



취리히 **Telecommunications Service Operations Management**

마지막 업데이트 날짜: 2025년 12월 17일

기
계
변
역

해당 자료는 사용자 편의를 위해 번역 소프트웨어를 사용하여 번역되었습니다. 정확한 번역을 제공하기 위해 합당한 노력을 기울였지만, 사람이 직접 번역하는 것을 대체할 수 있는 기계 번역은 없습니다. 번역은 "기계 번역 결과 그대로" 제공됩니다. 다른 언어로 번역한 내용의 정확성, 신뢰성 또는 무결성에 대해서 명시적이든 묵시적이든 어떠한 보증도 하지 않습니다. 일부 콘텐츠는 번역 소프트웨어의 한계로 인해 정확하게 번역되지 않을 수 있습니다. 해당 자료의 공식 언어는 영어입니다. 번역에서 발생한 불일치 또는 차이점은 구속력이 없으며 규정 준수나 시행을 위한 법적 효력이 없습니다.

여기에 표시된 일부 예와 그래픽은 설명을 위해서만 제공됩니다.
ServiceNow 제품 또는 서비스와의 실제 연관 또는 연결을 의도하지 않았으며 그렇게 유추해서는 안 됩니다.

ServiceNow, ServiceNow 로고, Now 및 기타 ServiceNow 표시는 미국 및/또는 기타 국가에서 ServiceNow, Inc.의 상표 및/또는 등록 상표입니다. 기타 회사 및 제품 이름은 관련된 해당 회사의 상표일 수 있습니다.

다음 사이트에서 ServiceNow 웹 사이트 이용 약관을 읽어보십시오.
www.servicenow.com/terms-of-use.html

본사
2225 Lawson Lane
Santa Clara, CA 95054
United States
(408) 501-8550

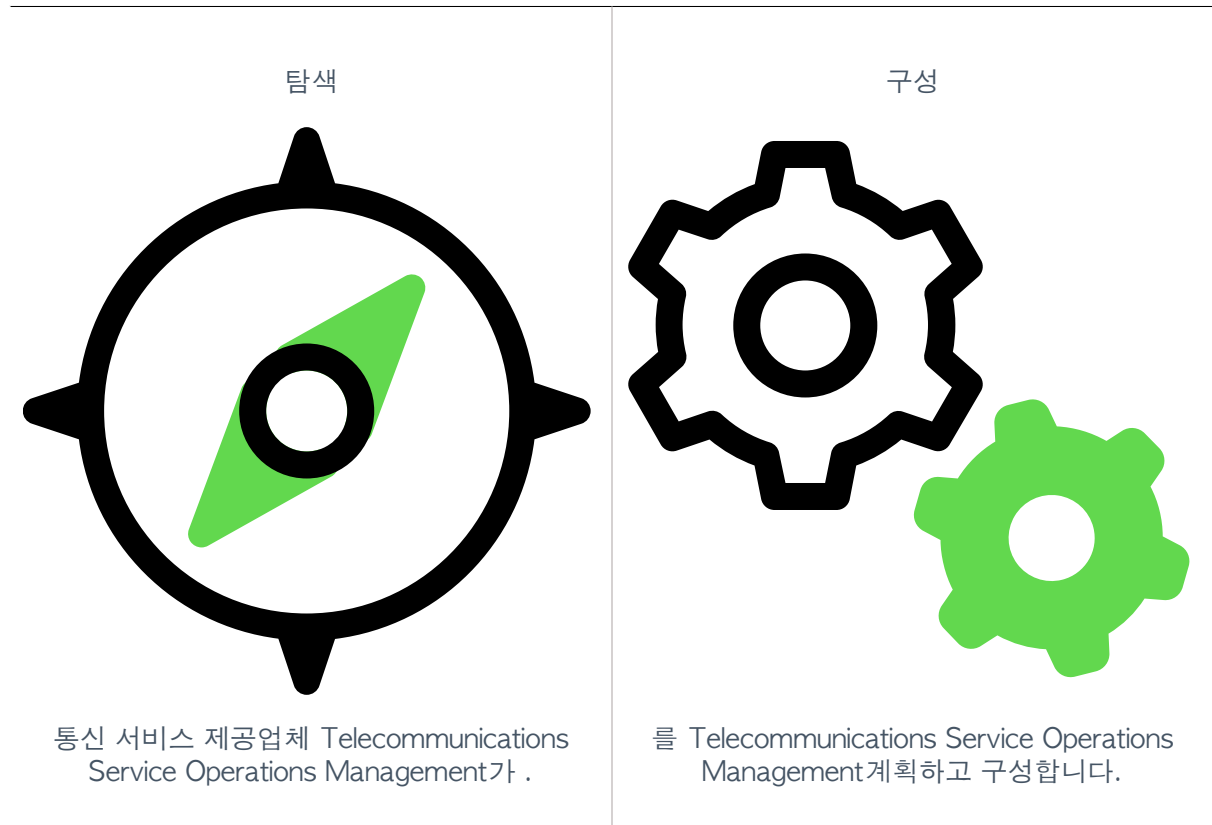
목차

Telecommunications Service Operations Management.....	4
Telecommunications Service Operations Management 탐색.....	6
통신 API 이벤트 알림.....	8
통신 가시성.....	9
Telecommunications Service Operations Management 구성.....	40
통신 API 알림 구성.....	41
통신 가시성 설정.....	44
Telecommunications Service Operations Management 사용.....	82
통신 검색 패턴 사용.....	82
Nokia Altiplano 임포트 실행.....	87
TSOM 스키마 검사기를 사용하여 JSON 페이로드 확인.....	89
통신 불일치 감사 실행.....	90
속성 값 불일치에 대한 보고서 생성.....	94
Telecommunications Service Operations Management 참조.....	95
통신 API 알림과 함께 설치되는 시스템 구성요소.....	95
Nokia Altiplano와 함께 설치되는 시스템 구성요소.....	96
Nokia NSP와 함께 설치되는 시스템 구성요소.....	97
(store-future: BEGIN review)	
(End)	
통신 불일치 식별 및 조정 구성요소.....	97

Telecommunications Service Operations Management

통신 서비스 제공업체(CSP)가 고객에게 영향을 미치기 전에 네트워크 및 서비스 문제를 사전 예방적으로 모니터링, 분석 및 해결할 수 있도록 (TSOM)을 지원하는 방법을 Telecommunications Service Operations Management 알아보십시오. 를 ServiceNow AI Platform기반으로 구축된 TSOM은 분산된 다중 도메인 통신 환경 전반에 걸쳐 통합 운영 뷰를 제공하여 팀이 서비스 가용성, 운영 효율성 및 고객 만족도를 개선할 수 있도록 지원합니다.

시작하기



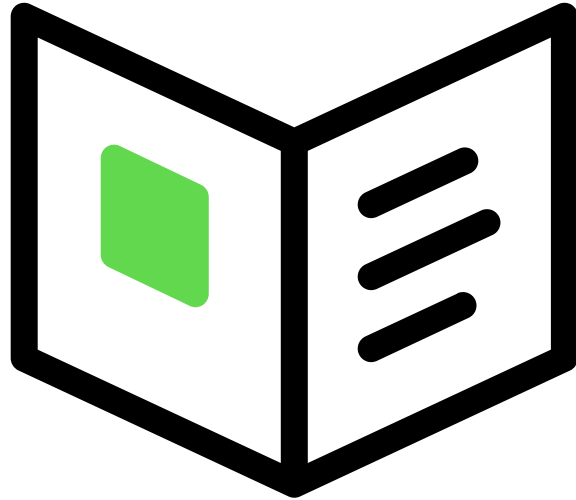
기계면역

사용



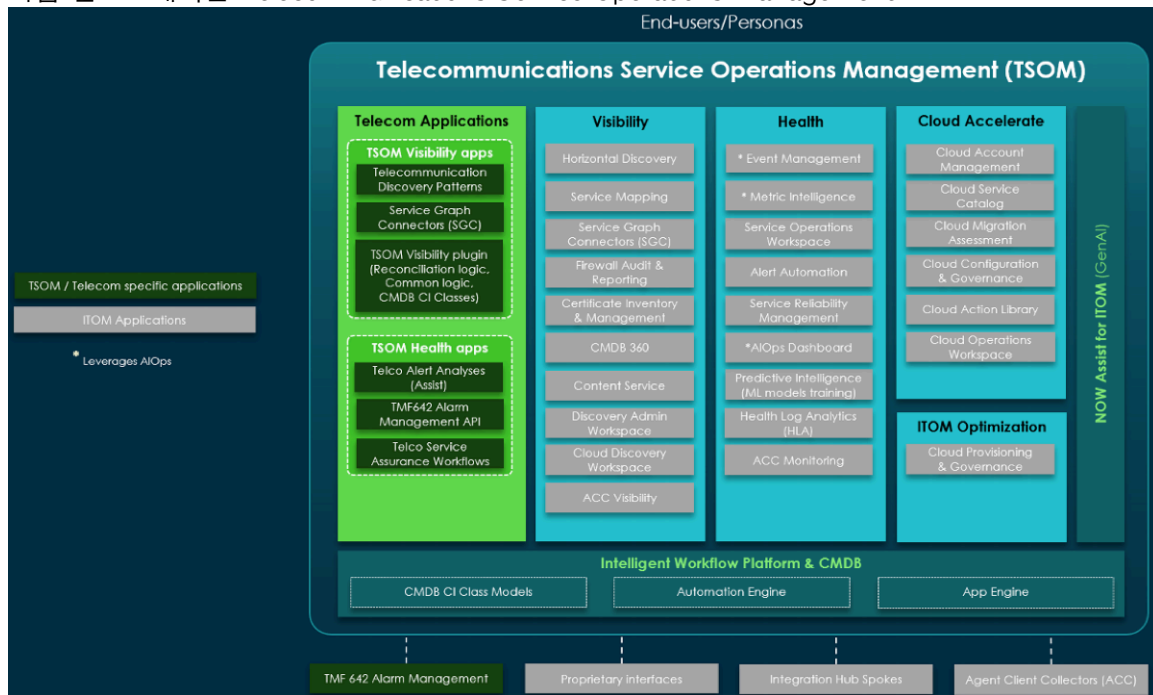
포괄적인 통신 서비스 운영을 추적하는 데 사용합니다 Telecommunications Service Operations Management .

참조



참조 정보를 가져옵니다 Telecommunications Service Operations Management .

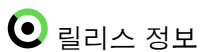
다음 인포그래픽은 Telecommunications Service Operations Management



아키텍처를 이해하는 데 도움이 됩니다.

추가 자원

유용한 정보를 제공할 수 있는 몇 가지 ServiceNow 리소스는 다음과 같습니다.



릴리스 정보

이 릴리스의 새로운 기능과 변경된 내용에 대한 자세한 내용은 다음을 참조하십시오.
[Telecommunications Service Operations Management \(TSOM\) release notes](#)



ServiceNow 커뮤니티

에서 다른 Telecommunications Service Operations Management 사용자와 커넥트
[ServiceNow 커뮤니티](#)



고객 성공 센터

다음에서 역할과 관련된 유용한 자원을 찾고 베스트 프랙티스를 살펴보십시오.
[Customer Success Center](#)



ServiceNow University

[ServiceNow 대학교](#) 에서 실시간 과정, 자기 주도형 교육 및 경력 자원에
 액세스하십시오.



지원

에서 고객 서비스 지원에 문의하십시오. <https://support.servicenow.com/now>

Telecommunications Service Operations Management

탐색

통신 서비스 제공업체(CSP)가 고객에게 영향을 미치기 전에 네트워크 및 서비스 문제를 사전 예방적으로 모니터링, 분석 및 해결할 수 있도록 (TSOM)을 지원하는 방법을 Telecommunications Service Operations Management 알아보십시오. 를 ServiceNow AI Platform기반으로 구축된 TSOM은 분산된 다중 도메인 통신 환경 전반에 걸쳐 통합 운영 뷰를 제공하여 팀이 서비스 가용성, 운영 효율성 및 고객 만족도를 개선할 수 있도록 지원합니다.

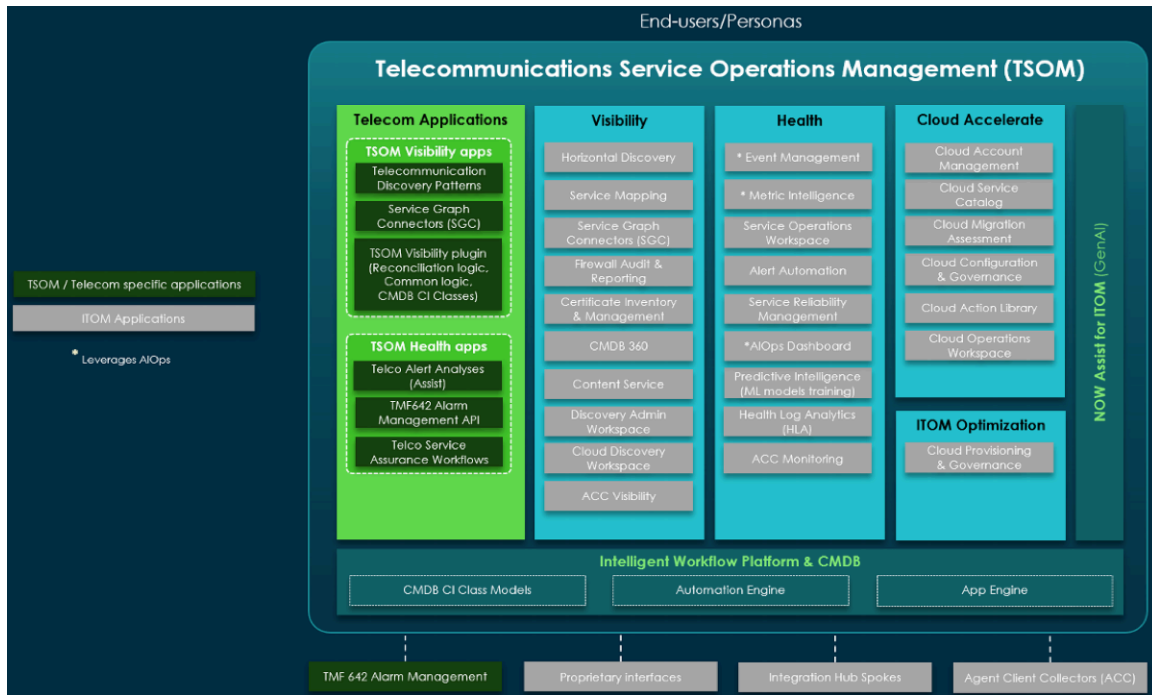
Telecommunications Service Operations Management 는 네트워크 및 서비스 상태에 대한 완벽한 가시성을 제공하는 통신 사업자를 위한 솔루션입니다. TSOM은 표준 API와 의 기능을 ServiceNow AI Platform사용하여 액세스, 전송 및 코어와 같은 네트워크 도메인 전반에서 이벤트를 수집하고, 상관 관계를 지정하고, 우선 순위를 지정합니다. TSOM은 실행 가능한 실시간 인사이트를 제공하여 프런트 라인 팀과 백오피스 팀 모두 서비스에 영향을 미치는 문제를 해결하고 일관된 성과를 유지할 수 있도록 지원합니다.

TSOM 작동 방식

Telecommunications Service Operations Management 는 기존 모니터링 및 원격 측정 플랫폼과 연결하고, 실행 가능한 패턴을 식별하고, 해결 워크플로우를 자동화하여 통신 운영을 단순화합니다. 다음을 활용합니다.

- 표준화된 경보 수집을 위한 통신 API 알림을 통한 외부 이벤트 관리.
- 이벤트 관리 이벤트 상관관계, 억제 및 우선순위 지정을 위해
- 임계치 위반, 성능 저하 및 예외 동작을 실시간으로 감지하는 메트릭 인텔리전스입니다.
- 상태 로그 분석을 통해 로그 데이터를 분석하고, 관련 문제를 분류하고, 사용자에게 영향을 미치기 전에 근본 원인을 식별합니다.
- 영향을 받은 인프라 및 비즈니스 서비스를 기반으로 서비스 중단을 평가하고 추적하는 서비스 영향 분석입니다.
- 서비스 그래프 커넥터 및 검색을 통해 동적인 통신 인식 CMDB를 구축합니다.

TSOM 아키텍처 및 전기 통신 애플리케이션



주요 기능

TSOM 핵심 기능

역량	설명
실시간 이벤트 모니터링	통신 API 알림을 통해 외부 이벤트 관리를 사용하여 다중 도메인 네트워크 모니터링 시스템에서 알람 및 이벤트를 수집합니다.
이벤트 상관 관계 및 분석	이벤트 관리 및 메트릭 인텔리전스를 활용하여 관련 이벤트의 상관 관계를 지정하고 노이즈를 줄이며 예외를 탐지합니다. 자세한 내용은 Event Notification Management Open API 문서를 참조하십시오.
통신 가시성	영향 추적 기능을 포함하여 네트워크 및 서비스 상태의 엔드 투 엔드 시각화를 확보합니다.
서비스 영향 분석	네트워크 또는 인프라 문제가 서비스에 미치는 영향을 이해하고 비즈니스 영향에 따라 정정 우선순위를 지정합니다.
메트릭 인텔리전스	성능 추세를 모니터링하고 임계치 위반 및 메트릭의 예외를 탐지하여 문제를 사전에 식별합니다. 자세한 내용은 Metric Intelligence 문서를 참조하십시오.
상태 로그 분석	ServiceNow 상태 로그 분석 애플리케이션은 사용자에게 영향을 미치기 전에 IT 문제를 방지하는 데 도움이 됩니다. 관련 로그를 분류하고 원시 데이터를 분석하여 사용자가 문제의 근본 원인을 식별할 수 있게 해 줍니다. 자세한 내용은 Health Log Analytics 문서를 참조하십시오.
자동 정정	안내 워크플로우와 플레이북을 사용하여 빠르고 일관되며 감사 가능한 문제 해결을 추진합니다.
통신 인식 CMDB	정확한 근본 원인 분석을 위해 통신별 모델을 사용하여 인프라, 서비스 및 물리적/논리적 CI(구성 항목)를 연결합니다.
경보 관리	상관 관계, 그룹화 및 자동화된 응답 작업을 통해 경보를 효율적으로 관리합니다. 자세한 내용은 Manage and monitor alerts 문서를 참조하십시오.

주요 이점

- MTTD 및 MTTR 감소: 에스컬레이션 전에 서비스에 영향을 주는 문제를 식별하고 대응합니다.
- 운영 효율성 개선: 안내 워크플로우를 통해 노이즈를 줄이고 운영을 간소화합니다.
- 엔드 투 엔드 가시성 확보: 인프라 문제가 고객 대면 서비스에 미치는 영향을 이해합니다.
- 원활한 통합: 개방형 표준을 사용하여 레거시 및 최신 NMS/EMS 도구에서 데이터를 수집합니다.
- 규정 준수 보장: 통신 서비스 보증을 위한 TM Forum 표준에 부합합니다.

주요 가상 사용자

- 네트워크 운영 센터(NOC) 에이전트: 네트워크 이벤트를 모니터링하고 해결 안내를 실행합니다.
- 서비스 보증 관리자: 서비스 추세를 분석하고 해결 메트릭을 추적합니다.
- 시스템 통합자/관리자: 데이터 수집, 상관관계 규칙 및 워크플로우를 구성합니다.
- 백오피스 분석가: CMDB에서 근본 원인을 조사하고 CI 관계를 추적합니다.

앱과의 통합 **ServiceNow AI Platform**

TSOM은 Now Platform에 내장되어 있으며 다음과 같은 핵심 기능과 완벽하게 통합됩니다.

- 디스커버리
- 서비스 매핑
- CMDB
- Flow Designer 및 Playbook

이러한 통합은 통신 도메인 전반에 걸쳐 일관된 워크플로우, 정확한 서비스 모델 및 통합 운영 관리를 보장합니다.

관련 정보

[통신 API 알림을 통한 외부 이벤트 관리](#)

[통신 가시성](#)

통신 **API** 알림을 통한 외부 이벤트 관리

통신 API 알림을 사용하여 고객 네트워크 시스템에서 발생하는 외부 이벤트를 수신하여 .ServiceNow

통신 **API** 알림 소개

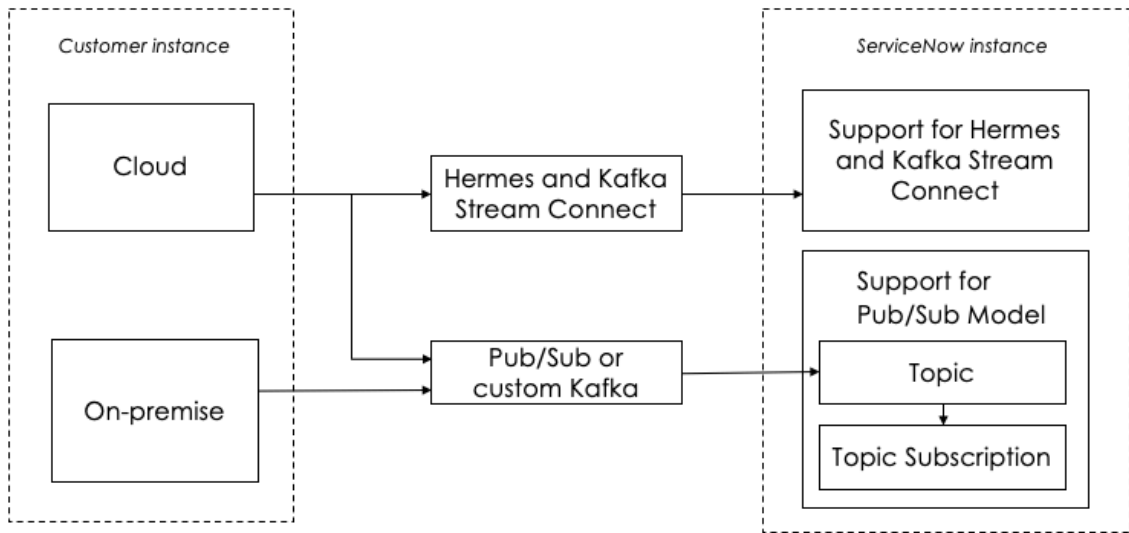
Telecommunications API 알림은 Telecommunications Alarm Management Open API 애플리케이션에서 사용할 수 있는 기능입니다. 통신 API 알림을 사용하면 ServiceNow 외부 네트워크 시스템에서 발생하는 수신 알림을 수신하고 적시에 응답할 수 있습니다. 지점 간 연결이 필요하지 않으므로 플랫폼 기능을 통해 외부 시스템에 이벤트를 브로드캐스트할 수 있습니다.

통신 API 알림은 네트워크에서 구독한 외부 시스템에서 들어오는 알림을 받습니다. 외부 시스템에서 알림을 받으면 애플리케이션을 사용하여 응답에 대한 이벤트를 만들 수 있습니다 이벤트 관리 .에서는 이벤트 관리 수집된 정보를 바탕으로 모든 서비스 영향 이벤트의 통합 보기를 보여주는 대시보드를 제공합니다.

통신 API 알림 데이터 모델

다음 다이어그램에서는 통신 API 알림에 대한 데이터 모델의 구성요소를 보여 줍니다.

API 알림 데이터 모델



통신 API 알림을 사용하면 ServiceNow 게시자/구독자(Pub/Sub) 구독 모델, Hermes 및 Kafka 스트림 커넥트와 같은 이벤트 기반 아키텍처를 통해 수신 알림을 받을 수 있습니다. 클라우드 고객은 두 아키텍처 중에서 유연하게 선택할 수 있지만 온프레미스 고객은 자체 Kafka 또는 Pub/Sub 구독 모델만 사용할 수 있습니다.

- Apache Kafka Stream용 Stream Connect에 대한 자세한 내용은 다음을 참조하세요 [Using Stream Connect for Apache Kafka](#).
- 메시징 서비스에 대한 자세한 내용은 다음 문서를 참조하십시오 [Hermes Messaging Service](#).

Pub/Sub 모델에서 수신 알림은 주제로 분류됩니다. 는 수신 알림을 이러한 주제에 게시하는 데 사용하며 ServiceNow 구독자(고객)는 구독할 주제를 유연하게 선택할 수 있습니다. 이 프로세스를 통해 구독자는 자신의 관심사에 맞는 메시지만 선택할 수 있습니다. 예를 들어, 외부 시스템에서 들어오는 메시지에 대한 주제가 10개 있는 경우, 고객은 요구 사항에 따라 그 중 2개를 구독하도록 선택할 수 있습니다. 따라서 외부 시스템에서 알림을 수신하면 고객이 구독한 두 주제에 대한 이벤트가 특별히 생성됩니다.

관련 정보

[통신 API 알림 구성](#)

[통신 API 알림과 함께 설치되는 시스템 구성요소](#)

통신 가시성

Telecom Visibility는 ITOM Visibility의 검색 및 조정 기능을 확장하여 통신 서비스 제공자의 특정 요구 사항을 충족합니다. 이는 실시간 네트워크 데이터를 검색하고, 불일치를 조정하고, CMDB 및 TNI(통신 네트워크 인벤토리)에서 일관된 통신 모델을 유지함으로써 정확한 통신 인식 네트워크 인벤토리를 가능하게 합니다.

통신 도메인 전반에서 물리적 및 논리적 네트워크 구성요소를 검색하고, 검색과 인벤토리 데이터 간의 불일치를 조정하고, 고급 자동화 및 보증 사용 사례를 지원하기 위해 통신별 CMDB 구조를 유지합니다.

통신 가시성 개요

Telecom Visibility는 통신 네트워크 자원을 검색하고 관리하기 위한 통합 솔루션입니다. 수평 검색, 통신 검색 패턴 및 SGC(서비스 그래프 커넥터)와 같은 입증된 검색 기술을 활용하여 다양한 시스템(예: CLI/SNMP 기반 독립 실행형 장치 또는 API 기반 EMS/NMS/컨트롤러)의 네트워크 데이터를 CMDB로 가져옵니다.

Telecom Visibility는 통신 서비스 제공업체(CSP)와 통신 사업자를 지원합니다.

- 통신별 네트워크 인벤토리 검색 및 유지 관리
- 계획된/설계 인벤토리와 실제 네트워크 상태 간의 불일치 조정
- ANO(Autonomous Network Operations), 서비스 이행 및 보증 워크플로우 지원

핵심 기능

TSOM의 핵심 기능

역량	설명
패턴을 사용한 통신 검색	사전 정의된 통신 Discovery 패턴을 통해 SNMP 및 CLI 프로토콜을 사용하는 라우터 및 스위치와 같은 독립 실행형 네트워크 기능(xNF)을 검색합니다.
서비스 그래프 커넥터 통합	노스바운드 API를 사용하여 외부 EMS/NMS/컨트롤러의 인벤토리 및 구성 데이터를 CMDB로 가져옵니다.
불일치 식별 및 조정	인벤토리된 데이터와 검색된 데이터 간의 불일치(예: 누락된 CI, 모델 불일치, 부실 항목)를 탐지하고 해결합니다.
통신 중심 데이터 모델링	인터페이스 카드, 슬롯, 하위 슬롯, VLAN 및 포트를 포함한 구조적이고 계층적인 통신 데이터를 유지 관리합니다.
CMDB 통합	CMDB 360 및 검색 관리자 작업 공간을 사용하여 검색 소스, 속성 변경 및 조정 결과를 모니터링합니다.

애플리케이션 및 플러그인

- 1. 통신 수평 Discovery 패턴:** SNMP/CLI를 사용하여 라우터 및 스위치와 같은 통신 장치(xNF)를 검색합니다. 라이브 네트워크 인프라를 스캔하여 정확한 인벤토리 업데이트를 가능하게 합니다. 다음과 같은 벤더별 패턴을 지원합니다.
 - Cisco(ASR1K, 7613, Nexus 9000, Nexus 3548)
 - 주니퍼 (MX80, MX104, MX240, MX460)
- 2. TSOM용 SGC(서비스 그래프 커넥터):** API를 통해 인벤토리 데이터를 임포트하기 위해 EMS/NMS/컨트롤러와 사전 구축된 통합입니다. 지원:
 - 예약된 검색 또는 요청 시 검색
 - 멀티 인스턴스 구성
 - Nokia Altiplano(GPON 네트워크)와 같은 시스템과의 통합
- 3. 통신 가시성 플러그인:** 검색 및 불일치 조정을 위해 공유된 로직과 기본 구성요소를 제공합니다. 보장:
 - TSOM 워크플로우 전반에서 일관된 동작
 - 조정 및 정정을 위해 재사용 가능한 구성요소
- 4. 통신용 CMDB CI 클래스 모델 (필수 버전: 1.69.0):** 다음을 포함하여 통신별 클래스 및 관계를 포함하도록 CMDB를 확장합니다.

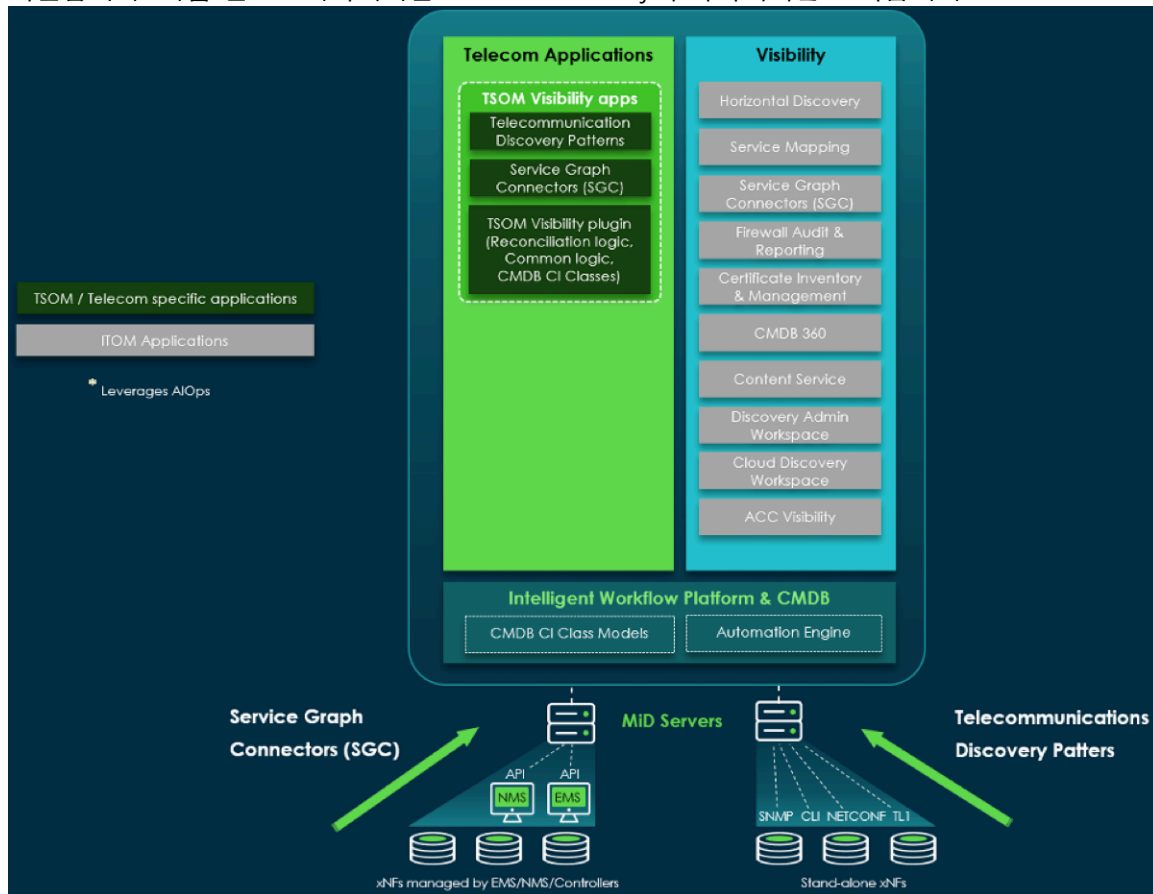
- 인터페이스 카드
- 슬롯 및 하위 슬롯
- 네트워크 인터페이스

통신 가시성 아키텍처

Telecom Visibility는 다음과 같은 직접 및 간접 검색 방법을 결합합니다.

- SNMP/CLI를 사용한 직접 검색(예: 독립 실행형 라우터/스위치)
- EMS/NMS/컨트롤러의 API를 사용한 간접 검색(예: Nokia Altiplano)

이는 CMDB 및 TNI에 공급되어 의사 결정, 자동화 및 네트워크 운영을 지원합니다. 다음 인포그래픽에서는 Telecom Visibility의 아키텍처를 보여줍니다.



사용 케이스

통신 가시성 사용 사례

사용 케이스	결과
자율 네트워크 운영	신뢰할 수 있는 네트워크 인벤토리는 AI/ML 기반 자동화를 지원합니다.
서비스 이행	정확한 CMDB/TNI는 주문의 정확성과 신속한 프로비저닝을 보장합니다.
서비스 보증	조정 인벤토리는 문제 해결 시간을 줄이고 가동 시간을 늘립니다.
감사 및 준수	Discovery 관리자 작업 공간 및 CMDB 360은 감사 준비 및 추적 가능성을 지원합니다.

주요 가상 사용자

통신 가시성은 다음에 사용됩니다.

- 네트워크 및 인프라 팀: 공급업체 및 기술 전반에서 실시간 네트워크 리소스 데이터를 검색하고 관리합니다.
- 인벤토리 및 이행 팀: 정확하고 완전한 CMDB/TNI를 유지하여 서비스 이행, 주문 관리 및 프로비저닝을 지원합니다.
- 플랫폼 관리자: 플러그인을 구성하고, 검색 소스를 관리하고, 조정 규칙을 확인하고, 검색 성능을 모니터링합니다.

모니터링 도구

검색 관리자 작업 공간 - 중앙 콘솔은 다음을 수행합니다.

- 디스커버리 작업 모니터링
- 조정 프로세스 확인
- 일정 조정 및 진단 보기

CMDB 360 - 시각화:

- 속성 수준 업데이트 이력
- 소스 속성
- 조정 규칙 실행 중

주요 이점

통신 가시성의 주요 이점

기능	혜택
CLI/SNMP 검색	외부 API에 의존하지 않고 네트워크 범위를 확장합니다.
Nokia Altiplano 통합	여러 컨트롤러 인스턴스에서 GPON 네트워크 검색을 지원합니다.
불일치 조정	인벤토리를 깨끗하고 실행 가능한 상태로 유지하며 실제 네트워크 상태에 맞게 조정합니다.

관련 정보

[통신 가시성 설정](#)

[Discovery Admin Workspace](#)

통신 가시성 vs. ITOM 가시성

통신 네트워크가 발전하고 점점 더 하이브리드되고 복잡해짐에 따라 인프라에 대한 가시성이 그 어느 때보다 중요해졌습니다. 다양한 환경의 ServiceNow AI Platform 고유한 요구 사항을 충족하기 위해 ITOM 가시성과 통신 가시성이라는 두 가지 목적에 맞게 구축된 가시성 솔루션을 제공합니다.

