



NSO ベースの設定管理

- [機能説明 \(1 ページ\)](#)
- [機能の仕組み \(2 ページ\)](#)
- [CUPS 設定 MOP \(13 ページ\)](#)
- [トラブルシューティング \(50 ページ\)](#)
- [付録 A : 互換性のない StarOS ネイティブ コマンド シンタックス \(51 ページ\)](#)
- [付録 B : RCM を使用した N:M 展開の設定例 \(54 ページ\)](#)

機能説明

4G CUPS の Cisco Network Service Orchestrator (NSO) ベースの設定管理は、以下をサポートします。

- Cisco Virtual Network Function (VNF) デバイスのオンボーディング : CP、UP、および RCM
- Day-N、Day-1、および Day-0.5 CUPS 設定プッシュ用の 4G ベース CP、UP、および RCM の一元的な設定管理
 - Day-0.5 は、RCM を使用する N:M UP 冗長性スキームに適用されます。Day-0.5 設定は、UP の RCM との通信を目的としているため、そのロールを定義し、適切な設定を後からプッシュできるようになっています。

NSO 自動化を使用した 4G CUPS 展開の顧客構成管理の管理は、再利用性、標準的な通知管理、および体系的なデバイス構成ガバナンスといった機能性も発揮します。

使用例

NSO 設定処理は、次の使用例に対応します。

1. 管理 IP を使用してすでに展開されている VNF (CP、UP、および RCM) の NSO オンボーディング :
 - すでに実行されている VNF (CP、UP、および RCM) をデバイスとして NSO にオンボーディングし、事後チェックを実行して、デバイスの到達可能性と機能を確認しま

す。これは、設定をプッシュまたは同期し、通知の通信を確立するための準備手順です。

VNF デバイスのオーケストレーション（インスタンス化と破棄）は個別のモジュールであり、設定モジュールに依存しません。デバイスをオンボーディングし、設定管理をサポートするには、特定の詳細（IP、ポート、管理ユーザー名/パスワード）が必要です。

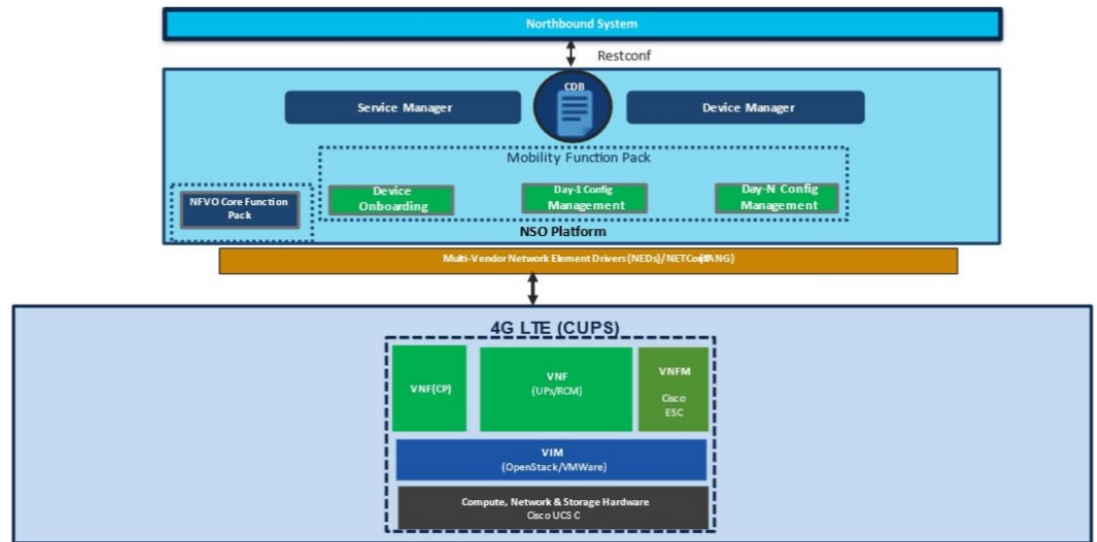
2. CP、UP、および RCM のネイティブ設定またはデバイステンプレートの保存を許可します。
 - RESTCONF/NSO-CLI を介してインターフェイスを提供し、論理名を持つデバイスの再利用可能な設定を管理します。また、ネットワーク SME やオペレータが設定を作成、変更、削除、無効化/有効化できる柔軟性を提供します。目的は、それらのアクティブな設定を選択し、CP、UP、および RCM の Day-0.5、Day-1、または Day-N の一部としてデバイスセットに適用することです。
3. デバイス論理グループ（CP/UP/RCM を含む）またはターゲットデバイスのカスタムリストに Day-N/Day-1/Day-0.5 設定を適用するための CLI/REST インターフェイスを提供します。
 - RESTCONF/NSO-CLI を介してネットワーク SME やオペレータにインターフェイスを提供し、Day-N/Day-1/Day-0.5 の設定を単一または一連のデバイス（CP、UP、または RCM）にプッシュします。このインターフェイスには、エンドユーザーに対する設定のプッシュの進行状況に関する通知やステータスが表示されます。
4. デバイスごとの設定管理のロジスティック管理（Day-N、Day-1、または Day-0.5 プッシュ）：
 - ダッシュボードユーティリティを提供し、デバイスごとに設定ログを管理します。このログは、デバイスで実行された最新のアクティビティを把握するのに役立ちます。
5. 5. (N:M の場合) RCM 通知管理の通知フレームワークを構築し、Day-N、Day-1、および Day-0.5 に関する UP への設定のプッシュを自動化します。
 - NSO で通知フレームワークを構築して、ステータスの変更に関する RCM NETCONF 通知をリッスンし、シナリオに基づいて設定を自動的にプッシュします。

機能の仕組み

アーキテクチャ

次の図は、ソリューションに関連するコンポーネントとフレームワークの概要を示しています。

NSO Based Automation Architecture



RCM と NSO

N:M UP 冗長性シナリオでは、NSO が設定を管理している間、RCM は引き続き UP のロール（アクティブまたはスタンバイ）の調停を行い、アクティブ UP のスイッチオーバーを処理するため、このソリューションでは、設定機能が RCM から NSO に移動されるだけで、RCM は引き続き必要です。RCM の詳細については、RCM コンフィギュレーションおよびアドミニストレーションガイド [英語] を参照してください。

コンポーネント

Cisco 4G CUPS VNF の導入と設定のワークフローは、NSO から実行されます。次に、NSO の重要なコンポーネントの一部を示します。

- **NSO デバイスマネージャ**：各 VNF コンポーネント（CP、UP、RCM）を管理し、各デバイス設定のコピーを保持し、デバイス設定プッシュの整合性を管理します。
- **NSO サービスマネージャ**：カスタマー/ユーザー入力の高レベルの抽象化ネットワークサービスモデルを定義する YANG 標準を提供します。
- **CDB**：ネットワーク設定と運用データを保存するための永続的な設定データベース。
- **モビリティ機能パック**：4G CUPS ベースの VNF オーケストレーションと設定を管理するためにカスタム構築された NSO パッケージ。
- **NFVO コア機能パック**：NSO コア NFV FP は、シスコまたは他のサードパーティの VNFM および VIM（OpenStack/VMWare など）と通信して VNF を展開するためのドライバソフトウェアです。

