습에서는 Nexus 스위치의 Ethernet1/10에서 192.168.100.1/24을 사용하고 텔레메트리 수신기에서 192.168.100.10/24을 사용합니다. 텔레메트리 관련 동작을 트러블슈팅하기 전에 기본 레이어 3 연결 가능성을 확인해야 합니다.

수신기가 연결 가능하고 TCP 포트 57000에서 GPB-over-gRPC 텔레메트리를 수락할 준비가 되면 NX-OS 텔레메트리 컨피그레이션을 적용할 수 있습니다.

NX-OS에서 스트리밍 텔레메트리 구성

이 구성에서는 세 가지 기본 구성 요소를 사용합니다.

- 대상 그룹: 텔레메트리 데이터가 전송되는 위치 및 전송되는 방법을 정의합니다.
- 센서 그룹: 수집되는 정보를 정의합니다.
- 서브스크립션: 센서 그룹을 대상 그룹과 연결하고 수집 동작을 정의합니다.

Lab에서는 다음과 같은 텔레메트리 대상을 사용합니다.

매개변수	가치
대상	192.168.100.10:57000
전송	gRPC
인코딩	GPB
VRF	기본값

초기 실습 컨피그레이션에서는 3개의 서브스크립션을 사용합니다.

서브스크립션	센서	컬렉션 유형	샘플 간격
1	Ethernet1/10 인터페이스 개체	정기	10000밀리초
2	Ethernet1/10 통계 및 운영 상태	정기	60000밀리초
3	Loopback100 개체	이벤트 기반	0

텔레메트리 기능 활성화

스트리밍 텔레메트리를 먼저 전역적으로 활성화해야 합니다.

```
N9K-TELEMETRY-SW1# configure terminal
N9K-TELEMETRY-SW1(config)# feature telemetry
```

원격 분석 컨피그레이션 모드를 시작합니다.

```
N9K-TELEMETRY-SW1(config)# telemetry
N9K-TELEMETRY-SW1(config-telemetry)#
```

나머지 텔레메트리 컨피그레이션은 이 모드에서 수행됩니다.

대상 그룹 구성

대상 그룹은 외부 텔레메트리 수신기를 식별하고 텔레메트리 데이터를 전송하는 데 사용되는 전송 및 인코딩을 정의합니다.

구성:

```
N9K-TELEMETRY-SW1(config-telemetry)# destination-group 1
N9K-TELEMETRY-SW1(conf-tm-dest)# ip address 192.168.100.10 port 57000 protocol gRPC encoding GPB
```

```
N9K-TELEMETRY-SW1(conf-tm-dest)# use-vrf default
```

이 컨피그레이션에서는 다음을 정의합니다.

- 대상 주소: 192.168.100.10
- 대상 포트: 57000
- 전송: gRPC
- 인코딩: GPB
- VRF: 기본값

텔레메트리 수신기가 기본 VRF에서 Ethernet1/10을 통해 연결할 수 있으므로 기본 VRF가 사용됩니다.

텔레메트리 전송을 위해 선택한 VRF는 구성된 수신기에 IP 연결성을 제공해야 합니다.

센서 그룹 1 구성

센서 그룹 1은 Ethernet1/10 DME 인터페이스 객체를 모니터링합니다.

```
N9K-TELEMETRY-SW1(config-telemetry)# sensor-group 1
N9K-TELEMETRY-SW1(conf-tm-sensor)# path sys/intf/phys-[eth1/10]
```

센서 경로는 DME 계층 구조에서 Ethernet1/10 관리 객체를 식별하고 해당 객체와 관련된 일반적인 인터페이스 정보를 제공합니다.

서브스크립션 1 구성

센서 그룹 1을 대상 그룹 1과 연결:

```
N9K-TELEMETRY-SW1(config-telemetry)# subscription 1
N9K-TELEMETRY-SW1(conf-tm-sub)# dst-grp 1
N9K-TELEMETRY-SW1(conf-tm-sub)# snsr-grp 1 sample-interval 10000
```

샘플링 간격은 밀리초 단위로 지정됩니다.

10000 ms = 10초

따라서 서브스크립션 1은 10초마다 주기적으로 Ethernet1/10 DME 객체를 수집하고 텔레메트리 데이터를 대상 그룹 1에 전송합니다.

센서 그룹 2 구성

센서 그룹 2는 Ethernet1/10에 대한 통계 및 운영 정보를 수집합니다.

구성:

```
N9K-TELEMETRY-SW1(config-telemetry)# sensor-group 2
N9K-TELEMETRY-SW1(conf-tm-sensor)# path sys/intf/phys-[eth1/10]/dbgIfIn
N9K-TELEMETRY-SW1(conf-tm-sensor)# path sys/intf/phys-[eth1/10]/dbgIfOut
N9K-TELEMETRY-SW1(conf-tm-sensor)# path sys/intf/phys-[eth1/10]/phys
```

센서 경로는 다음을 제공합니다.

센서 경로	정보
sys/intf/phys-[eth1/10]/dbgIfIn	입력 인터페이스 통계
sys/intf/phys-[eth1/10]/dbgIfOut	출력 인터페이스 통계
sys/intf/phys-[eth1/10]/phys	운영 인터페이스 정보

센서 그룹은 여러 관련 센서 경로를 포함할 수 있으므로 서브스크립션이 동일한 모니터링되는 인터페이스에서 여러 카테고리의 정보를 수집할 수 있습니다.

서브스크립션 2 구성

센서 그룹 2를 대상 그룹 1과 연결:

```
N9K-TELEMETRY-SW1(config-telemetry)# subscription 2
N9K-TELEMETRY-SW1(conf-tm-sub)# dst-grp 1
N9K-TELEMETRY-SW1(conf-tm-sub)# snsr-grp 2 sample-interval 60000
```

구성된 샘플링 간격은 다음과 같습니다.