이들은 동일한 기반 위에 구축되었지만 각각 기존 IT 인프라용 ITOM과 통신 네트워크용 TSOM 등 서로 다른 세계에 서비스를 제공하도록 맞춤화되어 있습니다.

ITOM 가시성과 통신 가시성은 모두 ServiceNow Discovery 엔진, IRE(식별 및 조정 엔진) 및 CMDB의 동일한 기본 기능을 기반으로 구축됩니다. 둘 다 다음을 제공합니다.

- 에이전트 기반 및 에이전트 없는 검색.
- 수평 및 패턴 기반 검색.
- 검색된 데이터를 CMDB로 조정
- 의존성 매핑 및 CI 관계 생성
- 검색 관리자 작업 공간 및 CMDB 360과의 통합.

이러한 공유 아키텍처에도 불구하고 지원하는 범위, 사용 사례 및 데이터 모델은 다릅니다.

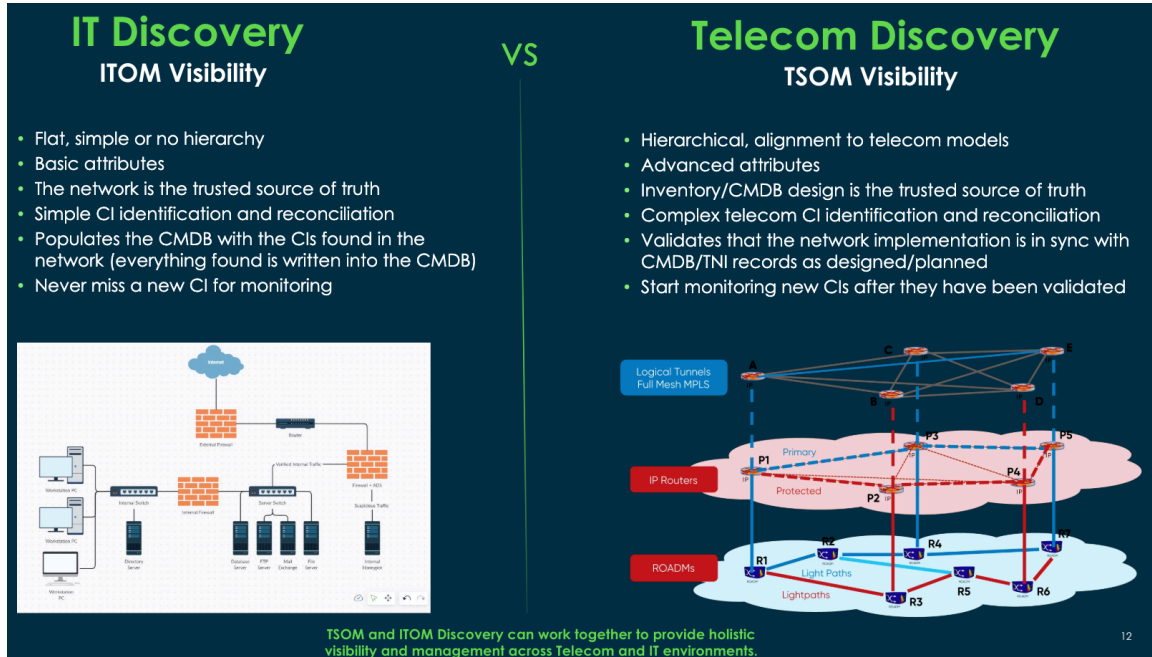
TSOM과 ITOM 가시성의 주요 차이점

통신 가시성 vs. ITOM 가시성

기능/포커스 영역	ITOM 가시성	통신 가시성
대상 환경	기존 IT 인프라(서버, 애플리케이션, 데이터베이스, 클라우드 자원)	통신 인프라(xNF, 네트워크 요소, EMS/NMS 컨트롤러)
디스커버리 방법	IT 패턴을 사용한 가로 검색	통신 Discovery 패턴(SNMP/CLI) 및 통신용 서비스 그래프 커넥터를 사용한 수평 검색
CMDB 모델	ITOM CMDB 클래스 (예: Windows Server, 애플리케이션, 네트워크 어댑터)	통신 인식 CMDB 클래스 및 TNI(Telecom Network Inventory) (예: 인터페이스 카드, 슬롯, LAG, 하위 슬롯, VLAN)
필요한 플러그인	com.snc.discovery 및 com.snc.itom.visibility	sn_tsom_core, sn_tsom_patterns 및 통신 관련 SGC 플러그인(예: sn_sgc_altiplano_connector)
사용 사례 포커스	애플리케이션 종속성 매핑, 서비스 모델링, 클라우드 인프라 검색	통신 네트워크 인벤토리 검색, 조정, 자율 네트워크 운영
불일치 처리	일반 IRE 조정 규칙	통신 관련 불일치 식별 및 조정 (예: 계층 구조 불일치, 속성 수준 충돌)
벤더 데이터 수집	주로 검색 패턴을 통해	SGC(EMS/NMS/CONTROLLERS)를 사용한 노스바운드 API 통합에 중점을 둡니다.
지원되는 네트워크 유형	엔터프라이즈 네트워크, 데이터 센터, 클라우드	다중 벤더, 다중 도메인 통신 네트워크(RAN, 코어, 전송, 액세스)

인포그래픽

다음 인포그래픽은 TSOM과 ITOM 가시성의 차이점을 이해하는 데 도움이 됩니다.



주요 이점

다음과 같은 경우 ITOM 가시성을 사용합니다.

- IT 인프라 구성요소(예: Windows 서버, 클라우드 VM, 데이터베이스, 부하 분산 장치)를 검색하고 있습니다.
- 주요 목표에는 서비스 매핑, 운영 복원성 또는 클라우드 최적화가 포함됩니다.
- ITSM, ITOM 또는 DevOps 사용 케이스에 중점을 두고 있습니다.

다음과 같은 경우 통신 가시성을 사용합니다.

- 기존 IT 시스템으로 관리되지 않는 장치를 포함한 통신 네트워크 인프라를 검색하고 있습니다.
- 카드, 포트, 하위 슬롯, LAG와 같은 통신별 네트워크 계층 구조를 처리하고 있습니다.
- EMS/NMS/컨트롤러를 신뢰할 수 있는 데이터 소스로 사용합니다.
- 통신 인벤토리 모델에 맞게 조정된 불일치 탐지 및 조정이 필요합니다.
- TM Forum 표준을 준수하거나, 자율 네트워크 운영을 지원하거나, 폐쇄 루프 보증을 활성화하고 있습니다.

예제

사용 케이스	ITOM 가시성	통신 가시성
AWS에서 가상 머신 플릿 검색	예	아니요
API를 사용하여 EMS에서 라우터 수집 및 스위치 데이터	아니요	예
애플리케이션 간 의존성 식별	예	아니요
통신 카드 계층 구조에서 불일치 감지 및 조정	아니요	예

ITOM 가시성과 통신 가시성은 모두 정확한 CMDB를 채우고 유지하는 역할을 하지만 서로 다른 도메인에 최적화되어 있습니다. ITOM 가시성은 엔터프라이즈 IT 환경에 맞게 조정된 반면 통신 가시성은 통신 인프라 검색, 불일치 관리 및 인벤토리 조정의 특수한 요구 사항에 맞게 조정됩니다.

올바른 가시성 솔루션을 선택하거나 두 가지를 동시에 사용하면 IT 및 통신 환경 모두에서 신뢰할 수 있는 도메인별 운영 가시성을 유지할 수 있습니다.

통신 디스커버리

ServiceNow AI Platform Telecom Discovery는 ITOM Visibility의 기능을 확장하여 통신 관련 사용 사례를 지원함으로써 통신 네트워크 인프라에 대한 포괄적인 가시성을 확보하는 데 도움이 됩니다.

통신 서비스 제공업체(CSP)를 위해 구축된 이 솔루션은 표준화된 프로토콜과 네트워크 관리 시스템과의 통합을 사용하여 멀티벤더 환경에서 네트워크 요소를 검색하고 매핑할 수 있습니다.

통신 검색 플러그인과 서비스 그래프 커넥터 및 검색 패턴의 기능을 결합하면 구성 관리 데이터베이스(CMDB)에서 통신 자원에 대한 정확한 기록을 자동으로 채우고 유지관리하여 IT 및 네트워크 인프라에 대한 통합 뷰를 제공할 수 있습니다.

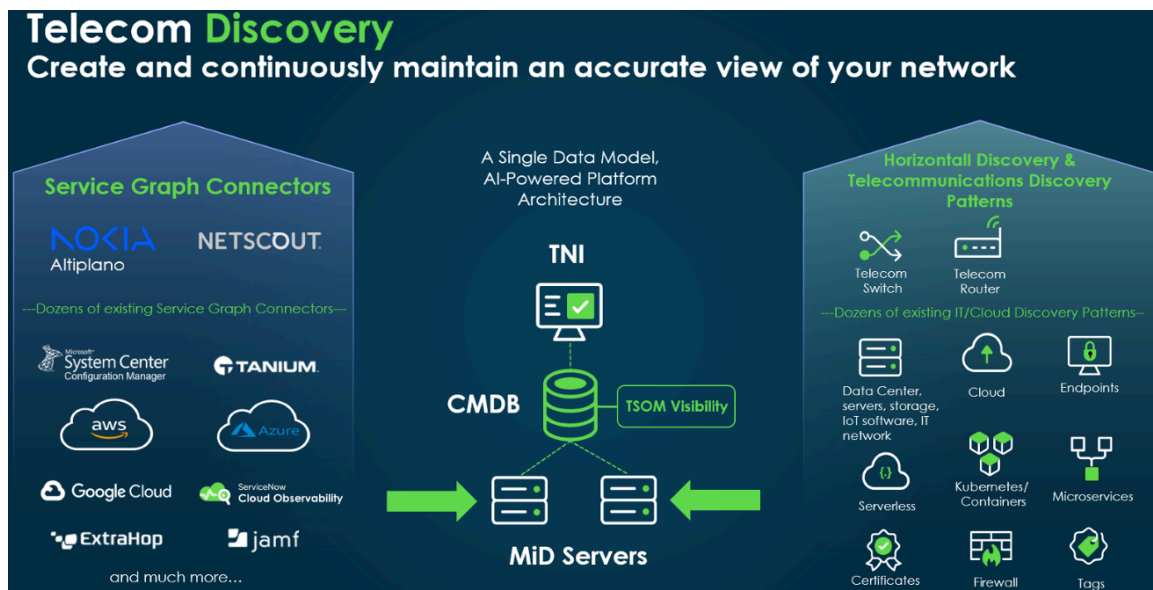
i 주:

Telecom Discovery는 TSOM Visibility 구독의 일부이며 TM Forum ANO(Autonomous Network Operations) 프레임워크와 일치합니다.

통신 검색을 통해 다음을 수행할 수 있습니다.

- 도메인과 벤더 전반에서 물리적 및 논리적 통신 네트워크 자원을 검색합니다.
- EMS(요소 관리 시스템), NMS(네트워크 관리 시스템) 및 SDN 컨트롤러와 통합합니다.
- 실시간 네트워크 데이터를 기반으로 CMDB/TNI 기록을 자동으로 채우고 업데이트합니다.
- SNMP 및 CLI(명령줄 인터페이스)를 사용하여 독립 실행형 xNF를 검색합니다.
- 서비스 그래프 커넥터 및 특수 검색 패턴을 사용하여 CMDB 데이터를 보강합니다.
- 검색된 네트워크 데이터와 인벤토리 기록 간의 불일치를 식별합니다.
- 일관되고 정확한 인프라 가시성을 통해 자동화 사용 사례를 지원합니다.

통신 검색 아키텍처



ITOM 가시성과의 통합

Telecom Discovery는 기존 ITOM 가시성 기능을 보완하도록 설계되었습니다. 다음을 수행할 수 있습니다.

- TSOM 플러그인과 함께 수평 검색 및 ITOM 기능을 활용합니다.
- IT 및 통신 리소스에 대해 일관된 검색 방법을 유지합니다.
- 동일한 CMDB 데이터 모델을 사용하여 교차 도메인 서비스 가시성을 관리합니다.

이 통합을 통해 IT 도메인과 네트워크 도메인 모두에서 통합 자산 관리, 더 빠른 문제 해결, 간소화된 운영을 확인할 수 있습니다.

로우코드/노코드 도구로 커스터마이제이션

코드를 작성하지 않고 검색 논리를 확장할 수 있는 직관적인 설계 도구를 제공합니다. 다음을 수행할 수 있습니다.

- 사용자 지정 서비스 그래프 커넥터를 빌드하거나 수정합니다.
- 벤더별 요구사항에 맞게 통신 Discovery 패턴을 확장합니다.
- 새 장치 유형 및 네트워크 도메인의 온보딩을 가속화합니다.

이러한 접근 방식을 통해 CSP는 민첩성을 유지하고 검색 공간을 확장할 때 가치 실현 기간을 단축할 수 있습니다.

핵심 구성요소

- 통신 검색 패턴(sn_tsom_patterns): 독립 실행형 라우터, 스위치 및 xNF의 SNMP 기반 검색을 위한 패턴을 제공합니다. Cisco 및 주니퍼 고유의 검색 로직을 포함합니다.
- Nokia Altiplano용 서비스 그래프 커넥터(sn_sgc_altiplano_connector): REST API를 통해 Nokia Altiplano Access SDN 컨트롤러에서 데이터를 수집할 수 있습니다.
- Telecom Core(sn_tsom_core): 불일치 식별, 정정 로직 및 공유 통신 검색 기능과 같은 기본 기능을 제공합니다.

통신 검색 작성기 프레임워크

통신 검색 작성기 프레임워크 ETL(추출, 변환, 로드)은 TSOM(통신 서비스 운영 관리) Core 애플리케이션과 함께 제공되는 재사용 가능하고 스키마에 부합하는 구성 요소입니다. 여러 SGC(서비스 그래프 커넥터)에서 통신 인벤토리 데이터를 ServiceNow CMDB(구성 관리 데이터베이스)로 수집하기 위한 일관되고 확장 가능한 방법을 제공합니다.

통신 검색 작성기 프레임워크 ETL은 통신별 CI(구성 항목) 데이터를 처리하도록 설계된 기존 데이터 수집 유틸리티 역할을 합니다. 커넥터 개발 팀은 각 커넥터에 대해 복제하고 사용자 지정할 수 있는 표준화된 변환 로직을 제공하여 ETL을 처음부터 빌드하지 않아도 됩니다.

TSOM Core 플러그인이 활성화되면 일반 ETL이 자동으로 프로비저닝되고 통합 허브 ETL 스튜디오에서 사용할 수 있게 되므로 커넥터별 요구 사항에 맞게 재사용하고 조정할 수 있습니다.

Telecom Discovery Builder 프레임워크를 사용하는 이유와 시기

각각의 새로운 통신 커넥터에 대해 ETL 논리 또는 변환 맵을 처음부터 빌드하는 것은 시간이 많이 걸리고 오류가 발생하기 쉬우며 일관성이 없을 수 있습니다. 통신 검색 작성기 프레임워크는 구현 전반에 걸쳐 일관성을 보장하는 표준화되고 재사용 가능한 기반을 제공하여 프로세스를 간소화합니다.

Telecom Discovery Builder 프레임워크를 사용하여 다음을 수행할 수 있습니다.

- 통신 환경에 새 서비스 그래프 커넥터를 배포하고 ETL을 처음부터 빌드하지 않으려고 합니다.
- 여러 커넥터 및 플랫폼 인스턴스에서 스키마에 부합하는 일관성을 유지합니다.
- TNI(Telecom Network Inventory)를 준수하는 검색 페이로드 및 CI 관계입니다. 자세한 내용은 [Telecommunications Network Inventory](#) 문서를 참조하십시오.
- SGC(서비스 그래프 커넥터) 애플리케이션 범위 내에서 작업하고 핵심 논리를 수정하지 않고 ETL 동작을 사용자 지정하려고 합니다.
- 미리 정의된 매핑, 유효성이 검사된 JSON 스키마 지원 및 UI 기반 구성 인터페이스의 이점을 활용합니다.

핵심 기능

- TSOM 코어로 자동 프로비저닝됨: 자동으로 설치되며 통신 커넥터에서 사용할 준비가 됩니다.
- 일반 스키마 기반 수집: 통신 CI에 대한 통합 데이터 스키마를 지원합니다.
- 중복 및 커스터마이제이션 지원: 통합 허브 ETL 스튜디오를 사용하여 통신 디스커버리 빌더 프레임워크 ETL을 애플리케이션 범위에 복제합니다.
- TNI 지원: TNI 데이터 모델 및 ETL 논리에 맞춰 TNI 엔터티를 생성하고 연결하도록 확장할 수 있습니다.
- 유연한 필드 매핑 인터페이스: UI 기반 환경을 통해 임포트 세트, 데이터 소스, 대상 및 변환 논리를 구성합니다.

주요 이점

Telecom Discovery Builder 프레임워크 ETL을 사용하여 다음을 얻을 수 있습니다.

- 통신 CI(구성 항목)가 구조화되고 CMDB에 로드되는 방식의 일관성 유지.
- 커넥터 간 복제 및 사용자 지정을 통한 재사용성입니다.
- TNI 스키마 및 검색 모델 요구 사항 준수
- 기본 스키마를 손상시키지 않고 데이터 변환을 확장하고 조정할 수 있는 유연성.

관련 정보

[커넥터에서 Telecom Discovery Builder 프레임워크 ETL 구성](#)

[중복 ETL에 대한 TNI 엔터티 지원 확장](#)

Discovery 패턴을 사용한 Direct Discovery

통신 검색 패턴 플러그인(TSOM 패턴이라고도 함)은 통신 검색을 확장하여 ServiceNow AI Platform 기존 네트워크 관리 시스템에 의존하지 않고도 라우터 및 스위치와 같은 독립 실행형 네트워크 요소를 직접 검색할 수 있도록 지원합니다. 이러한 패턴을 통해 CSP(Communication Service Provider)는 NETCONF를 지원하는 SNMP 및 CLI와 같은 프로토콜을 사용하여 멀티벤더 xNF를 식별하고 매핑할 수 있습니다.

통신 디스커버리 패턴은 네트워크 요소에서 직접 통신 네트워크 자원을 검색하고 관리할 수 있는 강력한 패턴 기반 접근 방식을 제공합니다. 이러한 패턴은 기존 EMS/NMS 시스템을 통해 관리되지 않는 독립형 xNF(예: 라우터 및 스위치)를 검색하는 데 특히 유용합니다.

이 기능은 통신 인프라 전반에서 가시성을 향상시키고, TNI 데이터 모델에 따라 벤더 중립적 및 벤더별 장치 데이터가 모두 캡처되고 CMDB 및 TNI(Telecom Network Inventory)에 반영되도록 합니다.

**주:**

통신 Discovery 패턴은 TSOM 가시성 구독의 일부이며 ServiceNow Store에서 고객에게 표시되는 플러그인으로 제공됩니다.

핵심 기능

- 직접 네트워크 요소 검색
 - SNMP, CLI 및 NETCONF를 사용하여 물리적 네트워크 요소와 직접 통신합니다.
 - EMS/NMS 중재에 의존하지 않고 인터페이스 카드, 슬롯, 포트 및 장치와 같은 물리적 네트워크 인벤토리를 검색합니다.
- 자동화된 CI 매핑 및 CMDB 통합
 - 검색된 데이터는 통신 정렬 CI(구성 항목) 클래스에 자동으로 매핑됩니다.
 - 식별 및 조정 엔진(IRE)과 직접 통합하여 CMDB 및 TNI(통신 네트워크 인벤토리)에서 중복 없는 정확한 CI 기록을 보장합니다.
- CMDB 정확성 및 규정 준수
 - 패턴은 CMDB 준수 인증 감사를 트리거하여 불일치 또는 오래된 기록을 탐지합니다.
 - 불일치 식별 및 정정을 지원하여 데이터 무결성을 유지합니다.
- 로우코드 패턴 커스터마이제이션: 로우코드 패턴 디자이너를 사용하여 검색 패턴을 확장하거나 사용자 지정하여 특정 네트워크 아키텍처 및 비즈니스 요구 사항을 충족합니다.

작동 원리

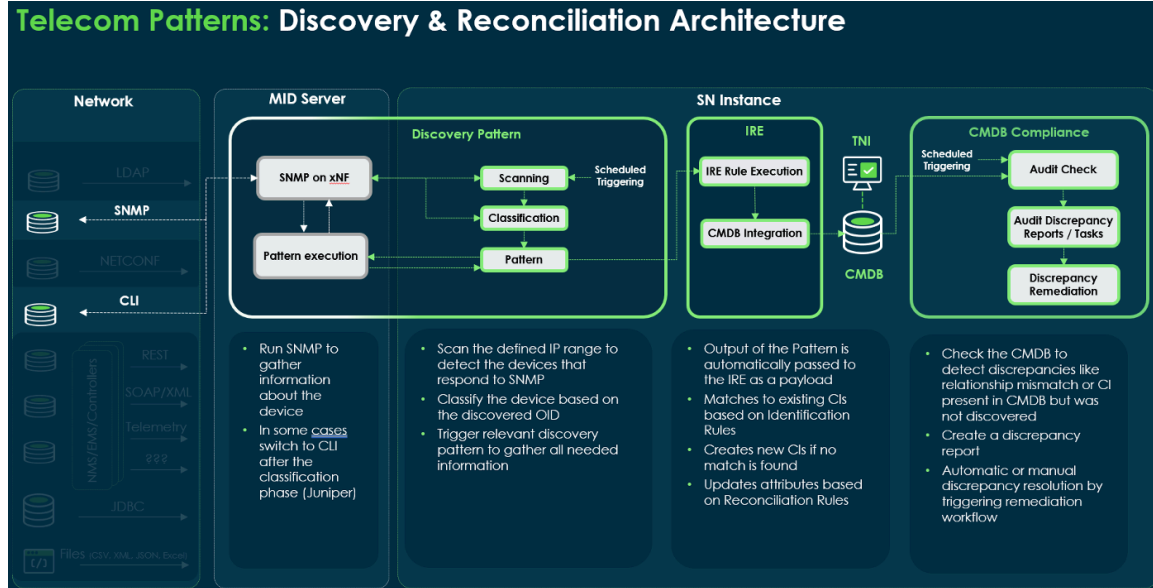
통신 Discovery 패턴은 NDL(Nebula Discovery Language)을 사용하여 수평 Discovery를 통해 실행됩니다. 패턴은 다음과 같은 일련의 단계입니다.

1. 대상 장치에 대한 연결을 설정합니다.
2. SNMP, CLI 또는 둘 다를 사용하여 명령을 실행합니다.
3. CI 속성 및 관계를 추출합니다.
4. 조정 및 CMDB 및 TNI에 삽입하기 위해 결과를 IRE로 보냅니다.

검색 로직은 복잡한 CI 관계(예: 카드 대 카드 시나리오)를 재구조화하여 통신 모델링 베스트 프랙티스에 맞추는 TNI 데이터 모델을 따릅니다. 예를 들어 상위 카드 내에서 하위 카드가 검색되면 통신 검색은 하위 슬롯을 합성하여 하위 카드를 삽입함으로써 잘못된 카드 온 카드 구성을 방지합니다.

수평 **Discovery** 및 통신 **Discovery** 패턴을 사용하는 아키텍처

다음 인포그래픽은 독립 실행형 SNMP 및/또는 CLI xNF를 구현한 예입니다.



가로 검색 애플리케이션

ServiceNow의 수평 검색 애플리케이션은 네트워크, IT 및 클라우드 환경에서 효과적으로 작동하도록 설계된 다목적 및 확장성이 뛰어난 검색 엔진으로, 여러 계층에서 데이터를 수집하여 인프라에 대한 전체적인 뷰를 제공합니다.

자세한 내용은 [Horizontal discovery process flow with patterns](#) 문서를 참조하십시오.

지원되는 검색 패턴

패턴은 CI(구성 항목)의 속성과 아웃 바운드 연결을 감지하도록 설계된 일련의 명령입니다. 통신 검색은 광범위한 네트워크 요소를 포괄하는 미리 구성된 패턴 세트를 제공합니다. ServiceNow 는 다음과 같은 여러 OOB(바로 사용 가능) TSOM 패턴을 제공합니다.

- 통신 라우터 패턴 - 일반 SNMP 기반 라우터 검색입니다.
- 텔레콤 Cisco 7613 라우터 패턴 - SNMP를 사용하는 Cisco 7613 라우터의 경우입니다.
- 통신 Juniper MX SSH 라우터 패턴 - Juniper MX 라우터의 SNMP + CLI 검색입니다.
- 통신 Cisco 스위치 패턴 - SNMP 기반 Cisco 스위치 검색입니다.
- 통신 스위치 패턴 - SNMP 기반 일반 스위치 검색입니다.

TNI 엔터티 생성

인스턴스에 TNI(Telecom Network Inventory) 플러그인이 설치되어 있는 경우:

- 검색된 모든 CI는 자동으로 TNI 엔터티 기록을 생성합니다.
- IRE 페이로드에는 cmdb_ci 및 tni_entity 테이블 모두에 대한 매핑이 포함됩니다.

이를 통해 운영 시스템과 재고 시스템 간의 원활한 조정이 보장되며, 이는 주문 이행, 보증 및 네트워크 계획에 필수적입니다.

MID 서버

MID 서버는 로컬 네트워크 내의 서버에서 Windows 서비스 또는 UNIX 디먼으로 실행되는 Java 애플리케이션입니다. 이를 ServiceNow[#] MID 서버 통해 ServiceNow 인스턴스와 외부 애플리케이션, 데이터 소스 및 서비스 간의 통신과 데이터 전송이 용이해집니다.

자세한 내용은 [MID Server](#) 문서를 참조하십시오.

IRE(식별 및 조정 엔진)

IRE는 여러 소스의 데이터를 식별하고 조정하기 위한 중앙 집중식 프레임워크를 제공합니다. 다양한 데이터 소스를 사용하여 CI 기록을 만들거나 업데이트할 때 CMDB 및 일부 비 CMDB 테이블의 무결성을 확인합니다.

관련 정보

[가로 검색 설치 및 검색 패턴 설정](#)

통신 라우터 패턴

ServiceNow[#] 통신 검색 애플리케이션은 통신 라우터 검색 패턴을 사용하여 네트워크에서 SNMP 기반 라우터를 찾습니다. 이러한 자원을 검색하려면 스토어에서 TSOM 패턴(통신 디스커버리 패턴)을 업데이트해야 합니다 ServiceNow[#].

통신 검색은 통신 검색 패턴을 사용하여 가로 검색을 실행합니다. 이 통신 검색 패턴은 SNMP 요청 세트를 사용하여 네트워크 요소를 찾고, 분류하고, 검색합니다.

통신 라우터 패턴은 TSOM 가시성의 일부인 sn_tsom_patterns(Telecommunications Discovery Patterns) 애플리케이션의 일부입니다.

스토어에서 앱 요청

[ServiceNow Store](#) 웹 사이트를 방문하면 사용 가능한 모든 앱을 확인하고 스토어에 요청을 제출하는 방법에 대한 정보를 참조할 수 있습니다. 출시된 모든 앱의 누적 릴리스 정보는 [ServiceNow Store 버전 기록 릴리스 정보](#)를 참조하십시오.

필수 구성요소

- TSOM 구독.
- 네트워크 라우터 장치에 SNMP 접근 권한이 있는지 확인합니다.
- ServiceNow 인스턴스에서 SNMP 자격 증명을 구성합니다. 자세한 내용은 [SNMP support for Discovery](#) 문서를 참조하십시오.
- TSOM 패턴 설정에 대한 자세한 내용은 문서를 참조하십시오 [가로 검색 설치 및 검색 패턴 설정](#).

영향을 받는 **CMDB CI** 및 **CI** 관계(물리적 계층)

CI	CI 관계
IP 라우터 CI	IP 라우터 장치는 IP 라우터 CI로 표시됩니다. 테이블 이름: cmdb_ci_ip_router IP 라우터 CI에는 슬롯 또는 네트워크 인터페이스가 포함되어 있습니다.

CI	CI 관계
슬롯 CI	슬롯은 슬롯 CI로 표시됩니다. 테이블 이름: cmdb_ci_container_slot 슬롯은 IP 라우터에 포함되어 있습니다. 슬롯 CI에는 인터페이스 카드가 포함되어 있습니다.
하위 슬롯 CI	하위 슬롯은 하위 슬롯 CI로 표시됩니다. 테이블 이름: cmdb_ci_container_subslot 하위 슬롯은 인터페이스 카드 CI에 포함되어 있습니다. 하위 슬롯 CI에는 인터페이스 카드 CI가 포함되어 있습니다.
인터페이스 카드 CI	다양한 유형의 카드가 인터페이스 카드 CI로 표시됩니다. 팬 및 전원 공급 장치도 카드 CI로 표시됩니다. 테이블 이름: cmdb_ci_interface_card 인터페이스 카드는 슬롯 또는 하위 슬롯으로 구성됩니다. 인터페이스 카드에는 네트워크 인터페이스 또는 하위 슬롯이 포함될 수 있습니다.
네트워크 인터페이스 CI	모든 유형의 네트워크 인터페이스는 네트워크 인터페이스 CI로 표시됩니다. 테이블 이름: cmdb_ci_ni_interface 네트워크 인터페이스는 인터페이스 카드, IP 라우터에 포함되어 있습니다.

통신 Cisco 7613 라우터 패턴

ServiceNow[#] Telecom Discovery 애플리케이션은 Telecom Cisco 7613 Router 검색 패턴을 사용하여 네트워크에서 SNMP 기반 Cisco 7613을 찾습니다. 이러한 자원을 검색하려면 스토어에서 TSOM 패턴(통신 디스커버리 패턴)을 업데이트해야 합니다 ServiceNow[#].

통신 검색은 통신 검색 패턴을 사용하여 가로 검색을 실행합니다. 이 통신 검색 패턴은 SNMP 요청 세트를 사용하여 네트워크 요소를 찾고, 분류하고, 검색합니다.

통신 Cisco 7613 라우터 패턴은 TSOM 가시성의 일부인 sn_tsom_patterns(Telecommunications Discovery Patterns) 애플리케이션의 일부입니다.

스토어에서 앱 요청

[ServiceNow Store](#) 웹 사이트를 방문하면 사용 가능한 모든 앱을 확인하고 스토어에 요청을 제출하는 방법에 대한 정보를 참조할 수 있습니다. 출시된 모든 앱의 누적 릴리스 정보는 [ServiceNow Store 버전 기록 릴리스 정보](#)를 참조하십시오.

필수 구성요소

- TSOM 구독.
- 네트워크 라우터 장치에 SNMP 접근 권한이 있는지 확인합니다.
- ServiceNow 인스턴스에서 SNMP 자격 증명을 구성합니다. 자세한 내용은 [SNMP support for Discovery](#) 문서를 참조하십시오.
- TSOM 패턴 설정에 대한 자세한 내용은 문서를 참조하십시오 [가로 검색 설치 및 검색 패턴 설정](#).

영향을 받는 **CMDB CI** 및 **CI** 관계(물리적 계층)

CI	CI 관계
IP 라우터 CI	IP 라우터 장치는 IP 라우터 CI로 표시됩니다. 테이블 이름: cmdb_ci_ip_router IP 라우터 CI에는 슬롯 또는 네트워크 인터페이스가 포함되어 있습니다.
슬롯 CI	슬롯은 슬롯 CI로 표시됩니다. 테이블 이름: cmdb_ci_container_slot 슬롯은 IP 라우터에 포함되어 있습니다. 슬롯 CI에는 인터페이스 카드가 포함되어 있습니다.
하위 슬롯 CI	하위 슬롯은 하위 슬롯 CI로 표시됩니다. 테이블 이름: cmdb_ci_container_subslot 하위 슬롯은 인터페이스 카드 CI에 포함되어 있습니다. 하위 슬롯 CI에는 인터페이스 카드 CI가 포함되어 있습니다.
인터페이스 카드 CI	다양한 유형의 카드가 인터페이스 카드 CI로 표시됩니다. 팬 및 전원 공급 장치도 카드 CI로 표시됩니다. 테이블 이름: cmdb_ci_interface_card 인터페이스 카드는 슬롯 또는 하위 슬롯으로 구성됩니다.

CI	CI 관계
	인터페이스 카드에는 네트워크 인터페이스 또는 하위 슬롯이 포함될 수 있습니다.
네트워크 인터페이스 CI	모든 유형의 네트워크 인터페이스는 네트워크 인터페이스 CI로 표시됩니다. 테이블 이름: cmdb_ci_ni_interface 네트워크 인터페이스는 인터페이스 카드, IP 라우터에 포함되어 있습니다.

통신 **Juniper MX SSH** 라우터 패턴

ServiceNow[#] 통신 검색 애플리케이션은 통신 Juniper MX SSHRouter 검색 패턴을 사용하여 네트워크에서 SNMP 및 CLI 기반 Juniper MX 시리즈 라우터를 찾습니다. 이러한 자원 중 일부를 검색하려면 저장소에서 ServiceNow[#] TSOM 패턴(통신 디스커버리 패턴)을 업데이트해야 합니다.

통신 검색은 통신 검색 패턴을 사용하여 가로 검색을 실행합니다. 이 통신 검색 패턴은 SNMP 요청 세트를 사용하여 SSH를 통해 CLI를 찾고 분류하여 네트워크 요소를 검색합니다.

통신 Juniper MX SSH 라우터 패턴은 TSOM 가시성의 일부인 sn_tsom_patterns(Telecommunications Discovery Patterns) 애플리케이션의 일부입니다.

스토어에서 앱 요청

[ServiceNow Store](#) 웹 사이트를 방문하면 사용 가능한 모든 앱을 확인하고 스토어에 요청을 제출하는 방법에 대한 정보를 참조할 수 있습니다. 출시된 모든 앱의 누적 릴리스 정보는 [ServiceNow Store 버전 기록 릴리스 정보](#)를 참조하십시오.

필수 구성요소

- TSOM 구독.
- 네트워크 라우터 장치에 SNMP 접근 권한이 있는지 확인합니다.
- ServiceNow 인스턴스에서 SNMP 자격 증명을 구성합니다. 자세한 내용은 [SNMP support for Discovery](#)  문서를 참조하십시오.
- TSOM 패턴 설정에 대한 자세한 내용은 문서를 참조하십시오 [가로 검색 설치 및 검색 패턴 설정](#).