- **StarOS NSO NED** : 設定をプッシュするために Cisco 4G CUPS VNF と接続する StarOS ベースの NSO ネットワーク エlement ドライバ (NED)。この NED は Cisco CLI をベースとしています。StarOS NSO NED は、セキュアシェル (SSH) を使用して StarOS 管理 CLI インスタンスと通信します。
- **RCM NSO NED** : RCM ベースの NETCONF NED は、設定管理のために RCM デバイスと通信する際に使用されます。

プラットフォーム、ハードウェア、およびソフトウェアの最小要件

以下に、一元化された設定管理をサポートするためのプラットフォームおよびソフトウェアの最小要件を示します。

- サポートされるオーケストレータ : NSO
- 次のネットワーク要素の設定管理 :
 - RCM : 冗長性と設定の管理
 - VPC-SI : 4G CUPS CP または UP として
 - VPC-DI : 4G CUPS CP としてのみ
- 最小ハードウェア要件 :
 - VM CPU : CPU コア x 8
 - VM RAM : 基準は 16 GB RAM、サポートする StarOS デバイスごとに + 10 MB RAM
 - VM 接続 : 10 Gbps ネットワークリンク x 1。これは、個別の VLAN またはその他のメカニズムを使用して、NSO HA と構成/展開の両方に使用できます。
 - VM ストレージ : 100 GB ディスク (SSD を推奨)
- 最小ソフトウェアのバージョン

ソフトウェア	最小バージョン
Cisco NSO	6.1.6.1
StarOS NSO NED	5.52.4
Cisco NSO HCC	6.0.1
モビリティ機能パック	3.5



(注) UP、CP、および RCM で推奨される StarOS ソフトウェア イメージバージョンは、リリース 21.23 以降です。リリースバージョンは密接に結合されておらず、NED のみに依存します。

ライセンス

NSOベースの設定管理は、ライセンスに基づくシスコの機能です。特定のライセンス要件の詳細については、シスコのアカウント担当者にお問い合わせください。

NSO のインストール

コール フロー

この項では、4G CUPS 設定管理機能の主要なコールフローについて説明します。

コールフローは、「connect」、「sync-from」などの NSO プリミティブを参照します。プリミティブの詳細については、NSO ユーザーガイド [英語] を参照してください。



(注) 次のコールフロー図では、「NSO ノースバウンド」は NCS CLI または RESTCONF インターフェイスを意味します。

既存の 4G CUPS VNF の NSO へのオンボーディング

この項では、既存の 4G CUPS デバイスを NSO に追加するためのフローについて説明します。

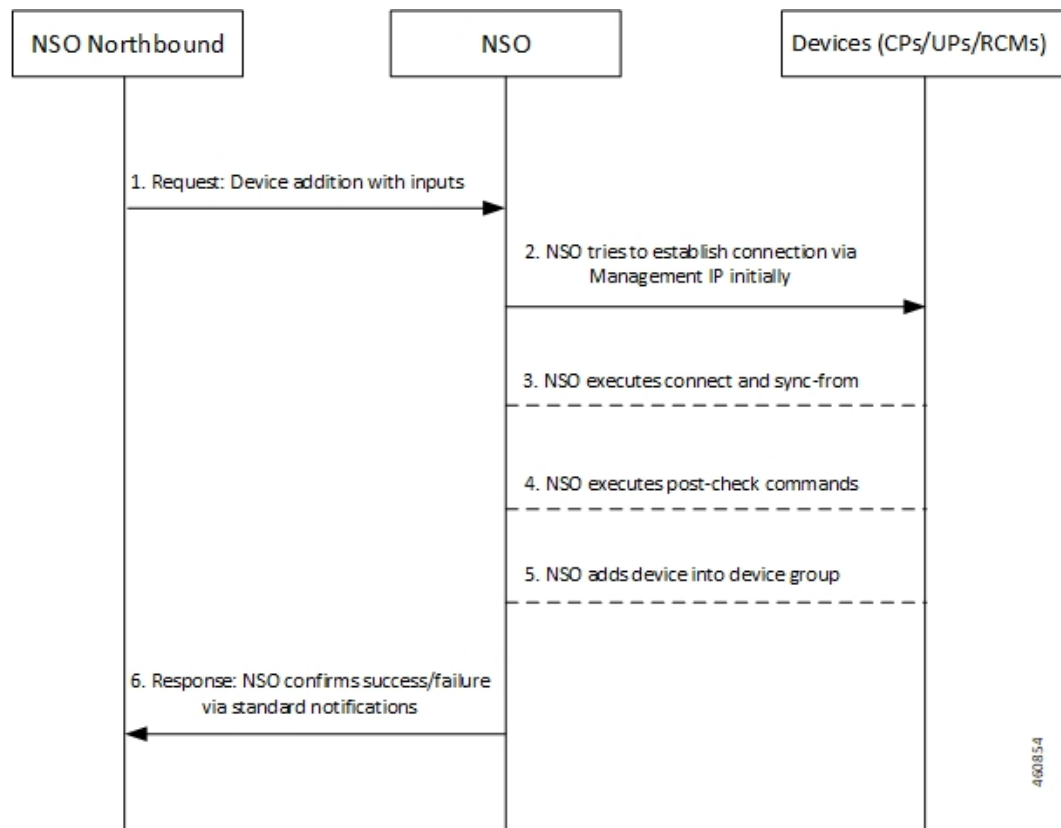


表 1: コールフローの説明

ステップ	説明
1	NSO ノースバウンドが、デバイスをデバイスグループに追加するための要求を NSO に送信します。 デバイスグループは、ほぼ同一の設定を共有するデバイス（VNF）の論理グループなので、場合によっては設定が簡素化されます。
2	NSO が最初に管理 IP を介して接続を確立しようとします。
3	NSO が connect および sync-from コマンドを実行します。 NSO がデバイス上の正確な設定を認識できるように、sync-from 操作によりデバイスや VNF から既存の設定がプルされて NSO に入力されます。デバイス設定は sync-from では変更されません。
4	NSO が post-check コマンドを実行します。
5	NSO がデバイスをデバイスグループに追加します。
6	標準通知を介したデバイスの追加の成功や失敗について、NSO が NSO ノースバウンドを更新します。

4G CUPS デバイス設定のプッシュ：手動

この項では、4G CUPS デバイスの手動設定のプッシュに関するフローについて説明します。このシナリオは、スタンドアロンまたは1:1冗長構成のCP、UP、またはRCMに当てはまりません。

設定をプッシュする前に、デバイスをオンボーディングする必要があります。[既存の4G CUPS VNF の NSO へのオンボーディング \(5 ページ\)](#) を参照してください。