60000 ms = 60초

따라서 서브스크립션 2는 60초마다 Ethernet1/10 통계 및 운영 정보를 수집합니다.

이 서브스크립션은 이 문서의 뒷부분에서 주기적인 텔레메트리 동작을 확인하는 데 사용됩니다.

루프백 인터페이스 구성

루프백 인터페이스는 추가 물리적 연결 없이 이벤트 기반 텔레메트리를 시연하는 데 사용됩니다.

구성:

```
N9K-TELEMETRY-SW1(config)# interface loopback100
N9K-TELEMETRY-SW1(config-if)# ip address 10.255.255.100/32
```

해당 DME 객체는 다음과 같습니다.

```
sys/intf/lb-[1o100]
```

루프백 인터페이스는 이벤트 기반 텔레메트리 검증 중에 제어된 컨피그레이션 및 관리 상태 변경을 생성하는 간단한 방법을 제공합니다.

센서 그룹 3 구성

Loopback100 DME 센서 경로를 구성합니다.

```
N9K-TELEMETRY-SW1(config-telemetry)# sensor-group 3
N9K-TELEMETRY-SW1(conf-tm-sensor)# path sys/intf/lb-[1o100]
```

센서 그룹 3은 Loopback100 Managed Object를 모니터링합니다.

서브스크립션 구성 3

센서 그룹 3을 대상 그룹 1과 연결하고 이벤트 기반 수집을 구성합니다.

```
N9K-TELEMETRY-SW1(config-telemetry)# subscription 3
N9K-TELEMETRY-SW1(conf-tm-sub)# dst-grp 1
N9K-TELEMETRY-SW1(conf-tm-sub)# snsr-grp 3 sample-interval 0
```

DME 기반 서브스크립션의 경우 샘플 간격이 0이면 이벤트 기반 동작이 구성됩니다.

따라서 모니터링되는 Loopback100 개체의 변경 사항은 반복 수집 타이머를 사용하지 않고 텔레메트리 알림을 생성할 수 있습니다.

이 서브스크립션은 나중에 설명 및 관리 상태 변경을 시연하는 데 사용됩니다.

초기 텔레메트리 컨피그레이션 완료

컨피그레이션이 완료되면 다음을 사용하여 확인합니다.

```
N9K-TELEMETRY-SW1# show running-config telemetry
feature telemetry
telemetry
  destination-group 1
    ip address 192.168.100.10 port 57000 protocol gRPC encoding GPB
    use-vrf default
  sensor-group 1
    path sys/intf/phys-[eth1/10]
  sensor-group 2
    path sys/intf/phys-[eth1/10]/dbgIfIn
    path sys/intf/phys-[eth1/10]/dbgIfOut
    path sys/intf/phys-[eth1/10]/phys
  sensor-group 3
    path sys/intf/lb-[lo100]
  subscription 1
    dst-grp 1
    snsr-grp 1 sample-interval 10000
  subscription 2
    dst-grp 1
    snsr-grp 2 sample-interval 60000
  subscription 3
    dst-grp 1
    snsr-grp 3 sample-interval 0
```

이 시점에서 스위치에는 구성된 수신기에 대한 텔레메트리 데이터 전송을 시작하는 데 필요한 대상, 센서 그룹 및 서브스크립션이 있습니다.

다음 섹션에서는 Ethernet1/10 서브스크립션에 의해 생성된 전송 세션 및 주기적인 텔레메트리를 확인합니다.

주기적인 원격 분석 확인

텔레메트리 컨피그레이션이 적용된 후 전송 세션이 설정되었고, 주기적인 센서 그룹이 활성 상태이며, 텔레메트리 데이터가 수집되어 수신자에게 전달되는지 확인합니다.

텔레메트리 전송 확인

다음 명령을 실행합니다.

```
N9K-TELEMETRY-SW1# show telemetry transport
Session Id  Dst Grp  IP Address      Port      Encoding      Transport  Status
-----
0           1         192.168.100.10  57000     GPB           gRPC       Connected
-----
Retry buffer Size:          10485760
Event Retry Messages (Bytes): 0
Timer Retry Messages (Bytes): 0
Total Retries sent:         0
Total Retries Dropped:      0
N9K-TELEMETRY-SW1#
```

Connected(연결됨) 상태는 구성된 텔레메트리 수신기에 대한 gRPC 전송 세션이 설정되었음을 확인합니다.

관련 전송 매개변수도 구성된 Destination Group과 일치합니다.

정기 센서 그룹 확인

Use:

```
N9K-TELEMETRY-SW1# show telemetry control database
```

Subscription Database size = 3

Subscription ID	Data Collector Type	Reachability	Configuration Method
3	DME	Reachable	CONFIG_DME-ADMIN
2	DME	Reachable	CONFIG_DME-ADMIN
1	DME	Reachable	CONFIG_DME-ADMIN

