

# CNCアップグレードの導入事例

## 内容

---

[はじめに](#)

[概要](#)

[背景](#)

[実稼働ネットワーク](#)

[CNC 4.1からCNC 7.1への移行ワークフロー](#)

[CNCアーキテクチャと他のコンポーネントとの統合](#)

[アーキテクチャ図](#)

[ネットワーク図](#)

[CNC 4.1→7.1詳細な移行ワークフロー](#)

[使用例](#)

[L2VPN \(EVPNベース\) サービスプロビジョニング](#)

[カスタムNSOテンプレート](#)

[L3VPN \(VRFベース\) サービスプロビジョニング](#)

[カスタムNSOテンプレート](#)

[トラフィック エンジニアリング](#)

[TC1トラフィック \(低遅延\)](#)

[TC4トラフィック \(認定帯域幅\)](#)

[sZTPを使用したデバイス起動](#)

[ZTP後のオーケストレーション \(自動化による\)](#)

[CNCでのBandwidthNotification Message\(BNM\)処理](#)

[一時的な \(一時的なイベント\) 変更](#)

[BNMのMDT](#)

[カスタム自動化プレイブックを通じてDay-2ネットワーク運用を標準化](#)

[Cisco CNC 7.1アップグレードにおけるTACACS+統合継続性](#)

[SplunkへのCNCおよびCDG Syslog転送](#)

[OneFMへのアラーム転送](#)

[毎日のCNCバックアップの自動化](#)

[課題](#)

[Crossworkバージョンでの大きなジャンプ](#)

[インプレースアップグレードなし](#)

[ロールバックオプションのない導入の落とし穴](#)

[導入後の診断検証の制約](#)

[HIカスタムKPI作成手順の変更](#)

[BNMプレイブックトリガースクリプトのAPIタイムアウト](#)

[BNM処理およびプレイブックのトリガー設計の変更](#)

[元のアラート設計の制限](#)

[KPIフレームワーク変更の影響](#)

[過剰なプレイブックトリガー](#)

[自動化ロジックの再設計](#)

[成果](#)

---

[デバイスアラームの抑制](#)

[アウトオブバンドの変更](#)

[L2/L3 VPNの調整](#)

[スケジュールの影響](#)

[観察](#)

[同様のアップグレードに関する推奨事項](#)

[メンテナンスモードの依存関係によるCNCバックアップの失敗](#)

[運用への影響](#)

[軽減のための戦略](#)

[結果と成果](#)

[Splunkへのsyslogの転送](#)

[デバイスのグループ化の移行の問題](#)

[帯域幅が著しく低下しているデバイスを特定](#)

[デバイステレメトリ構成の削除](#)

[MDT収集のトラブルシューティング](#)

[NSO 6.4.1.1におけるHA動作の変更とコンセンサスアルゴリズムの調整](#)

[NSOバージョンアップグレードおよびパッケージ互換性の強化](#)

[規模に応じたKPIの有効化に関する問題](#)

[RESTCONFノースバウンドAPIは管理者アクセスに制限されています](#)

[戦略的な成功要因としての自動化](#)

[学ぶべきレッスン](#)

[アップグレードは簡単ではない](#)

[CXは重労働を行わなければならない](#)

[自動化ツールキットの必要性](#)

[移行中のデュアルコントローラ競合の回避](#)

[MOPはSacrosanctではありません](#)

[TACケースの有効性](#)

[CNC BUと効果的なナレッジサポートを締結](#)

[CNCアップグレードのベストプラクティス](#)

[最適化されたアップグレード戦略の計画](#)

[特に不変のパラメータに対しては、厳格な導入前検証が不可欠](#)

[製品に触れる前に専用の検証環境を使用](#)

[分散したクロスワークコンポーネントの実績に基づいたサイジング](#)

[反復的で大量の作業の自動化](#)

[パラレル実行時の二重クローズドループ制御の回避](#)

[アップグレード構造化影響評価の実行](#)

[統合サーフェス全体での互換性と動作のテスト](#)

[堅牢な移行前データエクスポート戦略の確立](#)

[検証ゲートが組み込まれたデバイスの一括移行](#)

[NSO統合によるアウトオブバンド構成変更の処理](#)

[変更の凍結を強く強調する](#)

[結論](#)

[用語集](#)

[参照資料](#)

---

# はじめに

このドキュメントでは、Cisco CNC 4.1から7.1への固定ワイヤレスネットワークのリフトアンドシフトによる複雑で大規模な移行の事例を説明します。

## 概要

このホワイトペーパーでは、Cisco Crosswork Network Controller(CNC)バージョン4.1からバージョン7.1への大規模な固定ワイヤレスネットワークの移行に関する詳細なケーススタディを紹介します。インプレースアップグレードメカニズムが存在しないため、移行には完全なリフトアンドシフト導入が必要となり、2,000を超えるネットワークデバイスと相互に依存する複数のシステム間で、アーキテクチャ、運用、および統合の面で著しい複雑さが生じました。この調査では、複数の分野で直面する課題を調査しています。

主な成果として、特に大量のワークフローのスケラビリティ、精度、および運用上の決定性を確保するうえで、自動化が果たす重要な役割が明らかになります。その結果、実稼働環境は制御されたラボ条件からかなり逸脱しており、適応型のトラブルシューティング、反復的な検証、TACおよびビジネスユニットエンジニアリングチームとの継続的な関与が必要であることが明らかになりました。この作業は、将来のCNCアップグレードと大規模なオーケストレーションプラットフォーム移行の参照計画として役立つ実践的な洞察、検証済みの手法、推奨されるベストプラクティスに貢献します。

## 背景

5Gネットワークの急増、接続デバイスの急速な普及、企業およびコンシューマ環境のデジタル化により、トラフィック量が大幅に増加し、規模に応じて安全かつ確実に提供する必要があるサービスの多様性が増しています。コミュニケーションサービスプロバイダー(CSP)は、非常に動的なネットワークで運用を行うようになりました。従来のサイロ化された運用ツールでは、複雑さが増し、ユーザエクスペリエンスが低下し、運用コスト(OpEx)が増大することがよくあります。

競争力を維持するため、自動化、仮想化、SDNの原則、分析主導の自己最適化ネットワークに基づいて構築された最新の運用モデルを採用する事業者が増えています。

Cisco Crosswork Network Controller(CNC)は、運用ワークフローを簡素化し、総所有コスト(TCO)を削減し、マルチベンダーの転送ネットワーク全体でインテントベースの自動化を実現することで、この変革をサポートするように設計されています。CNCは、サービスプロビジョニング、ネットワークヘルスマニタリング、リアルタイム最適化のための統合プラットフォームを提供し、事業者が大規模なIPネットワークをよりプロアクティブかつ効率的に管理するための一元

化された窓口を提供します。

基盤となるCrossworkインフラストラクチャは、すべてのCNCアプリケーションが実行される、復元力のあるスケーラブルなクラスタフレームワークを提供します。CNC 7.1では、Optimization Engine、Active Topology、Change Automation、Health Insights、Element Management Functions(EMF)、Service Health、Crosswork Workflow Manager(CWM)などのモジュールが含まれ、それぞれがエンドツーエンドのオーケストレーションと保証に貢献します。

ただし、CNCのアップグレードには固有の課題があります。CNCはインプレースアップグレードをサポートしておらず、新しい環境と既存の環境が並行して構築され、すべてのデータとサービスが新しいバージョンに移行される、完全なリフトアンドシフト導入が必要です。このケーススタディでは、他のすべてのサービスプロバイダーにバックボーンサービスを提供することをサポートするオーストラリアの主要なサービスアグリゲータを対象に、CNC 4.1からCNC 7.1への大規模なアップグレードを検証します。

複数のカスタマイズされた変更自動化プレイブック、カスタムのヘルスインサイトKPI、L2/L3 VPNサービス調整要件、およびセキュアZTPの必要性により、移行は特に複雑でした。

大規模バージョンへの移行により、アーキテクチャや動作の内部の変化が生じたため、新しいリリースでの既存のユースケースの動作を予測することが困難になり、さらに不確かなものとなりました。このため、すべてのユースケースで包括的な検証と調整が必要でした。

ハイブリッド/ワーカーノード数、CDG配分、PCEサイジングなど、最適なりソース割り当てを決定し、既存のリソースのフットプリントを維持できるかどうかについて十分な計画を行いました。

CNC 7.1の初期導入と検証は、社内のCALOラボで実施され、実験、設定の調整、および信頼性の確立を行うための安全な環境が提供されました。その後、実稼働環境を厳密に反映した内部テスト環境で導入しました。最後のフェーズでは、CNC 7.1を実稼働環境に導入し、デバイスレベルの設定変更を適用し、すべてのデバイスと関連サービスを新しいコントローラに段階的に移行しました。

## 実稼働ネットワーク

エアギャップ生産ネットワークはオーストラリアの広い地域に広がっています。NCSからASR9Kまで、2,000以上のデバイスが存在するCNCは、トポロジのライブビューを提供することで、これらすべてのデバイスを管理しました。約2,000台のデバイスは、ローカルではIOS-XR 24.3.2を実行するSWR(Small Wireless Router)と呼ばれるNCS540で、ローカルではLWR(Large Wireless Router)と呼ばれるASR-9Ks (バージョン7.5.2) です。

Crossworkセットアップは、3つのハイブリッドノードと2つのワーカーノードで構成されました

。デバイスに対して合計5つのCDGがあり、そのうち4つがアクティブノード、1つがスタンバイノードでした。プールには1つのスタンバイCDGしかないため、この方法では保護が制限されていました。しかし、要件を考慮すると、これは先に進むことを与えられました。すべてのVMが単一のデータセンターにあるという事実も、1つのスタンバイのみで処理を進める方が容易な決定でした。

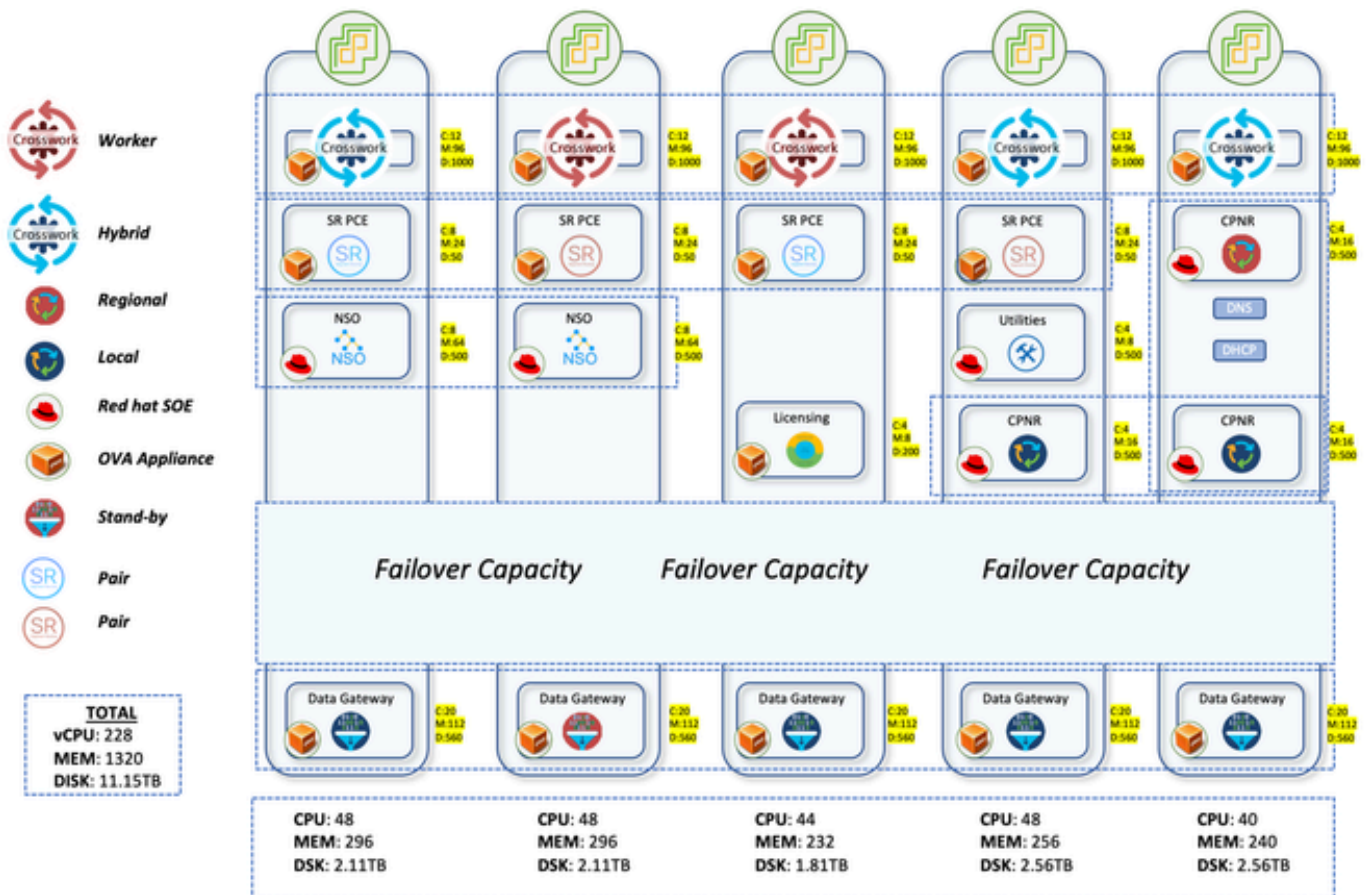
CDGは、SNMP、CLI、GNMIなどのさまざまなプロトコルを介してデバイスからのデータ収集を処理するコンポーネントです。CDGが収集したデータは、内部のkafkaを通じてCrossworkに公開されます。Crossworkにオンボーディングされたデバイスは、CDGに接続する必要があります。これにより、データゲートウェイはデバイスに接続し、デバイスデータを取得できます。