영향을 받는 **CMDB CI** 및 **CI** 관계 (물리적 계층)

CI	CI 관계
IP 라우터 CI	IP 라우터 장치는 IP 라우터 CI로 표시됩니다. 테이블 이름: cmdb_ci_ip_router IP 라우터 CI에는 슬롯 또는 네트워크 인터페이스가 포함되어 있습니다.
슬롯 CI	슬롯은 슬롯 CI로 표시됩니다. 테이블 이름: cmdb_ci_container_slot

CI	CI 관계
	슬롯은 IP 라우터에 포함되어 있습니다. 슬롯 CI에는 인터페이스 카드가 포함되어 있습니다.
하위 슬롯 CI	하위 슬롯은 하위 슬롯 CI로 표시됩니다. 테이블 이름: cmdb_ci_container_subslot 하위 슬롯은 인터페이스 카드 CI에 포함되어 있습니다. 하위 슬롯 CI에는 인터페이스 카드 CI가 포함되어 있습니다.
인터페이스 카드 CI	다양한 유형의 카드가 인터페이스 카드 CI로 표시됩니다. 팬 및 전원 공급 장치도 카드 CI로 표시됩니다. 테이블 이름: cmdb_ci_interface_card 인터페이스 카드는 슬롯 또는 하위 슬롯으로 구성됩니다. 인터페이스 카드에는 네트워크 인터페이스 또는 하위 슬롯이 포함될 수 있습니다.
네트워크 인터페이스 CI	모든 유형의 네트워크 인터페이스는 네트워크 인터페이스 CI로 표시됩니다. 테이블 이름: cmdb_ci_ni_interface 네트워크 인터페이스는 인터페이스 카드, IP 라우터에 포함되어 있습니다.

통신 Cisco 스위치 패턴

ServiceNow[#] Telecom Discovery 애플리케이션은 Telecom Cisco 스위치 검색 패턴을 사용하여 네트워크에서 SNMP 기반 Cisco 스위치를 찾습니다. 이러한 작업을 검색하려면 스토어에서 TSOM 패턴(통신 디스커버리 패턴)을 업데이트해야 합니다 ServiceNow[#].

통신 검색은 통신 검색 패턴을 사용하여 가로 검색을 실행합니다. 이 통신 검색 패턴은 SNMP 요청 세트를 사용하여 네트워크 요소를 찾고, 분류하고, 검색합니다.

통신 Cisco 스위치 패턴은 TSOM 가시성의 일부인 sn_tsom_patterns(Telecommunications Discovery Patterns) 애플리케이션의 일부입니다.

스토어에서 앱 요청

ServiceNow Store² 웹 사이트를 방문하면 사용 가능한 모든 앱을 확인하고 스토어에 요청을 제출하는 방법에 대한 정보를 참조할 수 있습니다. 출시된 모든 앱의 누적 릴리스 정보는 ServiceNow Store 버전 기록 릴리스 정보²를 참조하십시오.

필수 구성요소

- TSOM 구독.
- 네트워크 라우터 장치에 **SNMP** 접근 권한이 있는지 확인합니다.
- ServiceNow[#] 인스턴스에서 **SNMP** 자격 증명을 구성합니다. 자세한 내용은 [SNMP support for Discovery](#) 문서를 참조하십시오.
- TSOM 패턴 설정에 대한 자세한 내용은 문서를 참조하십시오 [가로 검색 설치 및 검색 패턴 설정](#).

영향을 받는 **CMDB CI** 및 **CI** 관계(물리적 계층)

CI	CI 관계
IP 스위치 CI	IP 스위치 장치는 IP 스위치 CI로 표시됩니다. 테이블 이름: cmdb_ci_ip_switch IP 스위치 CI에는 슬롯 또는 네트워크 인터페이스가 포함되어 있습니다.
슬롯 CI	슬롯은 슬롯 CI로 표시됩니다. 테이블 이름: cmdb_ci_container_slot 슬롯은 IP 스위치에 포함되어 있습니다. 슬롯 CI에는 인터페이스 카드가 포함되어 있습니다.
하위 슬롯 CI	하위 슬롯은 하위 슬롯 CI로 표시됩니다. 테이블 이름: cmdb_ci_container_subslot 하위 슬롯은 인터페이스 카드 CI에 포함되어 있습니다. 하위 슬롯 CI에는 인터페이스 카드 CI가 포함되어 있습니다.
인터페이스 카드 CI	다양한 유형의 카드가 인터페이스 카드 CI로 표시됩니다. 팬 및 전원 공급 장치도 카드 CI로 표시됩니다. 테이블 이름: cmdb_ci_interface_card 인터페이스 카드는 슬롯 또는 하위 슬롯으로 구성됩니다. 인터페이스 카드에는 네트워크 인터페이스 또는 하위 슬롯이 포함될 수 있습니다.

CI	CI 관계
네트워크 인터페이스 CI	모든 유형의 네트워크 인터페이스는 네트워크 인터페이스 CI로 표시됩니다. 테이블 이름: cmdb_ci_ni_interface 네트워크 인터페이스는 인터페이스 카드, IP 스위치에 포함되어 있습니다.

통신 스위치 패턴

ServiceNow[#] 통신 검색 애플리케이션은 통신 스위치 검색 패턴을 사용하여 네트워크에서 SNMP 기반 통신 스위치를 찾습니다. 이러한 자원을 검색하려면 스토어에서 TSOM 패턴(통신 디스커버리 패턴)을 업데이트해야 합니다 ServiceNow[#].

통신 검색은 통신 검색 패턴을 사용하여 가로 검색을 실행합니다. 이 통신 검색 패턴은 SNMP 요청 세트를 사용하여 네트워크 요소를 찾고, 분류하고, 검색합니다.

통신 스위치 패턴은 TSOM 가시성의 일부인 sn_tsom_patterns(Telecommunications Discovery Patterns) 애플리케이션의 일부입니다.

스토어에서 앱 요청

ServiceNow Store² 웹 사이트를 방문하면 사용 가능한 모든 앱을 확인하고 스토어에 요청을 제출하는 방법에 대한 정보를 참조할 수 있습니다. 출시된 모든 앱의 누적 릴리스 정보는 ServiceNow Store 버전 기록 릴리스 정보²를 참조하십시오.

필수 구성요소

- TSOM 구독.
- 네트워크 라우터 장치에 SNMP 접근 권한이 있는지 확인합니다.
- ServiceNow[#] 인스턴스에서 SNMP 자격 증명을 구성합니다. 자세한 내용은 SNMP support for Discovery² 문서를 참조하십시오.
- TSOM 패턴 설정에 대한 자세한 내용은 문서를 참조하십시오 가로 검색 설치 및 검색 패턴 설정.

영향을 받는 **CMDB CI** 및 **CI** 관계(물리적 계층)

CI	CI 관계
IP 스위치 CI	IP 스위치 장치는 IP 스위치 CI로 표시됩니다. 테이블 이름: cmdb_ci_ip_switch IP 스위치 CI에는 슬롯 또는 네트워크 인터페이스가 포함되어 있습니다.
슬롯 CI	슬롯은 슬롯 CI로 표시됩니다. 테이블 이름: cmdb_ci_container_slot 슬롯은 IP 스위치에 포함되어 있습니다.

CI	CI 관계
	슬롯 CI에는 인터페이스 카드가 포함되어 있습니다.
하위 슬롯 CI	하위 슬롯은 하위 슬롯 CI로 표시됩니다. 테이블 이름: cmdb_ci_container_subslot 하위 슬롯은 인터페이스 카드 CI에 포함되어 있습니다. 하위 슬롯 CI에는 인터페이스 카드 CI가 포함되어 있습니다.
인터페이스 카드 CI	다양한 유형의 카드가 인터페이스 카드 CI로 표시됩니다. 팬 및 전원 공급 장치도 카드 CI로 표시됩니다. 테이블 이름: cmdb_ci_interface_card 인터페이스 카드는 슬롯 또는 하위 슬롯으로 구성됩니다. 인터페이스 카드에는 네트워크 인터페이스 또는 하위 슬롯이 포함될 수 있습니다.
네트워크 인터페이스 CI	모든 유형의 네트워크 인터페이스는 네트워크 인터페이스 CI로 표시됩니다. 테이블 이름: cmdb_ci_ni_interface 네트워크 인터페이스는 인터페이스 카드, IP 스위치에 포함되어 있습니다.

서비스 그래프 커넥터를 사용한 간접 검색

SGC(Service Graph Connectors)를 사용하는 ServiceNow Telecom Discovery를 사용하면 EMS, NMS 및 SDN 컨트롤러와 같은 외부 관리 시스템의 네트워크 인프라 데이터를 CMDB(Configuration Management Database)에 원활하게 통합할 수 있습니다.

이 접근 방식은 CSP(통신 서비스 공급자)가 멀티벤더 통신 네트워크 리소스, 서비스 및 구성에 대한 최신 및 정확한 보기를 유지하는 데 도움이 됩니다.

사전 정의된 커넥터와 강력한 데이터 변환 도구를 활용하여 도메인 간에 인벤토리를 통합하고 CMDB 및 TNI(Telecom Network Inventory)에 통신 데이터 모델에 맞는 실시간 네트워크 인사이트가 반영되도록 할 수 있습니다.



주:

통신 서비스 그래프 커넥터는 TSOM 가시성 구독의 일부이며 표준 서비스 그래프 커넥터 프레임워크의 기능을 확장합니다.

서비스 그래프 커넥터 개요

통신 SGC를 통해 다음을 수행할 수 있습니다.

- 노스바운드 REST API를 사용하여 EMS/NMS/컨트롤러에서 데이터를 수집합니다.
- CMDB 및 TNI 기록을 풍부한 통신 정렬 데이터로 자동으로 채우고 업데이트합니다.
- IRE(식별 및 조정 엔진)를 사용하여 수신 데이터를 기존 CI 기록과 조정합니다.
- TNI 플러그인이 설치되면 TNI 엔터티 기록을 자동으로 생성합니다.
- 통신 불일치 식별 및 조정의 일환으로 불일치 탐지 및 정정을 지원합니다.
- 로우코드 도구를 사용하여 통합을 엔드 투 엔드로 구성, 테스트 및 관리합니다.

아키텍처 개요

통신 서비스 그래프 커넥터는 모듈식의 확장 가능한 아키텍처에 의존합니다.

서비스 그래프 커넥터의 주요 구성요소

구성요소	역할
서비스 그래프 커넥터	EMS/NMS 시스템(예: Nokia Altiplano 또는 Nokia NSP)에서 데이터를 추출하고 스테이징하는 통합 논리를 정의합니다.
MID 서버	ServiceNow 인스턴스와 외부 네트워크 시스템 간의 보안 브리지 역할을 합니다. 자세한 내용은 IntegrationHub ETL 문서를 참조하십시오.
통합 허브 ETL(3.2)	ETL 변환 맵을 작성, 테스트 및 관리하기 위한 안내 UI를 제공합니다. 자세한 내용은 MID Server 문서를 참조하십시오.
강력한 변환 엔진(RTE)	정의된 ETL 논리를 사용하여 준비된 소스 데이터를 CMDB 호환 기록으로 변환합니다. 자세한 내용은 Create a robust import set transformer 문서를 참조하십시오.
IRE(식별 및 조정 엔진)	CI를 식별하고 조정하여 데이터 일관성을 보장하고 중복을 방지합니다. 자세한 내용은 CMDB Identification and Reconciliation (IRE) 문서를 참조하십시오.
CMDB/TNI	가시성 및 다운스트림 프로세스를 위해 구조화되고 정확한 통신 인프라 데이터를 저장합니다.

지원되는 서비스 그래프 커넥터

- Nokia Altiplano SGC(sn_sgc_altiplano_connector): REST API를 통해 Nokia Altiplano 액세스 네트워크 SDN 컨트롤러와 통합합니다.
- [store-future: BEGIN review]Nokia NSP SGC: REST API를 통해 전송 네트워크 데이터를 수집하기 위해 NSP(Nokia Network Services Platform)와 통합합니다.[End]

**주:**

커넥터는 IT 및 클라우드 서비스 그래프 커넥터(예: 서버, 모니터링 도구, IoT 등)와 공존할 수 있습니다.

주요 이점

- 빠른 가치 실현 기간 – 최소한의 구성이 필요한 사전 정의된 지원 커넥터를 사용합니다.
- 멀티벤더 지원 – 액세스, 코어, 전송 네트워크 전반에서 다양한 관리 플랫폼과 통합합니다.
- 모델 정렬 가시성 – 통신사별 계층 구조와 관계가 CMDB에서 정확하게 모델링되도록 합니다.
- 불일치 탐지 준비 – 네트워크 데이터를 CMDB 준수를 위해 Telecom Discrepancy Identification & Reconciliation에 직접 공급합니다.
- 확장 가능한 통합 – 성능 및 확장성을 위해 구축된 ServiceNow의 입증된 통합 프레임워크를 활용합니다.

TNI(Telecom Network Inventory) 데이터 모델

통신 SGC에는 TNI(통신 네트워크 인벤토리) 데이터 모델과의 호환성을 보장하는 논리가 포함되어 있습니다.

- TNI 플러그인이 설치되면 검색된 각 네트워크 요소에 TNI 엔터티 기록이 자동으로 포함됩니다.
- `tni_entity`는 시스템 생성 페이로드 매핑(예: `inventory_category`)을 사용하여 해당 `cmdb_ci` 기록과 함께 생성됩니다.
- 이를 통해 운영 및 계획 시스템 전반에 걸쳐 일관성이 보장됩니다.

TNI가 설치되면 다음과 같은 페이로드가 각 항목에 대한 IRE 페이로드에 추가됩니다(`className`을 기반으로 `inventory_category` 채워짐).

```
related = [{
  "className": "tni_entity",
  "values": {
    "inventory_category": ""
  }
}];
```

결과적으로 검색된 CI는 `cmdb_ci` 테이블과 `tni_entity` 테이블 모두에 있습니다.

관련 정보

[Nokia Altiplano를 통한 통신 검색](#)

Nokia Altiplano를 통한 통신 검색

Nokia Altiplano용 서비스 그래프 커넥터는 Nokia Altiplano 액세스 네트워크 SDN 컨트롤러에서 ServiceNow CMDB로 실시간 네트워크 인벤토리를 가져오는 통신 인식 통합을 제공합니다.

서비스 프로바이더 및 통신 기업을 위해 설계된 이 커넥터를 사용하면 물리적 및 논리적 네트워크 인프라의 완전한 가시성, 제어 및 동기화가 가능합니다. 이러한 통합은 REST API와 MID Server를 사용하여 통신 모델에 부합하는 네트워크 뷰를 제공함으로써 보다 정확한 서비스 모델링, 인벤토리 관리 및 운영 효율성을 지원합니다.

주요 이점

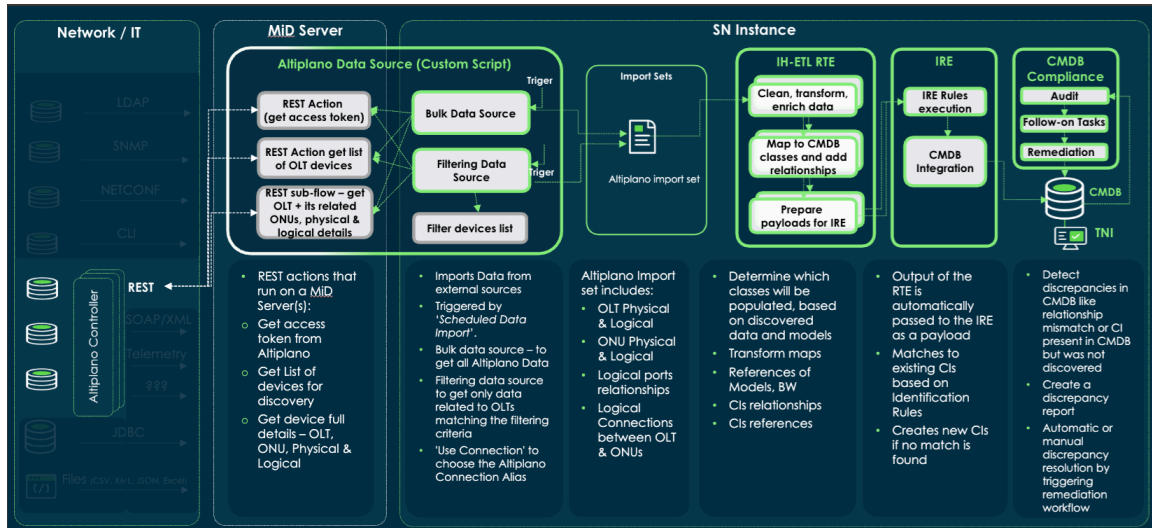
- 정확한 인벤토리 동기화: 안전한 REST API 기반 수집을 사용하여 Nokia Altiplano의 최신 실제 및 논리적 인벤토리로 CMDB를 자동으로 채웁니다. 여기에는 OLT, ONU/ONT, 인터페이스 카드, 포트, 슬롯 및 논리적 연결이 포함됩니다.
- 통신 인식 CI 모델링: 목적에 맞게 구축된 CI 클래스 및 관계를 사용하여 통신 연계 형식으로 네트워크 인프라를 모델링합니다. 이 커넥터는 장치와 장치의 종속성을 정확하게 표현하여 실제 네트워크 토폴로지를 미러링합니다.
- 단순화된 안내 설정: 연결 구성, 데이터 소스 관리 및 임포트 일정을 쉽게 안내하는 내장 안내 설정으로 가치 실현 기간을 단축합니다.
- 유연한 검색 옵션: 전체 대량 로드, 대상 필터링된 검색 또는 단계적 OLT 전용 가져오기 등 필요에 따라 검색을 실행할 방법과 시기를 선택합니다. 장치 IP 또는 이름으로 사용자 지정 필터를 적용합니다.
- 멀티 인스턴스 지원: 여러 Altiplano 인스턴스를 독립적으로 온보딩하여 손쉽게 확장할 수 있습니다. 완전한 운영 유연성을 위해 연결 별칭을 구성하고 인스턴스당 임포트 일정을 정의합니다.

i 주:

지원되는 Nokia Altiplano 컨트롤러 최소 버전은 24.6입니다. 서비스 그래프 커넥터 기술에 대한 일반적인 개요는 다음 문서를 참조하십시오 [Getting started with Service Graph Connectors](#).

Nokia Altiplano SGC 아키텍처

다음 인포그래픽은 Nokia Altiplano 서비스 그래프 커넥터의 아키텍처를 이해하는 데 도움이 됩니다.



사용 사례

다음은 Nokia Altiplano 서비스 그래프 커넥터를 사용하는 방법의 예입니다.

- 물리적 및 논리적 네트워크 요소 모두에 대해 네트워크 데이터를 ServiceNow의 CMDB로 자동 수집하고 구조화합니다. 이렇게 하면 실제 네트워크 환경을 반영하는 통신 모델에 맞는 CMDB 뷰가 생성됩니다.
 - 물리적 구성 요소: OLT, ONU/ONT, 슬롯, 하위 슬롯, 카드 및 포트
 - 논리적 구성 요소: 논리적 포트, VLAN, LAG(Link Aggregation Groups) 및 장치 간 논리적 경로
- 다음과 같은 예약된 자동 조정을 통해 네트워크 데이터의 무결성을 유지합니다.

- 실시간 변경 감지 및 대응
- 데이터 드리프트 또는 부실 기록 방지
- 서비스 보증, 주문 이행 및 네트워크 계획과 같은 운영 프로세스를 지원합니다.
- ServiceNow 내에서 논리적 연결을 캡처하고 관리하여 물리적 인프라 그 이상을 발견할 수 있습니다.
 - 논리적 포트, LAG, PON 및 VLAN 연결과 같은 논리적 경로 식별 및 모델링
 - 상위-하위 및 구성원 관계를 사용하여 논리적 CI를 물리적 구성요소에 연결
 - 엔드 투 엔드 논리적 토폴로지를 시각화하여 진단, 영향 분석 및 서비스 모델링을 개선합니다
- ServiceNow의 기본 제공 ETL(추출, 변환, 로드) 프레임워크를 활용하여 통합을 단순화하고 가속화합니다.
 - 사전 정의된 변환 맵과 CI 클래스 정의로 개발 오버헤드 감소
 - 기존 CMDB 구조를 재사용하여 사용자 지정을 최소화합니다.
 - 새로운 Altiplano 인스턴스를 신속하게 온보딩하고 네트워크 인프라 전반으로 확장

핵심 기능 및 구성요소

핵심 기능 및 구성요소

역량	설명	지원 구성요소
통신 인식 CMDB 모델링	통신별 CI 클래스 및 관계를 사용하여 Altiplano의 물리적 및 논리적 인벤토리(OLT, ONU/ONT, 포트, 슬롯, 카드, 인터페이스)를 CMDB에 매핑합니다.	RTE, IRE, CMDB 테이블 (cmdb_ci_optical_line_terminal, cmdb_ci_optical_network_terminal 등)
자동화된 데이터 수집	안전한 예약된 임포트가 있는 REST API를 통해 Nokia Altiplano에서 장치 인벤토리를 검색합니다.	MID 서버, 데이터 소스(SGC-Nokia Altiplano 대량/필터링된 디스커버리)
사용자 지정 디스커버리 통제	OLT 전용 선택 또는 ONU 데이터 포함, IP 또는 이름으로 필터 적용, 인스턴스당 작업 예약.	일정, 시스템 속성 (sn_sgc_altiplano.enable_onu_discovery 등) 임포트
멀티 인스턴스 지원	여러 Altiplano 컨트롤러에 대해 독립적으로 검색을 구성하고 관리합니다.	연결 별칭, 자격 증명 별칭
병렬 데이터 처리	큰 데이터 세트에 대해 동시 데이터 소스 작업을 실행하여 성능을 향상시킵니다.	시스템 속성: sn_sgc_altiplano.parallel_number_of_data_source 병렬 로드 활성화
안내 구성	연결, 자격 증명 및 작업을 생성하기 위한 단계별 인터페이스로 설정을 단순화합니다.	안내 설정 UI(탐색: 모든 > 서비스 그래프 커넥터 > Nokia Altiplano > 설정)
모델 기반 CI 분류	검색된 장치를 모델과 매칭하고 올바른 CI 클래스(OLT, ONU,	모델 테이블, 시스템 속성: sn_sgc_altiplano.onu_ci_class

핵심 기능 및 구성요소

역량	설명	지원 구성요소
	ONT)를 할당하거나 네트워크 기어로 폴백합니다.	
관계	통신별 CI 관계(예: 구성원, 포함, 논리 경로)를 설정합니다.	IRE, 논리적 연결 CI(cmdb_ci_ni_logical_path)
대시보드 및 모니터링	각 실행의 상태, 결과 및 오류를 봅니다. 커넥터 또는 시간 범위별로 필터링합니다.	CMDB 대시보드용 통합 커먼즈
확장 가능하고 재사용 가능한 아키텍처	손쉬운 확장과 커스터마이제이션을 위해 데이터 소스, 변환 및 CI 조정을 분리했습니다.	임포트 세트 (sn_sgc_altiplano_tsom_inventory), 변환 맵, 시스템 속성

CMDB 통합 대시보드

CMDB용 통합 커먼즈 스토어 앱은 설치된 모든 서비스 그래프 커넥터의 상태, 처리 결과 및 처리 오류에 대한 중앙 뷰가 포함된 대시보드를 제공합니다. 모든 통합 실행에 대한 메트릭을 볼 수 있습니다. 뷰를 특정 통합, 특정 기간 또는 특정 통합 실행으로 필터링할 수 있습니다. CMDB 통합 대시보드의 모니터링 통합에 대한 자세한 내용은 문서를 참조하십시오 [Integration Commons for CMDB](#).

관련 정보

[Nokia Altiplano 서비스 그래프 커넥터 구성](#)

[Nokia Altiplano와 함께 설치되는 시스템 구성요소](#)

[store-future: BEGIN review]

Nokia NSP를 통한 통신 검색

Nokia NSP용 서비스 그래프 커넥터는 Nokia NSP 액세스 네트워크 SDN 컨트롤러에서 ServiceNow CMDB로 실시간 네트워크 인벤토리를 가져오는 통신 인식 통합을 제공합니다.

서비스 프로바이더 및 통신 기업을 위해 설계된 이 커넥터를 사용하면 물리적 및 논리적 네트워크 인프라의 완전한 가시성, 제어 및 동기화가 가능합니다. 이러한 통합은 REST API와 MID Server를 사용하여 통신 모델에 부합하는 네트워크 뷰를 제공함으로써 보다 정확한 서비스 모델링, 인벤토리 관리 및 운영 효율성을 지원합니다.

주요 이점

- 정확한 인벤토리 동기화: 안전한 REST API 기반 수집을 사용하여 CMDB에 Nokia NSP의 최신 실제 및 논리적 인벤토리를 자동으로 채웁니다. 여기에는 IP 라우터, 인터페이스 카드, 포트, 슬롯 및 논리적 인터페이스(ENET, LAG), 논리적 연결 및 네트워크 토폴로지가 포함됩니다.
- 통신 인식 CI 모델링: 목적에 맞게 구축된 CI 클래스 및 관계를 사용하여 통신 연계 형식으로 네트워크 인프라를 모델링합니다. 이 커넥터는 장치와 장치의 종속성을 정확하게 표현하여 실제 네트워크 토폴로지를 미러링합니다.
- 단순화된 안내 설정: 연결 구성, 데이터 소스 관리 및 임포트 일정을 쉽게 안내하는 내장 안내 설정으로 가치 실현 기간을 단축합니다.

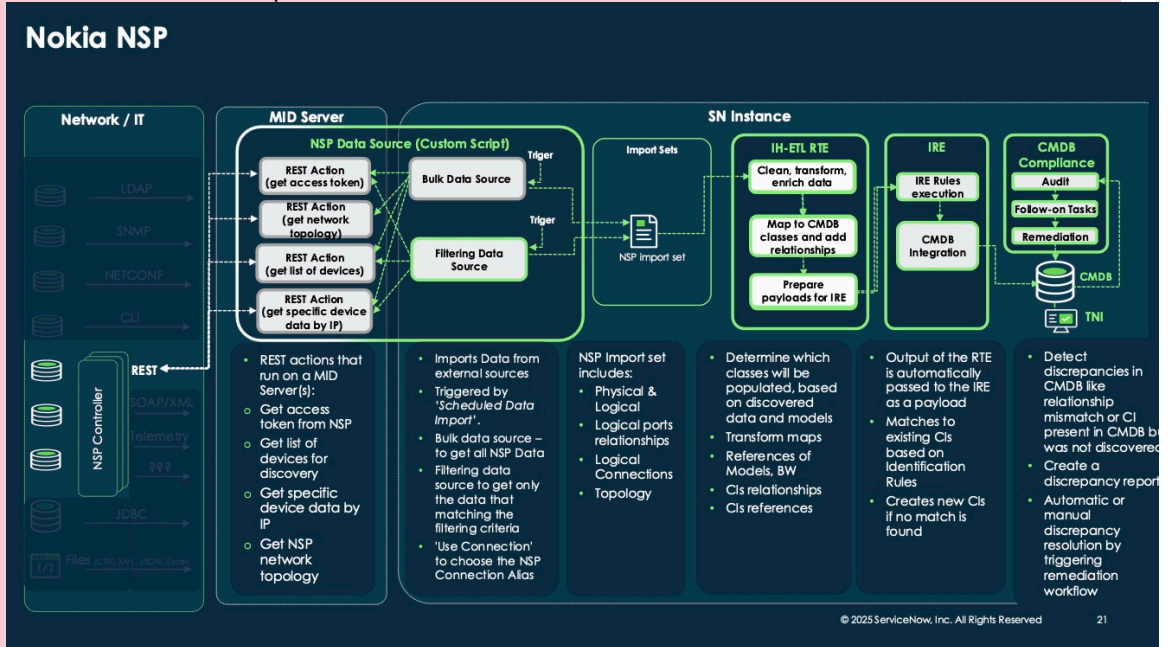
- 유연한 검색 옵션: 전체 대량 로드, 필터링된 대상 검색 등 필요에 따라 검색을 실행하는 방법과 시기를 선택합니다. 장치 IP 또는 이름으로 사용자 지정 필터를 적용합니다.
- 멀티 인스턴스 지원: 여러 NSP 인스턴스를 독립적으로 온보딩하여 손쉽게 확장할 수 있습니다. 완전한 운영 유연성을 위해 연결 별칭을 구성하고 인스턴스당 임포트 일정을 정의합니다.

i 주:

서비스 그래프 커넥터 기술에 대한 일반적인 개요는 다음 문서를 참조하십시오 [Getting started with Service Graph Connectors](#) .

Nokia Altiplano SGC 아키텍처

다음 인포그래픽은 Nokia Altiplano 서비스 그래프 커넥터의 아키텍처를 이해하는 데 도움이



됩니다.

사용 사례

다음은 Nokia NSP 서비스 그래프 커넥터를 사용하는 방법에 대한 예입니다.

- 물리적 및 논리적 네트워크 요소 모두에 대해 네트워크 데이터를 ServiceNow의 CMDB로 자동 수집하고 구조화합니다. 이렇게 하면 실제 네트워크 환경을 반영하는 통신 모델에 맞는 CMDB 뷰가 생성됩니다.
 - 물리적 구성요소: IP 라우터, 슬롯, 하위 슬롯, 카드 및 포트
 - 논리적 구성 요소: 논리적 포트, LAG(Link Aggregation Groups) 및 장치 간 논리적 경로, 네트워크 토폴로지
- 다음과 같은 예약된 자동 조정을 통해 네트워크 데이터의 무결성을 유지합니다.
 - 실시간 변경 감지 및 대응
 - 데이터 드리프트 또는 부실 기록 방지
 - 서비스 보증, 주문 이행 및 네트워크 계획과 같은 운영 프로세스를 지원합니다.
- ServiceNow 내에서 논리적 연결을 캡처하고 관리하여 물리적 인프라 그 이상을 발견할 수 있습니다.

- 논리적 포트, LAG, ENET 및 LAG 연결과 같은 논리적 경로를 식별하고 모델링합니다
- 상위-하위 및 구성원 관계를 사용하여 논리적 CI를 물리적 구성요소에 연결
- 엔드 투 엔드 논리적 토폴로지를 시각화하여 진단, 영향 분석 및 서비스 모델링을 개선합니다
- ServiceNow의 기본 제공 ETL(추출, 변환, 로드) 프레임워크를 활용하여 통합을 단순화하고 가속화합니다.
 - 사전 정의된 변환 맵과 CI 클래스 정의로 개발 오버헤드 감소
 - 기존 CMDB 구조를 재사용하여 사용자 지정을 최소화합니다.
 - 새로운 NSP 인스턴스를 신속하게 온보딩하고 네트워크 인프라 전반으로 확장

핵심 기능 및 구성요소

핵심 기능 및 구성요소

역량	설명	지원 구성요소
통신 인식 CMDB 모델링	NSP의 물리적 및 논리적 인벤토리(IP 라우터, 포트, 슬롯, 카드, 인터페이스)를 통신별 CI 클래스 및 관계를 사용하여 CMDB에 매핑합니다.	RTE, IRE, CMDB 테이블 (cmdb_ci_ip_router 등)
자동화된 데이터 수집	안전한 예약된 임포트를 통해 REST API를 통해 Nokia NSP에서 장치 인벤토리를 검색합니다.	MID 서버, 데이터 소스(SGC-Nokia NSP 대량/필터링된 검색)
사용자 지정 디스커버리 통제	IP 또는 이름으로 필터를 적용하고 인스턴스별로 작업을 예약합니다.	일정 임포트, 시스템 속성
멀티 인스턴스 지원	여러 NSP 컨트롤러에 대해 독립적으로 검색을 구성하고 관리합니다.	연결 별칭, 자격 증명 별칭
병렬 데이터 처리	큰 데이터 세트에 대해 동시 데이터 소스 작업을 실행하여 성능을 향상시킵니다.	시스템 속성: <code>sn_sgc_nsp.parallelNumberOfDataSourceJobs</code> , 병렬 로드 사용
안내 구성	연결, 자격 증명 및 작업을 생성하기 위한 단계별 인터페이스로 설정을 단순화합니다.	안내 설정 UI(탐색: 모든 > 서비스 그래프 커넥터 > Nokia NSP > 설정)
모델 기반 CI 분류	검색된 장치를 모델과 매칭하고 올바른 CI 클래스(IP 라우터)를 할당하거나 네트워크 기어로 대체합니다.	모델 테이블
관계	통신별 CI 관계(예: 구성원, 포함, 논리 경로)를 설정합니다.	IRE, 논리적 연결 CI(cmdb_ci_ni_logical_path)
대시보드 및 모니터링	각 실행의 상태, 결과 및 오류를 봅니다. 커넥터 또는 시간 범위별로 필터링합니다.	CMDB 대시보드용 통합 커먼즈

핵심 기능 및 구성요소

역량	설명	지원 구성요소
확장 가능하고 재사용 가능한 아키텍처	손쉬운 확장과 커스터마이제이션을 위해 데이터 소스, 변환 및 CI 조정을 분리했습니다.	임포트 세트 (sn_sgc_nsp_tsom_inventory), 변환 맵, 시스템 속성

CMDB 통합 대시보드

CMDB용 통합 커먼즈 스토어 앱은 설치된 모든 서비스 그래프 커넥터의 상태, 처리 결과 및 처리 오류에 대한 중앙 뷰가 포함된 대시보드를 제공합니다. 모든 통합 실행에 대한 메트릭을 볼 수 있습니다. 뷰를 특정 통합, 특정 기간 또는 특정 통합 실행으로 필터링할 수 있습니다. CMDB 통합 대시보드의 모니터링 통합에 대한 자세한 내용은 문서를 참조하십시오 [Integration Commons for CMDB](#).

관련 정보

[Nokia NSP 서비스 그래프 커넥터 구성](#)

[End]

통신 불일치 식별 및 조정

통신 불일치 식별 및 조정 기능을 사용하여 CMDB 또는 TNI(통신 네트워크 인벤토리)를 라이브 통신 네트워크와 동기화 상태로 유지합니다. 이 솔루션은 검색된 데이터를 인벤토리 레코드와 지속적으로 감사하고 비교하여 불일치가 서비스 품질, 보증 또는 이행 프로세스에 영향을 미치기 전에 불일치를 탐지, 분류 및 자동으로 정정하는 데 도움이 됩니다.

통신 불일치 식별 및 조정은 TSOM 가시성 플러그인에 포함된 통신 관련 기능입니다. 이는 실시간 네트워크 데이터(검색 또는 외부 시스템)와 CMDB 또는 TNI에 저장된 인벤토리 기록 간의 불일치를 식별하여 네트워크 인벤토리의 무결성을 확인하는 데 도움이 됩니다.