Sensor Group Database size = 3

Row ID	Sensor Group ID	Sensor Group type	Sampling interval(ms)	Linked subscriptions	SubID
--------	-----------------	-------------------	-----------------------	----------------------	-------

```
-----
1          3          Event /DME          0/No Timer          1          3
Collection Time in ms (Cur/Min/Max): 0/0/0
Encoding Time in ms (Cur/Min/Max): 0/0/0
Transport Time in ms (Cur/Min/Max): 2/1/414
Streaming Time in ms (Cur/Min/Max): 2/1/414
```

```
Collection Statistics:
collection_id_dropped      = 0
last_collection_id_dropped = 0
drop_count                 = 0
```

```
Configuration method: CONFIG_DME-ADMIN
```

```
2          2          Timer /DME          60000/Running          1          2
Collection Time in ms (Cur/Min/Max): 0/0/1
Encoding Time in ms (Cur/Min/Max): 0/0/1
Transport Time in ms (Cur/Min/Max): 2/1/415
Streaming Time in ms (Cur/Min/Max): 3/2/416
```

```
Collection Statistics:
collection_id_dropped      = 0
last_collection_id_dropped = 0
drop_count                 = 0
```

```
Configuration method: CONFIG_DME-ADMIN
```

```
3          1          Timer /DME          10000/Running          1          1
Collection Time in ms (Cur/Min/Max): 0/0/1
Encoding Time in ms (Cur/Min/Max): 0/0/1
Transport Time in ms (Cur/Min/Max): 2/1/415
Streaming Time in ms (Cur/Min/Max): 3/2/417
```

```
Collection Statistics:
collection_id_dropped      = 0
last_collection_id_dropped = 0
drop_count                 = 0
```

```
Configuration method: CONFIG_DME-ADMIN
```

Row ID	Subscribed	Linked Groups	Sec Groups	Retrieve level	Path(GroupId) : Query : Filter
1	No	1	0	Self	sys/intf/phys-[eth1/10](1) : NA : NA
<snip>					
2	No	1	0	Self	sys/intf/phys-[eth1/10]/dbgIfIn(2) : NA : NA
<snip>					
4	No	1	0	Self	sys/intf/phys-[eth1/10]/dbgIfOut(2) : NA : NA

센서 그룹 데이터베이스에서 정기 센서 그룹이 활성 상태인지 확인합니다.

센서 그룹	유형	샘플링 간격	서브스크립션
1	타이머/DME	10000 ms/실행 중	1
2	타이머/DME	60000 ms/실행 중	2

센서 그룹 1은 10초마다 Ethernet1/10 인터페이스 객체를 수집합니다.

센서 그룹 2는 60초마다 이더넷1/10 통계와 작동 정보를 수집합니다.

동일한 명령은 구성된 센서 경로도 표시합니다.

```
sys/intf/phys-[eth1/10]
sys/intf/phys-[eth1/10]/dbgIfIn
sys/intf/phys-[eth1/10]/dbgIfOut
sys/intf/phys-[eth1/10]/phys
```

이 Lab에서 사용된 센서 경로의 경우 수집 및 인코딩 시간은 일반적으로 0ms에서 1ms 사이였으며 텔레메트리 메시지 삭제는 관찰되지 않았습니다.

데이터 수집 확인

Use:

```
N9K-TELEMETRY-SW1# show telemetry data collector brief
```

Row ID	Collector Type	Successful	Payloads	Failed	Skipped
1	YANG	0	0	0	0
2	DME	513	513	0	44
3	NX-API	0	0	0	0

N9K-TELEMETRY-SW1#

Lab은 다음을 보고했습니다.

```
DME Successful Collections: 513 Payloads: 513 Failed: 0 Skipped: 44 Dropped: 0
```

Successful(성공) 및 Payload(페이로드) 카운터는 DME 텔레메트리 데이터가 수집되고 있으며 텔레메트리 페이로드가 생성되고 있음을 확인합니다.

건너뛴 컬렉션 44개는 Lab에서 원격 분석 대상을 일시적으로 사용할 수 없는 동안 관찰된 기록 카운터입니다. 이러한 카운터는 나중에 Basic Troubleshooting(기본 문제 해결) 섹션에서 검사됩니다.

추가 센서별 경로 정보는 다음을 사용합니다.

```
N9K-TELEMETRY-SW1# show telemetry data collector details
```

이 명령은 어떤 구성된 센서 경로가 성공적인 수집, 실패한 수집, 건너뛴 수집 또는 삭제된 수집에 기여했는지 식별할 수 있습니다.

수신기에서 정기 데이터 확인

서브스크립션 2는 60초마다 Ethernet1/10 통계를 수집합니다.

텔레메트리 수신기가 21:44:45 UTC(Coordinated Universal Time)에 이러한 입력 통계를 보고했습니다.

```
timestamp: 2026-09-17T21:44:45Z
source: N9K-TELEMETRY-SW1
subscription: 2
path: sys/intf/phys-[eth1/10]/dbgIfIn

broadcastPkts: 5074
discards: 0
errors: 0
multicastPkts: 143403
octets: 12870535
ucastPkts: 31256
```

1분 후, 다른 샘플을 받았습니다.

```
timestamp: 2026-09-17T21:45:45Z
source: N9K-TELEMETRY-SW1
subscription: 2
path: sys/intf/phys-[eth1/10]/dbgIfIn

broadcastPkts: 5075
discards: 0
errors: 0
multicastPkts: 143430
octets: 12875428
ucastPkts: 31292
```

두 샘플 간의 카운터 변경 사항은 아래에 요약되어 있습니다.

카운터	21:44:45	21:45:45
유니캐스트 패킷	31,256	31,292
멀티캐스트 패킷	143,403	143,430
브로드캐스트 패킷	5,074	5,075

카운터	21:44:45	21:45:45
옥텟	12,870,535	12,875,428
오류	0	0
폐기	0	0

타임스탬프는 약 60초 간격으로 분리되어 서브스크립션 2에 대해 구성된 샘플링 간격과 일치합니다.

패킷 및 바이트 카운터가 증가하면 업데이트된 인터페이스 통계가 수집되어 수신자에게 전달되고 있음을 확인할 수도 있습니다.

출력 및 운영 정보 확인

dbgIfOut 센서 경로는 Ethernet1/10에 대한 출력 통계를 제공합니다.

UTC 21:45:45에 수집된 샘플:

```

broadcastPkts: 8
discards: 0
errors: 0
multicastPkts: 4712
octets: 2444721
ucastPkts: 4092

```

물리적 센서 경로는 동일한 인터페이스에 대한 운영 특성을 제공합니다.