CDGのデバイス配分についても検討が行われました。以前の導入では、デバイスをCDG間でランダムに分散していましたが、このため、一部のCDGではより多くのデバイスを使用し、1 ~ 2のCDGではより少ないデバイスを使用するという、非常に偏った分布が生じていました。これにより、一部のCDGでは過剰消費と過負荷が発生し、プロビジョニングが不十分なCDGもあります。

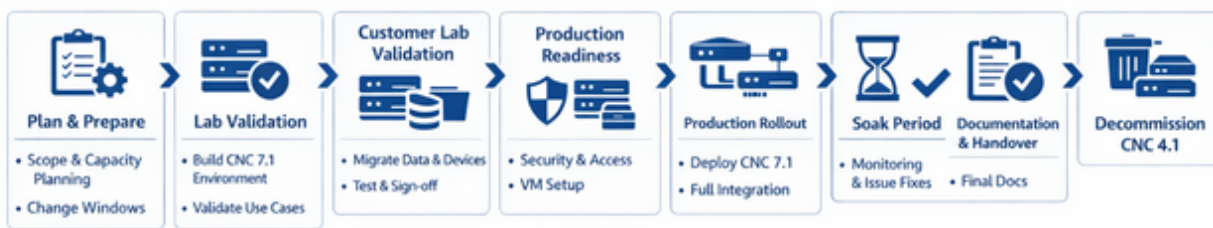
このアップグレードでは、4つのアクティブなCDGにそれぞれ700台のSWRを配布することを検討しました。この結果、最初の3つのCDGには2100台のSWRが収容されました。 インターフェイスの前面にある重いLWRはすべて、4番目のCDG用に予約されています。カウントが30の非常に小さい数ですが、この割り当てにより、これらのデバイスからさらに多くの収集が行われた場合でも、CDGに大きな負荷がかかりません。以降にSWRのオンボーディングを行う場合も、4<sup>th</sup> CDGに進みます。これにより、最初の3つのCDGでは均等な分散が確保され、4番目のCDGでは新しいデバイスを取り込むためのより多くのスペースが確保されました。

SR-PCEは2ペアで導入されました。つまり、4つのVMが異なるホストマシンに分散されました。一方のペアは7つのPOIサイトを管理し、もう一方のペアは残りの8つのPOIサイトを管理します。CNC GUIでのトポロジの更新は、SR-PCEを使用して行われます。ネットワークのトポロジは、他のLWRルータとのBGP-LSピアリングを通じて学習されます。このコンポーネントは、トラフィックを別のパスに誘導するコントローラの役割を果たす、すべてのトラフィックエンジニアリングのユースケースでも使用されます。

すべてのサービスプロビジョニングとデバイス設定のユースケースを処理するには、NSOをCNCと組み合わせて使用する必要があります。実稼働ネットワークでは、バージョン6.4.1.1の2つのNSOを導入し、ハイアベイラビリティモードで並行して動作させました。SR-PCE(Segment Routing Path Computation Element)は、CNCにトポロジアップデートを提供し、リアルタイムトラフィックエンジニアリングのユースケースを処理するために必要なコンポーネントです。ここでは、バージョン25.2.1の4つのSR-PCEが導入され、各PCEは2つの異なるLWRにピア接続されました。



## CNC 4.1からCNC 7.1への移行ワークフロー



CNCの導入では、Dockerベースの導入を進めることをお勧めします。しかし、クライアントがオンプレミスでDockerのセットアップを承認しなかったため、vCenterを使用して手動で導入する以外に選択肢はありませんでした。vCenter GUIで入力を複数回入力する必要があるため、スクリプトベースのスクリプトと比較すると、導入に時間がかかります。

CNCの導入が完了すると、必要なアプリケーションはすべて、BUが提供する自動アクションイン

ストールファイルを使用して導入されました。このファイルは、アプリケーションを一度にアップロードしてアクティブ化するため、手動で行う時間が短縮されます。プレミアム層は、Crosswork Optimization Engine、アクティブトポロジ、サービスの状態、要素管理機能、Crosswork Workflow Managerを含めて導入されました。これに加えて、変更の自動化とヘルスインスایتを含むアドオンパッケージもセットアップされました。

CWMとSHには使用例がありませんでした。しかし、これらのアプリケーションが次のバージョンで提供するユースケースの一部に関心があったため、それにもかかわらず導入されました。

アプリケーションがセットアップされたら、次のステップは古いバージョンのCNCからデータを移行することでした。これは主に、クレデンシャルプロファイル、プロバイダー、タグ、カスタムプレイブック、カスタムKPI、ルール、sZTPバウチャー、およびその他のデータで構成されます。CNCは、これらすべてについてエクスポートオプションを提供します。これらのオプションは、活用して新しいCNCにインポートできます。

これらを設定したら、デバイスの移行を開始することが賢明です。アップグレードの場合、新しいCNCが古いCNCよりも新しいサブネットに導入されると、新しいCNCとの到達可能性を提供するために、デバイスでACLを変更する必要があります。各デバイスに手動でログインして設定を変更する必要があるため、これは時間のかかるプロセスです。

これらのACLの変更が完了したら、次の手順では、デバイスを新しいCNCにインポートし、CDGに接続します。到達可能性が適切で、SSHおよびSNMPクレデンシャルが正しい場合、デバイスはCNCで到達可能と表示され、NSO ( ネットワークサービスオーケストレータ ) にもオンボーディングされます。

NSOフロントでは、CNCがNSOと通信できるようにするため、必要なすべてのパッケージが適切で運用上アップしている必要があります ( その逆も同様 )。たとえば、CNCからNSOにデバイスを自動的にオンボーディングするには、DLM機能パックが必須です。同様に、NSOがデバイスにMDTセンサーパスを設定する必要がある場合は、TM-TCパッケージをNSOに導入する必要があります。主な目的は、ユースケースに応じて、関連するパッケージをNSOに展開することです。

これらの必要なパッケージ、特にTransport-SDNパッケージを手動で導入する代わりに、プロビジョニング用の自動スクリプトが開発されました。CNC 7.1アップグレードでは、TSDNパッケージにアップデートが導入されました。これらの更新されたパッケージは、アップグレードされた環境でL2/L3プロビジョニングを継続的にサポートするために、NSOサーバに導入することを目的としています。このスクリプトは、更新されたTSDNパッケージのインストールを自動化し、必要なメタデータをNSOにロードして、必要に応じてサービスをプロビジョニングできるようにします。

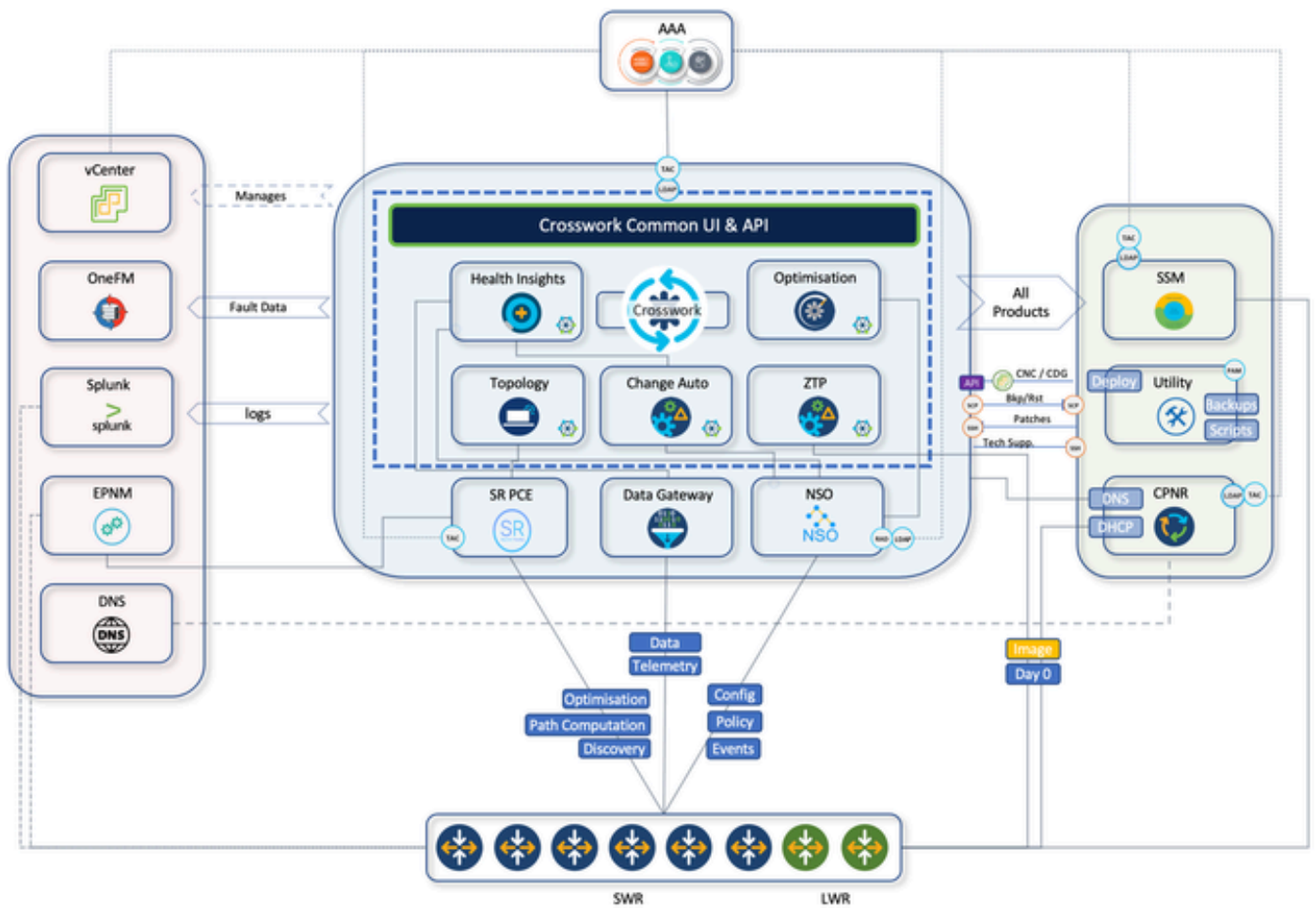
Cisco Smart Software Manager(SSM)Licensing Serverの1つのインスタンスとCisco Prime Network Registrar(CPNR)の3つのインスタンスも異なるホストに導入できます。