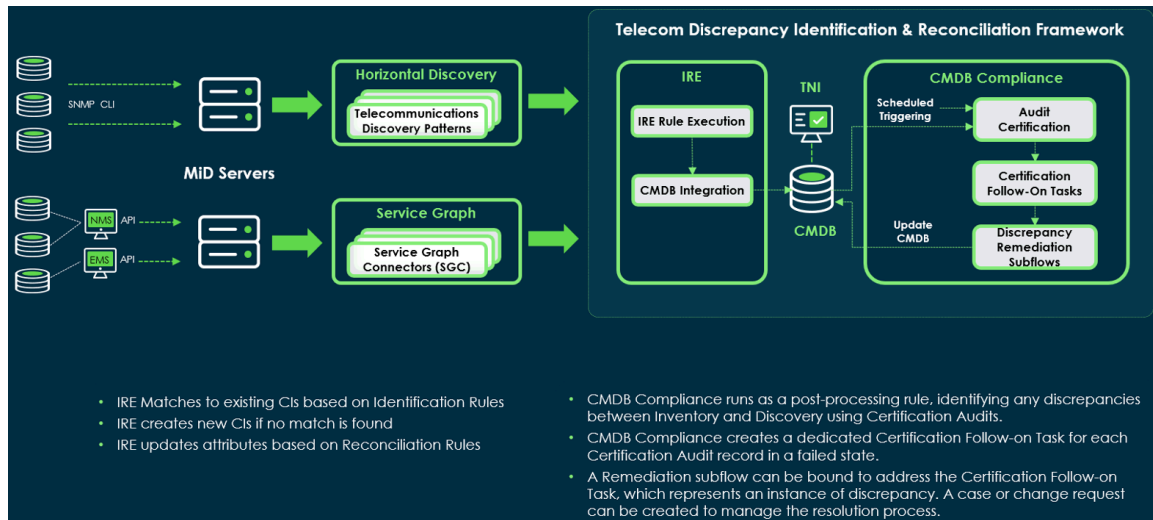
이 솔루션은 다음을 사용합니다.

- CMDB 준수 인증 감사는 번칙을 탐지합니다.
- 문제를 기록하고 추적하는 후속 작업입니다.
- 불일치를 해결하기 위한 자동화된 정정 하위 플로우입니다.

i 주:

재고를 정확하게 유지하면 자동화가 가능하고 서비스 오류가 줄어들며 규정 준수를 지원할 수 있습니다. 또한 TM Forum의 ANO(Autonomous Network Operations) 프레임워크의 기본 구성 요소이기도 합니다.

통신 불일치 식별 및 조정 아키텍처



주요 기능

Telecom Discrepancy Identification and Reconciliation의 주요 기능

기능	설명
지속적인 감사 확인	인증 감사를 사용하여 검색된 네트워크 데이터를 인벤토리 기록과 비교합니다.
불일치 탐지	계층 구조 또는 속성 값 측면에서 누락, 잘못 구성 또는 정렬된 CI를 식별합니다.
후속 작업 생성	각 불일치에 대한 작업을 자동으로 생성하여 추적 및 정정을 활성화합니다.
자동 정정	Flow Designer 하위 플로우를 사용하여 오래된 CI, 잘못된 관계, 누락된 검색 데이터와 같은 문제를 해결합니다.

통신 불일치 식별 및 조정의 작동 방식

Discovery 프로세스가 완료되면 CMDB 준수 프레임워크는 예약된 감사 또는 요청 시 준수 감사를 사용하여 검색된 데이터와 인벤토리 기록 간의 불일치를 식별합니다.

- 규정 준수 감사: 검색된 네트워크 데이터를 CMDB 인벤토리 기록과 비교하여 불일치를 탐지합니다.
- 후속 작업: 실패한 각 감사에 대해 자동으로 생성됩니다. 이러한 작업은 식별된 불일치를 문서화하고 범주화합니다.
- 정정 하위 플로우: 후속 작업에서 시작되어 CI(구성 항목)를 업데이트, 폐기 또는 재정렬하여 불일치를 해결합니다.

i 주:

자세한 내용은 [불일치 식별 - 불일치 유형](#) 문서를 참조하십시오.

CMDB 준수 및 통신 불일치 식별 및 조정

CMDB 준수는 관리자가 CMDB 데이터의 정확성을 인증하고 준수 감사 중에 감지된 불일치를 식별할 수 있는 도구 세트입니다. 또한 실패한 감사 기록에 대한 후속 작업을 자동으로 생성하고 할당할 수 있으며, 이는 불일치를 수정하기 위해 적절한 정정 하위 플로우를 트리거하는 작업 역할을 합니다. CMDB 규정 준수 감사는 통신 불일치 식별 및 조정의 기초를 형성합니다.

- CMDB 준수는 CMDB의 예외(불일치)를 식별하는 후처리 규칙으로 감사를 실행합니다.
- CMDB 준수는 실패한 상태(실패한 상태는 감사에서 CMDB의 예외 또는 불일치를 발견한 결과)의 각 감사 기록에 대해 후속 작업을 생성합니다. 불일치를 해결하고 해결하기 위해 각 후속 작업에 대해 정정 플로우를 설계하고 트리거할 수 있습니다.

통신 불일치 식별 및 조정의 논리와 정정 하위 플로우 예는 TSOM 가시성 플러그인에 자동으로 포함됩니다. 일반 CMDB 준수 도구 세트에 대한 자세한 내용은 다음 문서를 참조하십시오 [CMDB Compliance](#).

불일치 식별 시나리오(준수 감사 사용)

TSOM 가시성의 불일치 식별은 CMDB 준수(인증 감사) 사용에 의존하며, 모델 관계와 정보를 사용하여 불일치를 식별하는 특정 논리를 추가하여 이를 확장했습니다. 정정을 지원하기 위해 시스템은 발견된 각 문제에 대해 다음과 같은 특정 조정 작업 유형을 생성합니다.

- 슬롯 점유 불일치
- 최근 디스커버리 날짜가 설정되지 않음
- 최근 디스커버리 날짜가 구성된 임계치 내에 없음
- 모델 관계가 정의되지 않음
- CI 모델을 찾을 수 없습니다.
- 잘못된 관계 수입니다.
- 논리적 인터페이스에 대한 참조를 찾을 수 없음

다음 감사를 사용하여 검색된 물리적 및 논리적 엔터티의 불일치를 식별할 수 있습니다

- 통신 불일치 감사
- 통신 논리적 연결 불일치 감사
- 통신 네트워크 토폴로지 불일치 감사

i 주:

일반 준수 감사에 대한 자세한 내용은 다음 문서를 참조하십시오 [Certification audits](#).

자동화 및 UI 통합

- 실패한 모든 감사는 자동으로 후속 작업을 작성합니다.
- 작업은 플로우 디자이너를 사용하여 사전 구축된 하위 플로우 또는 사용자 지정 하위 플로우를 트리거할 수 있습니다.
- 작업 양식의 "정정" UI 작업 버튼을 사용하여 수동 정정을 시작할 수 있습니다.
- 정정 단계는 가시성과 감사를 위해 작업 메모에 기록됩니다.

i 주:

하위 플로우를 빌드하는 방법에 대한 자세한 내용은 다음 문서를 참조하십시오 [Building subflows](#).

실패한 감사 결과 기록에 대해 작성된 후속 작업 유형

조건과 일치하는 CI 관계 테이블(cmdb_rel_ci)에서 각 관계 기록의 상위 CI 및 하위 CI에 대해 다음과 같은 불일치 유형(감사 결과)을 찾을 수 있으며, 실패한 각 감사 결과에 대해 다음과 같은 후속 작업을 만들 수 있습니다.

1. 최근 디스커버리 날짜가 설정되지 않음 - CI의 최근 디스커버리 날짜 필드가 누락된 경우 작성됩니다.
2. 구성된 임계치 내에 없는 최근 디스커버리 날짜 - 상위 CI와 하위 CI 간의 최근 디스커버리 날짜 필드 값의 차이가 2.5일을 초과하는 경우 생성됩니다. 예를 들어 기본적으로 `sn_tsom_core.discovered_date.diff.threshold.in.days` 시스템 속성에서 2.5일로 설정되며 변경할 수 있습니다.
3. CI 모델을 찾을 수 없음 ("모델 ID" 필드가 설정되지 않았거나 데이터가 잘못되었습니다.) 해당 CI 모델을 찾을 수 없는 경우 생성됩니다. CI 모델을 찾을 수 없는 경우 CI 모델에 의존하기 때문에 다음 검증(4-6)은 관련이 없습니다. CI 모델이 발견되면 감사는 다음 유효성 검사(4-6)로 계속됩니다.
4. 점유한 슬롯 불일치 - 카드가 잘못된 수의 슬롯을 차지하는 경우 생성됩니다.
5. 모델 관계가 정의되지 않으며 TNI가 설치된 경우에만 관련이 있습니다. 감사가 네트워크 모델 관계 테이블에서 상위 CI 모델과 하위 CI 모델 간의 관계를 찾을 수 없는 경우 생성됩니다.
6. 잘못된 관계 수 - TNI가 설치된 경우에만 관련이 있습니다. 감사에서 검색된 하위 CI 기록 수가 네트워크 모델 관계 테이블의 모델 관계 카운트 필드에서 해당 상위 CI 기록의 최대 수를 초과하는 것으로 확인되면 생성됩니다.
7. 잘못된 관계 수 - 논리적 인터페이스가 둘 이상의 논리적 연결과 연결되어 예상되는 일대일 매핑을 위반할 때 논리적 연결 불일치 감사 중에 생성됩니다.
8. 잘못된 관계 수 - 네트워크 토폴로지 기록이 필요한 관계 기준을 충족하지 않을 때 네트워크 토폴로지 불일치 감사 중에 생성됩니다. 특히:
 - 기록에는 장비 CI와 하나 이상의 "Contains:Contained By" 관계가 있어야 합니다.
 - 또한 기록에는 논리적 연결 CI와의 "구성원:구성원" 관계가 하나 이상 있어야 합니다.
9. 논리적 인터페이스에 대한 참조를 찾을 수 없음 - 논리적 연결에 엔드포인트(포트 A 또는 포트 z 또는 둘 다) 중 하나가 누락될 때 논리적 연결 불일치 감사 중에 생성됩니다.

통신 조정

네트워크 인벤토리 불일치 조정을 자동화하고 운영 효율성을 개선합니다. 통신 조정은 다음을 지원합니다.

- 라이브 네트워크 인벤토리와 CMDB 인벤토리 간의 불일치를 식별하고 해결하여 조정을 보장하고 생산성을 높입니다.
- 불일치 유형을 자세히 설명하는 자동 생성된 불일치 보고서를 통해 사용자에게 권한을 부여합니다.
- 사용자에게 권장 시정 조치와 불일치를 해결하기 위해 수동 또는 자동 방법 중에서 선택할 수 있는 유연성 및 제어 기능을 제공합니다.
- 네트워크 자원의 운영 상태를 인벤토리 CMDB의 동일한 상태와 자동으로 정렬하여 운영을 개선합니다

관련 정보

[불일치 식별 - 불일치 유형](#)

[통신 불일치 식별 및 조정 활성화](#)

[통신 불일치 감사 실행](#)

[통신 불일치 식별 및 조정과 함께 설치되는 시스템 구성요소](#)

불일치 식별 - 불일치 유형

통신 불일치 식별 및 조정 기능은 네트워크 상태(TSOM 검색 또는 서비스 그래프 커넥터를 통해 검색됨)와 CMDB 또는 TNI에 저장된 인벤토리 데이터 간의 불일치를 식별하고 분류합니다.

불일치 식별은 다음과 같은 CMDB 준수 인증 감사를 통해 이루어집니다.

- CI 및 관계 데이터에서 실행됩니다.
- 검색된 기록과 인벤토리에 포함된 기록을 비교합니다.
- 불일치가 감지되면 후속 작업을 생성합니다.

불일치 유형

감사 프로세스는 검색된 네트워크 데이터를 CMDB/TNI의 기존 인벤토리와 비교하여 불일치를 식별합니다. 불일치는 두 가지 주요 범주로 나뉩니다.

네트워크에 없음 - 인벤토리에 있지만 네트워크에는 누락된 엔터티

정의: CI가 Discovery에 의해 탐지되었지만 CMDB/TNI에서 누락되었거나 잘못 표현되었습니다. 예를 들어 Discovery가 Slot04에 설치된 Card05를 탐지하지만 CMDB는 여전히 Card04를 나열하거나 더 나쁜 경우 Card04와 Card05를 모두 동일한 슬롯에 표시하여 카디널리티 또는 모델 제약 조건을 위반합니다.

영향도:

- 데이터 충돌을 강조 표시하기 위해 불일치 작업이 생성됩니다.
- 오래된 기록을 폐기하거나 슬롯 할당을 업데이트하여 데이터를 조정하기 위해 선택적 정정 하위 플로우를 트리거할 수 있습니다.

일치하지 않는 구성 항목(CI) - 인벤토리 및 네트워크에 존재하지만 속성 값과 계층적 관계가 다른 엔터티

정의: CI는 Discovery와 CMDB/TNI 모두에 존재하지만 관계, 계층 구조 또는 속성 값에는 불일치가 있습니다. 하위 유형은 다음과 같습니다.

- 계층 구조 불일치 - CI 간의 구조적 관계(예: 상위-하위 연결)가 일치하지 않을 때 발생합니다. 예를 들면 다음과 같습니다.
 - 새시에 모델 정의에서 허용하는 것보다 많은 하위 카드가 포함되어 있습니다.
 - 카드가 디스커버리 데이터와 일치하지 않는 CMDB의 슬롯과 잘못 연결되었습니다. 확인 소스는 다음과 같습니다.
 - Contains::Contained by ### 대한 cmdb_rel_ci 기록입니다.
 - 모델별 제약 조건을 적용하기 위한 sn_ni_core_network_model_relationship 테이블입니다.
- 속성 값 불일치 - CI 필드 수준 속성에 불일치가 있습니다. 일반적인 문제는 다음과 같습니다.
 - 오래되었거나 잘못된 디스커버리 날짜입니다.
 - 부정확한 모델 구성.
 - 모델 규칙을 위반하는 잘못된 슬롯 할당입니다.

영향으로 영향을 받는 기록이 감사 보고서에서 실패로 표시됩니다. 후속 작업은 컨텍스트별 정정 하위 플로우를 호출하여 인벤토리 데이터를 실제 네트워크 상태에 맞게 재정렬할 수 있습니다.

주:

자세한 내용은 [CMDB 360에서 속성 값 불일치 구성](#) 문서를 참조하십시오.

관련 정보

통신 불일치 식별 및 조정 활성화

통신 불일치 감사 실행

Telecommunications Service Operations Management 구성

외부 네트워크 모니터링 및 검색 시스템과 통합하여 실시간 이벤트 수집, 상관 관계 및 자동 정정을 사용하도록 구성 Telecommunications Service Operations Management (TSOM)합니다.

경보 수집, CMDB 채우기, 불일치 탐지 및 서비스 영향 가시성을 포함한 엔드 투 엔드 통신 서비스 운영을 사용하도록 TSOM을 설정합니다. 이 구성에는 통신 API 알림 활성화, 검색 및 서비스 그래프 커넥터 설정, 가시성 활성화, 감사 및 조정 프레임워크 구성이 포함됩니다.

구성 개요

Telecommunications Service Operations Management(TSOM)를 사용하려면 엔드 투 엔드 통신 운영을 지원하기 위해 경보 수집, 검색, 데이터 정규화 및 조정 전반에 걸쳐 여러 구성 요소를 구성해야 합니다. 구성 플로우에는 일반적으로 다음이 포함됩니다.

1. 시스템에 TSOM Core를 사용하려면 다음 플러그인을 활성화합니다.

- 패턴 디자이너(com.snc.pattern.designer)
- ServiceNow 통합 허브 시작됨 팩 설치 관리자(com.glide.hub.integrations)
- 검색(com.snc.discovery)
- CMDB CI 클래스 모델(sn_cmdb_ci_class) 1.69.0
- 가시성 콘텐츠(sn_pattern_design) 6.23.0
- Integration Commons for CMDB (sn_cmdb_int_util) 2.19.0

2. 경보 수집 활성화: 통신 API 알림을 활성화하고 외부 시스템에서 경보를 수신하도록 주제 구독을 설정합니다. 자세한 내용은 [통신 API 알림 구성](#) 문서를 참조하십시오.

3. 가시성 설정: 서비스-인프라 매핑을 보고 네트워크 상태를 모니터링하도록 통신 가시성을 구성합니다. 자세한 내용은 [통신 가시성 설정](#) 문서를 참조하십시오.

a. 통신 인식 CMDB를 채웁니다.

- 가로 검색을 설치하고 구성합니다. 자세한 내용은 [가로 검색 설치 및 검색 패턴 설정](#) 문서를 참조하십시오.
- 커넥터에서 TDB(Telecom Discovery Builder) ETL 플로우를 설정합니다. 자세한 내용은 [커넥터에서 Telecom Discovery Builder 프레임워크 ETL 구성](#) 문서를 참조하십시오.
- 서비스 그래프 커넥터(예: Nokia Altiplano)를 사용하여 토폴로지 및 구성 데이터를 임포트합니다. 자세한 내용은 [Nokia Altiplano 서비스 그래프 커넥터 구성](#) 문서를 참조하십시오.

b. 불일치 탐지 사용:

- 통신 불일치 식별 및 조정을 활성화합니다. 자세한 내용은 [통신 불일치 식별 및 조정 활성화](#) 문서를 참조하십시오.
- CMDB 360에서 통신 감사에 대한 필터를 정의하고 속성 값 불일치 검사를 구성합니다.

각 단계는 모듈식으로 진행되며 사용자 환경과 사용 가능한 통합에 따라 구성할 수 있습니다.

통신 API 알림 구성

인스턴스에서 통신 API 알림을 ServiceNow 구성합니다.

통신 API 알림 워크플로우 모델링

다음 단계는 인스턴스에서 통신 API 알림을 구성하는 데 도움이 됩니다 ServiceNow .

- 1. 항목 작성:** 외부 메시지 상세 정보를 수동으로 입력하거나 외부 시스템에서 사용 가능한 주제를 자동으로 수집하여 주제를 만들 수 있습니다.
- 2. 토픽 구독 작성:** 고객 기본 설정에 따라 외부 시스템에서 수신되는 알림에 사용 가능한 주제를 구독합니다. 또한 콜백 URL을 생성하고 구독을 등록합니다.
- 3. 통신 경보 관리 개방형 API 연결의 엔드포인트 활성화:** 외부 시스템으로부터 응답을 수신하려면 워크플로우 스튜디오 Telecommunications Alarm Management Open API 연결의 구독된 엔드포인트를 활성화하십시오.
- 4. 알림 수신**을 위해 외부 시스템에 콜백 URL을 제공합니다. 고객은 콜백 URL을 재사용할 수도 있습니다. TMF 688의 요청이 콜백 URL에 도달하면 ## ## #### ## #### 플로우를 시작하여 이벤트를 작성합니다.

이벤트 알림 관리를 처리하는 기능에 대한 자세한 내용은 이벤트를 [Event Notification Management Open API](#) [TMFTopicEventAPIUtilOOB - Scoped](#) 생성, 업데이트 및 삭제하기 위해 외부 트리거 정의에 의해 트리거되는 개방형 API 요청을 참조하십시오.

이 워크플로우는 이벤트 관리 애플리케이션에 이벤트를 생성합니다. 사용 이벤트 관리방법에 대한 자세한 내용은 [이벤트 관리를](#) 참조하십시오.

관련 정보

[통신 API 알림과 함께 설치되는 시스템 구성요소](#)

[통신 API 알림을 통한 외부 이벤트 관리](#)

항목 작성

주제를 생성하고 외부 시스템에서 수신 알림을 주제에 게시합니다. 주제를 만들면 구독자는 구독하려는 주제를 선택할 수 있습니다.

시작하기 전에

Telecommunications Alarm Management Open API(sn_ind_tmf642) 애플리케이션이 와 함께 ServiceNow AI Platform설치되어 있는지 확인하십시오.

필요한 역할: admin, sn_api_notif_mgmt.topic_creator

이 태스크 정보

외부 메시지 상세 정보를 수동으로 입력하거나 외부 시스템에서 사용 가능한 주제를 자동으로 수집하여 주제를 만들 수 있습니다. 주제를 만들면 주제 [sn_api_notif_mgmt_topic] 테이블에 기록이 만들어집니다.

프로시저

- 모두 > 통신 API 알림 > 주제.
- 새로 만들기를 선택합니다.
외부 시스템과 통합한 경우 주제 가져오기 를 선택하여 사용 가능한 주제를 자동으로 가져올 수 있습니다. 이 작업은 #### ## ## API 하위 플로우를 트리거합니다. 이 주제에서 기록을 쿼리하고 조작할 수 있는 기능에 대한 자세한 내용은 다음 문서를 참조하십시오 [TopicUtilOOB - Scoped](#) .
- 양식에서 필드를 채웁니다.

주제 양식

필드	설명
주제 ID	고유 주제 ID입니다.
주제 이름	주제의 이름입니다.
유형	주제의 유형입니다. 다음 중 하나를 선택합니다. - 수신: 인바운드 알림에 대한 옵션입니다. - 송신: 아웃바운드 알림에 대한 옵션입니다.
헤더 쿼리	인코딩된 헤더 쿼리 매개변수입니다. TMF 688 표준을 따르는 쿼리 매개변수에 대한 자세한 내용은 TM 포럼 을 참조하십시오.
콘텐츠 쿼리	인코딩된 콘텐츠 쿼리 매개변수입니다. TMF 688 표준을 따르는 쿼리 매개변수에 대한 자세한 내용은 TM 포럼 을 참조하십시오.
설명	주제에 대한 간략한 설명입니다.

4. 제출을 선택합니다.

결과

주제가 생성됩니다.

다음에 수행할 작업

고객 요구 사항에 따라 토픽 구독을 생성할 수 있습니다. 자세한 내용은 다음 문서를 참조하십시오 .

토픽 구독 작성

외부 시스템에서 들어오는 알림에 응답하려는 주제를 ServiceNow AI Platform 구독합니다. 주제를 구독하면 구독자는 구독하는 주제에 따라 알림을 받습니다.

시작하기 전에

- Telecommunications Alarm Management Open API(sn_ind_tmf642) 애플리케이션이 와 함께 ServiceNow AI Platform설치되어 있는지 확인하십시오.
- 수신 알림에 대한 주제를 생성합니다.

필요한 역할: admin, sn_api_notif_mgmt.subscription_creator

이 태스크 정보

고객 기본 설정에 따라 외부 시스템에서 들어오는 알림에 대해 사용 가능한 주제를 구독합니다. 고객과 공유할 콜백 URL을 생성합니다. 외부 시스템의 요청이 콜백 URL에 도달하면 애플리케이션에서 이벤트 이벤트 관리 생성이 시작됩니다.

또한 주제 구독을 등록하여 수신 알림 수신을 시작합니다. 토픽 구독을 만들면 토픽 구독 [sn_api_notif_mgmt_subscription] 테이블에 레코드가 만들어집니다. 토픽 구독에서 레코드를 쿼리하고 조작하는 방법에 대한 자세한 내용은 다음 문서를 참조하십시오 [TopicSubscriptionUtilOOB - Scoped](#) .

프로시저

1. 모두 > 통신 **API** 알림 > 구독.
2. 새로 만들기를 선택합니다.
3. 양식에서 필드를 채웁니다.

주제 구독 양식

필드	설명
주제	구독하려는 주제입니다.
CallbackURL	수신 알림을 캡처하기 위해 외부 시스템과 공유하는 콜백 URL입니다. CallbackURL 생성을 선택하면 URL이 자동으로 생성됩니다.
쿼리 필터링	주제의 인코딩된 콘텐츠 쿼리 매개변수입니다. 필터 쿼리를 수정할 수도 있습니다. TMF 688 표준을 따르는 쿼리 매개변수에 대한 자세한 내용은 TM 포럼 을 참조하십시오.
등록 상태	외부 시스템에 대한 주제 등록 상태입니다. 기본적으로 등록되지 않았습니다. 프로세스가 성공하면 필드 값이 등록됨으로 변경됩니다. 그렇지 않으면 오류입니다.
등록 메시지	외부 시스템의 등록 상태 메시지입니다.
구독 ID	외부 시스템의 고유 구독 ID입니다.

4. **CallbackURL** 생성을 선택하여 콜백 URL을 가져옵니다.

5. 등록을 선택하여 구독을 등록합니다.

결과

콜백 URL에 대한 트리거 정의가 생성되고 주제가 외부 시스템에 등록됩니다.

다음에 수행할 작업

에서 워크플로우 스튜디오 Telecommunications Alarm Management Open API 연결의 엔드포인트를 활성화합니다. 자세한 내용은 [통신 경고 관리 개방형 API 연결의 엔드포인트 활성화](#) 문서를 참조하십시오.

통신 경고 관리 개방형 **API** 연결의 엔드포인트 활성화

Telecommunications Alarm Management Open API 연결의 엔드포인트를 활성화합니다. 엔드포인트를 활성화하면 등록된 주제에 대해 외부 시스템에서 수신 알림을 수신합니다.

시작하기 전에

- 주제를 작성하고 구독하여 수신 알림을 받습니다.
- 콜백 URL을 작성하고 토픽 구독을 등록합니다.

필요한 역할: 관리자

이 태스크 정보

에서 워크플로우 스튜디오 Telecommunications Alarm Management Open API 연결의 구독된 엔드포인트를 활성화하여 외부 시스템으로부터 응답을 수신합니다.

프로시저

1. 다음으로 이동 모두 > 프로세스 자동화 > 플로우 디자이너.
2. 연결 탭에서 **Telecommunications Alarm Management Open API**를 선택합니다.
3. 활성화할 엔드포인트 기록을 엽니다.
4. 활성화를 선택합니다.

통신 가시성 설정

통신 가시성은 통신 디스커버리와 통신 불일치 식별 및 조정을 모두 지원하는 기본 기능을 제공합니다. 여기에는 공유 로직, 향상된 CI 클래스 모델, 통신 네트워크 요소에 맞게 조정된 IRE(식별 및 조정 엔진) 업데이트가 포함됩니다.

시작하기 전에

필요한 역할: 관리자

확인 사항:

- ServiceNow 인스턴스에는 TSOM에 대한 라이선스가 부여됩니다.
- MID Server가 작동 중이고 Discovery에 대해 검증되었습니다.
- 업그레이드 중에 재정의의 방지할 수 있도록 통신 CI에 적용된 사용자 지정 IRE 규칙을 검토합니다.

이 태스크 정보

통신 가시성을 구성하려면 필요한 플러그인을 설치하고 CMDB CI Class Models를 통신별 IRE 식별 규칙이 도입된 버전 1.69.0으로 업데이트해야 합니다. 이를 통해 통신 도메인 간에 정확한 CI 식별 및 조정이 이루어집니다.

- i** 주:
- 영향을 받는 통신 CI에 대해 IRE 식별 규칙을 사용자 지정한 경우 버전 1.69.0으로 업그레이드하면 해당 규칙이 재정의되거나 영향을 받을 수 있습니다.

Telecom Visibility 구독에 포함된 기능은 다음과 같습니다.

플러그인 이름	설명	스토어 앱
서비스 그래프 커넥터 sn_sgc_altiplano_connector	Nokia Altiplano용 서비스 그래프 커넥터	예
통신 Discovery 패턴sn_tsom_patterns	통신 Discovery 패턴	예
Telecom Visibility 플러그인 (핵심 논리)sn_tsom_core	통신 가시성 및 불일치 조정을 위한 핵심 로직	아니요

- i** 주:
- 이러한 플러그인은 CMDB CI 클래스 모델 애플리케이션의 설치 또는 업데이트를 버전 1.69.0으로 자동으로 트리거합니다.

프로시저

1. 다음으로 이동 모두 > 플러그인.
2. 다음을 검색하여 설치합니다.

- a. Nokia Altiplano용 서비스 그래프 커넥터(sn_sgc_altiplano_connector)
 - b. 통신 Discovery 패턴(sn_tsom_patterns)
 - c. Telecom Visibility (플러그인) (sn_tsom_core)
3. 옵션: CMDB CI 클래스 모델을 업데이트합니다.
- a. 다음으로 이동 모두 > 사용 가능한 애플리케이션 ServiceNow Store의 목록.
 - b. CMDB CI 클래스 모델(sn_cmdb_ci_class)을 검색합니다.
 - c. 버전 1.69.0을 설치하거나 이 버전으로 업그레이드합니다.



주:

위에 나열된 TSOM 플러그인 중 하나를 설치하면 CMDB CI Class Models 앱의 버전 1.69.0이 자동으로 업데이트되거나 설치됩니다. 인스턴스에 TSOM 플러그인이 포함되어 있지 않거나 Yokohama 이전 릴리스(예: Washington DC 또는 Xanadu)를 사용하는 경우 스토어 앱을 수동으로 설치하거나 업그레이드할 수 있습니다.

4. 다음으로 이동 **CMDB** > 식별 규칙.

a. 다음과 관련된 규칙 검토:

- cmdb_ci_interface_card
- cmdb_ci_slot
- cmdb_ci_subslot
- cmdb_ci_network_adapter

b. 사용자 지정 논리가 계속 작동하는지 확인합니다.

c. 필요에 따라 사용자 지정 규칙을 업데이트하거나 병합합니다.

결과

다음 항목이 활성으로 표시되는지 확인합니다.

- 통신 가시성(sn_tsom_core)
- 통신 Discovery 패턴(sn_tsom_patterns)
- Nokia Altiplano용 서비스 그래프 커넥터(sn_sgc_altiplano_connector)
- CMDB CI Class Models 버전 1.69.0

관련 정보

[통신 가시성](#)

가로 검색 설치 및 검색 패턴 설정

수평 Discovery 패턴 설치 의존성 및 요구 사항 이해.

시작하기 전에

필요한 역할: 관리자

TSOM에 대한 구독이 있는지 확인합니다.

이 태스크 정보

[ServiceNow Store](#) 웹 사이트를 방문하면 사용 가능한 모든 앱을 확인하고 스토어에 요청을 제출하는 방법에 대한 정보를 참조할 수 있습니다. 출시된 모든 앱의 누적 릴리스 정보는 [ServiceNow Store 버전 기록 릴리스 정보](#)를 참조하십시오.

의존성 및 요구사항:

- 텔레콤 코어(sn_tsom_core)
- 검색에 의해 자동으로 설치되는 Discovery Core 플러그인(com.snc.discovery.core)
- ITOM 검색 라이선스 플러그인(com.snc.itom.discovery.license). 이 플러그인을 활성화해야 합니다.
- ITOM 라이선싱 플러그인(com.snc.itom.license). 자세한 내용은 [Request Discovery](#) 문서를 참조하십시오.

패턴 실행 논리: 기본적으로 TSOM 패턴이 실행되면 TSOM 관련 패턴과 해당 ITOM 패턴(예: Telco 라우터가 라우터 실행)을 모두 실행합니다. 이를 통해 필요할 때 공유 ITOM 라이브러리를 재사용할 수 있습니다. 이 동작을 재정의하려면 다음을 수행합니다.

- 시스템 속성 sn_tsom_patterns.itom_pattern_enabled를 사용합니다.
- 이 속성을 false로 설정하면 TSOM 관련 패턴만 실행됩니다.

프로시저

1. 가로 검색 애플리케이션을 설치합니다.
를 참조하십시오 [Discovery setup](#). 이는 Telecommunications Discovery 패턴 실행의 기본입니다.
2. 통신 Discovery 패턴 구하고 설치합니다.
 - a. 스토어에서 sn_tsom_patterns(통신 검색 패턴)을 ServiceNow[#] 설치합니다.
3. MID 서버 및 동기화 패턴 설정:
 - a. 설치된 패턴을 적절한 MID Server와 동기화하여 사용할 준비가 되었는지 확인합니다.
 - i. 다음으로 이동 디스커버리 > **MID** 서버.
 - ii. MID에 패턴 동기화를 선택합니다.

주:
이 작업은 TSOM 및 ITOM 패턴을 모두 동기화합니다.

MID 서버를 구성하는 방법에 대한 자세한 내용은 을 참조하십시오 [Configuring MID Server](#).
4. TSOM 시스템 속성을 구성합니다.
 - a. 시스템 속성 sn_tsom_patterns.itom_pattern_enabled를 설정하여 TSOM 패턴만 사용할지 아니면 ITOM과 TSOM 패턴의 조합을 사용할지에 대한 논리를 정의합니다.
 - i. 다음으로 이동합니다.모두 > 시스템 속성 > 모든 속성.
 - ii. **sn_tsom_patterns.itom_pattern_enabled**를 선택합니다.
 - iii. 값이 **true** (기본값)로 설정되어 있는지 확인합니다.

TSOM이 TSOM 패턴만 실행하고 ITOM 패턴을 제외하도록 하려면 값을 **false**로 설정합니다.



주:
기본 설정은 TSOM 및 ITOM 패턴을 모두 사용하도록 구성됩니다.

5. 특정 MID Server에서 다양한 ITOM 패턴을 TSOM 패턴으로 교체할 수 있습니다.

예: 통신 라우터 패턴은 **mid.telecom.discovery.patterns.enabled** 가 해당 MID 서버에 대해 true로 설정된 경우 특정 MID 서버에 대한 네트워크 라우터 패턴을 대체합니다.

- a. 필터 탐색기로 이동하여 **ecc_agent_config.list**를 입력합니다.
- b. **mid.telecom.discovery.patterns.enabled**를 선택합니다(각 MID 서버에는 이 매개변수가 있음).
- c. 값이 **true**로 설정되어 있는지 확인합니다.

TSOM 패턴을 실행하는 데 사용할 각 MID 서버에 대해 이 구성을 반복합니다.

관련 정보

[Discovery 패턴을 사용한 Direct Discovery](#)

커넥터에서 **Telecom Discovery Builder** 프레임워크 ETL 구성

커넥터 범위에 복제하고, 적절한 데이터 소스를 할당하고, 새 서비스 그래프 커넥터의 일부로 배포하여 사전 구축된 통신 검색 빌더 프레임워크 ETL을 활용합니다.

TSOM(Telecom Service Operations Management) Core와 함께 제공되는 통신 검색 작성기 프레임워크 ETL은 통신 커넥터에서 CMDB로의 데이터 수집을 간소화하도록 설계된 즉시 사용 가능한 프레임워크입니다. 직접 구성하는 대신 관리자는 ETL을 커넥터의 애플리케이션 범위에 복제하고 커넥터의 검색 페이로드에 맞게 데이터 소스를 업데이트할 수 있습니다.