수신자가 보고한 내용:

```

adminSt: up
operSt: up
operSpeed: 1G
operDuplex: full
operMtu: 1500
operDescr: TELEMETRY-COLLECTOR

```

이러한 샘플은 센서 그룹 2가 구성된 3개의 DME 센서 경로를 통해 인터페이스 통계와 작동 정보를 모두 제공하는지 확인합니다.

정기 텔레메트리 확인 요약

주기적인 텔레메트리 확인은 다음을 확인합니다.

확인	결과
gRPC 전송 세션	연결됨
GPB 인코딩	확인됨
센서 그룹 1	10000ms에 실행
센서 그룹 2	60000ms에 실행
DME 컬렉션	성공
실패한 수집	0
삭제된 페이로드	0
주기적 수신기 샘플	수신됨
인터페이스 카운터	샘플 간 업데이트

이벤트 기반 텔레메트리 확인

주기적인 텔레메트리를 확인한 후 Subscription 3을 사용하여 Loopback100 Managed Object에 대한 제어된 변경 사항을 생성하여 이벤트 기반 텔레메트리를 검증합니다.

이벤트 기반 서브스크립션 확인

Use:

```
N9K-TELEMETRY-SW1# show telemetry control database
```

```
Subscription Database size = 3
```

Subscription ID	Data Collector Type	Reachability	Configuration Method
3	DME	Reachable	CONFIG_DME-ADMIN
2	DME	Reachable	CONFIG_DME-ADMIN
1	DME	Reachable	CONFIG_DME-ADMIN

<snip>

```
Sensor Group Database size = 3
```

Row ID	Sensor Group ID	Sensor Group type	Sampling interval(ms)	Linked subscriptions	SubID

1 3 Event /DME 0/No Timer 1 3

<snip>

Sensor Path Database size = 5

Row ID	Subscribed	Linked Groups	Sec Groups	Retrieve level	Path(GroupId) : Query : Filter
4	Yes	1	0	Self	sys/intf/lb-[lo100](3) : NA : NA

<snip>

Subscription Id: 3

Snapshot Stats:

Sent = 1
Error = 0
Drops = 0

The Sensor Group Database reports:

Sensor Group ID: 3
Sensor Group Type: Event / DME
Sampling Interval: 0 / No Timer
Subscription ID: 3

The Event / DME type and 0 / No Timer state confirm that Sensor Group 3 is configured for event-based c
The associated sensor path is:
sys/intf/lb-[lo100]

동일한 텔레메트리 데이터베이스에서 서브스크립션 3을 통해 센서 경로가 서브스크립션되었음을
확인합니다.

이벤트 기반 업데이트 생성

이벤트 생성을 확인하려면 Loopback100 설명을 수정합니다.

```
N9K-TELEMETRY-SW1(config)# interface loopback100
N9K-TELEMETRY-SW1(config-if)# description TELEMETRY-EVENT-DEMO
```

텔레메트리 수신기가 보고했습니다.

```
{
"timestamp": "2026-09-17T21:41:12Z",
"source": "N9K-TELEMETRY-SW1",
"subscription": "3",
"event": {
```

```
"descr": "TELEMETRY-EVENT-DEMO",
"dn": "sys/intf/lb-[lo100]",
"status": "modified"
}
}
```

dn은 모니터링되는 DME 객체를 식별하고, descr 특성은 변경된 속성을 식별합니다.

다음으로, 관리적으로 인터페이스를 비활성화하고 복원합니다.

```
N9K-TELEMETRY-SW1(config)# interface loopback100
N9K-TELEMETRY-SW1(config-if)# shutdown
```

그 다음:

```
N9K-TELEMETRY-SW1(config-if)# no shutdown
```

수신기가 두 상태 변경을 감지했습니다.

타임스탬프	변경된 특성	가치
21:41:12	설명	텔레메트리 이벤트 데모
21:41:19	관리자	아래로
21:41:24	관리자	위로

각 업데이트에서 동일한 모니터링된 개체를 참조했습니다.

```
sys/intf/lb-[lo100]
```

객체 상태를 수정된 것으로 보고했습니다.

이러한 결과를 통해 모니터링되는 관리 객체의 다른 속성을 변경하면 개별 텔레메트리 업데이트가 생성될 수 있음을 확인할 수 있습니다.

초기 스냅샷 및 이벤트 메시지 확인

이벤트 기반 서브스크립션이 설정되면 NX-OS는 모니터링되는 객체의 초기 스냅샷을 생성했습니다.

센서 경로 데이터베이스가 보고했습니다.

스냅샷 통계:

Sent = 1 Error = 0 Drops = 0

세 가지 제어된 변경 후, 동일한 센서 경로가 보고되었습니다.

메시지 통계:

Sent = 3 Error = 0 Drops = 0

따라서 결과는 다음과 같이 요약할 수 있습니다.

수집	개수
초기 스냅샷	1
설명 수정	1
관리 상태 작동 중지	1
관리 상태 up	1
Total	4

초기 스냅샷은 서브스크립션이 활성화될 때 모니터링되는 객체의 상태를 나타내며, 후속 메시지는 객체 변경에 해당합니다.

이벤트 수집 통계 확인

```
N9K-TELEMETRY-SW1# show telemetry event collector stats
-----
Row ID      Collection Count  Latest Collection Time    Sensor Path(GroupId)
-----
1           4                Thu Sep 17 21:41:24.045 UTC sys/intf/1b-[1o100](3)
N9K-TELEMETRY-SW1#
```

Lab은 다음을 보고했습니다.

Collection Count: 4
Latest Collection Time: Thu Sep 17 21:41:24.045 UTC
Sensor Path: sys/intf/1b-[1o100]

수집 수는 테스트 중에 생성된 1개의 초기 스냅샷 및 3개의 변경 사항과 일치합니다.