# CNCアーキテクチャと他のコンポーネントとの統合

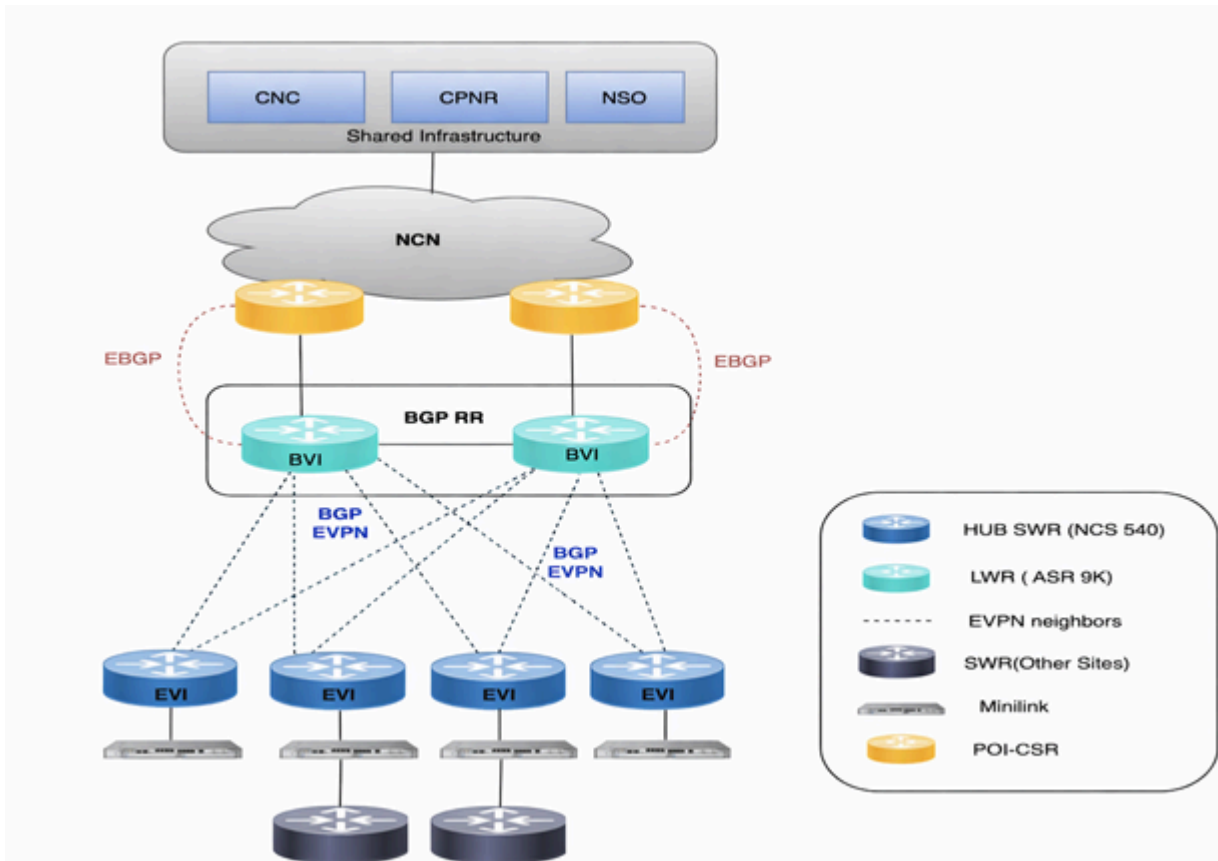
CNCは、統合されたUIを通じて、導入されたサービスのプロビジョニング、最適化、可視化を単一のプラットフォームで行います。ここでは、CNCプラットフォームスイートに存在するCNC内部コンポーネントと使用例の概要を示します。

- クロスワークトポロジ(CAT):
  - CNC VMノードに分散された内部コンポーネントアプリケーション
  - 調整済みインベントリのエンドツーエンドの可視性をリアルタイムで提供
  - 複数のデータソースのインベントリ情報を1つのディスプレイに統合
  - トランスポートネットワークパスの計算
  - トポロジディスカバリ
- クロスワーク最適化エンジン(COE):
  - CNC VMノードに分散された内部コンポーネントアプリケーション
  - リアルタイムネットワーク最適化
  - リアルタイムのトポロジ可視化
  - SR-TEの可視化とプロビジョニング
  - RSVP-TEの可視化とプロビジョニング
  - オンデマンド帯域幅
- Crosswork health insight(CHI):
  - CNC VMノードに分散された内部コンポーネントアプリケーション
  - KPIの監視
  - アラートダッシュボード
- Crosswork change automation(CCA):
  - CNC VMノードに分散された内部コンポーネントアプリケーション
  - すぐに使えるプレイブックを備えた開発運用ツール
  - 希望の時間にプレイを実行できるスケジューリング機能
  - 改善策として推奨プレイに組み込むKPIアラートが高い

## アーキテクチャ図



ネットワーク図



## CNC 4.1 → 7.1 詳細な移行ワークフロー

レガシーCNC 4.1からCNC 7.1へのエンドツーエンドの段階的移行 (バージョンに関係なく、CNCのアップグレードで同じフローに従うことができます)

計画	ラボ	お客様のラボ	実稼働準備	実稼働ロールアウト	ソーク期間	引き渡し	使用停止
フェーズ 1							
1 計画および準備							
範囲と計画 ・ スコープの定義 ・ キャパシティ プランニング				SCHEDULING ・ ウィンドウIDの変更 ・ 関係者との連携			
▼							

フェーズ 2

2 内部ラボ検証

<p>インフラストラクチャ</p> <ul style="list-style-type: none"><li>・ CNC 7.1の構築 (ハイブリッド/ワーカー)</li><li>・ アプリのインストール</li><li>・ NSOとHAの導入</li><li>・ SR-PCEペアを導入する</li></ul>	<p>検証</p> <ul style="list-style-type: none"><li>・ すべての使用例を検証</li><li>・ 機能サインオフ</li></ul>
---	--



フェーズ 3

3 カスタマーラボ検証

<p>インフラストラクチャの構築</p> <ul style="list-style-type: none"><li>・ CNC 7.1の構築 (ハイブリッド/ワーカー)</li><li>・ アプリのインストール</li><li>・ NSOとHAの導入</li><li>・ SR-PCEペアを導入する</li></ul>	<p>データ移行</p> <ul style="list-style-type: none"><li>・ CNC 4.1アーティファクトのエクスポート</li><li>・ デバイスグループの再作成</li><li>・ CNC 7.1へのインポート</li><li>・ NSOパッケージの展開</li></ul>	<p>デバイスの到達可能性</p> <ul style="list-style-type: none"><li>・ ACLの更新</li><li>・ デバイスのインポートとCDGの添付</li></ul>	<p>サービスと監視</p> <ul style="list-style-type: none"><li>・ サービスの調整と同期</li><li>・ KPIイネーブルメントおよび収集ジョブ</li><li>・ BNMプレイブックスクリプトの有効化</li><li>・ HI/Grafanaの観測可能性</li><li>・ Radiusの統合</li><li>・ Splunkの統合</li><li>・ OneFMの統合</li><li>・ CNCバックアップの有効化</li></ul>
--	---	--	--

✓ラボでATPを実行して承認を取得



フェーズ 4

4 実稼働準備

セキュリティとアクセス ・ セキュリティレビュー ・ アクセスコントロールの設定	インフラストラクチャ ・ 実稼働VMのサイジングとセットアップ ・ ネットワークの検証
--	---



フェーズ 5

5 実稼働カットオーバー

○実稼働環境でフェーズ3のすべてのステップを繰り返します。

インフラストラクチャの構築 ・ CNC 7.1の構築 (ハイブリッド/ワーカー) ・ アプリのインストール ・ NSOとHAの導入 ・ SR-PCEペアを導入する	データ移行 ・ CNC 4.1アーティファクト (プロバイダ、クレデンシャルプロファイル、プレイブック、タグ) のエクスポート ・ デバイスグループの再作成 ・ CNC 7.1へのインポート ・ NSOパッケージの展開	デバイスの到達可能性 ・ ACLの更新 ・ デバイスのインポートとCDGの添付	サービスと監視 ・ サービスの調整と同期 ・ KPIイネーブルメントおよび収集ジョブ ・ BNMプレイブックの有効化 ・ HI/Grafana、Splunk、OneFM ・ CNCバックアップの有効化
---	---	---	---

✓実稼働ロールアウト

▼	
<p>フェーズ 6</p> <p>6 ソーク時間</p>	
<p>モニタリング</p> <ul style="list-style-type: none"> <li>安定性モニタリング</li> <li>パフォーマンスベースライン</li> </ul>	<p>問題管理</p> <ul style="list-style-type: none"> <li>問題の追跡と解決</li> <li>エスカレーションプロセス</li> </ul>
▼	
<p>フェーズ 7</p> <p>7 文書化と引き渡し</p>	
<p>ドキュメンテーション</p> <ul style="list-style-type: none"> <li>MOP、設計文書および運用文書</li> <li>アーキテクチャ図</li> </ul>	<p>引き渡し</p> <ul style="list-style-type: none"> <li>講習セッション</li> <li>ハンドオーバーの承認</li> </ul>
▼	
<p>フェーズ 8</p> <p>8 レガシーCNC 4.1の撤去</p>	
<p>クリーンアップ</p> <ul style="list-style-type: none"> <li>CDGからすべてのデバイスを切断します</li> <li>4.1 CDG VMを指すMDTエントリを削除します</li> <li>本番VMの削除</li> </ul>	<p>アーカイブ</p> <ul style="list-style-type: none"> <li>すべてのCNC 4.1エクスポートをアーカイブします</li> <li>最終監査および承認</li> </ul>

## 使用例

## L2VPN ( EVPNベース ) サービスプロビジョニング

L2VPNサービスは、複数のSWR間でレイヤ2イーサネット接続を提供し、一部のサービスはLWRにアンカーされています。CNCアクティブトポロジはサービスプロビジョニングに使用され、すべての環境固有のロジックはNSOカスタムテンプレートを使用して実装されます。

L2VPNプロビジョニングはDay2設定アクティビティとして扱われ、オペレータ提供のサービス属性が必要です。

### カスタムNSOテンプレート

環境固有の命名規則とインターフェイスの動作に合わせて、複数のカスタムテンプレートが作成されました。

- CT-l2vpn-swr-hub-and-lwr  
SWRハブおよびLWR上でハブ側の命名規則の違い ( ブリッジグループとブリッジドメイン ) を処理します。
- CT-l2vpn-swr-nonhub-100/101/102/105  
VLAN固有の各EVIのデフォルトのEVPNブリッジグループとブリッジドメインからZTPアップリンクインターフェイスを削除します。

これらのテンプレートにより、ネットワーク全体で一貫したEVPN設定が保証され、ハードウェアレベルの違いが抽象化されます。

## L3VPN ( VRFベース ) サービスプロビジョニング

L3VPNの使用例では、エンドポイントとして複数のSWRにレイヤ3サービスを提供できます。プロビジョニングはCNCアクティブトポロジを通じて実行され、カスタムNSOテンプレートを使用して環境固有の要件が実装されます。

L2VPNと同様に、これはDay-2の設定操作であり、オペレータ入力が必要です。

### カスタムNSOテンプレート

- CT-l3vpn-swr  
VRF固有のパラメータ ( AS番号、VRF名、プレフィックスセット、ルートポリシー名、ルート識別子 ) を収集し、接続されたルートのユーザ定義のルートポリシーによる再配布を含む、必要なBGPインポート/エクスポートポリシーを構築します。

## トラフィック エンジニアリング

CNCスイートのCrosswork Optimization Engine(COE)アプリケーションは、目的に基づいてネットワーク内のトラフィックフローを制御するのに役立ちます。

異なるインテント ( SLAメトリック ) を必要とする2種類のトラフィックがあります。

- TC1トラフィック：遅延の影響を受けやすいSLAにより、トラフィックが最小の遅延パス上に存在することを保証
- TC4トラフィック：TC4トラフィックで専用帯域幅を常に使用できるようにする最小帯域幅SLA

### TC1トラフィック ( 低遅延 )

TC1トラフィックが常に最も低い遅延パスで行われるようにするには、パスの計算基準を遅延として使用して、ヘッドエンドSWRでセグメントルーティング(SR)ポリシーを作成する必要があります。

これは、SRポリシーの作成を促進するCNCを使用して、特定のカラー1001の各ヘッドエンドSWRでオンデマンドのネクストホップ(ODN)設定を定義することで実現されます。

### TC4トラフィック ( 認定帯域幅 )

TC4トラフィックが常に専用の帯域幅のパスで行われるようにするには、帯域幅としてパス計算基準を使用してヘッドエンドSWRでSRポリシーを作成する必要があります。

これは、次の方法で実現できます。

- CNCのオンデマンド帯域幅(BoD)機能パック
- これらの設定でのCNC SRポリシー作成を使用した、特定のカラー1004用の各ヘッドエンドSWRでのオンデマンド次ホップ(ODN)設定の定義

BoD機能パックは、パス計算の基準として帯域幅を持つSRポリシーのパスを計算するために使用されます。ポリシーに割り当てられた帯域幅を追跡し、ライフサイクル中にポリシーの現在のパスを監視し続けます。

いずれかの時点で、BWODポリシーの現在のパッチに、コミットされた帯域幅を満たすのに十分な容量がない場合、BWODポリシーパスが再計算され、ポリシーが新しいパスに再ルーティングされます。このBWODポリシーの再ルーティングは継続的なプロセスであり、手動による介入は

必要ありません。

ある意味、BWODは遅延に対するSR-PCEと同様に、帯域幅の最適化をオンザフライで行います。

## sZTPを使用したデバイス起動

従来のインストールおよびサポートモデルでは、新しいデバイスを起動するプロセスでは、新しいコンポーネントの実装をインストール、設定、およびトラブルシューティングするために、インストーラがある程度の専門知識を必要としていました。また、オフサイトの場所で機器を事前準備する長いプロセスもあり、ソリューションのさまざまな部分で作業する多くの人によってサポートされます。