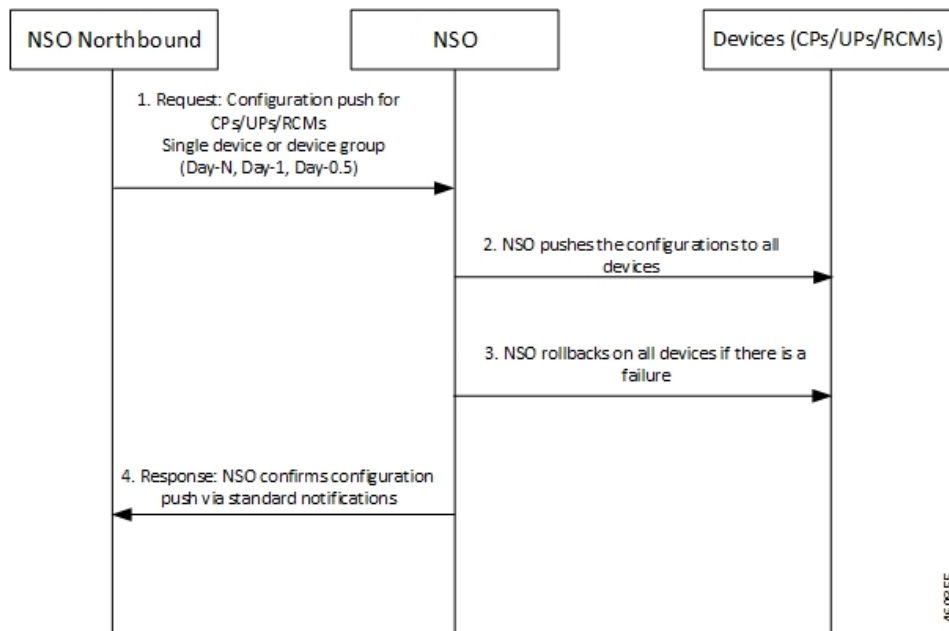


表 2: コール フローの説明

ステップ	説明
1	NSO ノースバウンドが、CP、UP、RCMなどのデバイスにおける設定のプッシュをNSOに要求します。デバイスは、単一のデバイスまたはデバイスグループ（Day-N、Day-1、Day-0.5）を指定できます。
2	NSO がすべてのデバイスに設定をプッシュします。
3	障害が発生した場合、NSO がすべてのデバイスの設定をロールバックします。ロールバック操作により、デバイスに適用された設定が元に戻り、デバイスが以前の状態（設定を適用する前）に復元されます。
4	NSO が、標準通知を介して設定のプッシュに関してNSO ノースバウンドを更新します。

N:M 冗長性での NSO から 4G CUPS UP への設定のプッシュ：自動化

ここでは、NSO から 4G CUPS デバイスへの自動設定プッシュのフローについて説明します。

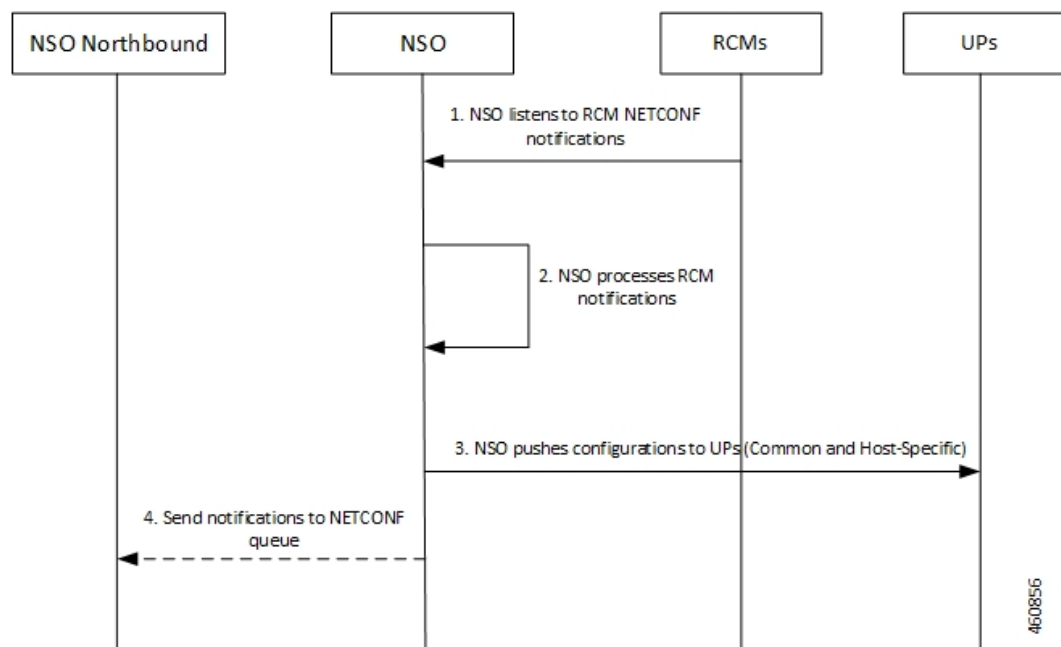


表 3: コールフローの説明

ステップ	説明
1	NSO は RCM NETCONF 通知をリッスンします。 UP が RCM に接続すると、RCM は UP のロール（アクティブまたはスタンバイ）を判断します。このロールは通知で伝えられます。UP にプッシュされる設定は、そのロールによって異なります。そのため、自動設定プッシュは RCM 通知に基づいて実行されます。
2	NSO は受信した RCM 通知を処理します。
3	NSO は共通設定およびホスト固有の設定を UP にプッシュします。 共通設定とは、冗長グループ内のすべての UP で共有される設定を指します。これは通常、Enhanced Charging Service (ECS) と Access Point Name (APN) の設定です。ホスト固有の設定は、アクティブ UP に固有です。各アクティブ UP には、ホスト固有の設定が必要です。スタンバイ UP はアクティブ UP を引き継ぐ必要があるため、すべてのスタンバイ UP にはすべてのアクティブ UP のホスト固有の設定が必要です。
4	NSO は NETCONF キューに通知を送信します。