이벤트 컬렉터 오류 확인

```
N9K-TELEMETRY-SW1# show telemetry event collector errors
```

Error Description	Error Count
Dme Event Subscription Init Failures	- 0
Event Data Enqueue Failures	- 0
Event Subscription Failures	- 0
Pending Subscription List Create Failures	- 0
Subscription Hash Table Create Failures	- 0
Subscription Hash Table Destroy Failures	- 0
Subscription Hash Table Insert Failures	- 0
Subscription Hash Table Remove Failures	- 0

N9K-TELEMETRY-SW1#

테스트 중에 이벤트 컬렉터 오류가 관찰되지 않았습니다.

이벤트 기반 텔레메트리 확인 요약

이벤트 기반 텔레메트리 확인에서는 다음을 확인합니다.

확인	결과
센서 그룹	3
센서 경로	sys/intf/lb-[lo100]
컬렉션 유형	이벤트/DME
샘플링 간격	0/타이머 없음
초기 스냅샷	전송됨
설명 변경	탐지됨
adminSt 중단	탐지됨
adminSet up(관리자 설정)	탐지됨
총 수집	4
이벤트 수집기 오류	0

제어된 루프백100 변화를 성공적으로 탐지하면 서브스크립션 3이 예상대로 작동하고 있음을 확인할 수 있습니다.

다음 섹션에서는 스트리밍 텔레메트리 작업을 평가하는 데 사용되는 기본 확인 및 문제 해결 명령

을 살펴봅니다.

기본 문제 해결

텔레메트리 데이터가 예상대로 수신되지 않을 경우, 문제가 전송 세션, 데이터 수집, 센서 컨피그레이션 또는 외부 수신자와 관련이 있는지 확인하여 트러블슈팅을 시작합니다.

이 워크플로에서는 이 실습 중에 관찰된 문제를 사용합니다. 여기서 NX-OS는 44개의 건너뛴 수집과 하나의 기록 gRPC 전송 오류를 보고했습니다.

현재 전송 상태 확인

```
N9K-TELEMETRY-SW1# show telemetry transport
Session Id  Dst Grp  IP Address      Port      Encoding  Transport  Status
-----
0           1         192.168.100.10  57000     GPB       gRPC       Connected

-----
Retry buffer Size:                10485760
Event Retry Messages (Bytes): 0
Timer Retry Messages (Bytes): 0
Total Retries sent:                0
Total Retries Dropped:            0

N9K-TELEMETRY-SW1#
```

최종 확인 과정에서 텔레메트리 세션이 다음을 보고했습니다.

```
Session ID: 0
Destination Group: 1
IP Address: 192.168.100.10
Port: 57000
Encoding: GPB
Transport: gRPC
Status: Connected
```

Connected(연결됨) 상태에서는 전송 세션이 현재 설정되었음을 확인합니다.

그러나 현재 세션 상태가 연결 문제가 이전에 발생했는지 여부를 반드시 나타내는 것은 아닙니다. 따라서 기록 수집 및 전송 카운터도 검토합니다.

데이터 수집 카운터 검토

N9K-TELEMETRY-SW1# show telemetry data collector brief

Row ID	Collector Type	Successful	Payloads	Failed	Skipped
1	YANG	0	0	0	0
2	DME	513	513	0	44
3	NX-API	0	0	0	0

N9K-TELEMETRY-SW1#

Lab은 다음을 보고했습니다.

DME Successful Collections: 513
Payloads: 513
Failed: 0
Skipped: 44
Dropped: 0

Successful 및 Payload 카운터는 DME 텔레메트리 수집 및 페이로드 생성이 발생했음을 확인합니다.

그러나 Skipped 카운터는 44개의 예약된 수집이 수행되지 않았음을 나타냅니다.

어떤 센서 경로가 영향을 받았는지 확인하려면 다음을 사용합니다.

N9K-TELEMETRY-SW1# show telemetry data collector details

Row ID	Successful	Payloads	Failed	Skipped	Dropped	Sensor Path(G
1	414	414	0	29	0	sys/intf/phys
2	54	54	0	5	0	sys/intf/phys
3	54	54	0	5	0	sys/intf/phys
4	54	54	0	5	0	sys/intf/phys

N9K-TELEMETRY-SW1#

보고된 자세한 출력:

센서 경로	건너뛰
sys/intf/phys-[eth1/10]	29
sys/intf/phys-[eth1/10]/dbgIfIn	5

센서 경로	건너뛸
sys/intf/phys-[eth1/10]/dbgIfOut	5
sys/intf/phys-[eth1/10]/phys	5
Total	44

건너뛸 컬렉션은 주기적인 센서 경로 전체에 배포되었으며, 이는 문제가 단일 DME 객체로 격리되지 않았음을 나타냅니다.



참고: 수집 카운터는 누적됩니다. 0이 아닌 기록 카운터가 반드시 동일한 조건이 현재 있을 수 나타내지는 않습니다.

컬렉션을 건너뛸 이유 확인

```
N9K-TELEMETRY-SW1# show telemetry control stats
```

Error Description	Error Count
<snip> Collections skipped due to destination unreachable	44
<snip> N9K-TELEMETRY-SW1#	

이 출력은 건너뛸 수집의 이유를 직접 식별합니다.

목적지 도달 불가 = 44

이 값은 DME 데이터 컬렉터에서 보고한 44개의 건너뛸 컬렉션과 일치합니다.

그러면 조사가 센서 경로 자체에서 벗어나 텔레메트리 대상 및 전송 경로로 이동할 수 있습니다.

전송 오류 검사

show telemetry transport에서 보고한 세션 ID를 사용합니다.

```
N9K-TELEMETRY-SW1# show telemetry transport 0 errors
Session Id: 0
Connection Errors
Connection Error Count: 0
Transmission Errors
```

Tx Error Count: 1
Last Tx Error: Thu Sep 17 21:22:16.008 UTC
Last Tx Return Code: UNAVAILABLE
N9K-TELEMETRY-SW1#

Lab은 다음을 보고했습니다.