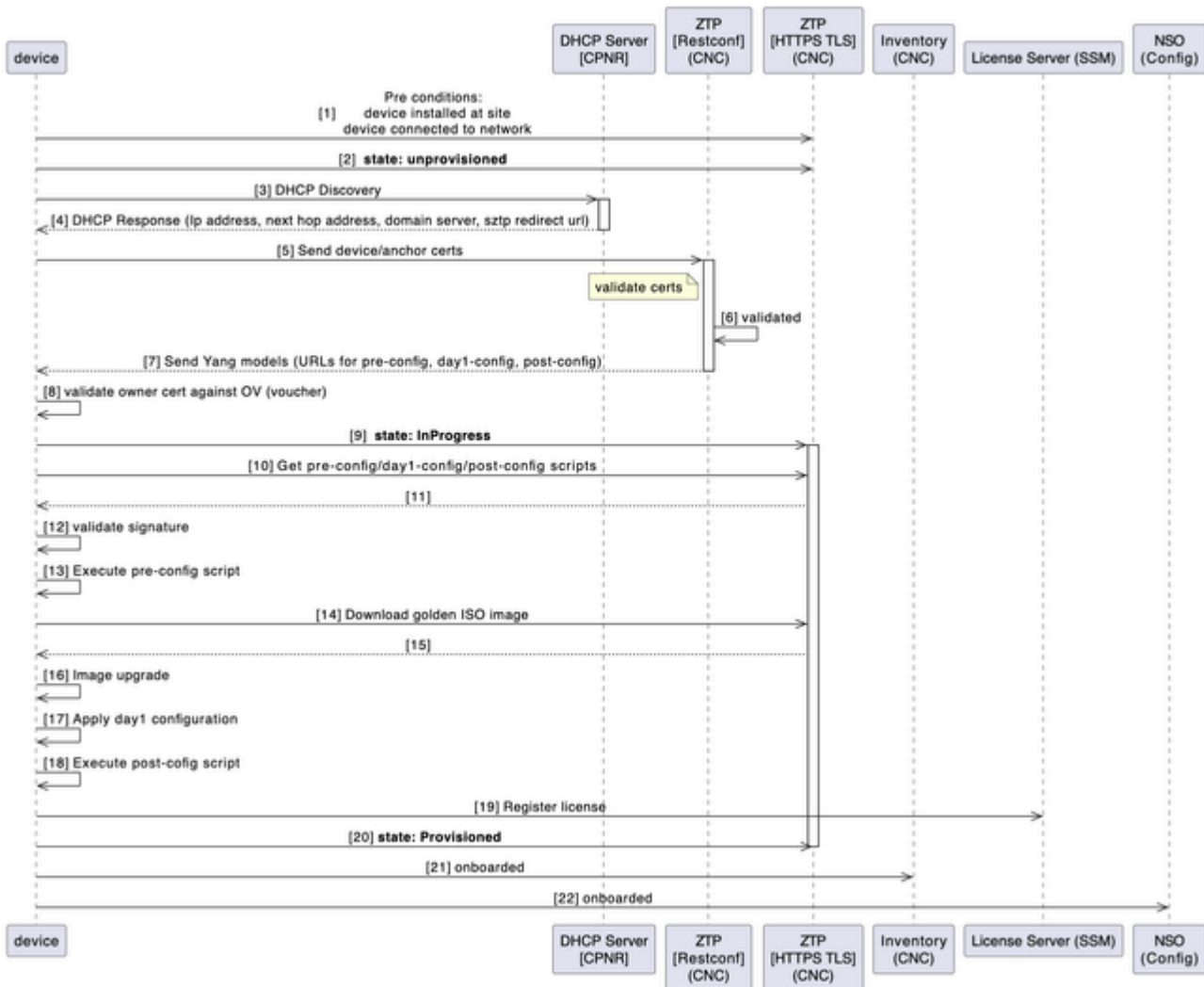
お客様の環境への導入が予定されている新しいSWRデバイスでは、CNCのセキュアなZTP (ゼロタッチプロビジョニング) アプリケーションを使用して、デバイスの起動プロセスが自動化されます。

ZTPワークフローはデバイスの初回起動時にトリガーされ、計画されたプラットフォームイメージと初期設定がダウンロードされます。この設定は手動による介入なしに適用する必要があります。

このデバイスは、CNCに自動オンボーディングされ、さらに調整されます。

次の図は、デバイス起動時のセキュアZTPプロセスのワークフローを示しています。

### Secure Zero Touch Provisioning



### ZTP後のオーケストレーション (自動化による)

ユーティリティホスト上のPython自動化は、構造化されたExcel入力 (チェーン単位) を使用して、エンドツーエンドのプロセスを調整および監査します。

- Day-1および構成後のアーティファクトを生成してCNCにアップロードします。
- CPNR予約 (SWRシリアルにバインドされたDHCPエントリ) を作成します。
- EPNMにデバイスを追加 (可視性/保証用)
- CNCでのZTP後のハウスキーピング：
  - SWRをCDG (テレメトリの宛先) に割り当てます。
  - デバイスグループとタグに接続します。
  - トポロジの可視化のために緯度/経度を更新
  - テレメトリストリーミングを有効にするためのBNM KPIプロファイルを添付

### CNCでの帯域幅通知メッセージ(BNM)処理

SWRは、WANポートの帯域幅に対応する同じ場所に配置されたMiniLinkスイッチからBNMを受信できます。これらの通知メッセージは標準ベースのCFMメッセージで、現在の実行レコード帯域幅(RBW)と、公称帯域幅(NBW)とも呼ばれる最大設定帯域幅が含まれます。

現在の帯域幅は、個々のマイクロ波リンクの集約帯域幅とそれらのマイクロ波リンクの実行中のQAMレベルに基づいて、マイクロ波WANリンクの実際の実行帯域幅です。公称帯域幅は、個々のマイクロ波リンクのそれぞれで設定された最大QAMの集約帯域幅に基づいて、設定可能な最大WAN帯域幅です。

帯域幅の最適化は、次のシナリオに基づいて行われます。

#### 一時的な ( 一時的なイベント ) 変更

- SWRに限定されたネットワーク/リンクで瞬間的な劣化または停止が発生した場合 (たとえば、マイクロ波の無線パスフェージングを引き起こす悪天候イベントや、変調方式の変更による使用可能な帯域幅の減少が原因で発生した場合)、影響を受けたネットワークインターフェイスのローカルSWRでトラフィックシェーピングの修正が発生します。
- これにより、影響を受ける伝送パスで発生するパケット損失を最小限に抑えることができます。

ZTP後のアクティビティの一部としてCNCのBNM KPIでSWRが有効になると、CNCはテレメトリ設定をSWRにプッシュします。

#### BNMのMDT

#### テレメトリモデル駆動型

```
destination-group <DGName>
```

```
vrf VRF-OMSLR-<エリアコード>1
```

```
address-family ipv4 <CDG IPv4Address>ポート9010
```

自己記述GPBの符号化

```
protocol tcp
```

!

!

sensor-group <グループ名>

sensor-path Cisco-IOS-XR-ethernet-cfm-oper:cfm/ノード/ノード/帯域幅通知/帯域幅通知

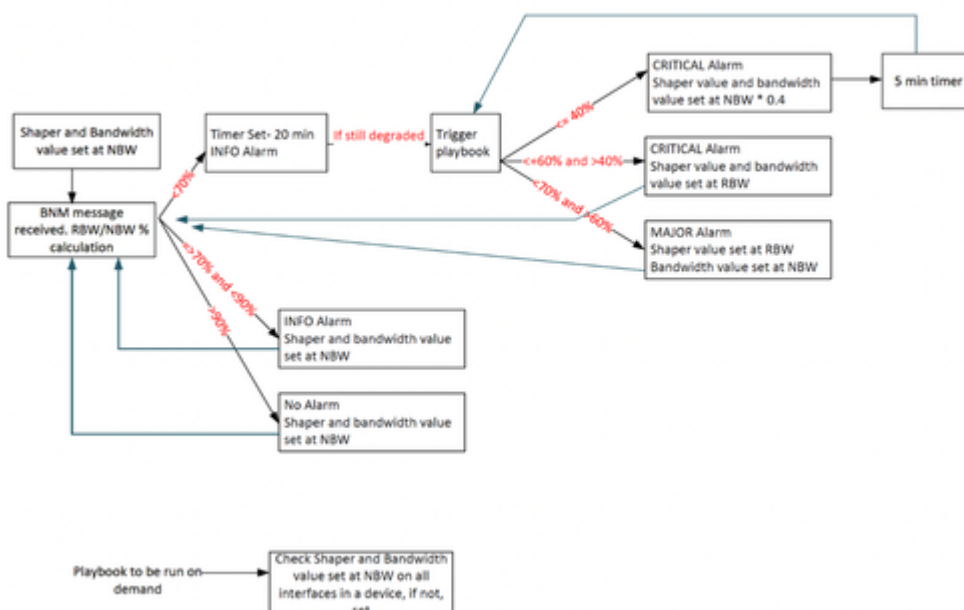
!

CNCは、テレメトリ経由で受信したこれらのBNMメッセージを処理し、必要に応じて修復処理を行います。CNCに関連する2つのコンポーネントを次に示します。

- ヘルスインサイト(HI):CNCアプリケーションは、BNMメッセージの特定のセンサーパスを監視するカスタムKPIによるBNM通知を取り込むために使用されます。Health Insightでは、帯域幅の変更が重要な場合にアラートを生成できます。
- 変更の自動化(CA):CNCアプリケーションは、HIアラートを引き起こすBNMメッセージのストリーミングに使用されます。影響を受けるインターフェイスでこれらの変更を行うために、2つのカスタムプレイブックが導入されています。
  - QoSシェーパーを新しいRBWに設定する
  - インターフェイスのキャパシティを新しいRBW値に設定する。

カスタムPythonスクリプトは、カスタムロジックを実行し、HI KPIが違反したときにCAプレイブックを自動的に実行するように開発されています。

プレイブックをトリガーするスクリプトは、次のアルゴリズムに基づいて動作します。



次の表に、帯域幅低下の程度に基づいて設定されたカスタムアラートレベルを示します。

報告された帯域幅= RBW

公称帯域幅= NBW

アラート間隔の値	通知レベル
$(RBW/NBW)*100 \geq 70$	Info
$(RBW/NBW)*100 < 70$ および $> 60$	warning
$(RBW/NBW)*100 \leq 60$	Critical

このセンサーのパスはCNCによって監視されます。

Cisco-IOS-XR-ethernet-cfm-oper:cfm/nodes/node/bandwidth-notifications/bandwidth-notification

カスタムKPIは、CNCでBNMセンサーパスを監視するために作成されます。このKPIは、120秒のケイデンスとアラートのしきい値で構成されたKPIプロファイルに追加されます。このプロファイルにSWRを接続すると、必要なテレメトリ設定がNSO経由でデバイスに自動的にプッシュされます。

イネーブルにすると、デバイスは設定された間隔で割り当てられたCDGにRBW/NBWデータをストリーミングします。Health Insight(HI)はRBW÷NBWの割合を計算し、しきい値を超えるとアラートを出します。オペレータは、HIでこれらのイベントを監視でき、Grafanaダッシュボードを使用できます。

CNCのアラートプロバイダーは、これらのアラートをPython自動化をホストするハイブリッドノードに転送します。このスクリプトは、デバイス/インターフェイス/RBW/NBWの詳細を解析し、定義された決定ロジックに基づいて、適切なChange Automationプレイブック(シェーパ調整、帯域幅更新、またはその両方)をトリガーします。

ワークフローで使用される2つのプレイブックを次に示します。

1. シェーパの値を変更するためのプレイブック
2. インターフェイス帯域幅を変更するためのプレイブック

すでに説明したように、スクリプトはWebサーバを起動し、REST APIを使用してCNCと通信するためのプロバイダーとして機能します。POSTリクエストに対して取得した応答は、ここでキャプチャされます。アラートはJSONの形式でキャプチャされ、必要なパラメータを取得するためにディクショナリに変換されます。

## カスタム自動化プレイブックを通じてDay-2ネットワーク運用を標準化

ネットワークライフサイクル全体で重要なDay-2運用を合理化および標準化するために、カスタム変更自動化(CA)プレイブックが開発されました。これには、バンドルイーサプロビジョニング、管理インターフェイスの記述の更新、CFMデジチチェーンオーケストレーション、シームレスなリンク容量拡張、eNodeBの使用停止、効率的なミニリンクオンボーディングが含まれます。これらのプレイブックは、運用のベストプラクティスを再利用可能なワークフローに組み込むことで、実行の一貫性を大幅に向上させ、人為的なミスリスクを最小限に抑え、手動による介入への依存を軽減します。Cisco CNCのアップグレードに関連して、この自動化フレームワークは、運用ターンアラウンドの加速、サービス継続性の確保、現代のネットワーク変革の目的に合わせたスケーラブルで反復可能なプロセスの実現において重要な役割を果たします。

## Cisco CNC 7.1アップグレードにおけるTACACS+統合継続性

Cisco CNC 4.1から7.1へのアップグレードの一部として、既存のTACACS+統合は、一元化された認証、許可の継続性を確保するために慎重に維持されました。アップグレードプロセスでは、Cisco CNC 7.1でTACACS+の設定の検証と複製を行い、確立された企業セキュリティポリシーとロールベースアクセスコントロール(RBAC)メカニズムとの整合性を維持しました。