設定メタデータの事前入力

ここでは、設定メタデータの事前入力フローについて説明します。

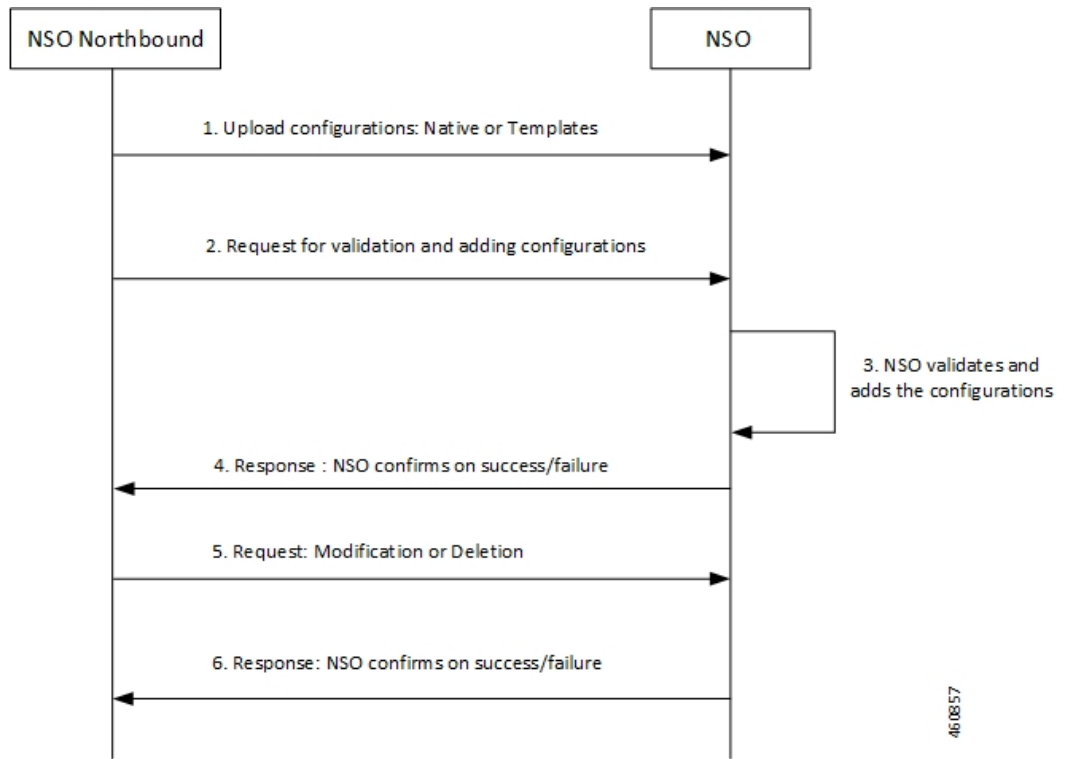


表 4: コールフローの説明

手順	説明
1	NSO のノースバウンドが、設定（ネイティブまたはテンプレート）を NSO にアップロードします。
2	NSO のノースバウンドが、設定の検証と追加を NSO に要求します。
3	NSO が設定を検証して追加します。
4	デバイス追加の成功または失敗について NSO が NSO のノースバウンドを更新します。
5	NSO のノースバウンドが、設定の変更または削除を NSO に要求します。
6	成功または失敗について NSO が NSO のノースバウンドを更新します。

NSO HA スイッチオーバーの処理

この項では、NSO HA スイッチオーバーの処理フローについて説明します。

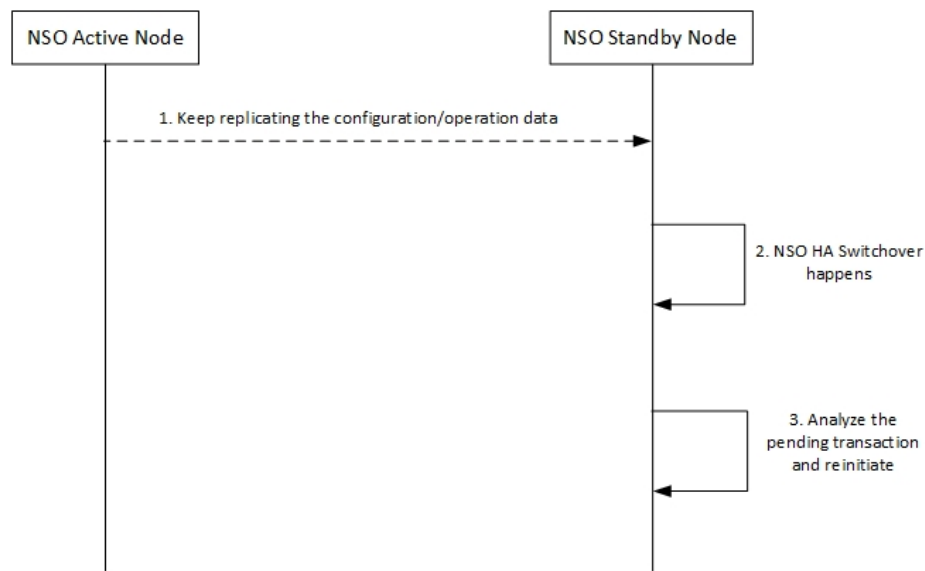


表 5: コールフローの説明

ステップ	説明
1	NSO スタンバイノードが、NSO アクティブノードからの設定や動作データの複製を続けます。
2	NSO HA スイッチオーバーが、NSO スタンバイノードで発生します。
3	NSO スタンバイノードが保留中のトランザクションを分析し、プロセスを再開します。

リカバリ

障害状態から以前の状態にリカバリするために、NSOにはプッシュされた設定に対する組み込みのロールバックメカニズムがあります。1つ以上のデバイスに設定をプッシュするには、次のオプションを使用できます。

- コミットまたは **dry-run** のみ
- コミット（ロールバックの生成あり）
- 単一または複数のトランザクションのスキーム
- 複数のトランザクションにおける障害処理のスキーム
- スタンバイノード、アクティブノード、または共通ノードのみをプッシュするスキーム

CP スイッチオーバー (1:1)

モビリティ機能パックは、アクティブな CP の積極的な追跡は行いません。ノースバウンドから設定のプッシュが開始されると、必要に応じて、オンデマンドで追跡します。いずれかの CP にプッシュされた設定は、その CP の起動設定として永続的に保存されます。



(注) 設定を永続的に保存するには、MOP オプションを使用する必要があります。

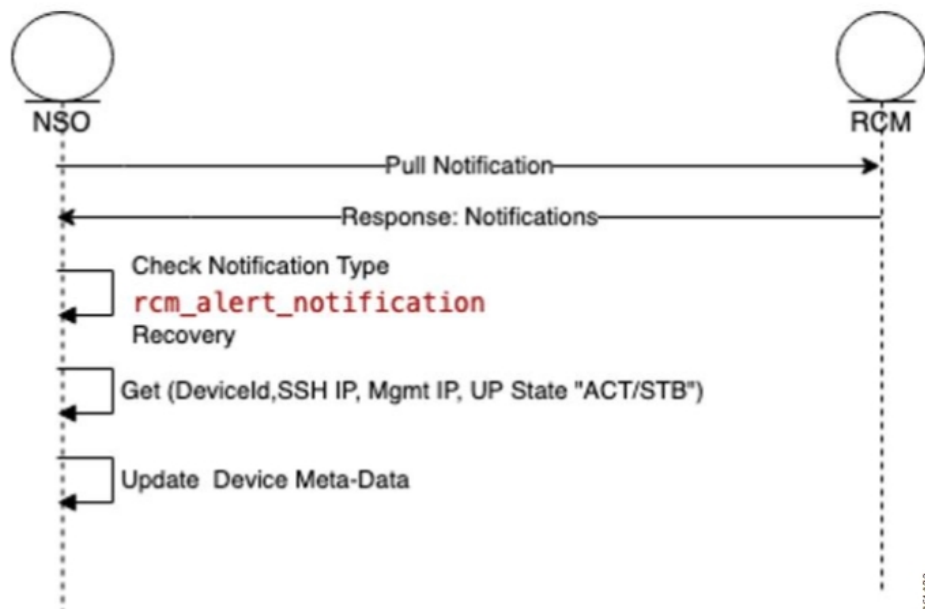
CP のスイッチオーバーでは、リブートされた CP は、必要なすべての設定を保持したまま起動することが求められます。モビリティ機能パックは、このシナリオで特別な処理を実行しません。