Connection Error Count: 0
Tx Error Count: 1
Last Tx Error: Thu Sep 17 21:22:16.008 UTC
Last Tx Return Code: UNAVAILABLE

UNAVAILABLE 반환 코드는 원격 분석 대상과 관련된 gRPC 전송 실패를 기록합니다.

이 Lab에서는 컨피그레이션을 수정하는 동안 외부 수신기가 의도적으로 중지되었다가 다시 시작되었습니다. 이 간격 동안 NX-OS는 텔레메트리 대상에 도달할 수 없습니다. 이는 위에서 관찰한 대상에 도달할 수 없는 수집 카운터에 해당합니다.

중요한 상관 관계는 다음과 같습니다.

관찰	결과
건너뛴 컬렉션	44
대상에 연결할 수 없음	44
전송 Tx 오류	1
마지막 Tx 반환 코드	사용 불가능
현재 전송 상태	연결됨

일치하는 건너뛴 및 대상 도달 불가 카운터는 컬렉션을 건너뛴 이유에 대한 직접적인 증거를 제공합니다.

기록 전송 오류는 동일한 랩 활동 중에 관찰된 전송 실패에 대한 추가 정보를 제공합니다.

전송 복구 확인

수신기에 대한 연결이 복원되면 다음을 사용합니다.

```
N9K-TELEMETRY-SW1# show telemetry transport 0 stats
Session Id: 0
Connection Stats
```

```

Connection Count          3
Last Connected:           Thu Sep 17 21:27:16.010 UTC
Disconnect Count          0
Last Disconnected:        Never
Transmission Stats
Compression:              disabled
Source Interface:         not set()
Transmit Count:           603
Last TX time:             Thu Sep 17 21:51:36.008 UTC
Min Tx Time:              1 ms
Max Tx Time:              414 ms
Avg Tx Time:              6 ms
Cur Tx Time:             1 ms

Flow Stats
Allowed Queued Msgs Size (bytes): 78643200
Current Queued Msgs Size (bytes): 0
Utilization (percent):      0
Max Queued Msgs Size (bytes): 3849
Total Queued Msgs Size (bytes): 865313
Current Msgs Held (# Msgs): 0
Total Msgs held (# Msgs): 604
Max Msgs Held (# Msgs): 5
Msgs Dropped (# Msgs): 0
Flow Control Apply Time (secs): 0
Flow Control Last Applied: Never
<snip>
N9K-TELEMETRY-SW1#

```

최종 실습 확인 결과:

```

Connection Count: 3
Disconnect Count: 0
Transmit Count: 603
Average Transmission Time: 6 ms
Current Transmission Time: 1 ms
Current Queued Messages Size: 0
Transport Utilization: 0%
Messages Dropped: 0
Flow Control Last Applied: Never

```

이 값은 전송 세션이 복구되었으며 대기 중이거나 삭제된 메시지 없이 텔레메트리 데이터를 적극적으로 전송했음을 나타냅니다.

가장 중요한 현재 상태 지표는 다음과 같습니다.

표시기	결과
전송 상태	연결됨
현재 대기열	0
삭제된 메시지	0
Flow Control	적용 안 함

이러한 구분은 텔레메트리 카운터를 트러블슈팅하는 데 중요합니다. 기록 오류는 기본 조건이 해결된 후에도 계속 표시될 수 있습니다.

컨피그레이션 및 이벤트 컬렉터 오류 확인

NX-OS에서 컨피그레이션 또는 이벤트 처리 문제를 보고하는지 확인하는 데 추가 명령을 사용할 수 있습니다.

```
N9K-TELEMETRY-SW1# show telemetry config errors
-----
Row ID          Path                               Sensor Group      Error
-----
Transport Errors
-----
Destination group ID      Source Interface      Configured VRF    Correct VRF
-----
N9K-TELEMETRY-SW1#
```

이벤트 기반 텔레메트리의 경우 다음을 사용합니다.

```
N9K-TELEMETRY-SW1# show telemetry event collector errors
-----
Error Description                               Error Count
-----
Dme Event Subscription Init Failures             - 0
Event Data Enqueue Failures                      - 0
Event Subscription Failures                      - 0
Pending Subscription List Create Failures         - 0
Subscription Hash Table Create Failures          - 0
Subscription Hash Table Destroy Failures         - 0
Subscription Hash Table Insert Failures          - 0
Subscription Hash Table Remove Failures          - 0
N9K-TELEMETRY-SW1#
```

모든 이벤트 수집기 오류 카운터도 0입니다.

이러한 결과는 건너뀀 정기 수집을 조사할 때 컨피그레이션 및 이벤트 수집기 실패를 배제하는 데 도움이 됩니다.

문제 해결 워크플로

이 조사 과정에서 사용된 명령은 다음과 같이 요약할 수 있습니다.

명령을 사용합니다	목적
원격 분석 전송 표시	현재 전송 세션 상태를 확인합니다.
텔레메트리 데이터 컬렉터 개요 표시	성공, 실패, 건너뛴 또는 삭제된 컬렉션을 식별합니다.
원격 분석 데이터 수집기 세부 정보 표시	영향을 받는 센서 경로를 확인합니다.
원격 분석 제어 통계 표시	수집을 건너뛴 이유를 확인합니다.
show telemetry transport <session-id> 오류	전송 실패를 검사합니다.
show telemetry transport <session-id> stats	전송 복구, 대기열 및 삭제를 검토합니다.
원격 분석 구성 오류 표시	텔레메트리 컨피그레이션 오류를 식별합니다.
원격 분석 이벤트 수집기 오류 표시	이벤트 컬렉터 오류를 식별합니다.