## SplunkへのCNCおよびCDG Syslog転送

Syslog転送は、アラーム、イベント、syslogをSplunkサーバに転送するように設定されます。これを実現するために、CNCのsyslogサーバを設定する設定済みの機能が使用されました。

## OneFMへのアラーム転送

CNCアラームは、CNC restconfコネクション型APIを使用して、OneFMなどのノースバウンドシステムにも転送されます。

```
curl -L --request GET \  
--url https://{server_ip}:30603/crosswork/notification/restconf/streams/v2/alarm.json \  
--header 'Accept: application/txt'). This API must be used over a websocket connection config.
```

## 毎日のCNCバックアップの自動化

自動スクリプトは、CNCバックアップAPIを使用してCNCの完全バックアップを作成し、そのバックアップファイルをユーティリティホストに保存します。この操作は毎日実行されます。

## 課題

### Crossworkバージョンでの大きなジャンプ

Cross work 4.4から7.1へのアップグレードでは、定期的な差分更新ではなく、大幅なバージョンアップが行われました。このような大幅な増加により、複数のアプリケーションに多数の新機能が導入され、大幅な改良とアーキテクチャの変更が行われました。このため、CNCアップグレードは単純なバージョン交換ではなく、すべての統合コンポーネントの互換性、安定性、および適切な機能を確認するための徹底的な検証が必要でした。拡張された機能セットと基盤となる改善により、既存のワークフロー、構成、統合には慎重な検証が必要になり、包括的なテストと検証がアップグレードの成功に不可欠なものになりました。

### インプレースアップグレードなし

CNCは、インプレースアップグレードモデルをサポートしていません。その代わりに、アップグレードはリフトアンドシフト方式に従う必要があります。この方式では、既存の展開を維持しながら、ターゲットバージョンでまったく新しい環境をゼロから構築します。新しいシステムをインストールしたら、古い環境を使用停止にする前に、構成、データ、統合を慎重に移行して検証する必要があります。

このアプローチでは、次のような運用上の課題が発生します。

- 並行環境：移行と検証が完全に完了するまで、古いCNC環境と新しいCNC環境の両方を同時に実行する必要があります。
- ハードウェアリソースへのプレッシャー：2つの完全な環境を並行して実行すると、コンピューティング、ストレージ、およびネットワークリソースの需要が大幅に増加し、使用可能なインフラストラクチャに負担がかかる可能性があります。
- 検証作業の延長：移行したすべてのデータ、構成、ポリシー、および統合を新しいバージョンで検証し、期待どおりに機能することを確認する必要があります。
- データ移行の複雑さ：履歴データ、アプリケーション設定、および運用設定を移行するには、不整合やデータ損失を回避するための慎重な計画が必要です。
- 使用停止の遅延：新しい導入の安定性が証明されるまで、古いシステムとそのVMを削除することはできません。これにより、リソースの使用率と運用オーバーヘッドが長引きます。
- 運用の調整：チームは、移行期間中に両方の環境間の同期を管理して、設定のずれや運用の

中断を防ぐ必要があります。

- クローズドループによる自動化の競合：CNCは、リアルタイムのネットワーク条件に基づいてアクションを動的にトリガーするクローズドループによる自動化のユースケースをサポートしています。移行時に古いコントローラと新しいコントローラの両方がアクティブな場合、両方のコントローラで同じ自動化ロジックが実行され、ネットワークで設定の変更が重複したり、アクションが競合したりする可能性があります。これには、移行ウィンドウ中に自動化ポリシーを慎重に制御する必要があります。
- 履歴アラーム、イベント、障害レコード、監査情報などの従来の運用データは、ネイティブのエクスポート機能がないため、新しい環境に移行されません。その結果、この履歴データはアップグレードされたシステムでは使用できず、移行後は回復不能として扱われる必要があります。

これらの要因により、リフトアンドシフトモデルでは、CNCアップグレードは標準的なインプレースアップグレードと比較して、リソースを大量に消費し、運用が複雑になります。

## ロールバックオプションのない導入の落とし穴

CNCの特定の導入および導入後の設定エラーには修復パスがなく、クラスタの完全なティアダウンと再導入が必要です。たとえば、sZTPのユースケースに必須であるCrosswork data VIP(CDVA)に対して誤ったFQDNが設定されていると、sZTPが機能しなくなります。この値は導入後に修正できないため、完全な再導入が必要でした。

同様に、変更の自動化のデバイス上書き資格情報の不適切な構成は、展開後に修正できず、別のクラスタの再構築につながります。ゲートウェイIPの設定ミスやサブネット定義ミスなど、その他のエラーも修復不可能と見なされます。

これらのシナリオでは、初期導入時に不変のパラメータをすべて検証することが非常に重要です。コストのかかる手直しやスケジュールへの影響を回避するには、綿密な計画と入力の検証が不可欠です。

## 導入後の診断検証の制約

CNCは、ディスクの読み取り/書き込み遅延、IOPS、同期遅延、ネットワークインターフェイス速度、CPUクロック周波数などのVMレベルのヘルスパラメータを評価する診断ユーティリティを提供します。このユーティリティは、測定値が予想されるしきい値と比較して報告され、各チェックに合格または不合格のマークが付けられます。ただし、これらの診断はクラスタの導入後のみ実行できるため、導入前にインフラストラクチャの準備状況を検証するメカニズムはありません。

インストール時に、「Ignore Diagnostic Checks」フラグがデフォルトでfalseに設定されます。実際には、1回のチェックが失敗すると、インストーラが停止し、展開を続行できなくなります。その結果、実稼働グレードの環境でも1つ以上のチェックに失敗することが多いため、フィールドエンジニアはこのフラグを有効にして診断を完全にバイパスしなければならないことがよくあります。そのため、運用に関するジレンマが生じます。つまり、チームは、導入をブロックする厳格な検証を実施するか、基盤となるインフラストラクチャが推奨されるパフォーマンスベンチマークを満たしているかを保証せずに進めるかを選択する必要があります。

## HIカスタムKPI作成手順の変更

Health Insight 4.1では、カスタムKPIの作成はTickスクリプトロジックに依存しており、KPIの定義と処理ロジックはTickフレームワーク内のスクリプトを使用して実装されていました。ただし、バージョン7.1では、このアプローチがKPIを定義および管理するためのトラックファイルベースのフレームワークに置き換えられました。

このアーキテクチャ上の変更により、既存のカスタムKPIを直接再利用できず、新しいトラックファイル形式に合わせて再作業する必要がありました。これには多大な時間と労力が必要でした。

- 新しいフレームワークの理解：チームは、7.1で導入されたトラックファイルベースのKPI定義モデルの構造、構文、および運用上の動作を調査する必要がありました。
- 既存のロジックの再設計：以前にTickスクリプトで実装されたロジックをトラックファイル形式に変換して適用する必要がありました。
- BNM KPIの再作成：新しいフレームワークを使用してカスタムBNM KPIを再作成し、以前と同じ結果と洞察が得られるようにする必要がありました。
- KPIの精度の検証：新しい実装が以前のバージョンと比較して一貫性のある正確なメトリックを生成したことを確認するために、広範な検証が必要でした。
- テストとチューニング：新しいモデルでは、実際のネットワーク条件下でのパフォーマンスと動作もテストする必要があり、必要に応じて調整が行われました。
- サポートの欠如：tickスクリプトで以前に動作していた一部の機能は、新しいトラックファイルの実装ではサポートされなくなりました。そのため、妥協が必要になりました。

このKPI作成メカニズムの変更により、アップグレードに必要な作業が大幅に増加しました。新しいシステムの学習と、運用上の洞察の継続性を確保するための既存のカスタムモニタリングロジックの再実装の両方が必要だったためです。

## BNMプレイブックトリガースクリプトのAPIタイムアウト

BNMプレイブックは、CNC APIと対話するカスタムスクリプトによってトリガーされます。アップグレードおよび検証プロセス中に、API認証と応答処理に関連するいくつかの問題が特定され

、対処されました。

CNC APIトークンの有効期間は8時間ですが、有効期限が切れたトークンを更新するための適切なロジックが元のスクリプトに含まれていませんでした。その結果、CNC 4.4のKPIアラートは正常に機能していましたが、トークンの期限が切れた後、プレイブックをトリガーするスクリプトの実行が停止しました。この問題は長い間認識されずにいたため、自動化スクリプトが1年以上にわたって確実に実行されていなかったことを意味します。この問題は、CNC 7.1での移行および検証アクティビティ中にのみ発生します。

そのため、次のような改善と改良が必要でした。

- トークン更新ロジック：適切なロジックが実装され、トークンの期限切れを検出してAPIトークンを自動的に更新し、スクリプトの実行が中断されないようにします。
- API応答の変更：CNCバージョン間の違いにより、さらに問題が発生しました。CNC 4.1では、有効期限が切れたトークンの応答には通常「expired」というメッセージが含まれていましたが、CNC 7.1では「Key not authorized」というメッセージが返されます。7.1の新しい応答パターンを正しく解釈するには、スクリプトロジックを更新する必要がありました。
- グローバルトークン処理：以前は、トークンは関数内にローカルで格納され、使用されてきました。これにより、関数の入力時にトークンが有効であったものの、その後のAPI呼び出しの前にトークンが期限切れになるシナリオが作成されました。実装はグローバルトークン処理を使用するように変更され、すべての機能の一貫性と適切な更新が確保されました。
- エラー処理の改善：場合によっては、NSO「check\_sync」APIが、不完全または予期された構造と異なる応答を返すことがあります。これによりKeyError例外が発生し、スクリプトの実行が中断されました。予期しないAPI応答を受信した場合でもスクリプトの実行を継続できるように、追加の例外処理および検証ロジックが導入されました。
- スクリプトの安定性の強化：APIの失敗、一時的な応答の問題、またはトークンの更新イベントによってスクリプトが予期せず終了しないようにするための追加の保護機能とチェックが追加されました。

これらの改善により、アップグレード中に明らかになった問題が解決されただけでなく、BNMプレイブック自動化フレームワークの信頼性、復元力、メンテナンス性が大幅に向上しました。

## BNM処理およびプレイブックのトリガー設計の変更

BNM自動化ロジックはイベント駆動型で、CNC内のHealth InsightアプリケーションのKPIによって生成されるアラートに依存します。ワークフロー全体は次のように動作します。

1. CNCは、デバイスからNB（公称帯域幅）とRBW（実帯域幅）の値を読み取ります。
2. これらの値を使用して帯域幅比率(BW%)を計算します。
3. Health Insight KPIは、この比率を事前定義されたアラートしきい値と比較して評価します。
4. アラートが生成されると、BNMプレイブックをトリガーするスクリプトによってアラート

が検出され、対応する修正プレイブックが実行されます

## 元のアラート設計の制限

設定されたアラートしきい値は次のとおりです。

- $BW\% < 60$  → 緊急
- $60 \leq BW\% \leq 70$  → 警告
- $BW\% > 90$  → 情報

この設計は帯域幅の劣化を特定する上では適切に機能しましたが、帯域幅の回復シナリオでは機能的なギャップが生じました。具体的には、70 ~ 90 %の範囲にはアラートレベルが定義されていません。

これにより、次の動作が発生します。

- BW %が70 %を下回ると、重大または警告のアラートが生成され、シェーパーと帯域幅の値を調整するプレイブックがトリガーされます。
- ただし、帯域幅が回復し、BW %が70 %を超えて増加した場合、値が関連するアラートレベルのない70 ~ 90 %の範囲に落ちたため、KPIではアラートが生成されませんでした。
- BNM自動化スクリプトはアラート生成に完全に依存してアクションをトリガーするため、更新されたNBW/RBW値を読み取ったり、復元アクションを開始したりする機会がありませんでした。
- その結果、十分な帯域幅が使用可能になっても、帯域幅の復元は自動的には行われませんでした。元の設計に復元ロジックはありませんでした。

この制限は、条件が改善された後も以前に帯域幅削減を行ったリンクが制限状態のままであった実稼働ネットワークで明らかになりました。

## KPIフレームワーク変更の影響

この問題は、CNC 7.1で導入されたフレームワークの変更によってさらに悪化しました。Health Insight 4.1では、TickベースのKPI実装で最大5つのアラートレベルがサポートされ、しきい値バンドをより詳細に制御でき、復元ロジックの実装が容易になりました。

ただし、CNC 7.1では、トラッカーファイルベースのKPIフレームワークでサポートされるアラートレベルは3つだけであるため、複数のリカバリしきい値を定義する柔軟性が低下し、これらの制約に適合するようにアラートロジックを再設計する必要がありました。

## 過剰なプレイブックトリガー

最初の実装で確認された別の問題は、プレイブックの実行の頻度が非常に高いことでした。自動化ロジックには、ホールドタイムや安定化ウィンドウは含まれていませんでした。CNCがアラート条件を満たしたデバイスから値を読み取るとすぐに、次の処理が行われます。