UP スイッチオーバー (1:1)

CP シナリオと同様に、いずれかの UP インスタンスにプッシュされた設定は、起動設定の一部として永続的に保存されます。設定を永続的に保存するには、MOP オプションを使用する必要があります。UP スイッチオーバーでは、再起動する UP は必要な設定がすべて設定された状態で自動的に起動することが期待されます。この場合、モビリティ機能パックでは特別なアクションは実行されません。

UP スイッチオーバー (N:M)

次の図は、UP リカバリ通知フローを示しています。



NSO が RCM デバイス通知に登録されると、NSO は [rcm-alert-notification] ストリームにパブリッシュされたすべての通知を取得できるようになります。

NSO は、UP リカバリ通知があるたびに、次の手順を実行します。

1. アラートステータスが [Recovery] になるまで待機します。
2. デバイスの詳細（デバイス名、SSH IP、管理 IP、UP ステータス）を取得します。
3. NSO のデバイスメタデータを更新します。

アウトオブバンド設定

NSO ベースの設定管理の基本となるのが、NSO で各デバイス（VNF）の設定のコピーを維持することです。設定の変更がノースバウンドから適用されると、NSO は適用された設定と設定のローカルコピーを比較して、デバイスにプッシュする必要がある設定を決定します。これを行うには、NSO にある設定のコピーがデバイス/VNF の実際の設定と同期している必要があります。

VNF 設定が NED をバイパスしてアウトオブバンドで実行されるケースには、理由が考えられます。たとえば、Day-0 設定は必ずデバイスを NSO にオンボーディングする前に行われるため、アウトオブバンドになります（適切な VNF マネージャを介してプッシュされます）。設定プッシュ MOP は、デバイスに設定をプッシュする前に「sync-from」操作を実行します。これにより、NSO はアウトオブバンド設定を NSO のローカルコピーにプルし、試行された設定のプッシュは最新のデバイス設定に適用されます。sync-from は、NED に認識されている設定のみをプルできます。また、暗号化されたデータを処理する場合の注意事項があります。

設定の機密要素

StarOS は、パスワード、キーなどの設定内にある機密要素を暗号化します。暗号化された項目は、StarOS によってのみ復号できるため、NSO には不透明です。さらに、基になるクリアテキストが変更されない場合でも、機密項目の暗号化フォームが変更される可能性があります。その結果、NSO は、そのような項目に加えられたアウトオブバンド変更を確実に検出できません。

次のいずれかを行うことが推奨されます。

- 対応する設定をアウトオブバンドで完全に管理します。

または

- NSO のみを使用して対応する設定を管理します。つまり、最初にノースバウンドから NSO にコマンドのクリアテキストフォーマットを設定し、その後は変更のたびに設定する必要があります。

同じ設定に NSO ベースの設定管理とアウトオブバンド管理を混在させないでください。

合法的傍受

合法的傍受 (LI) は、いくつかの方法で設定できます。1つの展開には、専用の LI コンテキスト機能を使用せずに、単一のコンテキスト内のすべての LI 設定が含まれ、一般的なシステム管理者に LI 管理者権限が付与されます。もう1つの展開には、専用の LI コンテキスト、分離された LI 設定、および一般的なシステム管理者とは別の専用の LI 管理者が含まれます。それらの中間に位置する他のバリエーションもあります。

NSO で LI 設定を管理できるようにするには、次の要件を満たす必要があります。

- LI 権限と一般的なシステム管理者権限を持っている
- クリアテキストで LI 設定を表示およびプルできる

シングル コンテキスト シナリオですべての LI 設定を含む展開の場合、NSO で LI 設定を管理する必要があります。それ以外の場合は、LI 設定をアウトオブバンドで維持し、Day-0 設定の一部として提供することを推奨します。

CUPS 設定 MOP

設定 MOP は、Cisco StarOS デバイスまたは RCM に設定を適用するための Method of Procedure (MOP) です。この操作はネットワークオペレータによって呼び出され、その応答として、NSO が要求の一意のタスク ID を提供します。ネットワークオペレータは、後からタスク ID を使用して NSO をポーリングし、ステータスを取得できます。

設定 MOP は、次の3つの手順で構成されています。

1. デバイスのオンボーディング
2. 設定メタデータの事前入力
3. モビリティ MOP を介した設定のプッシュ

デバイスのオンボーディング

デバイスのオンボーディング手順は、モビリティ オーケストレーション ソリューションの外部でインスタンス化またはオーケストレーションされるデバイスにのみ必要です。そうでない場合、インスタンス化された VNF は、NSO ベースのモビリティ オーケストレーション ソリューションによってデバイスとして NSO に暗黙的にオンボードされます。



- (注) この手順は、初回のみ必要です。後続の設定プッシュでは、この手順をスキップする必要があります。

次の例は、RESTCONF または CLI を使用して VNF をオンボードする方法をそれぞれ示しています。

RESTCONF

NSO URL のパッチ要求 (<http://<NSO-IP>:<PORT>/restconf/data>)

次に設定例を示します。

```
{
  "data": {
    "nfv-device-onboarding:nfv-devices": {
      "device": [
        {
          "name": "<Device-or-VNF-Name>",
          "address": "<Management-Address>",
          "username": "<Management-Username>",
          "password": "<Management-Password>",
          "ned-type": "<cisco-staros/RCM>",
          "retry-options": {
            "number-of-attempts": <no-of-attempts-to-ping>,
            "delay": <delay-between-pings>
          }
        }
      ]
    }
  }
}
```

次に設定例を示します。

```
{
  "data": {
    "nfv-device-onboarding:nfv-devices": {
      "device": [
        {
          "name": "vpc-si25",
          "address": "209.165.200.225",
          "username": "admin",
          "password": "Cisco@123",
          "ned-type": "cisco-staros",
          "retry-options": {
            "number-of-attempts": 2,
            "delay": 10
          }
        }
      ]
    }
  }
}
```

CLI

次の NSO CLI コマンドを使用して、デバイスのオンボーディングを作成/入力することもできます。

configure

```
nfv-devices device device_name address ip_address username user_name password
password ned-type cisco-staros retry-options delay delay_value
number-of-attempts value
```

commit

次に設定例を示します。

```
nfv-devices device dummy-device address 209.165.200.225 username admin password cisco@123
ned-type cisco-staros retry-options delay 10 number-of-attempts 2
```

NSO の既存のデバイスは、次の削除要求 URL（ペイロードなし）を使用して削除できます：
<http://<NSO-IP>:<NSO-PORT>/restconf/data/nfv-device-onboarding:nfv-devices/device=<device-name>>

次の NSO CLI を使用して削除することもできます。

```
configure
no nfv-devices device device_name
commit
```

設定メタデータの事前入力

設定プッシュ MOP では、変数の置換が可能です。これは、ほぼ同一の設定が複数のデバイス（たとえば、ICSR アクティブ/スタンバイペア）にプッシュされる場合に役立ちます。違いは、入力設定ファイルの変数として表すことができます。その後、各デバイスの特定の値を「variable: value」形式のメタデータとして入力できます。MOP は、実行時に適切な変数値を動的に置き換えます。