커넥터에서 통신 검색 작성기 프레임워크 ETL을 사용하는 단계

1. 임시 데이터 원본 만들기: TSOM Core의 일반 스키마 다중 원본 데이터 원본을 기반으로 커넥터의 애플리케이션 범위에서 데이터 원본을 만듭니다. 이렇게 하면 스키마에 정렬된 자리 표시자 데이터를 사용하여 중복된 ETL을 실행하고 테스트할 수 있습니다. 자세한 내용은 [통신 핵심 데이터 소스와 유사한 데이터 소스 작성](#) 문서를 참조하십시오.
2. 통신 검색 작성기 프레임워크 ETL 복제: ETL 스튜디오에 액세스하고 Telco 일반 스키마 ETL을 찾아 커넥터의 범위에 복제합니다. 중복 시: 새 이름을 제공합니다. 임시 데이터 소스를 할당합니다. importSet를 검색 소스로 사용합니다. 자세한 내용은 [통신 디스커버리 빌더 프레임워크 ETL을 커넥터 범위에 복제](#) 문서를 참조하십시오.
3. 복제 후 새 ETL을 열고 임시 데이터 소스를 커넥터의 실제 검색 데이터 소스로 바꿉니다. 자세한 내용은 [커넥터의 데이터 소스 업데이트](#) 문서를 참조하십시오.
4. 서비스 그래프 커넥터를 테스트하거나 배포합니다. 자세한 내용은 [기존 ETL을 사용하여 새 서비스 그래프 커넥터 배포](#) 문서를 참조하십시오.

통신 핵심 데이터 소스와 유사한 데이터 소스 작성

커넥터의 애플리케이션 범위에서 스키마 준수 데이터 소스를 설정하여 통신 데이터를 시뮬레이션하고 라이브 장치 데이터와 통합하기 전에 통신 디스커버리 빌더 프레임워크 ETL의 성공적인 테스트 및 검증을 보장합니다.

시작하기 전에

필요한 역할: 관리자

다음 사항을 확인하십시오.

- TSOM Core 애플리케이션 및 해당 데이터 소스에 대한 액세스 권한.
- 커넥터의 애플리케이션 범위로 전환합니다.
- Telco Generic Schema를 준수하는 샘플 페이로드가 있습니다(선택 사항이지만 테스트용으로 권장됨).

이 태스크 정보

통신 검색 작성기 프레임워크 ETL을 SGC(서비스 그래프 커넥터) 애플리케이션 범위로 복제하는 경우 먼저 TSOM Core 데이터 소스의 정확한 구조를 복제하는 데이터 소스를 생성해야 합니다. 이 복제된 데이터 소스는 ETL을 라이브 통신 데이터에 연결하기 전에 ETL을 테스트하고 유효성을 검사하는 데 필요한 스키마와 구조를 제공합니다.

통신 검색 작성기 프레임워크에는 통신 코어에 정의된 Telco 일반 스키마를 준수하는 데이터가 필요합니다. 원본 TSOM 데이터 소스를 복제하면 다음을 보장할 수 있습니다.

- ETL 복제 중 스키마 정렬.
- 테스트를 위한 유효한 임포트 세트 구조입니다.
- 커스터마이제이션 및 업그레이드 안전을 위해 코어와 커넥터 범위를 분리합니다.

TSOM Core 데이터 소스와 유사한 데이터 소스를 생성하는 경우:

- 통신 검색 작성기 프레임워크 ETL을 복제하기 전.
- 시뮬레이션된 통신 데이터 또는 자리 표시자 통신 데이터를 사용하여 테스트 로드를 실행하려는 경우
- 수집 구성을 위해 커넥터의 애플리케이션 범위를 준비하는 경우.

다음 스크린샷은 데이터 소스를 생성하는 동안 필드 값을 채우는 방법을 이해하는 데 도움이 될 수

There is a screenshot of the ServiceNow Data Source configuration interface. The 'Name' field is set to 'NSP Generic Data Source'. The 'Import set table label' is 'NSP Generic Data Source'. The 'Import set table name' is 'sn_sgc_nsp_generic_data_source'. The 'Type' is 'Custom (Load by Script)'. The 'Data in single column' checkbox is checked. The 'Data Loader' section shows a script for loading data from a table named 'import_set_table'.

```

1 (function loadData(import_set_table, data_source, import_log, last_success_import_time) {
2
3   const LOGICAL_COMPOSITES = [{"name": "generated_SR_7s_core1_LogicalComposite",
4     "class": "logical_composite", "key": "NSD08E595F50:LogicalComposite", "description": null, "devices": [{"key": "NSD08E595F50X",
5       "power_units": [{"name": "Power Shelf-1(ps-b10-shelf-ac/hv)", "class": "pdu", "unit_position": 1, "serial_number": "NSAA7C6E8507",
6         "model_name": "ps-b10-shelf-ac/hv", "model_number": null, "key": "NSD08E595F50NSD08E595F50:powerShelfSlot1/powerShelf=1",
7         "description": null, "slots": [{"name": "Power Module Slot-1/1(ps-b-ac/hv-6000)", "class": "slot", "description": null,
8           "model_name": "ps-slot", "model_number": null, "manufacturer": "Nokia", "unit_position": 1, "key": "NSD08E595F50:powerShelfSlot1/
9             powerShelf=1/powerModuleSlot=1", "cards": [{"name": "Power Module-1(ps-b-ac/hv-6000)", "class": "card", "unit_position": 1,
10              "key": "NSD08E595F50:powerShelfSlot1/powerModuleSlot=1/powerModule=1", "model_name": "ps-b-ac/hv-6000",
11              "model_number": null, "manufacturer": "Nokia", "firmware_version": null, "mac_address": null, "description": null,
12              "serial_number": "NS15E40AD9A1"}]}]}]}]}];
13
14   const DEVICES = [{"description": "Modular device, 1 card in slot 1, 1 MDA card in subslot 1, 36 connector ports
15     with 2 transceivers in port 1/1/c1 and 1/1/c2, 2 connector ports with 1 breakout ports each", "class": "device",
16     "model_name": "7750 SR-7s", "model_number": null, "manufacturer": "Nokia", "mac_address": "88:89:46:F1:11:31", "ip_address": "11.128.
17     28.27", "name": "generated_SR_7s_core1", "serial_number": "NSD08E595F50X", "key": "NSD08E595F50X", "firmware_version": "TIMOS-C-23.
18     18.R4-1", "ports": [{"name": "device_port_1", "class": "port", "model_name": "Ethernet", "description": null, "ip_address": "11.128.28.
19     101", "unit_position": 1, "key": "NSD08E595F50X:device_port_1", "virtual": false, "bandwidth_name": null,
20     "bandwidth_group": "Ethernet", "bandwidth": 400000, "mtu_size": 1400, "name": "device_port_1-ENET", "class": "port",
21     "model_name": "ENET", "description": null, "ip_address": "11.128.28.201", "unit_position": 1,
22     "key": "NSD08E595F50X:device_port_1-ENET", "virtual": true, "bandwidth_name": null, "bandwidth_group": null, "bandwidth": null,
23     "mtu_size": null, "name": "lag_1-VLAN", "class": "port", "model_name": "VLAN", "description": null, "ip_address": "11.128.28.251",
24     "unit_position": 1, "key": "NSD08E595F50X:lag_1-VLAN", "virtual": true, "bandwidth_name": null, "bandwidth_group": null,
  
```

프로시저

1. 다음으로 이동 모두 > 시스템 임포트 세트 > 관리 > 데이터 소스.
2. TSOM Core와 함께 제공되는 일반 스키마 다중 원본 v2 또는 유사한 기준 원본을 찾습니다.
3. TSOM Core 데이터 소스 기록을 열고 데이터 로더 필드에서 스크립트를 복사합니다.

4. 새로 만들기를 클릭하여 데이터 원본을 만듭니다.
5. 폼의 필드에 내용을 입력합니다.
자세한 내용은 [Create a Custom \(Load by Script\) type data source](#) 문서를 참조하십시오.
6. 유형 필드에서 사용자 지정(스크립트를 통해 로드)을 선택합니다.
7. 단일 열 필드에서 데이터를 선택합니다.
8. 데이터 로더 필드에 복사한 스크립트를 붙여 넣습니다.
9. 제출을 선택합니다.
데이터 소스가 생성됩니다.
10. 옵션: 데이터 소스 로드를 테스트하려면 다음을 수행합니다.
 - a. 20개 기록 또는 이와 유사한 기록 로드 테스트를 클릭하여 импорт 세트를 생성합니다.
 - b. 기록이 오류 없이 생성되었는지 확인합니다.
 - c. 논리 복합, 네트워크 장비 또는 포트와 같은 클래스가 스테이징 테이블에 나타나는지 확인합니다.

다음에 수행할 작업

데이터 소스가 생성되고 테스트된 후에는 다음이 수행됩니다.

- Telco Generic ETL을 복제할 때 импорт 소스로 사용합니다.
- 시뮬레이션 및 검증이 완료되면 실제 커넥터별 데이터 소스로 대체합니다.

관련 정보

[모든 서비스 그래프 커넥터를 지원하기 위한 표준화된 JSON 공통 데이터 세트](#)

[통신 디스커버리 빌더 프레임워크 ETL을 커넥터 범위에 복제](#)

통신 디스커버리 빌더 프레임워크 ETL을 커넥터 범위에 복제

통신 검색 빌더 프레임워크 ETL을 사용하면 SGC(서비스 그래프 커넥터) 팀이 ETL 논리를 처음부터 빌드하지 않고도 표준화된 스키마 준수 데이터 수집 파이프라인을 신속하게 채택할 수 있습니다.

시작하기 전에

필요한 역할: 관리자

다음을 확인하십시오.

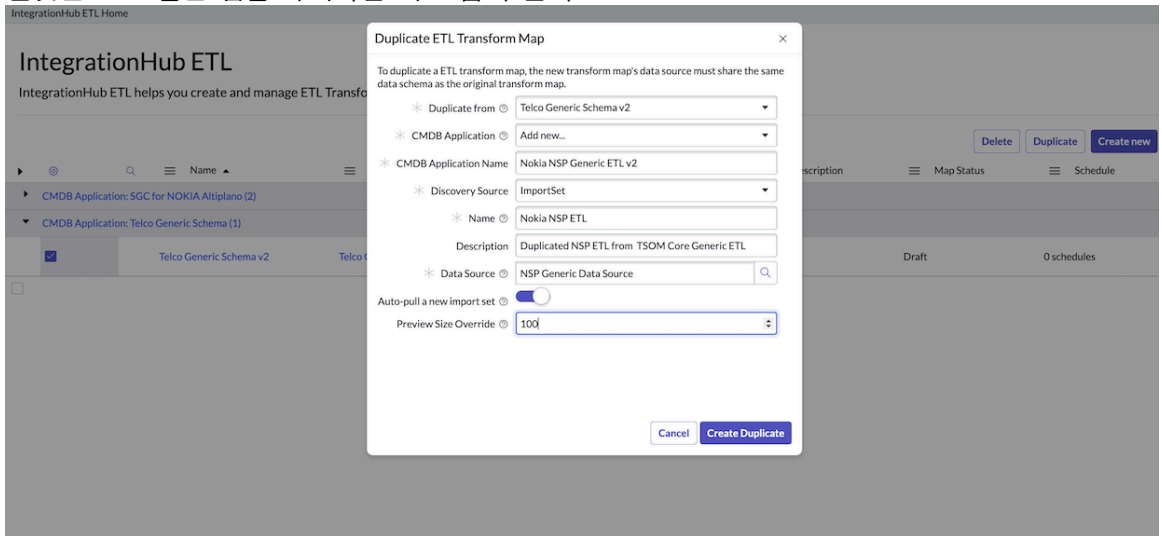
- Telecom Core 플러그인이 활성화되고 Telco Generic ETL v2(자동으로 설치됨)를 사용할 수 있습니다.
- 통합 허브 ETL 스튜디오에 대한 관리자 액세스 권한이 있습니다.
- 커넥터의 애플리케이션 범위에 임시 데이터 소스를 생성했습니다. 임시 데이터 원본을 생성한 후 일반 ETL을 커넥터의 애플리케이션 범위에 복제하여 특정 사용 사례에 맞게 사용자 지정하고 확장할 수 있습니다.

이 태스크 정보

Telco Generic ETL을 복제하면 다음을 수행할 수 있습니다.

- 여러 커넥터에서 표준화된 ETL 매핑을 재사용합니다.
- TSOM(Telecom Service Operations Management) 코어에서 제공하는 원래 기준선을 변경하지 않고도 ETL 동작을 사용자 지정합니다.
- 일관성 및 TNI 규정 준수를 위해 Telco 일반 스키마에 맞춥니다.
- 테스트되고 입증된 ETL 프레임워크로 작업하여 시간을 절약하고 오류를 줄입니다.

다음 스크린샷은 ETL 변환 맵을 복제하는 데 도움이 될 수



있습니다.

프로시저

1. 다음으로 이동 모두 > 통합 허브 **ETL**.

2. 통합 허브 **ETL** 페이지에서 커넥터의 애플리케이션 범위를 선택합니다.

i 주:

예를 들어 **Nokia** NSP용 서비스 그래프 커넥터입니다.

3. CMDB 애플리케이션 Telco Generic Schema를 확장하고 **Telco Generic Schema** 기록을 선택합니다.

4. 중복을 클릭합니다.
중복 ETL 변환 맵 페이지가 나타납니다.

5. 중복된 ETL 변환 맵 구성

a. **Duplicate** From(시작 복제) 필드에서 Telco Generic Schema(텔코 일반 스키마)가 선택되어 있는지 확인합니다.

b. 기존 CMDB 애플리케이션을 선택하거나 새로 추가... 를 클릭하여 새 CMDB 애플리케이션을 추가합니다.

c. 새 CMDB 애플리케이션의 경우 CMDB 애플리케이션 이름(예: Nokia NSP Generic ETL v2)을 입력합니다

d. 검색 소스 필드에서 임포트 세트를 선택합니다.

e. 이름 필드에 중복 ETL의 이름을 입력합니다.

f. 설명 필드에 중복 ETL에 대한 설명을 입력합니다.

g. 중복 ETL 변환 맵에 사용되는 데이터 소스를 선택합니다.

i 주:

이는 Telco 일반 스키마 ETL에 연결된 기존 기본 데이터 소스와 달라야 합니다. 자세한 내용은 [통신 핵심 데이터 소스와 유사한 데이터 소스 작성](#) 문서를 참조하십시오.

h. 데이터를 새 임포트 세트에 자동으로 가져오려면 새 임포트 세트 자동 끌어오기 옵션을 활성화합니다.

i. 미리 보기 크기 재정의 필드에서 테스트 및 확인을 위한 사용자 지정 미리 보기 크기를 설정합니다.

6. 중복 생성을 선택합니다.

Telco Generic ETL의 중복 ETL이 생성됩니다.

7. 옵션: 복제된 ETL 기록을 열고 매핑 및 설정을 검토합니다.

다음에 수행할 작업

1. 중복된 ETL 테스트(선택 사항이지만 권장됨):

- 임시 데이터 소스를 사용하여 테스트 로드 또는 시뮬레이션을 실행합니다.
- 임포트 세트가 성공적으로 처리되는지, CI가 Telco 일반 스키마에 따라 생성되는지, 관계가 올바르게 설정되었는지 확인합니다.

2. 복제 및 테스트에 성공한 후:

- 실제 프로덕션 데이터 소스(실제 디바이스 데이터의 경우)를 가리키도록 ETL의 데이터 소스 구성을 업데이트합니다.
- 커넥터 통합을 테스트 또는 프로덕션 환경에 배포합니다.
- 임포트 실행을 모니터링하여 인벤토리 데이터가 CMDB에 올바르게 수집되었는지 확인합니다.

관련 정보

[커넥터의 데이터 소스 업데이트](#)

예 - Telco 일반 ETL 스키마 복제

이 예에서는 통신 통합을 위해 사용자 지정된 SGC(서비스 그래프 커넥터) ETL을 설정하기 위해 Telco 일반 스키마 ETL을 복제하는 방법을 안내합니다. 표준화된 Telco 일반 스키마를 기반으로 커넥터의 애플리케이션 범위에서 기존 ETL을 생성하여 스키마 정렬, 일관성 및 더 빠른 배포를 보장하려는 경우 이 절차를 사용합니다.

시나리오

새 서비스 그래프 커넥터를 배포하고 있으며 Telco 일반 스키마 ETL을 커넥터의 애플리케이션 범위에 복제해야 합니다.

임시 데이터 소스를 생성하고, ETL을 복제하고, 커넥터의 디바이스 페이로드와 함께 작동하도록 구성합니다.

Telco 일반 스키마 ETL을 복제하는 단계

1. ETL을 복제할 커넥터의 애플리케이션 범위로 전환합니다(예: Nokia NSP 커넥터).

2. 데이터 소스를 생성합니다.

- 시스템 임포트 세트 > 관리 > 데이터 소스로 이동합니다.
- TSOM Core 애플리케이션에서 제공하는 일반 스키마 다중 소스 데이터 소스를 찾습니다.
- 이 데이터 소스를 커넥터의 애플리케이션 범위에 복사합니다.
- 테스트 로드 20 기록을 실행하여 복사된 데이터 소스를 테스트하고 샘플 임포트 세트를 만듭니다.

3. 중복 ETL 변환 맵 대화 상자 열기: ETL Studio에서 중복 ETL을 선택하여 중복 프로세스를 시작합니다.
4. **Duplicate** from(다음에서 복제) 목록에서 Telco Generic Schema(텔코 일반 스키마)를 선택합니다.
5. 새로 추가를 선택하고 중복된 ETL의 이름을 입력합니다.
6. 중복된 ETL의 디스커버리 소스로 importSet를 선택합니다.
7. 복제된 변환 맵의 새 이름을 입력합니다.
8. 일반 스키마 다중 원본에서 복사한 새로 만든 임시 데이터 원본을 지정합니다.
9. 필요한 경우 새 임포트 세트 자동 끌어오기를 활성화하여 복제 후 새 기록을 자동으로 로드합니다.
10. 복제를 완료하려면 **Create Duplicate**(중복 생성)를 클릭합니다.
11. 기본 세부 정보 업데이트:
 - 복제 후 새로 생성된 ETL을 엽니다.
 - 1단계: 기본 세부 정보 지정에서 임시 데이터 원본을 커넥터의 프로덕션 데이터 원본으로 바꿉니다.
12. ETL 구성 저장: 저장을 클릭하여 변경 사항을 완료합니다.
13. 데이터 소스 실행: 임포트 일정에서 커넥터의 데이터 소스를 실행하여 실제 장치 데이터를 수집합니다.

결과: 시스템은 Telco 일반 스키마 설정을 기반으로 새 ETL을 생성하고 지정된 커넥터 데이터 소스의 페이로드를 처리합니다. 임포트한 데이터는 구성 항목(CI)으로 변환되고 예상되는 관계와 구조로 ServiceNow CMDB에 삽입됩니다.

커넥터의 데이터 소스 업데이트

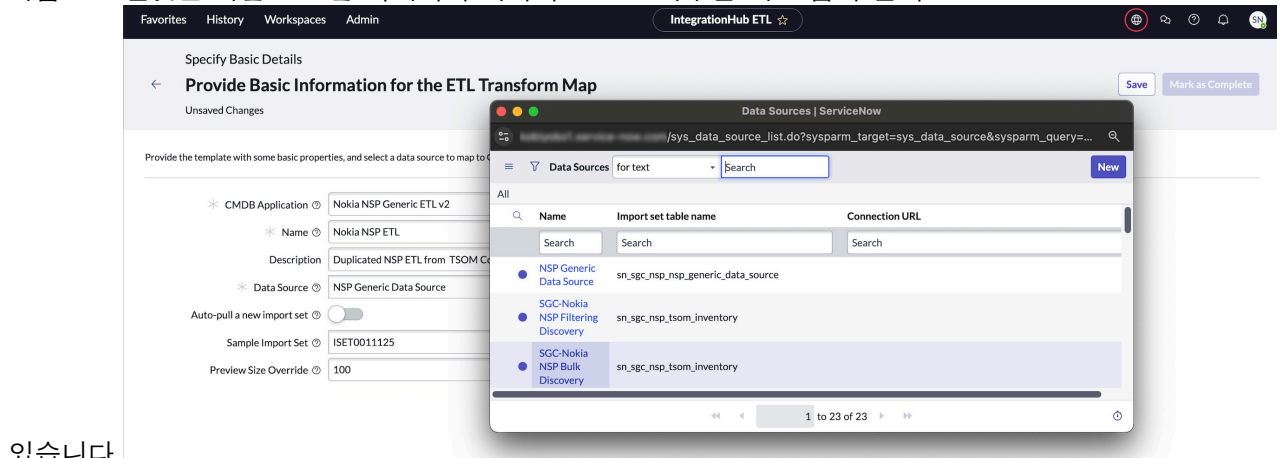
복제된 ETL을 특정 커넥터의 유효한 데이터 소스에 연결합니다.

시작하기 전에

필요한 역할: 관리자

이 태스크 정보

다음 스크린샷은 기본 소스를 커넥터의 데이터 소스로 바꾸는 데 도움이 될 수



있습니다.

프로시저

1. 다음으로 이동 모두 > 통합 허브 **ETL**.
2. 기본 세부 정보 지정에서 기본 데이터 원본을 커넥터의 데이터 원본으로 바꿉니다.

3. 저장을 선택합니다.

4. 임포트 일정을 열고 데이터 소스를 실행합니다.

시스템은 선택한 설정에 따라 새 변환 맵을 생성하고 페이로드에서 생성된 CI를 생성합니다.

관련 정보

[기존 ETL을 사용하여 새 서비스 그래프 커넥터 배포](#)

기존 ETL을 사용하여 새 서비스 그래프 커넥터 배포

새 SGC(서비스 그래프 커넥터)를 기존 ETL과 연결합니다.

시작하기 전에

필요한 역할: 관리자

필요한 서비스 그래프 커넥터 플러그인이 활성화되어 있는지 확인합니다.

프로시저

1. 다음으로 이동 모두 > 통합 허브 **ETL**.

2. 중복된 ETL 구성을 엽니다.

3. 서비스 그래프 커넥터의 데이터 소스를 선택하고 저장을 클릭합니다.

4. ETL을 실행하여 새 SGC에서 데이터를 변환하고 로드하려면 일정 임포트를 선택합니다.

5. 실행 상태를 모니터링합니다.

결과

CMDB 맵 또는 목록 뷰를 사용하여 CMDB에서 CI 기록과 관계가 생성되는지 확인합니다.

중복 ETL에 대한 **TNI** 엔터티 지원 확장

TNI(통신 네트워크 인벤토리)가 활성화된 경우 중복 ETL로 생성된 모든 CI(구성 항목)에는 해당 TNI 엔터티가 있어야 합니다.

시작하기 전에

필요한 역할: 관리자

- 인스턴스에 TNI가 설치되어 있고 활성 상태인지 확인합니다.

- 대상 애플리케이션 범위로 통신 디스커버리 빌더 프레임워크 ETL의 복제를 완료합니다.

프로시저

1. 다음으로 이동 모두 > 통합 허브 **ETL**.

2. 중복된 ETL과 연결된 CMDB 애플리케이션을 선택합니다.

새 페이지의 CMDB Integration Studio 애플리케이션

3. ETL 구성 페이지에서 스크립트 이전 실행 옵션을 선택합니다.

4. 기본 스크립트를 다음으로 바꿉니다.

```
(function(input, runId) {
    new
    sn_tsom_core.TelcoGenericMappingHelper().checkAndUpdatePayloadForTni(input);
})(input, runId);
```

5. 업데이트를 클릭하여 변경 사항을 저장합니다.

복제된 ETL은 검색된 CI에 대한 TNI 엔터티를 연결합니다.

모든 서비스 그래프 커넥터를 지원하기 위한 표준화된 **JSON** 공통 데이터 세트

통합 스키마와 재사용 가능한 ETL 논리를 사용하는 표준화된 서비스 그래프 커넥터를 지원하려면 TSOM 아키텍처를 사용합니다. 이를 통해 새 커넥터의 온보딩 시간이 단축되고 CMDB와의 통합이 단순화됩니다.

1. 공통 JSON 스키마 정의

모든 서비스 그래프 커넥터의 출력을 TNI 스키마를 준수하는 단일 JSON 형식에 맞게 표준화합니다.

다음 사항을 확인하십시오.

1. 수집기 또는 어댑터에서 변환 논리를 구현하여 데이터를 공통 스키마로 출력합니다.

2. 확인:

- 슬롯 상의 슬롯 또는 카드 온 카드 계층 구조는 제외됩니다.
- 논리적 인터페이스는 `virtual=true`로 명확하게 표시됩니다.
- 장비 유형은 모델-클래스 매핑에 맞춰집니다.

i 주:

스키마는 사용 가능한 경우 TNI 변경 사항에 대한 런타임 적응성을 지원해야 합니다.

다음은 모든 TSOM 서비스 그래프 커넥터를 지원하는 JSON 스키마 공통 데이터 세트입니다.

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "Telco Generic Schema",
  "version": "2.0.1",
  "oneOf": [
    {
      "type": "object",
      "properties": {
        "logical_composites": {
          "type": "array",
          "items": {
            "$ref": "#/$defs/logical_composite"
          }
        }
      },
      "required": [
        "logical_composites"
      ],
      "additionalProperties": false
    },
    {
      "type": "object",
      "properties": {
        "devices": {
          "type": "array",
          "items": {
            "$ref": "#/$defs/device"
          }
        }
      },
      "required": [
        "devices"
      ],
    }
  ]
}
```

```

    "additionalProperties": false
  },
  {
    "type": "object",
    "properties": {
      "logical_connections": {
        "type": "array",
        "items": {
          "$ref": "#/$defs/logical_connection"
        }
      }
    }
  },
  "required": [
    "logical_connections"
  ],
  "additionalProperties": false
},
{
  "type": "object",
  "properties": {
    "port_relations": {
      "type": "array",
      "items": {
        "$ref": "#/$defs/port_relation"
      }
    }
  }
},
"required": [
  "port_relations"
],
"additionalProperties": false
},
{
  "type": "object",
  "properties": {
    "logical_connection_relations": {
      "type": "array",
      "items": {
        "$ref": "#/$defs/logical_connection_relation"
      }
    }
  }
},
"required": [
  "logical_connection_relations"
],
"additionalProperties": false
},
{
  "type": "object",
  "properties": {
    "numbers": {
      "type": "array",
      "items": {
        "$ref": "#/$defs/number"
      }
    }
  }
},

```

```

    "required": [
      "numbers"
    ],
    "additionalProperties": false
  },
  {
    "type": "object",
    "properties": {
      "topologies": {
        "type": "array",
        "items": {
          "$ref": "#/$defs/network_topology"
        }
      }
    },
    "required": [
      "topologies"
    ],
    "additionalProperties": false
  },
  {
    "type": "object",
    "properties": {
      "topology_relations": {
        "type": "array",
        "items": {
          "$ref": "#/$defs/network_topology_relation"
        }
      }
    },
    "required": [
      "topology_relations"
    ],
    "additionalProperties": false
  }
],
"$defs": {
  "keyRef": {
    "type": "object",
    "properties": { "key": { "type": "string" } },
    "required": [ "key" ],
    "additionalProperties": false
  },
  "optionalKeyRef": { "type": [ "object", "null" ],
    "properties": {
      "key": { "type": "string" }
    },
    "additionalProperties": false
  },
  "value": {
    "type": "object",
    "properties": {
      "from": { "type": [ "integer" ], "default": 0, "minimum": 0 },
      "to": { "type": [ "integer" ], "default": 0, "minimum": 0 }
    },
    "required": [ "from", "to" ],

```

```

    "additionalProperties": false
  },
  "logical_composite": {
    "type": "object",
    "properties": {
      "class": { "type": "string", "enum": [ "logical_composite" ] },
      "key": { "type": "string" },
      "name": { "type": [ "string", "null" ] },
      "description": { "type": [ "string", "null" ] },
      "devices": { "type": "array", "items": { "$ref": "#/$defs/keyRef" } },
      "power_units": { "type": "array", "items": { "$ref": "#/$defs/pdu" } },
      "fan_shelves": { "type": "array", "items": { "$ref": "#/$defs/fan_shelf" } } },
    "required": [ "key", "name" ]
  },
  "pdu": {
    "type": "object",
    "properties": {
      "class": { "type": "string", "enum": [ "pdu" ] },
      "key": { "type": "string" },
      "name": { "type": [ "string", "null" ] },
      "description": { "type": [ "string", "null" ] },
      "model_name": { "type": [ "string", "null" ] },
      "model_number": { "type": [ "string", "null" ] },
      "unit_position": { "type": [ "integer", "null" ], "minimum": 1 },
      "slots": { "type": "array", "items": { "$ref": "#/$defs/slot" } } },
    "required": [ "key", "name" ]
  },
  "fan_shelf": {
    "type": "object",
    "properties": {
      "class": { "type": "string", "enum": [ "fan_shelf" ] },
      "key": { "type": "string" },
      "name": { "type": [ "string", "null" ] },
      "description": { "type": [ "string", "null" ] },
      "slots": { "type": "array", "items": { "$ref": "#/$defs/slot" } } },
    "required": [ "key", "name" ]
  },
  "device": {
    "type": "object",
    "properties": {
      "class": { "type": "string", "enum": [ "device" ] },
      "key": { "type": "string" },
      "name": { "type": [ "string", "null" ] },
      "description": { "type": [ "string", "null" ] },
      "ip_address": { "type": [ "string", "null" ] },
      "mac_address": { "type": [ "string", "null" ] },
      "serial_number": { "type": [ "string", "null" ] },
      "model_name": { "type": [ "string", "null" ] },
      "model_number": { "type": [ "string", "null" ] },
      "manufacturer": { "type": [ "string", "null" ] },
      "firmware_version": { "type": [ "string", "null" ] },
      "slots": { "type": "array", "items": { "$ref": "#/$defs/slot" } },
      "ports": { "type": "array", "items": { "$ref": "#/$defs/port" } } },
    "required": [ "key", "name", "serial_number" ]
  },

```

```

"slot": {
  "type": "object",
  "properties": {
    "class": { "type": "string", "enum": [ "slot" ] },
    "key": { "type": "string" },
    "name": { "type": [ "string", "null" ] },
    "description": { "type": [ "string", "null" ] },
    "model_name": { "type": [ "string", "null" ] },
    "model_number": { "type": [ "string", "null" ] },
    "manufacturer": { "type": [ "string", "null" ] },
    "unit_position": { "type": [ "integer", "null" ], "minimum": 1 },
    "cards": { "type": "array", "items": { "$ref": "#/$defs/card" } }
  },
  "required": [ "key", "name" ]
},
"card": {
  "type": "object",
  "properties": {
    "class": { "type": "string", "enum": [ "card" ] },
    "key": { "type": "string" },
    "name": { "type": [ "string", "null" ] },
    "description": { "type": [ "string", "null" ] },
    "mac_address": { "type": [ "string", "null" ] },
    "serial_number": { "type": [ "string", "null" ] },
    "firmware_version": { "type": [ "string", "null" ] },
    "model_name": { "type": [ "string", "null" ] },
    "model_number": { "type": [ "string", "null" ] },
    "manufacturer": { "type": [ "string", "null" ] },
    "unit_position": { "type": [ "integer", "null" ], "minimum": 1 },
    "slots": { "type": "array", "items": { "$ref": "#/$defs/slot" } },
    "ports": { "type": "array", "items": { "$ref": "#/$defs/port" } }
  },
  "required": [ "key", "name" ]
},
"port": {
  "type": "object",
  "properties": {
    "class": { "type": "string", "enum": [ "port" ] },
    "key": { "type": "string" },
    "name": { "type": [ "string", "null" ] },
    "description": { "type": [ "string", "null" ] },
    "model_name": { "type": [ "string", "null" ] },
    "ip_address": { "type": [ "string", "null" ] },
    "virtual": { "type": "boolean", "default": false },
    "unit_position": { "type": [ "integer", "null" ], "minimum": 1 },
    "bandwidth_name": { "type": [ "string", "null" ] },
    "bandwidth_group": { "type": [ "string", "null" ] },
    "bandwidth": { "type": [ "integer", "null" ], "minimum": 0 },
    "mtu_size": { "type": [ "integer", "null" ], "minimum": 0 }
  },
  "required": [ "key", "name" ]
},
"number": {
  "type": "object",
  "properties": {
    "class": { "type": "string", "enum": [ "number" ] },
    "key": { "type": "string" },

```

```

    "name": { "type": [ "string", "null" ] },
    "related_ci_type": { "type": "string", "enum": [ "Network Interface", "Physical Connection",
"Logical Connection", "Equipment", "Topology" ] },
    "related_ci": { "$ref": "#/$defs/keyRef" },
    "type": { "type": "string", "enum": [ "vlan_range", "vlan_subrange", "vlan", "lag_range",
"lag" ] },
    "vlan_type": { "type": "string", "enum": [ "inner", "outer" ] },
    "value": { "$ref": "#/$defs/value" }
  },
  "required": [ "key", "name", "type", "related_ci_type", "related_ci", "value" ] },
  "logical_connection": {
    "type": "object",
    "properties": {
      "class": { "type": "string", "enum": [ "logical_connection" ] },
      "key": { "type": "string" },
      "name": { "type": [ "string", "null" ] },
      "description": { "type": [ "string", "null" ] },
      "model_name": { "type": [ "string", "null" ] },
      "bandwidth_group": { "type": [ "string", "null" ] },
      "bandwidth_name_a_to_z": { "type": [ "string", "null" ] },
      "bandwidth_name_z_to_a": { "type": [ "string", "null" ] },
      "bandwidth_a_to_z": { "type": [ "integer", "null" ], "minimum": 0 },
      "bandwidth_z_to_a": { "type": [ "integer", "null" ], "minimum": 0 },
      "equipment_a": { "$ref": "#/$defs/optionalKeyRef" },
      "equipment_z": { "$ref": "#/$defs/optionalKeyRef" },
      "port_a": { "$ref": "#/$defs/keyRef" },
      "port_z": { "$ref": "#/$defs/keyRef" }
    },
    "required": [ "key", "name", "equipment_a", "equipment_z", "port_a", "port_z" ]
  },
  "port_relation": {
    "type": "object",
    "properties": {
      "class": { "type": "string", "enum": [ "port_relation" ] },
      "parent": { "$ref": "#/$defs/keyRef" },
      "child": { "$ref": "#/$defs/keyRef" }
    },
    "required": [ "parent", "child" ]
  },
  "logical_connection_relation": {
    "type": "object",
    "properties": {
      "class": { "type": "string", "enum": [ "logical_connection_relation" ] },
      "sequence": { "type": [ "integer", "null" ], "minimum": 1 },
      "route": { "type": [ "integer", "null" ], "minimum": 1 },
      "parent": { "$ref": "#/$defs/keyRef" },
      "child": { "$ref": "#/$defs/keyRef" }
    },
    "required": [ "parent", "child" ]
  },
  "network_topology": {
    "type": "object",
    "properties": {
      "class": { "type": "string", "enum": [ "network_topology" ] },
      "key": { "type": "string" },
      "name": { "type": "string" },
      "model_name": { "type": [ "string", "null" ] },

```