이 Lab에서 트러블슈팅 시퀀스는 DME 센서 경로 또는 컨피그레이션 오류가 아닌 임시 텔레메트리 대상 도달 조건을 식별했습니다.

수신자를 다시 사용할 수 있게 되면 텔레메트리 전송이 Connected(연결됨) 상태로 반환되고, 수집이 다시 시작되며, 현재 큐 또는 메시지 삭제 조건이 관찰되지 않았습니다.

실제 응용 프로그램 예: 시스템 리소스 모니터링

시스템 리소스 모니터링은 스트리밍 텔레메트리의 일반적인 활용 사례입니다. CPU 및 메모리 정보는 Nexus 스위치에서 외부 모니터링 플랫폼으로 주기적으로 보내져 기록 분석, 대시보드, 용량 모니터링 및 알림을 처리할 수 있습니다.

Cisco NX-OS는 일반적으로 모니터링되는 정보에 대해 미리 정의된 텔레메트리 경로 레이블을 제공합니다. 이 예에서는 리소스 경로 레이블을 사용하여 시스템 CPU 및 메모리 정보를 수집합니다.

시스템 리소스 원격 분석 구성

리소스 경로 레이블을 사용하여 새 센서 그룹을 생성합니다.

```
N9K-TELEMETRY-SW1(config)# telemetry
N9K-TELEMETRY-SW1(config-telemetry)# sensor-group 4
N9K-TELEMETRY-SW1(conf-tm-sensor)# path resources
```

서브스크립션 4를 생성하고 센서 그룹 4를 대상 그룹 1과 연결합니다.

```
N9K-TELEMETRY-SW1(config-telemetry)# subscription 4
N9K-TELEMETRY-SW1(conf-tm-sub)# dst-grp 1
N9K-TELEMETRY-SW1(conf-tm-sub)# snsr-grp 4 sample-interval 10000
```

센서 그룹 4는 미리 정의된 리소스 경로 레이블을 사용합니다. 서브스크립션 4는 센서 그룹을 기존 텔레메트리 대상과 연결하고 0이 아닌 샘플링 간격으로 정기적인 수집을 구성합니다.

관련 컨피그레이션은 다음과 같습니다.

```
telemetry
  destination-group 1
    ip address 192.168.100.10 port 57000 protocol gRPC encoding GPB
    use-vrf default

  sensor-group 4
    path resources

  subscription 4
    dst-grp 1
    snsr-grp 4 sample-interval 10000
```

리소스 경로 레이블 검토

리소스 경로 레이블로 표시되는 DME 경로를 검토하려면 다음 명령을 실행합니다.

```
N9K-TELEMETRY-SW1# show telemetry usability resources
```

```
1) label_name      : resources
   path_name       : sys/proc
   query_type      : poll
   <snip>

2) label_name      : resources
   path_name       : sys/procsys
   query_type      : poll
   <snip>

3) label_name      : resources
   path_name       : sys/procsys/systemem
   query_type      : event
   query_condition : query-target-filter=and(updated(procSysMem.memstatus),ne(procSysMem.memstatus))
```

출력은 리소스 레이블이 여러 기본 DME 경로를 나타낸다는 것을 보여줍니다.

sys/proc 및 sys/procsys 경로는 폴링 쿼리를 사용하고 프로세스 및 시스템 리소스 정보를 제공합니다. sys/procsys/systemem 경로는 모니터링되는 메모리 상태가 업데이트되어 더 이상 양호하지 않을 때 변경 사항을 보고할 수 있는 이벤트 쿼리를 사용합니다.

이는 개별 DME DN을 직접 지정하는 것과 사전 정의된 텔레메트리 경로 레이블을 사용하는 것의 차이를 보여 줍니다.

예를 들면 다음과 같습니다.

```
Individual DME path:  
path sys/intf/phys-[eth1/10]  
Predefined path label:  
path resources
```

시스템 리소스 원격 분석 확인

show telemetry control 데이터베이스를 사용하여 센서 그룹 4 및 서브스크립션 4의 상태를 확인합니다.

센서 그룹 데이터베이스는 다음을 보고합니다.

Sensor Group ID	Sensor Group type	Sampling interval(ms)	Linked subscriptions	SubID
4	Timer /DME	10000/Running	1	4

Timer /DME 유형 및 10000/Running 샘플링 간격은 센서 그룹 4가 주기적인 DME 텔레메트리 원본으로 작동하고 있음을 확인합니다.

센서 경로 데이터베이스에는 리소스 레이블과 연결된 기본 경로도 표시됩니다.

예를 들면 다음과 같습니다.

```
resources:sys/procsys(4)  
GPB Encoded Data size in bytes (Cur/Min/Max): 20221/20221/20229  
Subscription Id: 4  
Message Stats:  
Sent = 14  
Error = 0  
Drops = 0
```

프로세스 관련 경로는 활성 텔레메트리 수집도 보고합니다.

resources:sys/proc(4)

GPB Encoded Data size in bytes (Cur/Min/Max): 82284/81262/85098
Subscription Id: 4

Message Stats:

Sent = 14
Error = 0
Drops = 0

이러한 카운터는 리소스 정보가 수집되고 GPB로 인코딩되며 메시지 오류 또는 삭제 없이 전송됨을 확인합니다.

NX-OS 시스템 리소스와 비교

기존 NX-OS CLI를 사용하여 스위치의 현재 CPU 및 메모리 상태를 표시할 수 있습니다.

```
N9K-TELEMETRY-SW1# show system resources
Load average:  1 minute: 0.57   5 minutes: 0.58   15 minutes: 0.63
Processes   :   854 total, 2 running
CPU states  :   14.64% user,    3.50% kernel,   81.85% idle
<snip>
Memory usage: 24530808K total,   9315248K used,   15215560K free
Kernel buffers:  22104K Used
Kernel cached :  6255520K Used
Current memory status: OK
```

CLI는 시스템 리소스에 대한 즉각적인 보기를 제공합니다.