デバイスに事前入力されたデータがない場合、Config MOP は、設定プッシュ用に指定された設定ファイルに動的置換変数がないと想定します。設定ファイルで参照される属性値が欠落している場合、この手順は実行時に失敗します。

設定メタデータの事前入力の構造を以下に記載します。このデータの入りはネットワークスキームとデータセットに基づきます。強調表示された項目は設定をプッシュするための必須項目であり、その他の項目はオプションです。

```
container metadata-store{
    list config-metadata {
        key device-name;

        leaf device-name {
            tailf:info "onboarding device name";

            type string;
        }

        leaf schema {
            tailf:info "cluster-topology 1:1, N:M and N+2";

            type string;
        }

        list attributes {
            key attribute-name;
```

```
        leaf attribute-name {
            tailf:info "Attribute Name";

            type string;
        }

        leaf attribute-value {
            tailf:info "Attribute Value";

            type string;
        }
    }

list configuration-type {
    key config-type;

    tailf:info "Configuration type Day0.5, Day1 or DayN";

    leaf config-type {
        type string;
    }

    list files {
        key file-name;

        tailf:info "file name";

        leaf file-name {
            type string;
        }

        leaf config-scheme {
            type string;
        }
    }

    // CP device info

    list additional-files {
        key device; //cp device

        leaf device{
            tailf:info "device name";

            type string;
        }
    }
}
```



```
    }  
    list additional-file {  
        key additional-file-name;  
        leaf additional-file-name {  
            tailf:info "file name";  
            type string;  
        }  
    }  
}  
}  
}  
}
```

設定メタデータは、設定メタデータ要求を使用して入力されます。この要求は、次の YANG スキーマに従います。「input」セクションの項目は、オペレータが入力します。「output」セクションは、アクション要求の実行後に NSO によって返される内容を示します。

設定メタデータを入力または変更する NSO アクションの例を以下に示します。

```
tailf:action config-metadata-request {  
    tailf:info "Invoke upgrade action on the selected devices";  
    tailf:actionpoint config-metadata-request;  
    input {  
        list config-metadata {  
            key device-name;  
            leaf device-name {  
                tailf:info "onboarding device name";  
                type string;  
            }  
        }  
        list attributes {  
            key attribute-name;  
            leaf attribute-name {  
                tailf:info "Attribute Name";  
                type string;  
            }  
            leaf attribute-value {  
                tailf:info "Attribute Value";  
                type string;  
            }  
        }  
    }  
}
```

```

    output {
    leaf status {
        type string;
    }
    }
}
}
}

```

RESTCONF

RESTCONF からこのアクションを呼び出す例を以下に示します。

URI :

http://<NSO-IP>:<NSO-REST-PORT>/restconf/data/mobility-common:config-metadata/config-metadata-request

コンテンツタイプ : application/yang-data+json

ペイロード :

```

{
  "config-metadata": {
    "device-name": "test2",
    "schema" : "1:1",
    "attributes":{
      "attribute-name":"hostname",
      "attribute-value": "TEST"
    },
    "attribute-name":"BACKHAUL_IP",
    "attribute-value": "209.165.200.225"
  }
}
}

```

結果 :

```

{
  "mobility-common:output": {
    "status": "Success"
  }
}

```

CLI

次に、NCS CLI を使用してこのアクションを呼び出す例を示します。

```

ubuntu@ncs> request config-metadata config-metadata-request config-metadata { device-name
  staros-1 attributes { attribute-name hostname attribute-value TEST }
  status Success
[ok] [2021-07-12 08:05:01]

```

モビリティ MOP を介した設定のプッシュ

この手順は、設定 MOP の最後の手順で、新しい設定をプッシュしたり、以前にプッシュした設定をロールバックしたりできます。前述したとおり、プッシュされる設定は1つ以上のファイルに存在します。

設定 MOP プッシュ要求フロー

ネットワークオペレータが NSO API を呼び出して、デバイスの設定 MOP 自動化プロセスを開始

NSO は次の手順を実行します。

- デバイスで `check-sync` と `sync-from` または部分同期（必要な場合）を実行します。 `check-sync` により、デバイス構成の NSO コピーが実際のデバイス構成と同期しているかどうかを確認されます。
- NSO が MOP で指定されている場合、NSO はデバイス属性（変数）を設定メタデータのデバイスツリーから読み取ったノード固有の値に置き換えます。
- NSO は、MOP で指定された入力ファイルの設定をそのデバイスに適用するか、要求で指定されている順序で一連のデバイスに適用します。設定をデバイスにプッシュするときに障害が発生した場合、そのデバイスにそれ以上の設定はプッシュされません。
- NSO は、要求で指定されている `mop type` (`active/standby/common`) に基づいて MOP を適用します。

- `mop type` が「`common`」の場合、NSO は要求で指定されているすべてのデバイスに MOP を適用します。

障害が発生した場合、障害が発生したデバイスへの設定プッシュは停止されます。要求で指定されている他のデバイスへの設定プッシュは続行されます。「Status」には、障害が発生したデバイスの詳細が表示されます。オペレータは、障害が発生したデバイスの設定を個別にロールバックすることができます。

- `mop type` が「`active`」の場合、NSO は要求で指定されているすべての「アクティブ」デバイスに MOP を適用します。

`mop type` 「`active`」は、1:1 冗長性シナリオにのみ適用されます。

障害が発生した場合、プッシュされた設定はすべてロールバックされます。

- `mop type` が「`standby`」の場合、NSO は要求で指定されているすべての「スタンバイ」デバイスに MOP を適用します。

`mop type` 「`standby`」は、1:1 冗長性シナリオにのみ適用されます。

障害が発生した場合、プッシュされた設定はすべてロールバックされます。

- `mop type` が「`pair`」の場合、NSO は最初に「スタンバイ」デバイスに MOP を適用し、成功した場合は「アクティブ」デバイスに MOP を適用します。NSO はアトミックトランザクションを実行するため、設定は両方のデバイスに適用されるか、どちらにも適用されません。

`mop type` 「`pair`」は、1:1 冗長性シナリオにのみ適用されます。

障害が発生した場合、適用された設定はペアの両方のインスタンスからロールバックされます。

- mop type が「rcm-upf」の場合、NSO は入力デバイスに MOP を適用します。さらに、入力デバイスの RD グループを識別し、同じ RD グループに存在する他の UPF デバイスを見つけます。次に、入力デバイスに ECS/APN 設定を保存します。

障害が発生した場合、障害が発生したデバイスへの設定プッシュは停止されます。要求で指定されている他のデバイスへの設定プッシュは続行されます。[Status] には、障害が発生したデバイスの詳細が表示されます。オペレータは、障害が発生したデバイスの設定を個別にロールバックすることができます。