```

    "devices": { "type": "array", "items": { "key": "#/$defs/optionalKeyRef" } },
    "logical_connections": { "type": "array", "items": { "key": "#/$defs/optionalKeyRef" } }
  },
  "required": [ "key", "name", "devices", "logical_connections" ]
}
}
}

```

2. ETL에서 연결 분리

장치 상호작용 논리를 변환 논리(ETL 수집)와 분리하여 유연성을 보장합니다. 사용자는 ETL 논리와 독립적으로 자체 수집기를 개발할 수 있습니다.

다음 사항을 확인하십시오.

1. 통합 스키마로의 연결 및 데이터 변환에만 집중하도록 수집기를 설계합니다.
2. EMS/NMS 어댑터(Atrinet과 같은 타사 어댑터 포함)를 사용하거나 재사용합니다.
3. 표준화된 데이터를 ServiceNow 임포트 세트로 푸시합니다.

3. CMDB 업데이트에 대한 일반 ETL 구성

재사용 가능한 단일 ETL을 사용하여 표준화된 모든 임포트 세트를 처리하고 CMDB를 정확하게 업데이트합니다.

다음 사항을 확인하십시오.

1. 임포트 세트에 표준화된 JSON 형식의 데이터가 포함되어 있는지 확인합니다.
2. TSOM 일반 ETL을 사용하여 다음을 수행할 수 있습니다.
 - 엔터티 유형(예: 슬롯, 카드, 논리적 인터페이스)을 탐지합니다.
 - 모델 매핑 논리에 따라 각 엔터티를 올바른 CI 클래스에 매핑합니다.
 - 해당하는 경우 인벤토리 범주를 포함한 필드를 채웁니다.

특별 처리:

- 논리적 인터페이스 → 포트 테이블(virtual=true)
- 논리적 연결 → 논리적 연결 테이블
- PDU 테이블 → PDU 카드
- 장비 → 모델 기반 CI 클래스(여러 개가 있는 경우 최신 매핑)

4. 멀티 새시 및 복합 장치에 대한 지원 구성

올바른 CMDB 관계를 가진 멀티 새시 라우터와 같은 복잡한 장치를 모델링합니다.

다음 사항을 확인하십시오.

1. 논리 복합 구성을 사용하여 라우터 + PDU와 같은 그룹화된 엔터티를 나타냅니다.
2. 해당 계층 구조에서 개별 구성요소(팬, 관리, 슬롯)를 정의합니다.
3. TNI 모델링 지침에 따라 각 엔터티를 매핑합니다.

예: 7750-2s 멀티 새시 디바이스의 경우:

- 논리 복합 →에는 라우터와 PDU가 포함되어 있습니다.
- PDU →에는 슬롯 → 카드 → 하위 슬롯이 포함되어 있습니다.

5. 설치에 따라 TNI 엔터티 생성 활성화

불필요한 기록 생성 없이 TNI 표준과의 일관성을 보장합니다.

다음 사항을 확인하십시오.

- TNI가 설치된 경우:
 - 검색된 각 CI에 대해 TNI 엔터티를 자동으로 생성합니다.
 - 인벤토리 범주를 적절하게 설정합니다(예: "논리적 연결", "인터페이스").
- TNI가 설치되지 않은 경우: TNI 엔터티 생성을 건너뛰어 고아 기록 또는 잘못된 기록을 방지합니다.

Nokia Altiplano 서비스 그래프 커넥터 구성

실제 및 논리적 인벤토리 데이터를 Nokia Altiplano 액세스 네트워크 SDN 컨트롤러에서 ServiceNow 구성 관리 데이터베이스(CMDB)로 임포트하도록 Nokia Altiplano 서비스 그래프 커넥터를 구성합니다.

이 통합은 REST API(MID 서버를 통해)를 사용하여 CMDB가 TM 포럼 기반 데이터 모델에 부합하는 정확한 최신 통신 인벤토리를 반영하도록 합니다.

- i** 주:
 - 이 커넥터를 사용하려면 유효한 Telecommunications Service Operations Management 구독이 필요합니다.

필요 플러그인

플러그인	ID
통신 서비스 운영 코어	sn_tsom_core
패턴 디자이너	com.snc.pattern.designer
Discovery Core(검색과 함께 자동으로 설치됨)	com.snc.discovery.core
ITOM 디스커버리 라이선스	com.snc.itom.discovery.license
ITOM 라이선스	com.snc.itom.license
통합 허브 ETL(개발 환경에만 해당)	sn_int_studio

- i** 주:
 - 외부 요구 사항:
 - REST 노스바운드 API에 액세스할 수 있는 실행 중인 Nokia Altiplano 컨트롤러 인스턴스입니다.
 - Altiplano 인스턴스에 안전하게 연결되는 MID 서버.

구성 작업 개요

다음 섹션은 Nokia Altiplano 탐색 창에서 사용할 수 있습니다. 사후 안내 설정의 경우 다음 테이블을 사용하거나 수동 구성을 수행합니다.

섹션	설명
설정	MID 서버를 구성하고, Altiplano 연결을 정의하고, импорт 일정을 예약합니다.
데이터 소스	대량 및 필터링된 검색을 위해 미리 정의된 데이터 소스(SGC-Nokia Altiplano 대량 검색, SGC-Nokia Altiplano 필터링 검색). 필요한 경우 병렬 로드를 활성화합니다. 병렬 로드에 대한 자세한 내용은 문서를 참조하십시오 Nokia Altiplano에 대한 동시 импорт 및 병렬 로딩 구성 .
일정 импорт	각 Altiplano 연결 별칭에 대한 일정을 관리합니다. 수동으로 또는 정의된 간격에 따라 작업을 실행합니다.
연결 및 자격 증명 별칭	각 Altiplano 인스턴스에 대한 별칭을 정의합니다. 연결 메타데이터 및 자격 증명을 저장합니다.
연결	URL, 선택한 MID 서버, 자격 증명 참조, 연결 별칭 참조와 같은 Altiplano 인스턴스 상세 정보를 정의합니다.
자격 증명	기본 인증을 사용하여 Altiplano 자격 증명을 생성합니다.
필터	필터링된 검색에 사용되는 필터링 매개변수(예: 장치 IP 또는 이름별)를 구성합니다.
속성	커넥터별 속성을 사용하여 시스템 동작을 수정합니다. 자세한 내용은 Nokia Altiplano와 함께 설치되는 시스템 구성요소 문서를 참조하십시오.

지원되는 **Nokia Altiplano** 버전

- 지원되는 최소 버전: 24.6
- 지원 OLT: Lightspan MF-2

i 주:

자세한 내용은 를 참조하십시오 [CMDB에서 Nokia Altiplano CI 및 관계 매핑](#).

안내 설정에 액세스

안내 설정을 사용하여 구성 프로세스를 단순화합니다. 이 설정은 통합을 빠르고 정확하게 완료하는 데 도움이 되는 일련의 단계를 구성합니다. 안내 설정에 액세스하려면 다음을 수행합니다.

1. 다음으로 이동 모두 > 서비스 그래프 커넥터 > **Nokia Altiplano** > 설정.
2. 작업 순서에 따라 MID 서버 설정, Nokia Altiplano 연결, 필터 및 импорт 일정을 구성합니다. 자세한 내용은 [Nokia Altiplano 설치 및 구성](#) 문서를 참조하십시오.

관련 정보

[Nokia Altiplano 설치 및 구성](#)

[여러 Nokia Altiplano 인스턴스 설정](#)

[Nokia Altiplano에 대한 동시 импорт 및 병렬 로딩 구성](#)

[CMDB에서 Nokia Altiplano CI 및 관계 매핑](#)

[Nokia Altiplano 서비스 그래프 커넥터에 대한 импорт 실행 및 확인](#)

[Nokia Altiplano와 함께 설치되는 시스템 구성요소](#)

[Nokia Altiplano를 통한 통신 검색](#)

Nokia Altiplano 설치 및 구성

데모 데이터(옵선), 연결 설정, 데이터 수집 일정 등을 비롯하여 Nokia Altiplano용 서비스 그래프 커넥터를 설치하고 구성하는 방법에 대해 알아봅니다.

시작하기 전에

필요한 역할: 관리자

이 태스크 정보

Nokia Altiplano 애플리케이션 범위에 대한 서비스 그래프 커넥터에서 운영 중인지 확인합니다. Nokia Altiplano용 서비스 그래프 커넥터 범위로 전환하려면 다음을 수행합니다.

1. 헤더에서 애플리케이션 선택기()를 선택합니다.
2. **Nokia Altiplano**를 검색합니다.
3. 목록에서 Nokia Altiplano용 서비스 그래프 커넥터를 선택합니다.

프로시저

1. 다음으로 이동 모두 > 서비스 그래프 커넥터 > **Nokia Altiplano** > 설정.
2. 시작하기 페이지에서 시작하기를 선택합니다.
안내 설정 홈페이지가 새 탭에서 열립니다.
3. 옵선: 사용자 환경에 따라 생성하고 구성 MID 서버 하거나 건너뛩니다.
 - a. 구성을 선택하여 프로세스를 완료합니다.
 - b. 화면의 지시에 따라 다운로드하고 설치합니다.
 - c. 완료되면 완료로 표시를 선택합니다.
-  주:

MID 서버를 설치 및 구성하는 방법에 대한 자세한 내용은 문서를 참조하십시오 [Configuring MID Server](#).

 - d. 확인 단계를 구성하거나 건너뛩니다.
 - e. 확인되면 완료로 표시를 선택합니다.
모든 MID 서버 단계가 완료되면 연결 구성으로 진행합니다.
4. 연결 구성을 사용하여 Nokia Altiplano 인스턴스에 대한 연결 별칭, 자격 증명 및 HTTP 연결을 생성합니다
 - a. 시작하기를 선택합니다.
 - b. 연결 및 자격 증명에 대한 별칭을 생성하고 구성합니다.
 - i. 구성을 선택합니다.
 - ii. 이름 필드에 별칭 이름을 지정합니다.
 - iii. 나머지 필드는 기본값으로 두고 제출 을 선택한 다음 완료로 표시를 선택합니다.



주:

이렇게 하면 직접 연결이 아닌 이름으로 연결을 사용할 수 있으므로 수집기가 CMDB에서 모든 활성 별칭을 추출하고 바인딩된 HTTP 연결에서 데이터 수집을 시작할 수 있습니다.

c. 구성을 선택하여 Nokia Altiplano 컨트롤러에 액세스하기 위한 자격 증명을 생성합니다.

i. 이름 필드에 별칭 이름을 지정합니다.

ii. 사용자 이름 필드에 Nokia Altiplano 인스턴스 사용자 이름을 지정합니다.

iii. 암호 필드에 Nokia Altiplano 인스턴스 암호를 지정합니다.

i 주 :

Nokia Altiplano 인스턴스에서 사용하는 인증 방식에 따라 다른 인증 필드가 필요할 수 있습니다. 기본적으로 (안내 설정의 일부로)를 사용합니다 [Basic authentication credentials](#) .

iv. 나머지 필드는 기본값으로 두고 제출 을 선택한 다음 완료로 표시를 선택합니다.

d. 구성을 선택하여 HTTP 연결 생성

i. 이름 필드에 연결 이름을 지정합니다.

ii. 이전에 생성한 자격 증명 및 연결 별칭 을 선택합니다.

iii. Nokia Altiplano에 대한 연결 **URL** 을 지정합니다.

iv. **MID** 서버 사용을 선택하고 다음 중 하나를 선택합니다.

- 자동 선택
- 특정 MID 서버
- 특정 MID 클러스터

v. 나머지 필드는 기본값으로 두고 제출 을 선택한 다음 완료로 표시를 선택합니다.

5. 시작을 클릭한 다음 구성을 클릭하여 대량 검색 또는 필터링된 검색에 대한 데이터 수집 일정을 구성합니다.

6. 제출을 선택한 다음, 완료로 표시를 선택합니다.

결과

구성된 연결, 별칭, 자격 증명 및 임포트 일정은 탐색에서도 액세스할 수 있습니다. 다음으로 이동 모두 > 서비스 그래프 커넥터 > **Nokia Altiplano**.

다음 스냅샷은 Nokia Altiplano 서비스 그래프 커넥터의 구성 설정을 확인하는 데 도움이 됩니다.

100% Complete

Nokia Altiplano Integration with CMDB Filter Show all ▼

100%

Status: Completed

Edit

MID Server Skip

The MID Server runs as a Windows service or a UNIX daemon to facilitate communication and the movement of data between a ServiceNow instance and external applications, data sources, and services. Complete the activities in this category to create a user for the MID Server, download the installation package, and validate the MID Server after installation.

After completing each task, mark the step as complete or skip not relevant tasks.

3 / 3 Tasks completed

- ✓ Create MID User
- ✓ Download & Install MID
- ✓ Validate MID

100%

Status: Completed

Edit

Configure Connectivity

Configuration of information necessary for the Nokia Altiplano Collector to work properly.

After completing each task, mark the step as complete or skip not relevant tasks.

3 / 3 Tasks completed

- ✓ Create Connection Aliases
- ✓ Create Credentials
- ✓ Create HTTP Connection

100%

Status: Completed

Edit

Configure Data Collection schedule Skip

Configure Data collection schedule per Connection and Data Source for Nokia Altiplano inventory data integration. Use Bulk discovery data source for all devices in connection alias. Use Filtering discovery data source for specific OLT filter criteria for each connection alias.

After completing each task, mark the step as complete or skip not relevant tasks.

2 / 2 Tasks completed

- ✓ Schedule Data Collection for Bulk Discovery
- ✓ Schedule Data Collection for Filtered Discovery

관련 정보

[여러 Nokia Altiplano 인스턴스 설정](#)

[Nokia Altiplano에 대한 동시 임포트 및 병렬 로딩 구성](#)

[CMDB에서 Nokia Altiplano CI 및 관계 매핑](#)

[Nokia Altiplano와 함께 설치되는 시스템 구성요소](#)

여러 **Nokia Altiplano** 인스턴스 설정

단일 ServiceNow 환경 내에서 여러 Nokia Altiplano 인스턴스를 구성하고 관리하는 방법을 알아봅니다. 이를 통해 관리자와 통합자는 독립적인 연결 별칭을 생성하고 특정 필터링 또는 빈도 요구에 맞게 사용자 지정된 일정을 임포트할 수 있습니다.

시작하기 전에

필요한 역할: 관리자

이 태스크 정보

다음에 있는지 확인합니다.

- NOKIA Altiplano용 서비스 그래프 커넥터 애플리케이션 범위에서 운영하고 있습니다.
- Nokia Altiplano용 서비스 그래프 커넥터를 설치했습니다.
- 초기 Altiplano 인스턴스에 대한 안내 설정을 완료했습니다.
- MID 서버가 설정되고 검증되었습니다.

추가 Nokia Altiplano 인스턴스를 구성하거나 다른 연결 별칭 및 임포트 일정으로 동일한 Altiplano 인스턴스를 재사용할 수 있습니다. 새 Altiplano 인스턴스를 추가하려면 안내 설정을 다시 실행하여 새 연결 별칭을 구성합니다. 각 새 별칭에 대한 설정의 모든 연결 스테이지를 진행합니다.

프로시저

1. 다음으로 이동 모두 > 서비스 그래프 커넥터 > **Nokia Altiplano** > 설정.
2. 시작하기 페이지에서 시작하기를 선택합니다.
Configured Connectivity(구성된 연결) 섹션의 모든 단계를 반복합니다. Nokia Altiplano의 새 인스턴스에 대한 구성 항목을 작성합니다. 자세한 내용은 [Nokia Altiplano 설치 및 구성](#) 문서를 참조하십시오.
3. 연결 별칭, 자격 증명 및 HTTP 연결을 생성하여 연결을 구성합니다.

- a. 연결 구성 섹션에서 시작하기를 선택합니다.
 - b. 구성을 선택하여 고유한 별칭 이름(예: Altiplano_Prod_01)을 입력하여 새 연결 별칭을 만듭니다.
 - c. 제출을 선택하고 단계를 완료로 표시합니다.
4. 새 Altiplano 인스턴스에 대한 자격 증명을 구성합니다.
- a. Altiplano 컨트롤러의 사용자 이름과 암호를 지정합니다.
 - b. 단계를 제출하고 완료로 표시하십시오.
5. HTTP 연결을 구성합니다.
- a. 연결 이름을 제공합니다.
 - b. 새로 만든 자격 증명과 연결 별칭을 선택합니다.
 - c. Nokia Altiplano 인스턴스의 연결 URL을 입력합니다.
 - d. MID 서버 사용을 활성화하고 적절한 MID 옵션을 선택합니다.
 - e. 제출을 선택하고 단계를 완료로 표시합니다.
6. импорт 일정을 구성하여 대량 또는 필터링된 검색을 사용하여 데이터 임포트를 예약합니다.
- a. 필드에 내용을 입력합니다.
자세한 내용은 [Nokia Altiplano 설치 및 구성](#) 문서를 참조하십시오.
 - b. 연결 사용 필드에서 새 Altiplano 인스턴스를 선택합니다.

7. 제출을 선택합니다.

결과

새 인스턴스 설정이 성공적으로 완료되었는지 확인하기 위해 구성을 검증할 수 있습니다.

- 모든 > 서비스 그래프 커넥터 > Nokia Altiplano > 연결 별칭으로 이동하고 새 별칭이 나열되어 있는지 확인합니다.
- 모든 > 서비스 그래프 커넥터 > Nokia Altiplano > импорт 일정으로 이동하여 새 импорт 일정이 나열되는지 확인하고 각 별칭에 해당 작업이 있는지 확인합니다.

동일한 Nokia Altiplano 인스턴스를 가리키며 여러 연결 별칭을 구성할 수 있습니다. 이러한 유연성을 통해 다양한 빈도로 임포트를 실행하고 각 별칭에 서로 다른 필터를 적용할 수 있습니다.

예: Altiplano_Weekly_OLT_10.10.10.*: 매주 실행되며 IP가 10.10.10.* 범위에 있는 OLT를 필터링합니다. Altiplano_Daily_Prod: 매일 실행되며 이름에 "prod_olt"가 포함된 OLT를 필터링합니다.

관련 정보

[Nokia Altiplano에 대한 동시 импорт 및 병렬 로딩 구성](#)

[CMDB에서 Nokia Altiplano CI 및 관계 매핑](#)

[Nokia Altiplano와 함께 설치되는 시스템 구성요소](#)

Nokia Altiplano에 대한 동시 импорт 및 병렬 로딩 구성

동시 임포트를 구성하고 병렬 로드를 사용하도록 설정하여 Nokia Altiplano에서 대규모 데이터 임포트의 효율성을 개선합니다. 이 구성을 사용하면 플랫폼에서 여러 데이터 임포트 및 변환 작업을 동시에 실행할 수 있습니다.

시작하기 전에
필요한 역할: 관리자

이 태스크 정보

동시 가져오기 및 병렬 로드는 큰 데이터 세트를 더 작은 파티션으로 나누어 여러 변환 및 수집 작업을 병렬로 실행할 수 있도록 하여 성능을 향상시킵니다. 이 설정은 대량 또는 필터링된 검색 중에 대량의 장비, 논리적 연결 및 네트워크 토폴로지 데이터를 임포트할 때 유용합니다.

다음 지침을 사용하여 성능을 최적화합니다.

- 변환이 느린 경우: 임포트 일정에서 동시 임포트를 사용하도록 설정합니다.
- 데이터 수집이 느린 경우: 동시 임포트를 활성화하는 것 외에도 데이터 소스에서 병렬 로드를 구성하고 시스템 속성을 업데이트합니다.

i 주:

병렬 로드는 동시 임포트 외에도 작동합니다. 그것은 그것을 대체하지 않습니다.

프로시저

1. 다음으로 이동 모두 > 서비스 그래프 커넥터 > **Nokia Altiplano** > 임포트 일정.
2. 동시 임포트 또는 동시 임포트 구성

관련 정보

[Nokia Altiplano 설치 및 구성](#)

[CMDB에서 Nokia Altiplano CI 및 관계 매핑](#)

[Nokia Altiplano와 함께 설치되는 시스템 구성요소](#)

CMDB에서 **Nokia Altiplano CI** 및 관계 매핑

Nokia Altiplano용 서비스 그래프 커넥터를 사용하여 검색된 실제 및 논리적 네트워크 자원을 CMDB의 통신 정렬 CI(구성 항목) 클래스에 매핑합니다. 이 커넥터는 일관된 서비스 모델링, 새시 수준 구성요소에 대한 가시성, 논리적 및 물리적 관계의 자동화를 지원합니다.

정확한 CI 분류 및 삽입을 확인하기 위해 커넥터는 강력한 변환 엔진(RTE)과 식별 및 조정 엔진(IRE)을 사용합니다.

커넥터는 장치 유형, 기능 및 새시 구조에 따라 통신 관련 모델을 사용하여 검색된 CI를 분류하고 연관시킵니다. 이를 통해 벤더 간에 깔끔하고 표준화된 CMDB를 유지할 수 있습니다. Nokia Altiplano에서 검색된 모델 이름은 슬롯 및 하위 슬롯 구성요소에 대한 ServiceNow 표준 모델 식별자와 범주로 자동 변환됩니다.

CI 매핑 및 관계

다음 테이블에서는 Altiplano CI가 CMDB에 표시되는 방식과 물리적 및 논리적 계층에서 서로 어떤 관련이 있는지 설명합니다.

CMDB CI 매핑 및 관계(물리적 계층)

CI 유형	CMDB 테이블	설명 및 관계
OLT CI	cmdb_ci_optical_line_terminal	OLT 장치를 나타냅니다. 슬롯 CI와 논리적 네트워크 인터페이스 CI를 포함합니다.
ONU/ONT CI	cmdb_ci_optical_network_terminal 또는 cmdb_ci_optical_network_unit	ONU 또는 ONT 장치를 나타냅니다. 클래스는 ### ## sn_sgc_altiplano.onu_ci_class에

CMDB CI 매핑 및 관계(물리적 계층)

CI 유형	CMDB 테이블	설명 및 관계
		의해 결정됩니다. 네트워크 인터페이스 CI를 포함합니다.
슬롯 CI	cmdb_ci_container_slot	메인 쉘 슬롯을 나타냅니다. OLT CI에 포함됩니다. 인터페이스 카드 CI(예: LT/NT, PSU, 팬)를 포함합니다. 모델 변환은 데이터 소스에 적용됩니다. 자세한 내용은 위에서 언급한 슬롯 및 하위 슬롯 CI 테이블에 대한 모델 변환을 참조하십시오.
하위 슬롯 CI	cmdb_ci_container_subslot	인터페이스 카드 내의 하위 구성 요소를 나타냅니다(예: SFP용 케이징). LT/NT 카드에 포함되어 있습니다. 트랜시버 카드 CI를 포함합니다. 자세한 내용은 위에서 언급한 슬롯 및 하위 슬롯 CI 테이블에 대한 모델 변환을 참조하십시오.
인터페이스 카드 CI	cmdb_ci_interface_card	LT/NT 카드, 트랜시버 및 제어 장치를 나타냅니다. 하위 슬롯과 네트워크 인터페이스를 포함할 수 있습니다.
네트워크 인터페이스 CI	cmdb_ci_ni_interface	물리적(예: PON, 이더넷) 및 논리적(예: VLAN) 포트를 모두 나타냅니다. 인터페이스 카드 또는 ONU/ONT CI에 포함되어 있습니다. 논리 포트는 Members: :Member of를 사용하여 물리적 포트와 관련됩니다.
논리적 연결 CI	cmdb_ci_ni_logical_path	OLT와 ONU 간의 PON 또는 VLAN과 같은 논리적 경로를 나타냅니다. 네트워크 인터페이스 CI를 종료하는 포트 A 및 포트 Z 속성으로 정의됩니다. VLAN 경로는 PON 경로를 사용합니다.
IP 주소 CI	cmdb_ci_ip_address	OLT에 대해 검색된 IP 주소를 나타냅니다. 해당 OLT CI가 소유합니다.

주요 관계 예

- 포함 관계
 - OLT CI → 슬롯 CI를 포함합니다.
 - 슬롯 CI →에는 인터페이스 카드 CI가 포함되어 있습니다.
 - 인터페이스 카드 CI →에는 하위 슬롯 CI가 포함되어 있습니다.
 - 서브슬롯 CI → 트랜시버 포함 인터페이스 카드 CI
 - ONU/ONT CI → 네트워크 인터페이스 CI를 포함합니다.
- 인터페이스 관계:
 - 논리적 연결 - > 종료자 네트워크 인터페이스
 - 네트워크 인터페이스 - 네트워크 인터페이스의 > 구성원
- 논리 경로 관계:

- VLAN 경로(상위) → → PON 경로(하위)를 사용합니다.
- 논리 경로는 포트 A와 포트 Z를 통해 → 네트워크 인터페이스 C에서 종료→
- 소유권: IP 주소 OLT 장치가 소유한 CI →

지원되는 모델

1. 네트워크 장비 모델(sn_ent_nw_equipment_model)

- 지원되는 OLT는 Nokia Lightspan MF-2이며 기본 모델은 "Nokia MF-2"입니다.
- ONU/ONT 모델은 제조업체 + ONU/ONT입니다. 시스템 속성 sn_sgc_altiplano.onu_ci_class는 ONU 또는 ONT를 사용할지 여부를 정의합니다.
- 모델 테이블에서 모델을 찾을 수 없는 경우 CI에 새 모델이 생성됩니다. CI는 "네트워크 장비"로 생성됩니다.

2. 장비 홀더 모델: (sn_ent_nw_holder_model)

- 슬롯 모델: "트래픽 슬롯", "팬 슬롯", "전원 슬롯"
- 서브슬롯 모델: "SFP 서브슬롯"
- 사용된 모델 이름은 Altiplano 확장점(sn_sgc_altiplano. AltiplanoCustomizedModels)
- 모델 테이블에서 모델을 찾을 수 없는 경우 CI에 새 모델이 생성됩니다.

3. 네트워크 카드 모델(sn_ent_nw_card_model)

- 카드 모델은 Altiplano API에서 검색한 모델 이름, 제조업체, 모델 번호로 찾을 수 있습니다
- 모델 테이블에서 모델을 찾을 수 없는 경우 CI에 새 모델이 생성됩니다.

4. 네트워크 인터페이스 모델: (sn_ent_nw_interface_model).

- 이더넷 포트 모델은 네트워크 인터페이스 테이블(sn_ent_nw_interface_model)의 "포트 대역폭" 열에서 찾을 수 있습니다. 포트 CI의 포트 대역폭은 대역폭 테이블(대역폭)에서 검색된 포트 속도에 의해 위치합니다.
- PON 물리적 포트 모델: "PON 액세스 인터페이스", "PON 네트워크 인터페이스"
- 논리적 포트 모델: "ENET 인터페이스", "VLAN 인터페이스", "LAG 인터페이스", PON 논리적 인터페이스"
- 모델 테이블에서 모델을 찾을 수 없는 경우 "모델 ID"에 대한 참조는 비어 있는 상태로 유지됩니다.

5. 논리 네트워크 연결 모델(sn_ent_logical_nw_connection_model)

- PON 접근 경로
- VLAN 경로
- 모델 테이블에서 모델을 찾을 수 없는 경우 "모델 ID"에 대한 참조는 비어 있는 상태로 유지됩니다.

i 주:

- 커넥터가 검색된 장비를 제품 모델 테이블의 기존 모델과 일치시킬 수 없는 경우 CI는 기본적으로 ##### 로 생성됩니다.
- 데모 데이터가 설치되면 OLT, ONU, ONT, 슬롯, 하위 슬롯, 카드, 네트워크 인터페이스 및 논리적 연결에 대한 기본 모델이 생성됩니다.
- 확장점을 사용하여 장비 및 장비 홀더 모델 이름을 사용자 지정할 수 있습니다.

슬롯 및 하위 슬롯 CI에 대한 모델 변환

수집 중에 검색된 특정 모델 이름이 미리 정의된 CMDB 모델 식별자에 매핑되어 일관된 슬롯 범주화를 확인합니다. 변환 논리는 SGC 데이터 소스 스크립트에 포함되며 Nokia Altiplano 소스에 적용됩니다.

팬, 전원 및 트래픽 슬롯과 같은 슬롯 구성요소는 CMDB의 ## 범주에 매핑됩니다. SFP 케이지 또는 합성 하위 슬롯과 같은 하위 슬롯 구성 요소는 CMDB의 ## ## 범주에 매핑됩니다.

예: **Nokia Altiplano** 슬롯 및 하위 슬롯 모델 매핑

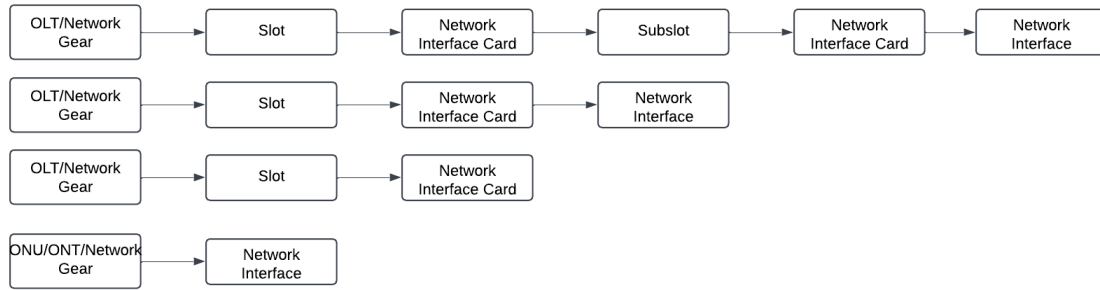
소스	검색된 모델 이름	대상 CMDB 모델 ID	모델 범주
알티플라노	슬롯 팬	팬 슬롯	슬롯
알티플라노	slot-lt	트래픽 슬롯	슬롯
알티플라노	농	SFP 하위 슬롯	하위 슬롯
알티플라노	슬롯-NT	트래픽 슬롯	슬롯
알티플라노	합성 nt-슬롯	SFP 하위 슬롯	하위 슬롯
알티플라노	slot-psu	전원 슬롯	슬롯

CI 관계 구조

다음 인포그래픽에서는 CI 관계를 설명합니다.

CIs Relationships

Contains::Contained by:



Owns::Owned by (Dependent Relationship)



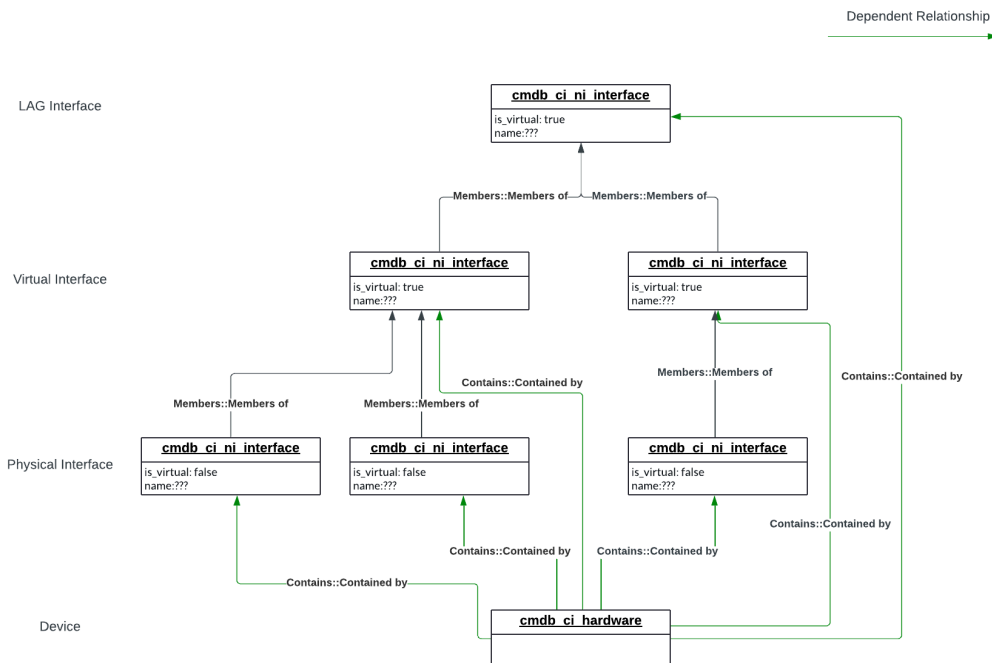
Members::Member of



Consumes::Consumed by

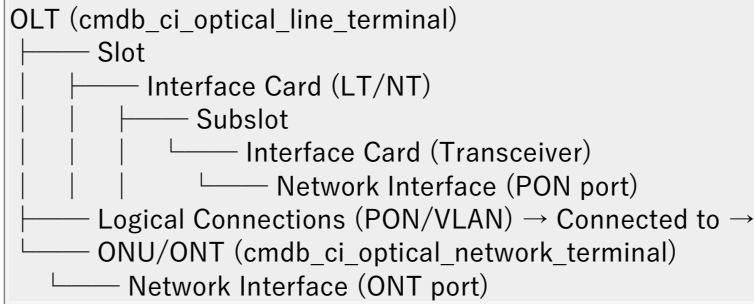


Terminates::Terminated by



예: OLT에서 ONU로의 구조

다음 구조를 사용하면 OLT에서 하드웨어 계층을 통해 연결된 ONU 및 관련 논리 경로까지 종단 간 추적이 가능합니다.



[store-future: BEGIN review]

Nokia NSP 서비스 그래프 커넥터 구성

실제 및 논리적 인벤토리 데이터를 Nokia NSP 액세스 네트워크 SDN 컨트롤러에서 ServiceNow 구성 관리 데이터베이스(CMDB)로 임포트하도록 Nokia NSP 서비스 그래프 커넥터를 구성합니다.

이 통합은 REST API(MID 서버를 통해)를 사용하여 CMDB가 TM 포럼 기반 데이터 모델에 부합하는 정확한 최신 통신 인벤토리를 반영하도록 합니다.

주:

이 커넥터를 사용하려면 유효한 Telecommunications Service Operations Management 구독이 필요합니다.

필요 플러그인

플러그인	ID
통신 서비스 운영 코어	sn_tsom_core
패턴 디자이너	com.snc.pattern.designer
Discovery Core(검색과 함께 자동으로 설치됨)	com.snc.discovery.core
ITOM 디스커버리 라이선스	com.snc.itom.discovery.license
ITOM 라이선스	com.snc.itom.license
통합 허브 ETL(개발 환경에만 해당)	sn_int_studio

주:

외부 요구 사항:

- REST 노스바운드 API에 액세스할 수 있는 실행 중인 Nokia NSP 컨트롤러 인스턴스입니다.
- NSP 인스턴스에 안전하게 연결되는 MID 서버.

구성 작업 개요

다음 섹션은 Nokia Altiplano 탐색 창에서 사용할 수 있습니다. 사후 안내 설정의 경우 다음 테이블을 사용하거나 수동 구성을 수행합니다.

섹션	설명
설정	MID 서버를 구성하고, Altiplano 연결을 정의하고, импорт 일정을 예약합니다.
데이터 소스	대량 및 필터링된 검색에 대해 미리 정의된 데이터 소스(SGC-Nokia Altiplano 대량 검색, SGC-Nokia NSP 필터링 검색). 필요한 경우 병렬 로드를 활성화합니다.
일정 импорт	각 NSP 연결 별칭에 대한 일정을 관리합니다. 수동으로 또는 정의된 간격에 따라 작업을 실행합니다.
연결 및 자격 증명 별칭	각 NSP 인스턴스에 대한 별칭을 정의합니다. 연결 메타데이터 및 자격 증명을 저장합니다.
연결	URL, 선택한 MID 서버, 자격 증명 참조 및 연결 별칭과 같은 NSP 인스턴스 상세 정보를 정의합니다.
자격 증명	기본 인증을 사용하여 NSP 자격 증명을 생성합니다.
필터	필터링된 검색에 사용되는 필터링 매개변수(예: 장치 IP 또는 이름별)를 구성합니다.
속성	커넥터별 속성을 사용하여 시스템 동작을 수정합니다. 자세한 내용은 Nokia NSP와 함께 설치되는 시스템 구성요소 문서를 참조하십시오.

지원되는 모델

- 네트워크 인터페이스 모델
- 제품 모델
- 논리적 연결 모델
- 네트워크 토폴로지 모델

장치 모델이 누락된 경우 CI는 ##### ##로 생성됩니다.

안내 설정에 액세스

안내 설정을 사용하여 구성 프로세스를 단순화합니다. 이 설정은 통합을 빠르고 정확하게 완료하는데 도움이 되는 일련의 단계를 구성합니다. 안내 설정에 액세스하려면 다음을 수행합니다.

1. 다음으로 이동 모두 > 서비스 그래프 커넥터 > **Nokia NSP** > 설정.
2. 작업 순서에 따라 MID 서버 설정, 데이터 소스, 자격 증명, 필터, импорт 일정을 구성합니다. 자세한 내용은 [Nokia NSP 설치 및 구성](#) 문서를 참조하십시오.

[End]

[store-future: BEGIN review]

Nokia NSP 설치 및 구성

옵션 데모 데이터, 연결 설정 및 데이터 수집 일정을 포함하여 Nokia NSP용 서비스 그래프 커넥터를 설치하고 구성하는 방법에 대해 알아보니다

시작하기 전에

필요한 역할: 관리자

이 태스크 정보

NOKIA NSP용 서비스 그래프 커넥터 애플리케이션 범위에서 운영 중인지 확인합니다. Nokia NSP용 서비스 그래프 커넥터 범위로 전환하려면 다음을 수행합니다.

1. 헤더에서 애플리케이션 선택기(🌐)를 클릭합니다.
2. **Nokia NSP**를 검색합니다.
3. 목록에서 Nokia NSP용 서비스 그래프 커넥터를 선택합니다.

프로시저

1. 다음으로 이동 모두 > 서비스 그래프 커넥터 > **Nokia NSP** > 설정.
2. 시작하기 페이지에서 시작하기를 선택합니다.
안내 설정 홈페이지가 새 탭에서 열립니다.
3. 옵션: 사용자 환경에 따라 생성 및 구성 MID 서버 또는 건너뛰기
 - a. 구성을 선택하여 프로세스를 완료합니다.
 - b. 화면의 지시에 따라 다운로드하고 설치합니다.
 - c. 완료되면 완료로 표시를 클릭합니다.

(Optional) MID 서버를 설치 및 구성하는 방법에 대한 자세한 내용은 문서를 참조하십시오 [Configuring MID Server](#).
 - d. 확인 단계를 구성하거나 건너뛵니다.
 - e. 확인되면 완료로 표시를 클릭합니다.
모든 MID 서버 단계가 완료되면 연결 구성으로 진행합니다.
4. 연결 구성을 사용하여 Nokia NSP 인스턴스에 대한 연결 별칭, 자격 증명 및 HTTP 연결 작성
 - a. 시작하기를 선택합니다.
 - b. 연결 및 자격 증명에 대한 별칭을 생성하고 구성합니다.
 - i. 구성을 선택합니다.
 - ii. 이름 필드에 별칭 이름을 지정합니다.
 - iii. 나머지 필드는 기본값으로 두고 제출 을 선택한 다음 완료로 표시를 선택합니다.



주:

이렇게 하면 직접 연결이 아닌 이름으로 연결을 사용할 수 있으므로 수집기가 CMDB에서 모든 활성 별칭을 추출하고 바인딩된 HTTP 연결에서 데이터 수집을 시작할 수 있습니다.

- c. 구성을 클릭하여 Nokia NSP 컨트롤러에 액세스하기 위한 자격 증명을 생성합니다.
 - i. 이름 필드에 별칭 이름을 지정합니다.
 - ii. 사용자 이름 필드에 Nokia NSP 인스턴스 사용자 이름을 지정합니다.
 - iii. 암호 필드에 Nokia NSP 인스턴스 암호를 지정합니다.

i 주:

Nokia NSP 인스턴스에 사용되는 인증 방법에 따라 다른 인증 필드가 필요할 수 있습니다. 기본적으로 (안내 설정의 일부로)를 사용합니다 [Basic authentication credentials](#) .

iv. 나머지 필드는 기본값으로 두고 제출 을 선택한 다음 완료로 표시를 클릭합니다.

d. 구성을 클릭하여 HTTP 연결 생성

- i.** 이름 필드에 연결 이름을 지정합니다.
- ii.** 이전에 생성한 자격 증명 및 연결 별칭 을 선택합니다.
- iii.** Nokia NSP에 대한 연결 **URL** 을 지정합니다.
- iv.** **MID** 서버 사용을 선택하고 다음 중 하나를 선택합니다.
 - 자동 선택
 - 특정 MID 서버
 - 특정 MID 클러스터

v. 나머지 필드는 기본값으로 두고 제출 을 선택한 다음 완료로 표시를 클릭합니다.

5. 시작을 클릭한 다음 구성을 클릭하여 대량 검색 또는 필터링된 검색에 대한 데이터 수집 일정을 구성합니다.

[End]

[store-future: BEGIN review]

여러 **Nokia NSP** 인스턴스 설정

단일 ServiceNow 환경 내에서 여러 Nokia NSP 인스턴스를 구성하고 관리하는 방법에 대해 알아봅니다. 이를 통해 관리자와 통합자는 독립적인 연결 별칭을 생성하고 특정 필터링 또는 빈도 요구에 맞게 사용자 지정된 일정을 임포트할 수 있습니다.

시작하기 전에

필요한 역할: 관리자

이 태스크 정보

다음에 있는지 확인하십시오.

- NOKIA NSP용 서비스 그래프 커넥터 애플리케이션 범위에서 운영하고 있습니다.
- Nokia NSP용 서비스 그래프 커넥터를 설치했습니다.
- 초기 NSP 인스턴스에 대한 안내 설정을 완료했습니다.
- MID 서버가 설정되고 검증되었습니다.

추가 Nokia NSP 인스턴스를 구성하거나 다른 연결 별칭 및 임포트 일정으로 동일한 NSP 인스턴스를 재사용할 수 있습니다. 새 NSP 인스턴스를 추가하려면 안내 설정을 다시 실행하여 새 연결 별칭을 구성합니다. 각 새 별칭에 대한 설정의 모든 연결 스테이지를 진행합니다.

프로시저

- 1.** 다음으로 이동 모두 > 서비스 그래프 커넥터 > **Nokia NSP** > 설정.
- 2.** 시작하기 페이지에서 시작하기를 선택합니다.

Configured Connectivity (구성된 연결) 섹션의 모든 단계를 반복합니다. Nokia NSP의 새 인스턴스에 대한 구성 항목을 생성합니다. 자세한 내용은 [Nokia NSP 설치 및 구성](#) 문서를 참조하십시오.

3. 새 연결 별칭, 자격 증명 및 HTTP 연결을 생성하여 연결을 구성합니다.
 - a. 연결 구성 섹션에서 시작하기를 클릭합니다.
 - b. 구성을 클릭하여 고유한 별칭 이름(예: NSP_Prod_01)을 입력하여 새 연결 별칭을 만듭니다.
 - c. 제출을 선택하고 단계를 완료로 표시합니다.
4. 새 NSP 인스턴스에 대한 자격 증명을 구성합니다.
 - a. NSP 컨트롤러에 대한 사용자 이름 및 암호를 지정합니다.
 - b. 단계를 제출하고 완료로 표시하십시오.
5. HTTP 연결을 구성합니다.
 - a. 연결 이름을 제공합니다.
 - b. 새로 만든 자격 증명과 연결 별칭을 선택합니다.
 - c. Nokia NSP 인스턴스의 연결 URL을 입력합니다.
 - d. MID 서버 사용을 활성화하고 적절한 MID 옵션을 선택합니다.
 - e. 제출을 선택하고 단계를 완료로 표시합니다.
6. импорт 일정을 구성하여 대량 또는 필터링된 검색을 사용하여 데이터 임포트를 예약합니다.
 - a. 필드에 내용을 입력합니다. 자세한 내용은 다음 문서를 참조하십시오 [Nokia NSP 설치 및 구성](#).
 - b. 연결 사용 필드에서 새 NSP 인스턴스를 선택합니다.
7. 제출을 선택합니다.

결과

새 인스턴스 설정이 성공적으로 완료되었는지 확인하기 위해 구성을 검증할 수 있습니다

- 모든 > 서비스 그래프 커넥터 > Nokia NSP > 연결 별칭으로 이동하고 새 별칭이 나열되어 있는지 확인합니다.
- 모든 > 서비스 그래프 커넥터 > Nokia NSP > импорт 일정으로 이동하여 새 импорт 일정이 나열되어 있는지 확인하고 각 별칭에 해당하는 작업이 있는지 확인합니다.

동일한 Nokia NSP 인스턴스를 가리키는 여러 연결 별칭을 구성할 수 있습니다. 이러한 유연성을 통해 다양한 빈도로 임포트를 실행하고 각 별칭에 서로 다른 필터를 적용할 수 있습니다.

관련 정보

[Nokia NSP를 통한 통신 검색](#)

[CMDB에서 Nokia NSP CI 및 관계 매핑](#)

[End]

[store-future: BEGIN review]

Nokia NSP에 대한 동시 импорт 및 병렬 로드 구성

동시 임포트를 구성하고 병렬 로드를 사용하도록 설정하여 Nokia NSP에서 대규모 데이터 임포트의 효율성을 개선합니다. 이 구성을 사용하면 플랫폼에서 여러 데이터 импорт 및 변환 작업을 동시에 실행할 수 있습니다.

시작하기 전에

필요한 역할: 관리자

이 태스크 정보

동시 가져오기 및 병렬 로드는 큰 데이터 세트를 더 작은 파티션으로 나누어 여러 변환 및 수집 작업을 병렬로 실행할 수 있도록 하여 성능을 향상시킵니다. 이 설정은 대량 또는 필터링된 검색 중에 대량의 장비, 논리적 연결 및 네트워크 토폴로지 데이터를 임포트할 때 유용합니다.

다음 지침을 사용하여 성능을 최적화합니다.

- 변환이 느린 경우: 임포트 일정에서 동시 임포트를 사용하도록 설정합니다.
- 데이터 수집이 느린 경우: 동시 임포트를 활성화하는 것 외에도 데이터 소스에서 병렬 로드를 구성하고 시스템 속성을 업데이트합니다.



주:

병렬 로드는 동시 임포트 외에도 작동합니다. 그것은 그것을 대체하지 않습니다.

프로시저

1. 다음으로 이동 모두 > 서비스 그래프 커넥터 > **Nokia NSP** > 임포트 일정.
2. 동시 임포트 또는 동시 임포트 구성

관련 정보

[Nokia NSP 설치 및 구성](#)

[CMDB에서 Nokia NSP CI 및 관계 매핑](#)

[여러 Nokia NSP 인스턴스 설정](#)

[End]

[store-future: BEGIN review]

CMDB에서 **Nokia NSP CI** 및 관계 매핑

Nokia NSP용 서비스 그래프 커넥터를 사용하여 검색된 실제 및 논리적 네트워크 자원을 CMDB의 통신 정렬 CI(구성 항목) 클래스에 매핑합니다. 이 커넥터는 일관된 서비스 모델링, 새시 수준 구성요소에 대한 가시성, 논리적 및 물리적 관계의 자동화를 지원합니다.

정확한 CI 분류 및 삽입을 확인하기 위해 커넥터는 강력한 변환 엔진(RTE)과 식별 및 조정 엔진(IRE)을 사용합니다.

커넥터는 장치 유형, 기능 및 새시 구조에 따라 통신 관련 모델을 사용하여 검색된 CI를 분류하고 연관시킵니다. 이를 통해 벤더 간에 깔끔하고 표준화된 CMDB를 유지할 수 있습니다. Nokia NSP에서 검색된 모델 이름은 슬롯 및 하위 슬롯 구성요소에 대한 ServiceNow 표준 모델 식별자 및 범주로 자동 변환됩니다.

CI 매핑 및 관계

다음 테이블에서는 NSP CI가 CMDB에 표시되는 방식과 물리적 및 논리적 계층에서 서로 어떻게 관련되는지에 대해 설명합니다.

CMDB CI 매핑 및 관계(물리적 계층)

CI 유형	CMDB 테이블	설명 및 관계
IP 라우터 CI	cmdb_ci_ip_router	IP 라우터 장치를 나타냅니다. 네트워크 토폴로지에 포함됩니다. 슬롯 CI와 논리 네트워크 인터페이스 CI를 포함합니다.
슬롯 CI	## Clcmdb_ci_container_slot	메인 새시 슬롯을 나타냅니다. IP 라우터 CI에 포함되어 있습니다. 인터페이스 카드 CI를 포함합니다.

CMDB CI 매핑 및 관계(물리적 계층)

CI 유형	CMDB 테이블	설명 및 관계
하위 슬롯 CI	cmdb_ci_container_subslot	인터페이스 내의 하위 구성요소를 나타냅니다. 인터페이스 카드에 포함되어 있습니다. 트랜시버 및 하위 카드 CI를 포함합니다.
인터페이스 카드 CI	cmdb_ci_interface_card	인터페이스 카드, 트랜시버 및 제어 장치를 나타냅니다. 하위 슬롯과 네트워크 인터페이스를 포함할 수 있습니다.
네트워크 인터페이스 CI	cmdb_ci_ni_interface	물리적(예: 이더넷) 및 논리적(예: ENET) 포트를 모두 나타냅니다. 인터페이스 카드 또는 IP 라우터 CI에 포함되어 있습니다. 논리 포트는 Members: :Member of를 사용하여 물리적 포트와 관련됩니다.
논리적 연결 CI	cmdb_ci_ni_logical_path	ENET 또는 LAG와 같은 논리 경로를 나타냅니다. 네트워크 인터페이스 CI를 종료하는 포트 A 및 포트 Z 속성으로 정의됩니다.
IP 주소 CI	cmdb_ci_ip_address	IP 라우터에 대해 검색된 IP 주소를 나타냅니다. 해당 IP 라우터 CI가 소유합니다.
네트워크 토폴로지	cmdb_ci_network_topology	네트워크 토폴로지를 나타냅니다. IP 라우터 및 논리적 연결을 포함합니다.

관련 정보

[Nokia NSP와 함께 설치되는 시스템 구성요소](#)

[Nokia NSP 서비스 그래프 커넥터 구성](#)

[End]

통신 불일치 식별 및 조정 활성화

라이브 네트워크에서 검색된 통신 네트워크 자원과 CMDB 또는 TNI(통신 네트워크 인벤토리)에 표시된 데이터 간의 일관성을 보장하기 위해 통신 불일치 식별 및 조정 기능(Telecom Visibility 플러그인의 일부)을 활성화합니다. 이 기능은 불일치를 자동으로 탐지하고 정정하여 통신 환경 전반에서 서비스 정확성과 데이터 무결성을 지원합니다.

시작하기 전에

필요한 역할: 관리자

이 기능을 사용하려면 조직에 TSOM에 대한 활성 구독이 있어야 합니다. 통신 검색, 통신 가시성 및 이 기능은 함께 라이선스가 부여됩니다.

이 태스크 정보

플러그인 의존성 - 다음 플러그인이 설치되고 활성화되었는지 확인합니다.

플러그인	ID/앱 ID	유형
통신 서비스 운영 코어	sn_tsom_core	스토어
CMDB CI Class Models	sn_cmdb_ci_class	스토어
확장된 모델 및 자산 클래스	sn_ent	스토어
가시성 콘텐츠	sn_pattern_design	스토어
CMDB용 통합 커먼즈	sn_cmdb_int_util	스토어
디스커버리 코어	com.snc.discovery.core	제품군

플러그인	ID/앱 ID	유형
ITOM 디스커버리 라이선스	com.snc.itom.discovery.license	제품군
ITOM 라이선스	com.snc.itom.license	제품군

프로시저

1. 통신 디스커버리 패턴 또는 Nokia Altiplano 서비스 그래프 커넥터를 설치할 때 sn_tsom_core 플러그인이 자동으로 설치되는지 확인합니다.
2. 언급된 종속 플러그인이 활성화되어 있는지 확인합니다.

관련 정보

[불일치 식별 - 불일치 유형](#)

[통신 불일치 식별 및 조정과 함께 설치되는 시스템 구성요소](#)

[통신 불일치 식별 및 조정](#)

[통신 불일치 감사 실행](#)

[조정 규칙을 사용하여 CI 속성 업데이트 제어](#)

감사에 대한 필터 구성

필터링 조건을 정의하여 통신 불일치 감사 범위를 제어합니다. 이러한 필터를 사용하면 원하는 CI 집합에 대해서만 감사가 실행되어 성능과 대상 지정 정확도가 향상됩니다.

시작하기 전에

필요한 역할: 관리자

이 태스크 정보

통신 불일치 감사의 필터링 조건 이해

- 감사에서는 필터링 조건을 사용하여 평가 대상 CI의 범위를 좁힙니다.
- 필터링 조건은 감사 대상의 범위를 좁히는 데 필수적입니다.
- 특정 감사 요구 사항을 충족하도록 감사 인스턴스별로 조건을 정의하고 수정할 수 있습니다.
- `discovery_source`별 필터링은 특정 통합 소스(예: TSOM 및 Altiplano)와 연결된 기록을 필터링하는 일반적인 조건입니다. 또한 감사 요구 사항에 따라 필터를 사용자 지정할 수 있습니다.
- 감사의 각 버전마다 필터링 조건이 다를 수 있습니다.

다음은 감사 필터입니다.

1. 통신 논리적 연결 불일치 감사: 이 감사는 논리적 연결 테이블에서 실행됩니다. 기본 필터 조건은 TSOM과 같은 `discovery_source`로 정의됩니다. 이 조건은 논리적 연결 테이블의 각 CI에 적용됩니다.
2. 통신 네트워크 토폴로지 불일치 감사: 이 감사는 네트워크 토폴로지 테이블에서 실행됩니다. 기본 필터 조건은 TSOM과 같은 `discovery_source`로 정의됩니다. 이 조건은 네트워크 토폴로지 테이블의 각 CI에 적용됩니다.
3. 통신 불일치 감사: 이 감사는 관계 테이블에서 실행됩니다. 기본 필터 조건은 TSOM과 같은 `discovery_source`로 정의됩니다. 이 조건은 장비 수준에서 각 CI에 적용됩니다.

다음 스크린샷은 인증 필터의 사용자 인터페이스를 이해하는 데 도움이 됩니다.

Number: CFLR0010004 Active

* Name: Filter for TSOM Logical Connections Discrepancy Audit

Description: Filter for TSOM Logical Connections Discrepancy Audit

Table: Configuration Item [cmdb_ci] Version: 2

Filter condition: No. of records matching the condition: 1

Add Filter Condition Add OR Clause

All of these conditions must be met

Discovery source contains TSOM AND OR

Name is pon_path_PON_L_GPON AND OR

Update

Related Links

Delete inactive versions

Other Versions Version Search

Name	Description	Table	Version
Filter for TSOM Logical Connections Disc...	Filter for TSOM Logical Connections Disc...	Configuration Item [cmdb_ci]	1

1 to 1 of 1

프로시저

1. 다음으로 이동 모두 > 인증 > 필터.
2. 새로 만들기를 선택합니다.
인증 필터 인터페이스가 나타납니다.
3. 필드에 내용을 입력합니다.
자세한 지침은 다음 문서를 참조하십시오 [Create a certification filter](#).
4. 필터 조건을 추가합니다.
예를 들어 검색 소스에 TSOM이 포함되어 있습니다.
5. 제출을 선택합니다.
필터가 생성됩니다.

결과

논리적 연결 감사에서 필터를 사용할 수 있습니다.

관련 정보

[통신 불일치 감사 실행](#)

[통신 불일치 감사 및 정정에 대한 예](#)

[통신 불일치 식별 및 조정과 함께 설치되는 시스템 구성요소](#)

[CMDB 360에서 속성 값 불일치 구성](#)

[조정 규칙을 사용하여 CI 속성 업데이트 제어](#)

CMDB 360에서 속성 값 불일치 구성

CMDB 360에서 속성 비교 설정을 구성하여 여러 검색 소스 간에 데이터 불일치를 감지합니다.

시작하기 전에

필요한 역할: 관리자

인스턴스에서 CMDB 360이 활성화되고 구성되어 있는지 확인합니다.

이 태스크 정보

다음 스크린샷은 불일치를 식별하도록 필드를 설정하는 데 도움이 될 수

있습니다.

프로시저

1. 테이블로 이동 모두 > **cmdb_multisource_column_metadata.list**.

2. 새로 만들기를 선택하여 새 기록을 생성합니다.

3. 멀티소스 열 필드에서 비교할 속성을 선택합니다.

4. 속성을 필드와 비교할 테이블을 선택합니다.



주:

필요한 테이블이 나열되지 않으면 테이블 레이블을 길게 누르거나 마우스 오른쪽 단추로 클릭하고 사전 구성을 선택합니다.

5. 옵션: 디렉터리 항목을 구성하려면 다음을 수행합니다.

a. 디렉터리 항목에서 보기를 선택한 다음, 고급을 선택합니다.

b. 속성 필드에서 속성을 `base_start=true` 및 `allow_public=true`로 설정합니다.

결과

CMDB 360 쿼리에서 구성된 속성을 사용할 수 있습니다.

관련 정보

[조정 규칙을 사용하여 CI 속성 업데이트 제어](#)

[속성 값 불일치에 대한 보고서 생성](#)

조정 규칙을 사용하여 CI 속성 업데이트 제어

CI(구성 항목)의 특정 속성이 디스커버리 또는 기타 데이터 소스에 덮어쓰기되지 않게 하려면 조정 규칙을 사용하십시오. 이러한 규칙은 여러 소스에서 값을 제공할 때 특정 속성을 업데이트하도록 신뢰할 수 있는 데이터 소스를 정의합니다.

시작하기 전에

필요한 역할: 관리자

이 태스크 정보

조정 규칙은 IRE(식별 및 조정 엔진)에 의해 처리되며 CMDB에서 데이터 무결성을 유지하는 데 필수적입니다.

사용 사례: 검색에서 속성 세트를 업데이트하지 못하게 하고 다른 소스(예: SCCM 또는 수동 입력)가 속성 세트를 업데이트하지 못하게 하려면 검색 소스를 제외하는 규칙을 정의합니다.



주:

충돌을 방지하려면 특정 CI 클래스 및 속성 조합에 대해 하나의 조정 규칙만 활성화해야 합니다.

프로시저

1. 다음으로 이동 모두 > **CI** 클래스 관리자.

2. 대상 CI 클래스(예: `cmdb_ci_computer`)를 선택합니다.

3. 조정 규칙 탭에서 새로 만들기를 클릭하여 규칙을 생성합니다.
4. 규칙 이름을 입력하고 규칙을 적용할 디스커버리 소스를 선택합니다.
5. 이 규칙이 지배할 속성을 지정하십시오.
6. 선택한 소스만 지정된 속성을 업데이트할 수 있도록 소스 우선순위를 설정합니다.

결과

특정 CI 속성은 Discovery와 같이 신뢰할 수 없거나 우선순위가 낮은 데이터 소스에 의해 더 이상 업데이트되지 않습니다. 규칙에 정의된 소스(예: SCCM, 수동 업데이트)만 해당 속성을 업데이트할 수 있습니다.

관련 정보

[CMDB 360에서 속성 값 불일치 구성](#)

Telecommunications Service Operations Management 사용

(TSOM)를 활용하여 Telecommunications Service Operations Management 통신 서비스를 사전 예방적으로 모니터링하고, 데이터 무결성을 검증하고, 네트워크 인벤토리 및 디스커버리 소스 전반의 불일치를 조정합니다. TSOM을 통해 운영 팀은 정확한 통신 인식 CMDB를 유지하고 실시간 네트워크 통찰력에 따라 조치를 취할 수 있습니다.

TSOM의 공통 운영 작업

TSOM이 구성되면 해당 기능을 사용하여 다음을 수행할 수 있습니다.

- 통신 불일치 감사 실행: 검색된 데이터 및 조정 규칙을 기반으로 인증 감사를 실행하여 네트워크 관계 및 구성의 불일치를 식별합니다.
- CMDB 360에서 속성 값 불일치 확인: CMDB 360의 속성 값 불일치 기능을 사용하여 다양한 검색 소스 간의 속성 값을 비교하고 상충 또는 오래된 기록을 감지합니다.
- 통신 디스커버리 패턴 사용: 통신 디스커버리 패턴을 적용하여 통신별 구성 항목(CI)과 그 관계를 정확하게 검색하고 채웁니다.
- 서비스 그래프 커넥터에서 임포트 작업 실행 및 확인: 임포트 일정(예: Nokia Altiplano 커넥터)을 수동으로 트리거하여 토폴로지 및 장치 데이터가 CMDB로 성공적으로 임포트되고 올바른 CI 클래스에 매핑되는지 확인합니다.

이러한 작업은 통신 네트워크가 ServiceNow 내에서 지속적으로 동기화되고 정확하며 운영상 볼 수 있도록 하는 데 도움이 됩니다.

통신 검색 패턴 사용

텔레콤 디스커버리 패턴을 사용하여 라우터, Cisco 및 주니퍼와 같은 벤더 장치에서 네트워크 기능(xNF)을 식별하고 분류합니다. 패턴 기반 검색을 활용하여 통신 자원을 CMDB에 매핑합니다.

통신 라우터 패턴에 대한 xNF 분류

분류될 OID의 전체 목록에 액세스하려면 다음을 수행합니다.

시작하기 전에

필요한 역할: 관리자

분류 규칙 이름: 표준 네트워크 라우터.

프로시저

1. 다음으로 이동 모두 > 디스커버리 정의 > **CI 분류** > **SNMP**.
2. 목록에서 표준 네트워크 라우터를 선택합니다.
3. **SNMP OID** 분류 탭을 열고 OID 목록을 봅니다.

SNMP Classification
Standard Network Router

After the Sensor processes the results of the SNMP - Classify Probe, secondary to SNMP System OIDs, each SNMP Classification defines the order, match criteria, and Probes to trigger. [More Info](#)

Name: Standard Network Router
Active: ☒
Order: 20
Table: IP Router
Match criteria: All
Manufacturer:
Model:

On classification script
1 // This script gets run when something gets classified

Update Delete

Related Links
[Run Point Scan](#)

Classification Criteria (1) **SNMP OID Classifications (199)** Triggers probes (3) Versions (8)

Classifier - Standard Network Router

OID	Operator	Table	Manufacturer	Model	Active
1.3.6.1.4.1.141.1.1.3220	Is	IP Router [cmdb_ci_ip_router]	NetScout Systems, Inc.	Packet Probe 3220	true
1.3.6.1.4.1.94.1.2.1.2	Is	IP Router [cmdb_ci_ip_router]	Nokia	IP4xx	true
1.3.6.1.4.1.9.1.2411	Is	IP Router [cmdb_ci_ip_router]	Cisco Systems	ciscoNC5501	true
1.3.6.1.4.1.664.1.466	Is	IP Router [cmdb_ci_ip_router]	Adtran	NETVANTA3205	true
1.3.6.1.4.1.43.1.16.4.2.12	Is	IP Router [cmdb_ci_ip_router]	Hewlett-Packard	R6080	true
1.3.6.1.4.1.9.1.758	Is	IP Router [cmdb_ci_ip_router]	Cisco Systems	1250	true
1.3.6.1.4.1.9.1.1448	Is	IP Router [cmdb_ci_ip_router]	Cisco Systems	ciscoAS5515K7xc	true
1.3.6.1.4.1.2636.1.1.2.24	Is	IP Router [cmdb_ci_ip_router]	Juniper Networks	J2350	true
1.3.6.1.4.1.2636.1.1.1.25	Is	IP Router [cmdb_ci_ip_router]	Juniper Networks	M5	true

i 주:

분류 규칙에 OID를 추가하는 방법에 대한 자세한 내용은 다음 문서를 참조하십시오
[Discovery 패턴을 사용한 Direct Discovery](#).

xNF에서 사용되는 MIB 테이블:

- 시스템MIB
- 엔터티실제 MIB
- IfMIB
- IfXMIB
- IpMIB

통신용 xNF 분류 **Cisco 7613** 라우터 패턴

분류될 OID의 전체 목록에 액세스하려면 다음을 수행합니다.

시작하기 전에
필요한 역할: 관리자

분류 규칙 이름: 표준 네트워크 라우터.

프로시저

1. 다음으로 이동 모두 > 디스커버리 정의 > **CI 분류** > **SNMP**.
2. 목록에서 표준 네트워크 라우터를 선택합니다.

3. SNMP OID 분류 탭을 열고 OID 목록을 봅니다.

After the Sensor processes the results of the SNMP - Classify Probe, secondary to SNMP System OIDs, each SNMP Classification defines the order, match criteria, and Probes to trigger. [More Info](#)

Name: Standard Network Router
 Active: ☒
 Order: 20
 Table: IP Router
 Match criteria: All
 Manufacturer:
 Model:

On classification script
 1 // This script gets run when something gets classified

Update Delete

Related Links
[Run Point Scan](#)

Classification Criteria (1) **SNMP OID Classifications (199)** Triggers probes (3) Versions (8)

Classifier - Standard Network Router

Classifier	Old	Operator	Table	Manufacturer	Model	Active
1.3.6.1.4.1.141.1.1.3220	Is	IP Router [cndb_ci_ip_router]	NetScout Systems, Inc.	Packet Probe 3220	true	
1.3.6.1.4.1.94.1.2.1.2	Is	IP Router [cndb_ci_ip_router]	Nokia	IP4ox	true	
1.3.6.1.4.1.9.1.2411	Is	IP Router [cndb_ci_ip_router]	Cisco Systems	ciscoNCS5011	true	
1.3.6.1.4.1.664.1.466	Is	IP Router [cndb_ci_ip_router]	Adtran	NETVANTA3205	true	
1.3.6.1.4.1.43.1.16.4.2.12	Is	IP Router [cndb_ci_ip_router]	Hewlett-Packard	R6080	true	
1.3.6.1.4.1.9.1.758	Is	IP Router [cndb_ci_ip_router]	Cisco Systems	1250	true	
1.3.6.1.4.1.9.1.1448	Is	IP Router [cndb_ci_ip_router]	Cisco Systems	ciscoASA5515K7ac	true	
1.3.6.1.4.1.2636.1.1.1.2.24	Is	IP Router [cndb_ci_ip_router]	Juniper Networks	J2350	true	
1.3.6.1.4.1.2636.1.1.1.2.5	Is	IP Router [cndb_ci_ip_router]	Juniper Networks	M5	true	

주:

분류 규칙에 OID를 추가하는 방법에 대한 자세한 내용은 다음 문서를 참조하십시오
[Discovery 패턴을 사용한 Direct Discovery](#).

이 패턴을 호출할 특정 OID 목록입니다.

벤더	모델	OID	패턴
Cisco	7613	1.3.6.1.4.1.9.1.528	통신 시스코 7613 라우터

xNF에서 사용되는 MiB 테이블:

- 시스템MIB
- 엔터티실제 MIB
- IfMIB
- IfXMIB
- IpMIB

통신용 xNF 분류 **Juniper MX SSH** 라우터 패턴

분류될 OID의 전체 목록에 액세스하려면 다음을 수행합니다.

시작하기 전에
 필요한 역할: 관리자

분류 규칙 이름: 표준 네트워크 라우터.

프로시저

1. 다음으로 이동 모두 > 디스커버리 정의 > **CI** 분류 > **SNMP**.
2. 목록에서 표준 네트워크 라우터를 선택합니다.

3. SNMP OID 분류 탭을 열고 OID 목록을 봅니다.

After the Sensor processes the results of the SNMP - Classify Probe, secondary to SNMP System OIDs, each SNMP Classification defines the order, match criteria, and Probes to trigger. [More Info](#)

Name: Standard Network Router
Active: ☒
Order: 20
Table: IP Router
Match criteria: All
Manufacturer:
Model:
On classification script:

```
1 // This script gets run when something gets classified
```

Update Delete
Related Links
Run Point Scan

Classification Criteria (1) **SNMP OID Classification (1996)** Triggers probes (3) Versions (8)

Classifier - Standard Network Router

Old	Operator	Table	Manufacturer	Model	Active
1.3.6.1.4.1.141.1.1.3220	Is	IP Router [cnmb_c_i_ip_router]	NetScout Systems, Inc.	Packet Probe 3220	true
1.3.6.1.4.1.94.1.21.2.2	Is	IP Router [cnmb_c_i_ip_router]	Nokia	IP4xx	true
1.3.6.1.4.1.9.1.2411	Is	IP Router [cnmb_c_i_ip_router]	Cisco Systems	ciscoNC55011	true
1.3.6.1.4.1.664.1.466	Is	IP Router [cnmb_c_i_ip_router]	Adtran	NETVANTA205	true
1.3.6.1.4.1.43.1.16.4.2.12	Is	IP Router [cnmb_c_i_ip_router]	Hewlett-Packard	R6080	true
1.3.6.1.4.1.9.1.758	Is	IP Router [cnmb_c_i_ip_router]	Cisco Systems	1250	true
1.3.6.1.4.1.9.1.1448	Is	IP Router [cnmb_c_i_ip_router]	Cisco Systems	ciscoASA5515K7sc	true
1.3.6.1.4.1.2636.1.1.2.24	Is	IP Router [cnmb_c_i_ip_router]	Juniper Networks	J2350	true
1.3.6.1.4.1.2636.1.1.1.2.5	Is	IP Router [cnmb_c_i_ip_router]	Juniper Networks	M5	true

주:

분류 규칙에 OID를 추가하는 방법에 대한 자세한 내용은 다음 문서를 참조하십시오
[Discovery 패턴을 사용한 Direct Discovery](#).

이 패턴을 호출할 특정 OID 목록입니다.

벤더	모델	OID	패턴
Juniper	MX80	1.3.6.1.4.1.2636.1.1.1.2.57	통신 Juniper MX SSH 라우터
Juniper	MX104	1.3.6.1.4.1.2636.1.1.1.2.97	통신 Juniper MX SSH 라우터
Juniper	MX240	1.3.6.1.4.1.2636.1.1.1.2.29	통신 Juniper MX SSH 라우터
Juniper	MX480	1.3.6.1.4.1.2636.1.1.1.2.25	통신 Juniper MX SSH 라우터

xNF에서 사용되는 MiB 테이블: SystemMIB.

CLI 명령이 사용되었습니다.

- 새시 하드웨어 | 더 이상 | XML 표시
- 인터페이스 미디어 표시 | 더 이상 | XML 표시

통신용 xNF 분류 Cisco 스위치 패턴

분류될 OID의 전체 목록에 액세스하려면 다음을 수행합니다.

시작하기 전에
필요한 역할: 관리자

분류 규칙 이름: 표준 네트워크 스위치.

프로시저

1. 다음으로 이동 모두 > 디스커버리 정의 > **CI 분류** > **SNMP**.
2. 목록에서 표준 네트워크 스위치를 선택합니다.
3. **SNMP OID** 분류 탭을 열고 OID 목록을 봅니다.

SNMP Classification
Standard Network Switch

After the Sensor processes the results of the SNMP - Classify Probe, secondary to SNMP System OIDs, each SNMP Classification defines the order, match criteria, and Probes to trigger. [More Info](#)

Name: Table:

Active: ☒ Match criteria:

Order: Manufacturer:

Model:

On classification script:

Update Delete

Related Links
[Run Point Scan](#)

Classification Criteria (2) **SNMP OID Classifications (4273)** Triggers probes (3) Versions

for text Search

Classifier - Standard Network Switch

OID	Operator	Table	Manufacturer	Model	Active
1.3.6.1.4.1.1916.2.98	Is	IP Switch [cnmdb_ci_ip_switch]	Extreme Networks	SUMMITX250-24XDC	true
1.3.6.1.4.1.14179.2.2.1.1.22.57	Is	IP Switch [cnmdb_ci_ip_switch]	Airspaces, Inc	Cisco 1830i Unified Access Point	true
1.3.6.1.4.1.9.1.1071	Is	IP Switch [cnmdb_ci_ip_switch]	Cisco Systems	ciscoPwrC3900Poe	true
1.3.6.1.4.1.6527.6.2.1.2.1	Is	IP Switch [cnmdb_ci_ip_switch]	Alcatel-Lucent Enterprise	7210 SAS-E	true
1.3.6.1.4.1.25506.1.246	Is	IP Switch [cnmdb_ci_ip_switch]	Hewlett-Packard	H3C S3100-26TP-EI	true
1.3.6.1.4.1.25506.1.38	Is	IP Switch [cnmdb_ci_ip_switch]	Hewlett-Packard	H3C S3600-52P-EI	true
1.3.6.1.4.1.9.1.1637	Is	IP Switch [cnmdb_ci_ip_switch]	Cisco Systems	ciscoE140D	true
1.3.6.1.4.1.25506.1.654	Is	IP Switch [cnmdb_ci_ip_switch]	H3C Technologies	S3100V2-16TP-SI	true
1.3.6.1.4.1.1991.1.3.39.1	Is	IP Switch [cnmdb_ci_ip_switch]	Brocade Communications Systems	Netron IMR 440 Switch	true
1.3.6.1.4.1.2.6.157	Is	IP Switch [cnmdb_ci_ip_switch]	IBM	8275-416	true



주:

분류 규칙에 OID를 추가하는 방법에 대한 자세한 내용은 다음 문서를 참조하십시오
[Discovery 패턴을 사용한 Direct Discovery](#).

이 패턴을 호출할 특정 OID 목록입니다.

벤더	모델	OID	패턴
Cisco	넥서스 9000	1.3.6.1.4.1.9.1.12.3.1.3.1954	텔레콤 Cisco 스위치
Cisco	넥서스 3548	1.3.6.1.4.1.9.1.12.3.1.3.1666	텔레콤 Cisco 스위치

xNF에서 사용되는 MiB 테이블:

- 시스템MIB
- 엔터티실제 MIB
- IfMIB
- IfXMIB
- IpMIB

통신 스위치 패턴에 대한 xNF 분류

분류될 OID의 전체 목록에 액세스하려면 다음을 수행합니다.

시작하기 전에

필요한 역할: 관리자

분류 규칙 이름: 표준 네트워크 스위치.

프로시저

1. 다음으로 이동 모두 > 디스커버리 정의 > **CI** 분류 > **SNMP**.
2. 목록에서 표준 네트워크 스위치를 선택합니다.
3. **SNMP OID** 분류 탭을 열고 OID 목록을 봅니다.

After the Sensor processes the results of the SNMP - Classify Probe, secondary to SNMP System OIDs, each SNMP Classification defines the order, match criteria, and Probes to trigger. [More Info](#)

Name: Table:

Active: ☒ Match criteria:

Order: Manufacturer:

Model:

On classification script