스트리밍 텔레메트리를 사용하면 동일한 유형의 CPU 및 메모리 정보를 외부 수신기로 내보내 여러 샘플을 저장하고 시간이 지남에 따라 분석할 수 있습니다.

CPU 사용률이 빠르게 변할 수 있습니다. 따라서 CLI에서 표시하는 CPU 값과 텔레메트리를 통해 수신하는 값은 서로 다른 시간에 샘플을 수집할 때 다를 수 있습니다.

수신자의 리소스 데이터 확인

텔레메트리 수신기가 서브스크립션 4와 관련된 시스템 리소스 정보를 디코딩했습니다.

다음 예에서는 하나의 텔레메트리 샘플의 CPU 및 메모리 정보를 보여 줍니다.

```
{
  "timestamp": "2026-09-18T23:15:08Z",
  "source": "N9K-TELEMETRY-SW1",
  "cpu": {
    "user": 11,
    "kernel": 2,
    "idle": 85,
    "averageLast60Seconds": 7.099999904632568
  },
  "memory": {
    "status": "OK",
    "utilization": 38.18336486816406,
    "usedKB": 9366688,
    "freeKB": 15164120,
    "totalKB": 24530808
  }
}
```

수신기 출력은 서브스크립션 4의 CPU 및 메모리 정보가 성공적으로 디코딩되고 외부 모니터링에 사용 가능함을 확인합니다.

비교를 위해 NX-OS CLI는 총 메모리 약 24.5GB 중 사용된 메모리 약 9.3GB와 현재 메모리 상태가 정상이라고 보고했습니다. 텔레메트리 샘플은 동일한 총 메모리 값, 약 38%의 메모리 사용률, 동일한 정상 메모리 상태를 보고합니다.

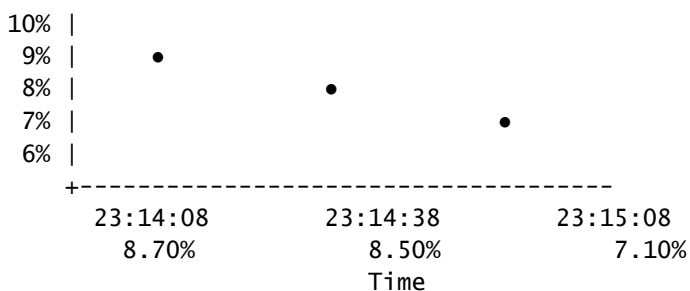
CPU 사용률이 동적으로 변경되고 측정값이 동일한 순간에 반드시 수집되지 않기 때문에 CLI 및 텔레메트리 샘플 간에 CPU 값이 달라질 수 있습니다.

시간에 따른 리소스 사용률 관찰

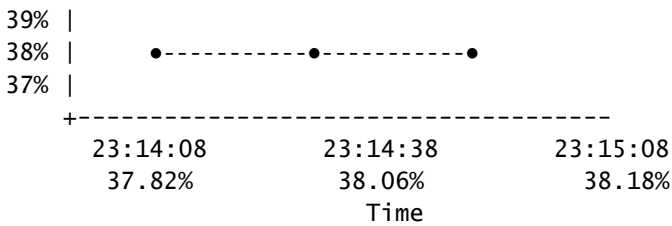
여러 텔레메트리 샘플을 사용하여 시스템 리소스 사용률이 시간에 따라 어떻게 변하는지 관찰할 수 있습니다.

이 샘플은 실습 중에 서브스크립션 4에서 받았습니다.

CPU 평균 사용률 - 지난 60초



메모리 사용률



이러한 샘플은 주기적인 텔레메트리를 통해 단일 순간 측정 대신 시스템 동작의 시간 기반 보기를 제공하는 방법을 보여줍니다. 프로덕션 환경에서는 모니터링 또는 관찰 가능 플랫폼이 더 많은 수의 샘플을 저장하고 샘플을 사용하여 트렌드를 식별하고 알림을 생성하며 기록 대시보드를 구축할 수 있습니다.

결론

Cisco NX-OS Streaming Telemetry는 Cisco Nexus 9000 스위치에서 외부 텔레메트리 수신기로 운영 정보를 내보내기 위한 구조화된 메커니즘을 제공합니다.

이 문서에서는 DME 센서 경로, GPB 인코딩, gRPC 전송, 대상 그룹, 센서 그룹, 서브스크립션을 비롯한 스트리밍 텔레메트리의 기본 구성 요소를 살펴보았습니다.

랩 환경을 사용하여 주기적 및 이벤트 기반 텔레메트리를 모두 구성하고 검증했습니다. 정기적인 서브스크립션은 Ethernet1/10 통계 및 운영 정보를 수집하는 데 사용되었고, 이벤트 기반 서브스크립션은 Loopback100 Managed Object에 대한 제어된 변경 사항을 탐지하는 데 사용되었습니다.

실용적인 시스템 리소스 모니터링 예는 또한 미리 정의된 리소스 경로 레이블을 사용하여 CPU 및 메모리 정보를 외부 텔레메트리 수신기로 내보내는 방법도 시연했습니다.

확인 및 트러블슈팅 예는 NX-OS 텔레메트리 명령을 사용하여 전송 연결, 데이터 수집, 이벤트 처리 및 기록 수집 실패를 검증하는 방법을 시연했습니다.

이러한 개념은 Cisco Nexus 9000 NX-OS의 기본 스트리밍 텔레메트리 구축을 이해, 구현, 확인 및 트러블슈팅할 수 있는 기반을 제공합니다.

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.