- NSO は、指定された MOP のドライランおよびリバース（ロールバック）設定をネイティブフォーマットで生成し、2つの個別のファイルに保存します。NSO は応答として、両方のファイル名と絶対ファイルパスをネットワークオペレータに返します。
 - ドライランファイルの名前は <MOP File Name>-<Device Name>-dryrun.txt になります。
 - ロールバックファイルの名前は <MOP File Name>-<Device Name>-rollback.txt になります。ファイルはタスク ID フォルダの下に生成されます。
- ネットワークオペレータがドライランの要求のみを送信した場合、NSO はドライランファイルとロールバックファイルを生成しますが、デバイスに MOP を適用しません。
- ネットワークオペレータが MOP の適用要求を送信すると、NSO はドライランファイルとロールバックファイルを生成し、MOP をデバイスに適用します。
- ネットワークオペレータは、MOP 自動化ステータスを取得するために NSO のポーリングを継続します。
- NSO はホスト（デバイス）のリストとともに、ドライランファイルとロールバックファイルの場所、およびステータス（完了/進行中/失敗）を返します。

設定 MOP ロールバック要求フロー

- ネットワークオペレータは、NSO API を呼び出して、以前に適用された設定のロールバックプロセスを開始します。
- NSO は次の手順を実行します。
 - デバイスで check-sync と sync-from または部分同期（必要な場合）を実行します。
 - NSO は、ネットワークオペレータによって指定されたタスク ID、MOP ファイル名、およびデバイス名の逆の順序で MOP ファイルのロールバックを実行します。
- MOP タイプが「pair」の場合、NSO は最初に「スタンバイ」デバイスでロールバックを実行し、ロールバックが成功すると、NSO は「アクティブ」デバイスでロールバックを実行します。
- タスク ID のみが指定されている場合は、トランザクション全体がロールバックされます。タスク ID と MOP ファイル名が指定されている場合、指定された MOP のみがすべてのデバイスに対してロールバックされます。タスク ID、MOP ファイル名、およびデバイス名

が指定されている場合、指定されたデバイスに対して指定された MOP のみがロールバックされます。

- NSO は、ロールバックを実行するためのドライランおよびリバース（ロールバック）設定をネイティブフォーマットで生成し、ファイルに保存します。NSO は応答として、両方のファイル名と絶対ファイルパスをネットワークオペレータに返します。
 - ドライランファイルの名前は <MOP File Name> -<Device Name> -dryrun.txt になります。
 - ロールバックファイルの名前は <MOP File Name> -<Device Name> -rollback.txt になります。

ファイルはタスク ID フォルダの下に生成されます。

- ネットワークオペレータがドライラン要求のみを送信した場合、NSO はドライランファイルとロールバックファイルを生成します。
- ネットワークオペレータが MOP のロールバック要求を送信すると、NSO はドライランファイルとロールバックファイルを生成し、ロールバックを実行します。
- ネットワークオペレータは、ロールバックステータスを取得するために NSO のポーリングを継続します。
 - NSO はホスト（デバイス）のリストとともに、ドライランファイルとロールバックファイルの場所、およびステータス（完了/進行中/失敗）を返します。

MOP の自動化

モビリティ設定 MOP は、NSO からモビリティデバイスを設定するために使用できる一連のコマンドです。モビリティ設定 MOP では、エンドユーザーが場所を指定して、MOP 関連の入力ファイルと出力ファイルを検索または保存できます。また、MOP のグローバル設定可能なパラメータも設定できます。

設定要件

- NSO CLI に移動し、静的アクションを使用して以下のパラメータを設定します。
 1. Dry-run-mop location : Dry-run-mop ファイルには、デバイスにプッシュされた設定が含まれています。MOP の dry-run ファイルを保存する場所を入力します。
 2. Rollback-mop location : ロールバックファイルは生成された構成ファイルで、デバイスの設定をロールバックするために必要です。MOP のロールバックファイルを保存する場所を入力します。
 3. Config-mop-file location : 入力設定 MOP ファイルを取得する場所を入力します。
 4. Netconf-to-cli Conversion : NETCONF 設定をデバイス CLI 形式に変換するには、フラグを「true」に設定します。フラグが「false」に設定されている場合、dry-run ファイルはネイティブ NETCONF xml 形式で生成されます。

- 設定内の static action call コマンド :


```
static dry-run-mop /var/opt/ncs/
static rollback-mop /var/opt/ncs/
```

 確認するには、次の CLI コマンドを使用します。


```
show full-configuration static
```
- mop-file location のグローバルパラメータ設定 :


```
configure
  configurable-parameters config-mop-file-loc /var/opt/ncs/
```

 確認するには、次の CLI コマンドを使用します。


```
show full-configuration configurable-parameters config-mop-file-loc
```
- StarOS レベルの NED 設定の例
 1. デバイスのシステム cfg ブートファイルの設定が更新されないようにするには、次のコマンドを使用して、NCS CLI で write-memory-setting が無効になっていることを確認します。


```
devices global-settings ned-settings cisco-staros
write-memory-setting disabled
```
 2. デバイスに設定をコミットする際の警告を除外するには、次のコマンドを使用します。


```
devices global-settings ned-settings cisco-staros behaviour
config-warning-ignore.*Standby card not ready.*
```



(注) .*Standby card not ready.* は警告メッセージに置き換え可能で、無視できます。

Mop タイプペアの前提条件

- デバイスの状態（アクティブ/スタンバイ）の識別に基づいて、デバイス名の 1 つを target-device-name として指定できます。
- 次のコマンドを使用して、ピアデバイスと srp_loopback を設定します。

```
config-metadata config-metadata-request config-metadata { device-name
up2-SI device-type vpc attributes { attribute-name srp_loopback
attribute-value 209.165.200.225 } scheme 1:1 }
config-metadata config-metadata-request config-metadata { device-name
up2-SI device-type vpc attributes { attribute-name Peer_Device_Name
attribute-value up1-SI } scheme 1:1 }
```

NSO API

NSO API は、設定プッシュ機能に関連する NSO モビリティ機能パックを通じて公開されます。これらの API には、RESTCONF または CLI のいずれかを介してアクセスできます。

設定プッシュ MOP の自動化

この API は、MOP を開始して 1 つ以上のデバイスに設定をプッシュするために使用されます。これは非同期操作であり、別の API を使用してステータスをクエリできます。

API :

mop-automation

要求の詳細

パラメータ	書式	必須	説明
mop-file-name	リスト	—	
file-name	文字列	キー	UP、CP、または RCM に対応するデバイスの名前。
execution-order	数値	必須	MOP の実行順序。1 が最初で、使用される順序は 1、2、3... です。
target-devices	リスト	必須	デバイスのリスト
target-device-name	Leafref	キー	VNF の NSO デバイス名。NSO がオーケストレーションに使用されている場合、デバイス名は VNF 名と同じです。
operation-type	文字列	必須	dry-run 、または commit