```
1 // This script gets run when something gets classified
```

Update Delete

Related Links

Run Point Scan

Classification Criteria (2) **SNMP OID Classifications (427)** Triggers probes (3) Versions

for text Search

Actions on selected rows... New Sync Candidates Sync From CDS

OID	Operator	Table	Manufacturer	Model	Active
1.3.6.1.4.1.1916.2.98	Is	IP Switch [cmdb_ci_ip_switch]	Extreme Networks	SUMMITX250-24XDC	true
1.3.6.1.4.1.14179.2.2.1.1.22.57	Is	IP Switch [cmdb_ci_ip_switch]	Airespace, Inc	Cisco 1830I Unified Access Point	true
1.3.6.1.4.1.9.1.1071	Is	IP Switch [cmdb_ci_ip_switch]	Cisco Systems	ciscoPwrC3900Poe	true
1.3.6.1.4.1.6527.6.2.1.2.1	Is	IP Switch [cmdb_ci_ip_switch]	Alcatel-Lucent Enterprise	7210 SAS-E	true
1.3.6.1.4.1.25506.1.246	Is	IP Switch [cmdb_ci_ip_switch]	Hewlett-Packard	H3C S3100-26TP-EI	true
1.3.6.1.4.1.25506.1.38	Is	IP Switch [cmdb_ci_ip_switch]	Hewlett-Packard	H3C S3600-52P-EI	true
1.3.6.1.4.1.9.1.1637	Is	IP Switch [cmdb_ci_ip_switch]	Cisco Systems	ciscoE140D	true
1.3.6.1.4.1.25506.1.654	Is	IP Switch [cmdb_ci_ip_switch]	H3C Technologies	S3100V2-16TP-SI	true
1.3.6.1.4.1.1991.1.3.39.1	Is	IP Switch [cmdb_ci_ip_switch]	Brocade Communications Systems	NetIron IMR 640 Switch	true
1.3.6.1.4.1.2.6.157	Is	IP Switch [cmdb_ci_ip_switch]	IBM	8275-416	true

주:

분류 규칙에 OID를 추가하는 방법에 대한 자세한 내용은 다음 문서를 참조하십시오
[Discovery 패턴을 사용한 Direct Discovery](#).

xNF에서 사용되는 MiB 테이블:

- 시스템MIB
- 엔터티실제 MIB
- IfMIB
- IfXMIB
- IpMIB

Nokia Altiplano 서비스 그래프 커넥터에 대한 импорт 실행 및 확인

Nokia Altiplano 서비스 그래프 커넥터에 대해 구성된 импорт 일정을 수동으로 실행합니다. Imports를 실행하여 커넥터 설정의 유효성을 확인하거나, 임시 Imports를 실행하거나, 새로 구성된 연결 별칭을 테스트할 수도 있습니다. 이 작업은 Altiplano의 데이터를 CMDB에서 성공적으로 импорт하거나 업데이트하는 데 도움이 됩니다.

시작하기 전에

필요한 역할: 관리자

이 태스크 정보

Nokia Altiplano 서비스 그래프 커넥터를 구성하고 하나 이상의 импорт 일정을 설정한 후에는 импорт 작업을 실행할 수 있습니다. 이 작업은 통합이 예상대로 작동하고 CMDB에서 CI를 성공적으로 импорт 또는 업데이트했는지 확인하는 데 도움이 됩니다.

импорт 일정이 정의된 빈도에 따라 자동으로 실행되도록 설정하거나, 즉시 실행 및 확인을 위해 수동으로 트리거할 수 있습니다.

주:

여러 환경 또는 필터에 대해 여러 Altiplano 연결 별칭을 구성한 경우 각 별칭에 대해 импорт 일정이 올바르게 설정되었는지 확인합니다.

다음 스크린샷은 예약된 데이터 импорт 프로세스를 이해하는 데 도움이 되며 импорт 세트 형식으로 실행을 표시합니다.

Number	Execution Context	Import Set	Created
EXE00001043	EC00001043	Import Set: ISET0010043	2025-07-13 00:00:00
EXE00001042	EC00001042	Import Set: ISET0010042	2025-07-12 23:36:01
EXE00001040	EC00001040	Import Set: ISET0010040	2025-07-12 00:00:00

프로시저

- 다음으로 이동 모두 > **Nokia Altiplano** > 일정 импорт.
- 실행할 импорт 일정을 선택합니다.
예약된 데이터 импорт 목록이 나타납니다.
- 예약된 데이터 Imports를 선택합니다.
- 다음 중 하나를 수행합니다.
 - 일정이 활성화된 경우 정의된 간격으로 작업이 자동으로 실행됩니다. 그러니 실행될 때까지 기다리세요.
 - 지금 실행을 선택하여 Imports를 즉시 실행합니다.
- 실행을 모니터링하고 결과를 확인하려면 다음을 수행합니다.
 - 실행 목록 또는 импорт 일정에 연결된 테이블까지 아래로 스크롤합니다.
 - 실행으로 생성된 가장 최근의 импорт 세트 기록을 엽니다.
- импорт 세트 로그를 검토하여 읽은 행 수, CMDB에 삽입되거나 업데이트된 행 수 또는 변환 성공 상태를 확인합니다.
- 옵션: 다음으로 이동 모두 > **CMDB** > **CI** 클래스 또는 사용자 지정 통신 테이블(예: TNI 기본 항목)을 사용하여 CI가 예상대로 생성되거나 업데이트되었습니다.

결과

임포트 일정이 즉시 실행되며, CMDB에서 임포트 세트 실행 및 해당 CI 업데이트를 확인할 수 있습니다. 올바르게 구성된 경우 커넥터는 Nokia Altiplano의 네트워크 인벤토리 데이터를 ServiceNow 인스턴스로 가져옵니다.

예: 임포트 일정을 실행한 후:

- 로그에 "12개 행 읽기, 0개 삽입/업데이트됨"이 표시됩니다. 이는 CI가 있음을 나타냅니다.
- 기존 CI를 삭제하고 임포트를 다시 실행하면 로그에 엔드 투 엔드 기능을 확인하는 여러 삽입 및 업데이트가 표시될 수 있습니다.

i 주:

- 임포트 일정에서 동시 임포트 옵션이 활성화된 경우 표준 임포트 세트 테이블 대신 동시 임포트 세트 테이블에 기록이 표시됩니다.
- 각 동시 작업은 자체 임포트 세트 및 로그 항목을 만듭니다. 실행 레코드의 구조는 동일하게 유지됩니다.

관련 정보

[Nokia Altiplano 서비스 그래프 커넥터 구성](#)

[Nokia Altiplano를 통한 통신 검색](#)

TSOM 스키마 검사기를 사용하여 JSON 페이로드 확인

데이터를 임포트하기 전에 TsomSchemaValidator 유틸리티 클래스를 사용하여 TSOM 스키마에 대해 JSON 페이로드를 확인합니다. 이를 통해 오류를 조기에 식별하고, ETL 오류를 줄이고, 데이터 품질을 확인할 수 있습니다.

이 확인 도구를 사용하여 임포트 세트를 생성하기 전에 JSON 페이로드가 장치, 논리적 연결 및 토폴로지와 같은 통신 객체에 대해 예상되는 스키마를 준수하는지 확인합니다. 이 사전 유효성 검사 단계는 스키마 불일치 오류를 방지하고 디버깅을 개선하는 데 도움이 됩니다.

지원되는 스키마 유형

유효성 검사기는 다양한 통신 데이터 구조에 대해 여러 스키마 유형을 지원합니다.

- 논리 복합 - 구성요소 그룹화 표시: 장비, PDU, 팬 선반
- 장치 - 장비 및 포함된 구성요소
- 논리적 연결 - 네트워크 인터페이스 간 연결
- 포트 관계 - 네트워크 인터페이스 간의 관계: 물리적, 논리적, 지연
- 논리적 연결 관계 - 논리적 연결 간의 관계
- 토폴로지 - 네트워크 토폴로지

클래스 구조

```
let TsomSchemaValidator = Class.create();
TsomSchemaValidator.prototype = {
  initialize: function() {
    this.schemas = new TsomGenericSchema();
  },
  isValidJson: function(payload) {
    // Validation logic that determines if the JSON structure is valid
    // Returns boolean (true/false)
  }
}
```

```

},
checkJsonValidation: function(payload) {
  // Validation logic that determines if the JSON structure is valid
  // Returns a JSON object containing errors (if exist)
},
type: 'TsomSchemaValidator'
};

```

단계

1. 스키마 유효성 검사기 인스턴스화 `javascriptCopyEdit`

```
var TsomSchemaValidator = new sn_tsom_core.TsomSchemaValidator();
```

2. 부울 유효성 검사 실행

```

if (!TsomSchemaValidator.isValidJson(target_json)) {
  gs.error('Invalid JSON: ' + JSON.stringify(target_json));
  return;
}

```

3. 상세 확인 확인 실행

```

let result = TsomSchemaValidator.checkJsonValidation(target_json);
if (!result.valid) {
  gs.error('Invalid JSON: ' + JSON.stringify(result, null, 2));
  return;
}

```

예제 출력

```

Example Output
{
  "schemaName": "devices",
  "errors": [
    {
      "message": "Missing required property: model_name",
      "params": { "key": "model_name" },
      "code": 302,
      "dataPath": "/devices/0/ports/0",
      "schemaPath": "/properties/devices/items/properties/ports/items/required/4"
    }
  ],
  "valid": false
}

```

관련 정보

[커넥터에서 Telecom Discovery Builder 프레임워크 ETL 구성](#)

[통신 검색 작성기 프레임워크](#)

통신 불일치 감사 실행

통신 불일치 감사는 CMDB 준수 프레임워크를 사용하여 CI(구성 항목)의 무결성과 통신 인벤토리 전반의 관계를 검증합니다.

시작하기 전에
필요한 역할: 관리자

이 태스크 정보

통신 불일치 감사는 CMDB 준수 프레임워크의 일부이며 ServiceNow 디스커버리 또는 서비스 그래프 커넥터(예: Nokia Altiplano)를 통해 검색된 CI 관계 및 속성의 불일치를 감지하고 해결하기 위한 단계별 감사를 지원합니다.

감사를 수동으로 실행하거나 정기적으로 실행하도록 설정할 수 있습니다. 수동 실행 중에 기존 필터를 선택하여 감사 범위를 특정 CI로 제한할 수 있어 테스트 또는 대상 조정에 유용합니다.

통신 불일치 감사는 두 단계로 작동합니다.

- 1. 초기 준수 실행:** 다음 기본 기준을 사용하여 CI 관계와 선택한 CI 테이블의 유효성을 확인합니다.
 - 지원되는 CI 클래스: 슬롯, 하위 슬롯, 카드, 인터페이스, 네트워크 기어(모든 확장 테이블 포함).
 - 소스: CI는 디스커버리를 통해 검색되었습니다(예: `discovery_source = SG-TSOM-Altiplano`).
 - 관계 유형: `Contains::Contained By` (`sn_tsom_core.audit.relationship_types` 속성을 통해 사용자 지정).
- 2. 후속 준수 실행:** 초기 검사 외에도 관계 또는 관련 CI의 업데이트된 타임스탬프가 감사 기록의 마지막 실행 날짜보다 최신인지 평가합니다.



주:

감사에 실패할 때마다 수동 또는 자동 정정을 위한 후속 작업이 생성되어 통신 CMDB 데이터가 네트워크 상태에 맞게 유지되는지 확인합니다.

다음 스크린샷은 통신 불일치 감사를 이해하는 데 도움이

The screenshot displays the configuration interface for the 'Telecom Discrepancy Audit' task. Key elements include:

- Name:** Telecom Discrepancy Audit
- Filter:** Filter for TSOM Discrepancy Audit - 1
- Table:** Configuration Item [cmdb_ci]
- Application:** Telecom Service Operations Core
- Active:** Checked
- Run:** On Demand
- Last run date:** 2025-06-30 04:07:00
- Assignment type:** User Field
- Assign to:** Assigned to
- Assign to empty:** Create Unassigned Task
- Audit type:** Scripted
- Short description:** Telecom CMDB Discrepancy Audit
- Task description:** Remedy the discrepancies listed in the Audit Results below
- Run this script:** Turn on ECMAScript 2021 (ES12) mode
- Script:** A JavaScript code block for processing discrepancies, including comments and logic for setting audit dates and processing relationships.

됩니다.

프로시저

- 1. 다음으로 이동** 모두 > 준수 > 감사
- 2. 통신 불일치 감사(cert_audit 테이블) 기록을 엽니다.**
- 3. 옵션:** 필터를 선택합니다.
 - 구성 중에 여러 감사 필터가 정의된 경우 감사를 실행하기 전에 하나를 선택할 수 있습니다.
 - 필터를 사용하면 검색 소스, CI 클래스 또는 특정 CI 속성과 같은 기준에 따라 감사 범위를 제한할 수 있습니다.
 - 이 단계는 기록의 하위 세트 문제를 해결하거나 특정 검색 결과를 확인할 때 특히 유용합니다.

4. 감사를 실행하려면 다음 중 하나를 수행합니다.

- 감사 실행을 클릭하여 수동 실행을 트리거합니다.
 - 되풀이 작업을 설정하여 일정에 따라 감사가 실행되도록 구성합니다.
- 감사는 CMDB 준수를 사용하여 통신 인벤토리 전반에서 관계와 CI 무결성을 확인합니다.

결과

감사 결과 검토

- 감사에서는 누락된 관계 또는 잘못 정렬된 속성과 같은 준수 실패를 식별합니다.
- 실패한 각 감사 기록에 대해 후속 작업이 자동으로 생성됩니다.
- 작업에는 CI 업데이트 또는 해제와 같은 권장 또는 자동화된 정정 작업이 포함됩니다.

사용 사례 예: 구성된 필터는 다양한 검색 소스에 사용할 수 있습니다. 예를 들어 Nokia Altiplano가 있습니다. 감사를 수동으로 실행하는 경우 적절한 필터를 선택하여 해당 특정 소스에서 검색한 CI만 확인하여 효율적인 목표 감사를 수행할 수 있습니다.

관련 정보

[감사에 대한 필터 구성](#)

[통신 불일치 감사 및 정정에 대한 예](#)

통신 불일치 감사 및 정정에 대한 예

다음 예제에서는 시나리오에서 감사가 작동하는 방식을 보여 줍니다.

시작하기 전에

필요한 역할: 관리자

이 태스크 정보

카드(Card04)가 Slot04에서 처음 발견되었습니다. 나중에 Card04는 물리적 네트워크에서 Card05로 대체되었지만 CMDB는 여전히 Slot04에 Card04를 표시합니다. 새 검색 실행이 실행되면 Card05가 검색되고 동일한 슬롯에 추가되어 CMDB에서 데이터 충돌이 발생합니다.

감사 동작:

- 통신 불일치 감사는 이러한 불일치를 탐지하고 실패한 감사 기록(예: AUDR0001283)을 작성합니다.
- 불일치에 대한 자세한 설명과 함께 후속 작업(#: TASK0020215)이 자동으로 생성됩니다.

작업 설명 예(잘못된 관계 수):

Card04는 2.5일 전에 마지막으로 발견되었습니다.

다음 CI 간의 관계

CI	모델
슬롯04	DEMO 20532트리
카드04	노키아 7360 FANT-F 카드 모듈
카드05	<모델이 식별되지 않음>

다음 스냅샷은 불일치 및 정정 프로세스를 이해하는 데 도움이 됩니다.

Run Audit

Related Links

[View Dashboard](#)
[Run Point Scan](#)

Audit Results (45)

Follow On Tasks (39)

Number

Search

Actions on selected rows...

Audit Results

	Created	Document	State	Column name	Desired value	Discrepancy value	Follow on task	Number	Template	Threshold	Stability
	2024-11-26 04:34:33	Configuration Item: Card40	Failed	Most recent discovered date	is not within threshold 2024-11-25 00:59:44	2024-11-22 00:59:45	TASK0020215	AUDR0001283	(empty)	Pending	Pending

UI Actions

Search

Actions on selected rows...

None

All + Name contains name + Table starts with cert_follow_on_task

Name	Table	Comments	Form action	List action	Active	Order	Condition	Updated
tsome	cert_follow_on_task	Search	Search	Search	Search	Search	Search	Search
Remediate	Follow On Task (cert_follow_on_task)		true	false	true		1,000 new TsomRemediation() performedLateCondit...	2024-11-21 08:27:48

SM Service-now: Yaron Nechushtan [maint...]

Work notes • 2024-11-26 04:37:39

TSOM CI Decommission

=====

Card: Card40

LCS set to 'End of life'

LCSS set to 'Retired'

Relationships removed:

Slot40 ==> Card40

SA System Administrator

Field Changes • 2024-11-26 04:34:33

Description

Card40 was last discovered more than 2.5 days ago.

Relationships between the following CIs:

CI: Slot40 (8b2beb4247ceda10f04f83ac416d4398), Model: DEMO 20532Tree (1ba577524c1b3110f8772646dabeb9bb),

CI: Card40 (0b2beb4247ceda10f04f83ac416d4399), Model: Nokia 7360 FANT-F CARD MODULE (3af9617de5928110f877657a333391e0),

CI: Card41 (832beb4247ceda10f04f83ac416d439a), Model: ()

SA System Administrator

Field Changes • 2024-11-26 04:34:32

Active

true

Audit

Service Operation CMDB Compliance Audit

Configuration item

Card40

Impact

3 - Low

Number

TASK0020215

Opened by

System Administrator

Priority

4 - Low

Short description

Most recent discovery date not within configured threshold.

State

Open

프로시저

- 이동 모두 > 준수 > 감사 > 통신 불일치 감사 > 감사 실행.
실패한 각 감사 기록(예: TASK0020215)에 대해 후속 작업이 자동으로 생성됩니다.
- TASK0020215** 선택하고 불일치 설명을 검토합니다.

주:

다음은 “잘못된 관계 수” 시나리오에 대해 생성된 TASK0020215 설명의 예입니다. 다른 시나리오와 환경에는 다른 설명이 있을 수 있습니다.

후속 작업에는 불일치에 대한 자세한 설명이 포함되어 있습니다. 설명에서 볼 수 있듯이 Card04 CI에 불일치가 있습니다.

- 정정하려면 다음 단계를 수행합니다.

a. 이동 모두 > 시스템 정의 > **UI** 작업.

b. 정정을 선택합니다.

정정 버튼은 TSOM CI ## 하위 플로우를 트리거하는 UI 작업입니다. 이 하위 플로우는 다음과 같습니다.

- 오래된 Card04 CI를 폐기합니다.
- 잘못된 Slot04 → Card04 관계를 제거합니다.
- CMDB 기록을 네트워크 상태와 동기화합니다.

결과
정정 후:

- 후속 작업(TASK0020215)은 해결 단계를 요약하는 작업 메모로 업데이트됩니다.
- Card04 CI가 폐기됨으로 표시되고 잘못된 관계가 제거됩니다.
- 이제 CMDB가 네트워크에서 검색된 최신 상태에 맞춰집니다.

i 주:

정정 사용자 지정 - 통신 불일치 감사는 정정 하위 플로우 예시와 함께 제공됩니다. 운영 요구 사항에 맞게 Flow Designer를 사용하여 사용자 지정 하위 플로우를 생성하고 첨부할 수 있습니다.

관련 정보

[통신 불일치 식별 및 조정](#)

[불일치 식별 - 불일치 유형](#)

[통신 불일치 식별 및 조정과 함께 설치되는 시스템 구성요소](#)

[통신 불일치 식별 및 조정 활성화](#)

[Defining UI actions](#)

속성 값 불일치에 대한 보고서 생성

CMDB 360을 사용하여 서로 다른 검색 소스 간 또는 검색 소스와 CMDB 기준선 간의 속성 값 불일치를 강조하는 보고서를 생성합니다.

시작하기 전에

필요한 역할: 관리자

이 태스크 정보

속성 값 불일치 보고서는 동일한 CI(구성 항목)를 업데이트하는 여러 소스에서 상충하는 데이터를 식별하는 데 도움이 됩니다. 이를 통해 CMDB 전체에서 데이터 품질과 무결성을 개선할 수 있습니다.

프로시저

1. 다음으로 이동 모두 > **CMDB** 작업 공간 > **CMDB 360**.
2. 쿼리 만들기를 선택합니다.
3. 쿼리 유형 창에서 **Compare attribute** values(특성 값 비교)를 선택합니다.
4. 쿼리 매개변수를 정의합니다.
 - a. 분석할 **CI** 클래스 (예: cmdb_ci_computer)를 선택합니다.
 - b. 필터를 적용하여 범위를 좁힙니다(예: 검색 소스에 "TSOM"이 포함되어 있음).
 - c. 속성을 선택하여 여러 검색 소스 간의 값을 비교합니다.
 - d. 비교할 소스(예: 검색, SCCM)를 선택합니다.
 - e. 옵션: 디스커버리 소스 데이터를 현재 CMDB 기준선과 비교하려면 CMDB와 비교 를 선택합니다.
 - f. 저장을 클릭하고 쿼리 이름을 제공한 다음 실행을 클릭하여 결과를 생성합니다.
5. 옵션: 일정을 클릭하여 정의된 간격으로 실행되도록 쿼리를 설정합니다.

결과

이 보고서에는 여러 소스의 특성 값 비교가 표시됩니다.

- 소스 1: 현재 CMDB 기록.
- 소스 1 값: 최근 디스커버리 소스에서 업데이트한 값입니다.
- 소스 2: 이전 검색 소스.
- 소스 2 값: 이전 디스커버리 소스에서 제공한 값입니다.

CI가 여러 소스에서 업데이트된 경우 각 불일치는 별도의 기록으로 나열됩니다.

관련 정보

[CMDB 360에서 속성 값 불일치 구성](#)

[조정 규칙을 사용하여 CI 속성 업데이트 제어](#)

Telecommunications Service Operations Management

참조

여러 유형의 구성요소가 애플리케이션 및 플러그인과 함께 Telecommunications Service Operations Management 설치됩니다.

통신 API 알림과 함께 설치되는 시스템 구성요소

관리자는 사용자 역할을 할당하여 API 알림 데이터베이스 테이블에 대한 액세스 권한을 부여할 수 있습니다. 토픽 [sn_api_notif_mgmt_topic] 및 토픽 구독 [sn_api_notif_mgmt_subscription] 테이블에 대한 다음과 같은 표준 역할이 시스템에 포함됩니다 ServiceNow .

통신 API 알림 역할

역할	설명
sn_api_notif_mgmt.topic_subscription_viewer	주제 및 주제 구독 테이블에 대한 읽기 권한을 사용 가능하게 하는 역할입니다.
sn_api_notif_mgmt.topic_creator	주제 테이블에 대한 생성, 읽기 및 편집 권한을 활성화하는 역할입니다.
sn_api_notif_mgmt.subscription_creator	주제 구독 테이블에 대한 작성 및 읽기 권한을 활성화하는 역할입니다.
sn_api_notif_mgmt.subscription_admin	다음 권한을 사용할 경우 활성화하는 역할입니다. • 주제 및 주제 구독 테이블에 대한 접근 권한을 생성하고 읽기 권한입니다. • 등록 상태를 변경하여 토픽 구독의 등록을 취소합니다.

관련 정보

[통신 API 알림을 통한 외부 이벤트 관리](#)

[통신 API 알림 구성](#)

Nokia Altiplano와 함께 설치되는 시스템 구성요소

시스템 속성은 검색 옵션 및 성능 설정을 포함하여 커넥터의 작동 방식을 제어합니다.

Nokia Altiplano와 함께 설치되는 속성

속성	설명
<code>sn_sgc_altiplano.enable_onu_discovery</code>	ONU 장치 검색 및 OLT와 ONU 간의 논리적 연결을 활성화하거나 비활성화합니다. - 기본값: True - 위치: 모두 >서비스 그래프 커넥터>Nokia Altiplano>속성 또는 '*altiplano'로 필터링하는 시스템 속성 [sys_properties] 테이블
<code>sn_sgc_altiplano.devices_list_batch_size</code>	Altiplano REST API 호출의 배치 크기를 제어합니다. - 기본값: 1000 - 위치: 모두 >서비스 그래프 커넥터>Nokia Altiplano>속성 또는 '*altiplano'로 필터링하는 시스템 속성 [sys_properties] 테이블
<code>sn_sgc_altiplano.parallel_number_of_data_source_jobs</code>	Altiplano 데이터를 수집하기 위한 병렬 작업의 수입니다 ("병렬 로드 사용" 설정 필요). - 기본값: 2 - 위치: 모두 >서비스 그래프 커넥터>Nokia Altiplano>속성 또는 '*altiplano'로 필터링하는 시스템 속성 [sys_properties] 테이블
<code>sn_sgc_altiplano.onu_ci_class</code>	ONU가 ONU 또는 ONT CI 클래스로 저장되는지 여부를 정의합니다. - 기본값: ONU - 위치: 모두 >서비스 그래프 커넥터>Nokia Altiplano>속성

Nokia Altiplano와 함께 설치되는 속성

속성	설명
	또는 '*altiplano'로 필터링하는 시스템 속성 [sys_properties] 테이블

관련 정보

[Nokia Altiplano를 통한 통신 검색](#)

[Nokia Altiplano 서비스 그래프 커넥터 구성](#)

[store-future: BEGIN review]

Nokia NSP와 함께 설치되는 시스템 구성요소

시스템 속성은 검색 옵션 및 성능 설정을 포함하여 커넥터의 작동 방식을 제어합니다.

Nokia NSP와 함께 설치되는 속성

속성	설명
<code>sn_sgc_nsp.parallelNumberOfDataSourceJobs</code>	NSP 데이터를 수집하기 위한 병렬 작업 수입니다("병렬 로드 사용" 설정 필요). - 기본값: 1 - 위치: 모든 > 서비스 그래프 커넥터 > Nokia NSP > 속성 또는 시스템 속성 [sys_properties] 테이블을 이름 '*nsp'로 필터링하여

[End]

통신 불일치 식별 및 조정과 함께 설치되는 시스템 구성요소

시스템 속성은 TSOM Visibility 플러그인(sn_tsom_core)의 일부이며 통신 불일치 식별 및 조정 로그(TSOM CMDB 감사)를 제어합니다. TSOM 가시성 플러그인은 TSOM 가시성 애플리케이션의 지원 서비스 역할을 하며, 통신 디스커버리 및 통신 불일치 식별 및 조정 솔루션에서 공유되는 논리를 포함합니다.

통신 불일치 식별 및 조정 시스템 속성(CMDB 감사에 영향을 미침)

속성 이름	권장/기본값	설명
<code>sn_tsom_core.audit.log.level</code>	통신 불일치 감사의 로깅.	디버그, 정보, 경고 기본값: 정보
<code>sn_tsom_core.audit.relationship_types</code>	통신 불일치 감사에서 처리할 관계 유형의 표시된 이름입니다.	범표로 구분된 관계 유 기본값: Contains::Co

통신 불일치 식별 및 조정 시스템 속성(CMDB 감사에 영향을 미침)

속성 이름	권장/기본값	설명
sn_tsom_core.audit.discovered_date.diff.threshold.in.days	가장 최근에 검색된 날짜 임계치 (일)입니다. 통신 불일치 감사에 사용됩니다.	문자열 기본값: 2.5
sn_tsom_core.audit.suppress_CI_Model_missing_discrepancy_task_creation	테이블에 대한 CI 모델 누락 불일치 작업의 생성을 억제합니다. 통신 불일치 감사에 사용됩니다.	cmdb_ci_ni_telco_equip cmdb_ci_container_slot
sn_tsom_core.audit.equipment_tables	통신 불일치 감사에서 처리할 장비 테이블입니다.	섬표로 구분된 장비 테이블 기본값: cmdb_ci_network
sn_tsom_core.audit.interface_card_tables	통신 불일치 감사에서 처리할 인터페이스 카드 테이블입니다.	섬표로 구분된 카드 테이블 기본값: cmdb_ci_interface
sn_tsom_core.audit.interface_tables	통신 불일치 감사에서 처리할 네트워크 인터페이스 테이블입니다.	섬표로 구분된 인터페이스 테이블 기본값: cmdb_ci_ni_interface
sn_tsom_core.audit.slot_tables	통신 불일치 감사에서 처리할 슬롯 테이블 이름입니다.	슬롯 테이블 이름의 섬표 기본값: cmdb_ci_container_slot
sn_tsom_core.audit.subslot_tables	통신 불일치 감사에서 처리할 하위 슬롯 테이블 이름입니다.	섬표로 구분된 하위 슬롯 테이블 기본값은 cmdb_ci_container_slot

통신 불일치 식별 및 조정 감사

감사 이름	설명
통신 불일치 감사	cmdb_rel_ci 테이블의 기록을 감사합니다. Nokia Altiplano 또는 NSP에서 검색한 기반으로 상위 및 하위 CI를 확인합니다. - 가장 최근 검색 날짜 - CI 모델 - CI 모델 (예외 경우).

통신 불일치 식별 및 조정 감사

감사 이름	설명
	주: 모든 불일치 감사는 장비 CI에 적용되는 필터링 메커니즘을 사용합니다. 기본 필터링은 discovery_source CONTAINS TSOM입니다.
통신 논리적 연결 불일치 감사	cmdb_ci_ni_logical_path 테이블의 기록을 감사합니다. Nokia Altiplano 또는 NS-3000에 대해 다음을 확인합니다. - 빈 엔드포인트(포트 A 또는 포트 Z) - 각 엔드포인트 연결에만 연결되어야 합니다.
통신 네트워크 토폴로지 불일치 감사	cmdb_ci_network_topology 테이블의 기록을 감사합니다. Nokia Altiplano 또는 NS-3000에 대해 다음을 확인합니다. - 최소 1개 이상의 Contains:Contained Members:Member Of 논리적 연결과의 관계.

관련 정보

[통신 불일치 식별 및 조정](#)[통신 불일치 식별 및 조정 활성화](#)[통신 불일치 감사 실행](#)