- 警報がすぐに上がった。
- 自動化スクリプトは、即座に修正プレイブックをトリガーしました。

テレメトリの値はライブネットワークで頻繁に変動するため、このために1時間に数百冊のプレイブックがトリガーされます。これは、ネットワークの安定性とアプリケーションパフォーマンスの両方の観点から見ると、理想的ではありませんでした。

## 自動化ロジックの再設計

これらの制限に対処するために、BNMオートメーション設計にはいくつかの改良が加えられています。

- アラートしきい値ロジックの改訂：リカバリ帯域が3つのアラートレベル内で取得されるように、ロジックが変更され、90%を超える値のみがINFOとして分類されていた以前のアプローチに代わって、70%を超えるBW%がINFOレベルのアラートとして扱われるようになりました。これにより、70 ~ 90 %のリカバリ帯域をアクティブに監視し、帯域幅の状態が改善したときに復元プレイブックをトリガーできるようになりました。
- ホールドタイムの導入：プレイブックをトリガーする前に帯域幅条件が一定の期間安定していることを確認するために、20分のホールドタイムメカニズムが導入されました。これにより、短期的な変動に自動化が影響を受けるのを防ぎます。
- 制御されたプレイブック実行：ロジックとホールド時間が変更されたため、プレイブック実行の頻度が大幅に減り、不要な自動化アクションが防止されました。
- 重度の劣化に対するブースターメカニズム：重度の帯域幅劣化の場合には、ブースターアプローチが導入されました。このようなシナリオでは、自動化によってトラフィックシェーパと帯域幅の割り当てがNBWの40 %にプロアクティブに調整され、輻輳からより迅速に回復できます。
- 自動化の安定性の向上：再設計されたワークフローにより、トラックベースのKPIフレームワークの制限内であっても、帯域幅削減と帯域幅復元の両方のシナリオが効果的に処理されます。

## 成果

これらの設計変更に加え、以前のAPI処理、トークン管理、スクリプトの堅牢性の改善により、BNM自動化フレームワークは、はるかに安定し、効率的で予測可能な方法で動作するようになりました。このシステムは、輻輳状態と復旧状態の両方に正しく対応でき、プレイブックの実行が

過剰になるのを回避して、信頼性の高いネットワーク帯域幅の最適化を実現します。

## デバイスアラームの抑制

CNC 4.1では、アラームはRESTCONF APIを介してOneFMと呼ばれるノースバウンドシステムに転送されました。CNC 4.1スタックにはEMF機能が含まれていないため、プラットフォームはシステムレベルのアラームのみを生成しました。これらのアラームは、アラームの分類に関連する複雑さをなしで、アップストリームに転送されました。

CNC 7.1の導入に伴い、EMFアプリケーションが導入され、アラームモデルが大幅に拡張されました。アラームは、次の3つのタイプに分類されるようになりました。

- システムアラーム：CNCプラットフォームとアプリケーションの健全性に関連するアラーム
- ネットワークアラーム：ネットワークサービスの状態に関連する
- デバイスアラーム：ネットワークデバイスから直接生成され、CNC経由で転送されます。

ただし、デバイスレベルのアラームの収集と管理を担当するEPNMはすでに存在していました。CNCがこれらのアラームもOneFMに転送すると、両方のシステムから重複したアラームが受信されます。したがって、要件は、システムアラームとネットワークアラームを転送しながら、CNCからデバイスアラームを除外することでした。

主な課題は、アラームをOneFMに転送するために使用されるRESTCONFノースバウンドAPIの制限でした。APIは、アラームカテゴリに基づくアラームのフィルタリングをサポートしませんでした。このようなフィルタリングが使用可能であれば、解決策は簡単です。デバイスのアラームをノースバウンドシステムに転送する前に、APIレベルで単純に除外します。

いくつかの潜在的なソリューションを評価し、検討しました。

- 発信元でのデバイスのトラップの停止：デバイスがCNCにトラップを送信できないようにします。
- ノースバウンドシステム(OneFM)でのアラームのフィルタリング：CNCがすべてのアラームを送信できるようにしますが、OneFM内のデバイスアラームはフィルタリングします。
- アラームを転送する前にCNC内でフィルタリングする。

CNCはデバイスのイベントを検出し、ネットワーク状態の運用認識を維持するためにこれらのトラップに依存しているため、デバイスレベルでトラップを停止することは現実的ではないと考えられていました。トラップを無効にすると、CNCがネットワークの問題に対応する能力が大幅に低下します。

このソリューションは、最終的に実装され、デバイスアラーム抑制と呼ばれる組み込みCNC機能を利用しました。この機能を使用すると、管理者はデバイスグループに基づいて特定のタイプのデバイスアラームを抑制し、それらのアラームが処理されたり、アップストリームに転送されるのを防ぐことができます。

デバイスアラーム抑制ポリシーを設定することで、システムは次のことができるようになりました。

- CNC内のデバイス生成アラームを抑制します。
- システムアラームとネットワークアラームの処理と転送を続行します。
- 重複したデバイスアラームがOneFMシステムに届かないようにします。

このアプローチは、デバイスからのトラップを受信するCNCの機能を中断することなく、クリーンでスケーラブルなソリューションを提供しました。その結果、OneFMへのアラームフローが合理化され、EPNMのデバイスアラーム管理による重複を回避しながら、関連するシステムアラームとネットワークアラームのみが転送されるようになりました。

## アウトオブバンドの変更

既存のセットアップでは、運用チームは直接CLIベースのスクリプトを使用して、特にACLの変更やデバッグアクティビティなどのタスク用に、設定の更新をネットワークデバイスにプッシュしていました。NSO以外で行われた変更はシステム内で追跡されなかったため、このアプローチは短期的には効果的でしたが、設定の不整合が発生しました。その結果、意図した（モデル化された）状態と実際のデバイス設定の間の不整合が原因で、NSOのプロビジョニングワークフローに影響が生じ、障害や運用の非効率性が発生しました。

## L2/L3 VPNの調整

アウトオブバンド設定の変更：ネットワーキングチームは、CNC/NSOおよびTSDNワークフロー外のデバイスのVPN関連の設定を更新しました。その結果、NSO（CNC 4.1時代から）に保存されている状態は、デバイスの状態と常に一致するわけではありません。

これらの不一致により、複数の調整の失敗と不整合が発生しました。一部のケースでは、NSOにはデバイスに存在しない（またはNSOが反映しない方法で変更された）VPNサービスデータが含まれていました。NSOをネットワークに合わせるには、NSOにのみ存在し、デバイスには存在しないVPNサービスエントリを削除し、アウトオブバンドの変更によって発生するその他の不一致を修正する必要がありました。

## スケジュールの影響

これらの問題の解決には、当初の調整計画に加えて約2週間が必要でした。余分な時間は、不一致の特定、デバイス状態の検証、NSO CDBデータの安全なクリーニングまたは修正に費やされました。

## 観察

1. 設定権限：VPN（またはTSDNで管理される設定）に対するアウトオブバンドの変更により、NSOとネットワーク間の不整合が生じ、調整が複雑になります。
2. 移行前のベースライン：移行前のNC/NSO管理状態とデバイスのみ状態の明確なベースラインにより、不一致の検出と解決が容易になりました。
3. 自動化と変換：ペイロード変換スクリプトとユーザ固有のカスタマイズは、4.1と7.1の形式とモデルの違いを一貫した方法で処理するために不可欠でした。

## 同様のアップグレードに関する推奨事項

1. 調整ウィンドウ中にVPN（およびその他のTSDN管理）サービスに対する変更の凍結を強制します。ただし、制御されたプロセスを介してのみ例外が発生します。
2. NSO CDBとデバイス設定を比較する調整前監査を実行し、調整を開始する前に不一致を定量化して一覧表示します。
3. VPNの変更は、アップグレード後にCNC/NSO TSDNを通過して、帯域外ドリフトの再発を回避する必要があることを文書化し、周知します。
4. 変換スクリプトと調整スクリプトを保持し、将来のアップグレードやトラブルシューティングで再利用できるようにします。

## メンテナンスモードの依存関係によるCNCバックアップの失敗

CNCバックアップメカニズムでは、バックアップオペレーションを開始する前に、プラットフォームをメンテナンスモードにする必要があります。設計上、バックアップAPIはこの前提条件を適用します。CNCがメンテナンスモードへの移行に失敗すると、バックアッププロセスは自動的に中断されます。

実際には、次のようなシステムアクティビティが進行中のため、メンテナンスモードへの移行は頻繁に失敗しました。

- Active Change Automation(MOP)の実行
- 継続的なsZTPワークフロー
- DLMサービスオペレーション
- KPIプロファイルの添付または分離アクティビティ
- オンデマンドのshowtechコレクション
- バックグラウンドオーケストレーションタスク

このようなアクティビティがあると、CNCがメンテナンスモードに入ることができず、バックアップ操作が実行前に失敗します。

## 運用への影響

コンプライアンスと運用保証のために必要な毎日のCNCバックアップしかし、自動化アクティビティ、特にBNMでトリガーされるプレイブックが頻繁に発生するため、システムがメンテナンスモードになることはほとんどありませんでした。その結果、バックアップの失敗が繰り返し発生し、重大な運用上のリスクが生じるため、手動による介入が必要になりました。

## 軽減のための戦略

1. バックアップスケジューリングの最適化：システムのアクティビティが最小限のメンテナンス時間帯を特定しました。トラフィックと自動化の分析に基づき、バックアップジョブは午前5時(AEST)にスケジュールされました。オーケストレーションとプレイブックの実行が最もアクティブになる可能性が低かったためです。

2. バックアップ前のアクティビティ検証：バックアップAPIを呼び出す前に、自動化された事前チェックが導入されました。

- このスクリプトは、CNC APIを照会して、実行中の変更オートメーションMOPジョブを検出します。
- いずれかのジョブがasRunningと報告された場合、スクリプトは5秒間待機して再試行します。
- このループは、システムがアクティブなジョブを報告しなくなるまで続きます。
- 環境がアイドルであることが確認された後にのみ、スクリプトはメンテナンスモードを有効にしてバックアップをトリガーします。

これにより、システムがビジー状態の運用中に不要なバックアップが試行されることがなくなりました。

3. リトライとレジリエンスのメカニズム：一時的なシステム状態に対応するために、追加のセーフガードが追加されました。

- バックアップAPIが障害を返した場合、最大3回の再試行
- 再試行間の短い遅延間隔
- スクリプトの終了を回避する適切なエラー処理

## 結果と成果

この緩和策の組み合わせにより、バックアップの信頼性が大幅に向上しました。

- バックアップの失敗が大幅に減少
- 実装後に確認された障害は2つだけで、どちらもスクリプトの制御外にあるsZTPプロセスのスタックが原因でした。
- BNMプレイブックの自動化に実行遅延が導入されたことにより、メンテナンスモードとの競合がさらに減少しました。

## Splunkへのsyslogの転送

syslogの宛先は、TLS経由でSplunkにログを転送するようにCNCで設定されました。ただし、受信後にSplunk側でログを読み取ることができませんでした。Splunk環境に起因するこの問題により、UDPトランスポートに戻すオプションが選択され、ログが正常に処理されました。

## デバイスのグループ化の移行の問題

ユーザは以前にCNC 4.1で18のデバイスグループを作成しましたが、このリリースでは、デバイスグループをエクスポートまたはインポートするためのUIベースまたはAPI駆動型のメカニズムは提供されませんでした。そのため、これらのグループをCNC 7.1に移行するには、非標準のアプローチが必要でした。2つの内部CNC APIが特定されました。1つはデバイスグループ階層を公開するもので、もう1つは各階層ノードにマップされたデバイスを一覧表示するものです。これらのAPIを使用して、すべてのデバイスグループとそれらに関連付けられているデバイスが抽出され、JSON出力として保存されました。その後、カスタムスクリプトを開発して応答を解析し、各グループからデバイスホスト名だけを抽出しました。

CNC 7.1では、CSVベースのインポートテンプレートを含む、デバイスグループのネイティブなインポート/エクスポート機能が導入されました。レガシーシステムからホスト名を抽出した後、必要な形式でCSVテンプレートを入力するための2つ目の自動化スクリプトが作成され、各デバイスグループを正確かつ独立してインポートできるようになりました。この自動化は不可欠でした。この自動化がなければ、デバイスグループをCNC 7.1に移行するのにかなりの時間がかかり、運用も複雑になっていました。

## 帯域幅が著しく低下しているデバイスを特定

低いRBW/NBW比率を自動的に修正するためにBNMのユースケースを実装したにもかかわらず、一部のデバイスは長時間にわたって深刻な機能低下状態のままでした。シェーパと帯域幅調整のプレイブックでは、通常は品質低下イベントの直後にデバイスが復元されますが、一部のデバイスは1週間以上もCritical状態のままであり、手動による介入が必要でした。しかし、これらの

デバイスの特定には課題がありました。CNC UIは、アラートと帯域幅メトリックを明確に可視化しますが、長期にわたってCritical状態のままになっているデバイスを簡単に明らかにすることはできません。

この運用上のギャップに対処するために、APIベースのソリューションが開発されました。CNCは、設定可能な期間（7日間、1ヵ月など）にわたって、アラートを生成する上位デバイスのリストを取得するAPIを提供します。このデータを取得し、選択した期間を通じて重大なアラートのみを生成したデバイスをフィルタリングすることで、チームは手動による修復が必要なデバイスを迅速に特定できました。この自動化されたアプローチにより、トラブルシューティングの効率が大幅に向上し、継続的な品質低下のケースを特定するために必要な時間が短縮されました。

## デバイステレメトリ構成の削除

CNC 4.1では、デバイスがHealth Insight(HI)KPIプロファイルに関連付けられたときに、NSOからthetm-tcfunctionパックを介してプッシュされたテレメトリ設定が自動的に適用されました。ただし、CDG VIP参照を含むこれらの構成は、後でKPIプロファイルが分離されたときに削除されませんでした。その結果、デバイスは時間の経過とともに古い冗長なテレメトリエントリを蓄積しました。

この問題は、CNC 7.1へのアップグレード時に顕著になりました。デバイスは、CNC 7.1によって生成された新しいエントリとともに、CNC 4.1の従来CDG VIPテレメトリ設定を保持していることが多く、2,000を超えるデバイスで複数のテレメトリ設定の競合が発生していました。CNC 7.1 CDG VIP構成だけがアクティブである必要があるため、運用への影響と構成の衛生状態に関する懸念が生じました。

この問題に対処するために、各デバイスのテレメトリ設定から古いCDG VIP参照を特定して削除する自動スクリプトが開発されました。このソリューションにより、設定の不整合が排除され、予想される7.1テレメトリモデルとの整合性が復元され、大規模なデバイス群で手作業で数日かかっていたクリーンアップ作業が防止されました。

## MDT収集のトラブルシューティング

CNC 7.1では、ほとんどのHealth Insight(HI)KPIコレクションがモデル駆動型テレメトリ(MDT)に依存しています。デバイスでKPIプロファイルを有効にすると、NSOは必要なセンサーパスを自動的にプログラムし、テレメトリの宛先としてCDG VIPを設定します。この設定が適用されると、対応するCDG収集ジョブが作成され、デバイスのテレメトリステータスが追跡されます。

検証中、100台を超えるデバイスでテレメトリ設定が欠落していることが報告されました。UIは

デバイスごとのフィルタリングしかサポートしておらず、2,000台を超えるデバイス群に対して効率的に拡張できないため、CNCユーザインターフェイスでこれらのデバイスを特定することは実際的ではありませんでした。このため、テレメトリの設定が不足しているデバイスと必要なKPIの有効化が不足しているデバイスを判別する自動方法が必要でした。

この問題に対処するために、KPIプロファイルがアクティブになるたびにデバイスに割り当てられるBNMタグを活用しました。まず、BNMタグを持つすべてのデバイスのエクスポートが生成されました。その後、CNCコレクションAPIと対話するためにPythonスクリプトが開発され、ページネーションロジックを組み込んでコレクションジョブの完全なセットを取得しました（各APIコールは最大100エントリを返します）。スクリプトは、収集ジョブデータからホスト名を抽出し、エクスポートされたBNMタグ付きデバイスリストと比較しました。

この比較により、タグ付けされたがBNM収集ジョブに表示されなかったデバイスのリストが生成され、MDTテレメトリ設定が適用されなかったことが示されました。その後、これらのデバイスでKPIプロファイルが再度有効になり、対応するすべての収集ジョブが正しく作成されたことが検証によって確認されました。

この自動化により、トラブルシューティングプロセスが大幅に合理化され、影響を受けるすべてのデバイスの特定と修復を1日で完了できるようになりました。この作業は、手作業での検査では不可能でした。

## NSO 6.4.1.1におけるHA動作の変更とコンセンサスアルゴリズムの調整

Cisco CNC 7.1への移行の一環としてCisco NSO 5.7.5.1から6.4.1.1にアップグレードする際、新しいNSOバージョンでコンセンサスアルゴリズムが暗黙的に有効になっているため、ハイアベイラビリティ(HA)動作に顕著な変更が見られました。これはNSO 5.7.5.1のデフォルト動作ではなく、アップグレード後のフェールオーバー特性の変化を招きました。特に、プライマリノードがダウンすると、セカンダリノードは読み取り専用状態に移行し、プロビジョニングアクティビティを処理できなくなります。同様に、セカンダリノードがダウンすると、プライマリノードがアクティブなプライマリ状態から「none」状態に移行し、サービスの継続性に影響が及びます。

以前の導入に合わせたHA動作を復元するために、NSO 6.4.1.1ではコンセンサスアルゴリズムが明示的に無効にされました。この調整により、フェールオーバーシナリオ中にプライマリノードとセカンダリノードが目的のロールを再開し、中断のないプロビジョニングが可能になり、以前のNSOバージョンと一貫した運用安定性が維持されます。

## NSOバージョンアップグレードおよびパッケージ互換性の強化

Cisco CNC 4.1から7.1への移行の一環として、基盤となるCisco NSOバージョンが5.7.5.1から6.4.1.1にアップグレードされました。このバージョンアップグレードにより、既存のNSOパッケ

ージ内のXMLテンプレート構造に変更が加えられ、従来のテンプレートの動作に依存する特定の  
回帰テストで障害が発生しました。

これらの互換性のギャップに対処するため、影響を受けるNSOパッケージテンプレートを分析し、  
NSO 6.4.1.1のスキーマおよび処理要件の改訂に合わせて更新しました。これらの機能拡張によ  
り、すべての自動化ワークフローとサービスモデルが期待どおりに機能し続け、回帰の安定性が  
回復し、アップグレードされたCNC環境全体の一貫性が維持されます。

## 規模に応じたKPIの有効化に関する問題

CNCは、デバイス上でKPIプロファイルを有効にするための、設定済みのUIメカニズムを提供し  
ます。このアプローチは小艦隊には適していますが、大規模な場合には非効率で信頼性が低くな  
ります。この導入では、2,000台を超えるSWRデバイスでKPIを有効にする必要があり、UIはデバ  
イスを一括で選択または処理する効果的な方法を提供しませんでした。

当初は、タギングベースのアプローチが試みられました。すべてのSWRデバイスにSWRタグが  
割り当てられ、KPIの有効化は手動でのデバイス選択ではなく、タグ選択を使用して実行されま  
した。しかし、2,000台を超えるデバイスを1つのワークフローで処理することは、運用上の大き  
な課題をもたらしました。このジョブは3時間以上にわたって実行され、何百回もの失敗を伴って  
完了しました。すべてのデバイスが意図に含まれていましたが、KPIの有効化に成功したのはわ  
ずか750台で、繰り返し試行しても徐々に進歩しただけです。このアプローチは、スケーラブル  
でも反復可能でもありませんでした。負荷に重大な問題が発生しました。

2つ目の課題は、NSOデバイスの同期に関する問題です。多くの障害は、NSOが対応するデバイ  
スと同期されていないことを示していました。手動の同期元の操作を試みてからKPIを再有効化  
するのは実際的ではなく、オペレータの多大な労力が必要でした。

これらの制限に対処するために、自動化されたバッチ駆動型ワークフローが開発されました。

1. CNCインベントリ全体をエクスポートします。
2. デバイスを50個ずつ処理します (チューニングによって最適なサイズとして識別されま  
す)。
3. 各バッチで、デバイスUUIDを使用して自動同期をトリガーします。
4. CNC APIを通じてKPIの有効化を実行します。
5. KPIジョブ履歴とログの失敗をプログラムで監視します。
6. 同期およびKPI有効化の手順を繰り返して、障害が発生したデバイスを再処理します。
7. バッチが正常に完了したら、次の50台のデバイスのセットに進みます。

自動化には、KPIプロファイルを無効にする機能も含まれており、ライフサイクル全体の管理が  
可能です。

このソリューションは、KPIのプロビジョニングを合理化し、決定論的で、拡張性の高いプロセスで行いました。手動による介入を排除し、一貫性のある成果を確保し、運用作業を何日間も節約しました。BNMの設計変更後にKPIプロファイルを無効にして再度有効にする必要があり、2,000台のデバイス全体を迅速かつエラーなく再設定できたのと同じ自動化が非常に重要であることが証明されました。

## RESTCONF ノースバウンドAPIは管理者アクセスに制限されています

CNCからアラームとイベントを転送するために使用されるRESTCONFベースのノースバウンドAPIには、adminアカウントを使用してのみ起動できるという制限があります。サービスアカウントを使用してAPIにアクセスしようとしたが、アカウントに必要な操作ロールが割り当てられていたにもかかわらず、失敗しました。回避策として、ユーザはノースバウンドシステムへのアラーム転送に管理者クレデンシャルを使用する必要があり、運用上の制約が生じ、最小権限アクセス原則への準拠が制限されていました。

## 戦略的な成功要因としての自動化

CNCのアップグレードおよび移行プログラムは規模が大きく、複雑であるため、運用タスクを手動で実行することは非常に困難です。デバイスのオンボーディング、KPIのプロビジョニング、設定の調整、調整、テレメトリの検証などのアクティビティには、何千ものネットワーク要素と、手動で実行すると人的エラーが発生しやすい繰り返しワークフローが含まれます。したがって、自動化は、実行を加速させるためだけでなく、一貫性を確保し、運用リスクを軽減し、時間のかかる繰り返しタスクからデリバリチームを解放するためにも不可欠でした。

スクリプト化されたワークフローとAPI主導の操作を通じてこれらのプロセスをシステム化することで、アップグレードプログラムは大幅な効率向上を達成しました。自動化により、タスク完了の迅速化、精度の向上、すべてのセクションにわたる予測可能な成果が実現しました。その結果、導入スケジュール全体が短縮されただけでなく、エンジニアは日常的な運用タスクよりも価値の高い検証と設計作業に集中できるようになりました。

自動化アクティビティの一部は、アップグレードプロジェクトが開始される前に特定されましたが、課題が発生したときに一部が変化しました。プロジェクトの過程で発生した問題によって必要になったものもいくつかありました。

この表は、自動化がプログラム全体に大きな影響を与えた分野を示しています。

タスクの説明	手動による作業 ( 日数 )	自動化の取り組み ( 日数 )	推定節約額 ( 日数 )
ACLアップデート (SWR/LWR)(2K+)	30.0	2.0	28.0
デバイスの移行とCDGへの接続 ( 2,000以上 )	5	1.0	4.0
デバイスへのBNM KPIの取り付け ( 2,000以上 )	4.0	1.5 ( 平均 )	2.5
サービスの調整	7	2.5	4.5
デバイスグループの移行	4	0.5	3.5
帯域幅が著しく低下しているデバイスを特定	3	0.5	2.5
MDT収集のトラブルシューティング	3	0.5	2.5
合計	56 日	8.5 日	47.5 日

## 学ぶべきレッスン

### アップグレードは簡単ではない

CNCはインプレースアップグレードをサポートしておらず、リフトアンドシフトモデルでは運用が大幅に複雑になります。特にバージョンジャンプが大きい場合は、プロセスが単純であると仮定しないでください。予期しない問題は、アプリケーション、統合、およびワークフロー全体に現れ、それぞれ時間、分析、および慎重な軽減が必要です。メジャーバージョンのleapはこの課題を増幅し、綿密な計画、検証、段階的な実行が不可欠になります。TACのケースやトラブルシューティングの作業に多くの時間を費やさなければなりませんでした。これに対してバッファ時間を維持しなかったため、困難な作業になりました。

## CXは重労働を行わなければならない

導入、統合、移行、エンド・ツー・エンドの使用例の検証において、CXに多大な労力を期待できません。古いリリースで実証されたワークフローが新しいリリースでも同じように動作するとは限りません。-動作させるには、トラブルシューティングと分析が大量に必要になります。

## 自動化ツールキットの必要性

このアップグレードの過程で、自動化はオプションの利便性ではなく、大規模なCNC導入の基本要件であることが実証されました。早い段階で必要な候補の自動化を計画しましたが、それだけでは十分とは考えられません。プロジェクトの途中で、前のセクションで示したように、自動化によって確実に付加価値が生まれるユースケースで問題を特定できます。

## 移行中のデュアルコントローラ競合の回避

アップグレードの際には、古いCNC環境と新しいCNC環境の両方が同時にアクティブにならないようにすることが重要です。検証には短いソーク期間が必要ですが、このプロジェクトで2か月以上続いたように、大幅に延長すると運用リスクが生じます。両方のCNCが15 ~ 20日以上有効な状態で、オンデマンド帯域幅などのクローズドループ自動化機能によって、ネットワーク全体で一貫性のない競合するアクションが生成されました。これは、自動化ロジックが2台のコントローラから同時に実行されているためです。

重要なレッスンは、移行中に明確なガードレールを実装することです。古いCNC内のデバイスを管理上無効にする、自動化ワークフローを一時停止する、テレメトリサブスクリプションを制限するなどの措置によって、これらの競合は回避されました。今後のアップグレードでは、デュアルコントローラ干渉を回避し、予測可能なネットワーク動作を確保するために、コントローラを厳密に分離する計画を明示的に立てる必要があります。

## MOPはSacrosanctではありません

手順説明書(MOP)は、導入、統合、およびユースケースごとに作成されますが、ラボ条件で検証されたMOPが実稼働環境でも同様に動作すると仮定するのは非現実的です。実稼働環境では常にばらつきが見られ、わずかながら重大なものもあり、制御テスト中には見られなかったギャップが浮き彫りになっています。実際のネットワーク、従来の動作、外部依存性、およびライブトラフィックの状態によって、ラボシミュレーションでは常に再現できない可変要素が生じます。

ここで重要なのは、チームは予期しない動作、エッジケース、新しい発見に遭遇することを期待

して実稼働環境への導入に取り組む必要があるということです。規模に応じて適切に実行するには、柔軟性、迅速なトラブルシューティング機能、および迅速な手順の適応性が不可欠です。

## TACケースの有効性

生産後の問題は避けられず、納品チームによる最初のトラブルシューティングは貴重ですが、社内の取り組みだけに頼ることは不要な遅延につながる可能性があります。特に製品に関連する問題や、すぐに診断できない複雑な動作については、TACケースをセーフティネットとして並行してオープンすることが賢明です。TACの調査には時間がかかることが多く、ケースの作成を数日遅らせると、プロジェクトの推進力が大幅に失われる可能性があります。TACを早期に利用することで、必要なときに専門家のサポートを受けることができ、根本原因の特定を迅速化し、回避可能なスケジュールの遅れを防ぐことができます。

## CNC BUと効果的なナレッジサポートを締結

CNC事業部門からの強力なサポートは、CNCプロジェクトの期間中に非常に貴重です。ユーザは、製品に関する詳細な洞察や明確さを必要とすることがよくあります。これらは、デリバリーチームだけでは簡単に入手できません。エンゲージメント全体を通じてBUの担当者と連絡を取ることができるため、問題解決が促進され、技術的な正確性が強化され、信頼とユーザラポールが向上します。

## CNCアップグレードのベストプラクティス

### 最適化されたアップグレード戦略の計画

CNCはインプレースアップグレードをサポートしないため、並行導入は避けられません。新しい環境を新規インストールとして扱い、2つの環境を同時に実行するために十分なコンピューティング、ストレージ、および管理能力を割り当てます。検証ステージ、移行シーケンス、カットオーバー作業を事前に計画します。

### 特に不変のパラメータに対しては、厳格な導入前検証が不可欠

多くの経験から、初期導入時の徹底した作業の重要性が浮き彫りになっています。コストのかかる再導入を防ぎ、スケジュールへの影響を防ぐには、すべての主要な入力、特に不変の設定パラメータを事前に検証することが不可欠です。したがって、元に戻せない設定エラーのリスクを最

小限に抑えるために、構造化された導入前チェックリスト、ピアレビュー、および実行不足の検証を使用することを強くお勧めします。

## 製品に触れる前に専用の検証環境を使用

プロジェクトの早い段階で内部CALO/テスト環境を確立することで、チームは実稼働に触れる前に、ワークフローを実験、検証し、バージョン固有の変更を発見し、自信を持つことができます。これにより、最終的なロールアウト時の未知数が大幅に減少します。

## 分散したクロスワークコンポーネントの実績に基づいたサイジング

クラスタ、CDG配分、およびPCE割り当てを設計する際は、単純なデバイス数ではなく、デバイスタイプ、インターフェイスのスケール、トポロジの複雑さ、および収集強度に基づいて決定します。分散のバランスを取ることで、過負荷を防止し、クラスタ全体で予測可能なパフォーマンスを確保できます。

## 反復的で大量の作業の自動化

反復的、大量、または運用上重要なキックオフタスクで自動化バックログを確立し、自動化が必須である場合に投資する。SIT環境で自動化を最初に検証し、改善することで、最終段階での修正に依存しない運用を実現します。規模の拡大により手作業のコストが増大。標準化された自動化により、品質、速度、制御が向上成果を再利用可能な資産（文書化されたインターフェイス、パラメータ化されたジョブ、共有ライブラリ）としてパッケージ化し、チームが今後のCrossworkのアップグレードや隣接プロジェクトに同じ自動化を活用できるようにします。これにより、再作業やオンボーディングの時間が短縮されます。

## パラレル実行時の二重クローズドループ制御の回避

共存している間は、クローズドループの自動化を単一ライター機能として扱います。1つのオーケストレーションパスだけで、修復やポリシー主導の設定をアクティブに実行できます。古いスタックと新しいスタックに対して同時CLAを実行すると、トリガーと意図が重複し、デバイスの状態が不安定になる可能性があります。機能の検証と新しいコントローラへのカットオーバーが完了した後、フェーズの後半のマイルストーンとしてCLAの実施を計画します。

## アップグレード構造化影響評価の実行

メジャーバージョンジャンプでは、古い機能を廃止または変更する際に、新しい機能が導入されます。これらの変化を考慮に入れることが非常に重要です。多くの場合、アップグレードされたバージョンのリリースノートに変更が記載されず、フィールドに移動するとポップアップ表示されます。次の項目に関する構造化された評価の実施：

- 非推奨のAPI
- KPIフレームワークの変更
- アプリケーションレベルの動作の違い
- 構成モデルの誤差
- アラート、トポロジ処理、プレイブック実行の変更

## 統合サーフェス全体での互換性と動作のテスト

CNCは、NSO、SSM、CPNR、EPNM、OneFM、Splunk、オーケストレーションフレームワークなどの複数の外部システムと通信します。

移行前：

- バージョンの互換性の検証
- すべてのノースバウンド/サウスバウンド統合をテスト
- データモデル、トラップ、テレメトリフローの確認
- SSL/RESTCONF認証動作の確認

移行後に統合障害が検出されると、運用上の盲点が生じます。

## 堅牢な移行前データエクスポート戦略の確立

移行を開始する前にすべてエクスポートする：

- クレデンシャルプロファイル
- プロバイダ
- タグ
- カスタムプレイブック
- カスタムKPI
- ロールとRBAC
- ZTPバウチャー
- デバイスグループ
- 履歴サービスメタデータ

## 検証ゲートが組み込まれたデバイスの一括移行

数千ものデバイスを移行する場合は、管理されたバッチで移行を実行します。

- 固定コホートでのデバイスの移動（領域、CDGロード、デバイスタイプなど）
- 次のバッチに移行する前に、テレメトリ、NSO同期状態、および到達可能性を検証する
- 永続的な異常が発生した場合は、バッチをロールバックします

これにより、短い間隔でCDGとCNCに高い負荷がかかるのを防ぐことができます。

## NSO統合によるアウトオブバンド構成変更の処理

CNC 4.1から7.1へのアップグレードの一環としてアウトオブバンドの課題に対処するために、NSO主導の運用への構造的な移行が実施されました。運用チームは、制御されたユーザベースのNSO CLIへのアクセスを提供されましたが、デバイスCLIへの直接管理アクセスは、アウトオブバンドの変更を防ぐために制限されていました。さらに、従来のCLISクリプトは、NSOと統合されたRESTCONFベースの自動化に体系的に変換され、ドライラン検証やトランザクションロールバックなどの機能を有効にしました。このアプローチにより、すべての構成変更が一元的に管理され、監査可能で、NSOのサービスモデルと整合性が保たれていることが保証され、構成の不整合が効果的に排除され、プロビジョニングの信頼性が復元されます。

## 変更の凍結を強く強調する

重要な移行の時間帯：

- ユーザが開始したネットワーク変更を凍結する
- 構成プッシュの制限
- フィールドチームおよびNOCチームとの同期
- CNC/ZTPを使用したデバイスの交換などの緊急作業に対応するための時間枠を計画します
- 

これにより、ノイズが軽減され、アップグレード中もネットワークの状態が安定します

## 結論

CNC 4.1からCNC 7.1への移行は、大規模なネットワークオーケストレーションプラットフォームのアップグレードに固有の複雑さにおける重要なケーススタディです。このプロジェクトは、このような移行が単なるバージョンの進化ではなく、アーキテクチャレイヤ、運用ワークフロー

、オートメーションエコシステム全体にわたる包括的な変革であることを実証しています。インプレースアップグレードパスが存在しないため、完全なリフトアンドシフト導入が必要となり、並行する環境の課題が生じ、CNC、NSO、SR-PCE、CDG、および外部システム統合の全体にわたって綿密な調整が必要になりました。その結果得られた運用状況から、堅牢な移行手法、包括的な検証サイクル、厳しく制御されたカットオーバープロセスが重要であることが明らかになり、実稼働環境でのリスクが軽減されました。

さらに、拡張性と正確性を確保する上で欠かせない要素である自動化の重要性も明らかになりました。2,000を超えるデバイス、広範なテレメトリ構成、複数の依存コンポーネント、動的なクローズドループ自動化ワークフローを備えたプロジェクトでは、この規模の環境における手動手順の限界が浮き彫りになりました。ACLのアップデート、デバイスオンボーディング、KPIのプロビジョニング、テレメトリクリーンアップ、および障害分離を網羅する専用の自動化は、決定性を確保し、人的エラーを削減し、効率を大幅に向上させるために不可欠であることが証明されました。自動化フレームワークは、移行時の運用継続性を実現しただけでなく、継続的なネットワーク最適化のための持続可能な基盤を確立しました。

同様に重要なのは、生産行動が管理された実験室の状態から著しく逸脱しているという認識でした。TickベースのKPIロジックからトラックベースの定義への移行などのフレームワークの変更により、再設計、再テスト、および反復的な改良を必要とする予期しない動作の移行が生じました。同様に、クローズドループの自動化、テレメトリの信頼性、APIの動作に関する運用上の課題から、適応型のトラブルシューティング、予防的なリスク評価、TACおよび事業部門の専門家との継続的な連携の必要性が浮き彫りになりました。これらの要因は、メジャーバージョンの移行に技術的な専門性と組織の準備状況の両方が必要であることを示しています。次のcrossworkバージョン7.2で解決される予定の未解決の問題はほとんど残っていません。

全体として、このアップグレードは、大規模なCNC移行を成功させるために、4つの基本的な柱が必要であることを示しています。それは、厳密な導入前検証、体系的で復元力のある自動化、強力な機能横断的な調整、ラボ環境と実稼働環境の間の相違を予測する適応型の運用ポスチャです。この取り組みから得られた知見は、安定したCNC 7.1の導入に貢献しただけでなく、将来の移行に向けたブループリントを提供し、ベストプラクティスを通知し、アーキテクチャのガードレールを強化し、ネットワーク自動化工コシステムのその後の進化に向けて制度的知識を強化しました。

## 用語集

用語	定義
BNM	帯域幅通知メッセージ。
CAT	Crosswork Active トポロジ

CCA	Crosswork Change Automation
CDG	Crosswork Data Gateway
チー	Crossworkヘルスインサイト
CNC	Cisco Crosswork Network Controller
COE	Crosswork最適化エンジン
CPNR	Cisco Prime Network Registrar
CWM	Crossworkワークフローマネージャー
起電力	エレメント管理機能
KPI	主要業績評価指標(KPI)
LWR	大規模ワイヤレスルータ
MDT	モデル駆動型テレメトリ
MOP	手順の方法
NBW	公称帯域幅
nso	ネットワーク サービス オーケストレータ
RBW	記録された帯域幅
SR-PCE	セグメントルーティングパス計算要素
SSM	Cisco Smart Software Manager
SWR	小型ワイヤレスルータ

TAC	Technical Assistance Center ( TAC )
TSDN	Software-Defined Networking(SDN)の転送
ZTP	ゼロタッチプロビジョニング
RR	ルート リフレクタ
RP	ルートプロファイル
POI	相互接続のポイント
EVPN	Ethernet Virtual Private Network ( イーサネットバーチャルプライベートネットワーク )。

## 参照資料

- [Cisco Systems, Cisco Crosswork Network Controllerリリースノート、リリース7.1.0](#)
- [シスコシステムズ、Cisco Crosswork Infrastructure 7.1インストールガイド](#)
- [Cisco Systems, Cisco Crosswork Infrastructure 7.1 Administration Guide : 概念の概要:](#)
- [シスコシステムズの『Crosswork Network Controller Traffic Engineering and Optimization Guide, Release 7.1』](#)
- [『Cisco Systems, Cisco Crosswork Health Insights User Guide, Release 7.1』](#)
- [Cisco Systems, Crosswork Zero Touch Provisioning\(ZTP\)導入ガイド](#)
- [シスコシステムズ『Cisco NSO Transport SDN Function Pack Bundle Installation Guide, Release 7.1.0』](#)
- [Cisco Systems, Cisco SR-PCEコンフィギュレーションガイド](#)

## 翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。