パラメータ	書式	必須	説明
mop-type	列挙体	—	

パラメータ	書式	必須	説明
			<p>設定をプッシュする先のデバイスを決定します。使用できる値は、次のとおりです。</p> <ul style="list-style-type: none"> • active <p>1:1 冗長性ペアのアクティブインスタンスにプッシュします。冗長性ペアのいずれかのインスタンスを選んでデバイス名を入力できます（アクティブかスタンバイかは問いません）。MOPが該当するペアの現在アクティブなインスタンスを自動的に判別して、プッシュします。</p> • standby <p>1:1 冗長性ペアのスタンバイインスタンスにプッシュします。冗長性ペアのいずれかのインスタンスを選んでデバイス名を入力できます（アクティブかスタンバイかは問いません）。MOPが該当するペアの現在スタンバイ中のインスタンスを自動的に判別して、プッシュします。</p> • pair <p>1:1 冗長ペアのアクティブインスタンスとスタンバイインスタンスの両方にプッシュします。冗長性ペアのいずれかのインスタンスを選んでデバイス名を入力できます（アク</p>

パラメータ	書式	必須	説明
			<p>タイプかスタンバイかは問いません)。MOPは、指定されたインスタンスをもとに該当するペアの2つのインスタンスを自動的に判別し、最初にスタンバイインスタンスにプッシュしてから、アクティブインスタンスにプッシュします。</p> <ul style="list-style-type: none">• common 要求で指定されたすべてのデバイスにプッシュします。これがdefaultです。• rcm-upf 単一またはすべての関連付けられたUPFに設定をプッシュします。

パラメータ	書式	必須	説明
transaction-type	列挙体	—	

パラメータ	書式	必須	説明
			<p>使用できる値は、次のとおりです。</p> <ul style="list-style-type: none"> • single-transaction 指定されたすべての入力ファイルの設定が結合され、単一のトランザクションとしてデバイスにプッシュされます。 • multiple-transaction 各入力ファイルが、個別のトランザクションとしてデバイスにプッシュされます。これはデフォルト値です。 <p>トランザクションは、設定変更の最小単位です。トランザクション内の設定はすべて、正常にプッシュされるか、プッシュ中に障害が発生した場合は自動的にロールバックされます。</p> <p>(注) 1つのトランザクションが複数のデバイスにまたがることはありません。各トランザクションは、1つのデバイスに固有です。</p> <p>たとえば、オペレータが3つのファイルをそれぞれ2つのデバイスにプッシュする場合、次のようになります。</p> <ul style="list-style-type: none"> • multiple-transactions オプションを使用すると、デバイスごと、ファイルごとに1つなので、トランザクショ

パラメータ	書式	必須	説明
			<p>ンの合計数は $3 \times 2 = 6$ となります。</p> <ul style="list-style-type: none"> • single-transaction オプションを使用すると、デバイスごとに1つずつなので、トランザクションの合計数は2となります。
save-config-permanently	ブール値	—	デフォルト値は「false」です。「true」に設定すると、設定を「system.cfg」に保存できます。
timeout	数値	オプション。デフォルト値は 600 秒	デバイスがロックされるまで待機する最大秒数。

Timeout パラメータ

NSO 6.0バージョンは、楽観的同時実行を使用してパラレルリズムを向上させます。ただし、サービスを並行して実行すると、トランザクションの競合が発生する可能性があります。単一のデバイスで同時実行が設定されている場合、最初のトランザクションによってデバイスがロックされるため、後続のトランザクションは失敗します。

timeout パラメータは、デバイスロックの取得など、デバイスに関連する一部の操作が実行されるまで MFP が待機する時間を規定します。timeout パラメータが直接関係するのは、同じデバイスに同時に設定をプッシュする複数の操作がある場合です。



(注) 単一のデバイスでの同時並行処理設定は推奨されません。

mobility-mop 自動化 CLI または **postman** API を使用して設定をプッシュする際に **timeout** パラメータが使用されていない場合、システムは自動的にデフォルト値である 600 秒を呼び出します。

多数の設定または大きなサイズの設定をプッシュする場合は、**timeout** パラメータを使用して、デフォルト値を超えるタイムアウト値を設定するコールを実行できます。

次の設定例に示すように、任意の値を指定できます。

```
cloud-user@ncs# mobility-mop:action mop-automation generate-dry-run true operation-type
  commit save-config-permanently true mop-type common mop-file-name { file-name
ABC.cfg order
  1 target-devices-list { target-device-name XYZ } } timeout 900
```

タイムアウト値に -1 を指定すると、**timeout** パラメータを無限に設定できます。

```
cloud-user@ncs# mobility-mop:action mop-automation generate-dry-run true operation-type
commit save-config-permanently true mop-type common mop-file-name { file-name
ABC.cfg order
1 target-devices-list { target-device-name XYZ } } timeout -1
```

設定がプッシュされるとすぐに、別のユーザーまたは後続の設定のプッシュに向けてデバイスが解放されます。たとえば、`timeout` が 600 秒で、設定のプッシュが 100 秒で完了する場合、100 秒後には別のユーザーがそのデバイスに設定をプッシュできます。

応答の詳細

パラメータ	書式	必須	説明
Task-id	文字列	必須	タスクの一意の識別子。Task-id は、操作のステータスのクエリに使用されます。
Time stamp	文字列	必須	Time stamp
Error Code	文字列		エラーコード
Error Message	文字列		エラーメッセージ

CLI

NCS CLI を使用した要求の例を以下に示します。

```
mobility-mop:action mop-automation mop-type common transaction-type
multiple-transaction operation-type commit mop-file-name { file-name
dayN.txt order 1 target-devices-list { target-device-name up2-SI } }
```

REST API 要求：トランザクションタイプを指定しない場合

以下に、トランザクションタイプを指定せずに `postman` を使用する REST API 要求の例を示します。

```
POST /restconf/data/mobility-mop:action/mop-automation
Host: localhost:8080
Authorization: Basic YWRtaW46YWRtaW4= Content-Type: application/vnd.yang.data+json
cache-control: no-cache
```

```
{
  "mop-automation": {
    "operation-type": "commit",
    "mop-type": "active",
    "generate-dry-run": "true",
    "save-config-permanently": "true",
    "mop-file-name": [
      {
        "file-name": "up_dayN.txt" ,
        "order": 1,
        "target-devices-list": [
          {
            "target-device-name": "up2-SI"
          }
        ]
      }
    ]
  }
}
```

```

    ]
  }
}

```

REST API 要求：トランザクションタイプを指定する場合

次に、トランザクションタイプを指定する REST API 要求の例を示します。

```

POST /restconf/data/mobility-mop:action/mop-automation
Host: localhost:8080
Authorization: Basic YWRtaW46YWRtaW4= Content-Type: application/vnd.yang.data+json
cache-control: no-cache
Postman-Token: d2d2ddb6-5dff-4917-972a-146db6dc175f

```

```

{
  "mop-automation": {
    "operation-type": "commit",
    "mop-type": "active",
    "transaction-type": "single-transaction",
    "mop-file-name": [
      {
        "file-name": "load3.txt",
        "order": 1,
        "target-devices-list": [
          {
            "target-device-name": "test3"
          }
        ]
      },
      {
        "file-name": "load4.txt",
        "order": 2,
        "target-devices-list": [
          {
            "target-device-name": "test3"
          }
        ]
      }
    ]
  }
}

```

REST API 要求：トランザクションタイプと *mop-type* をペアとして指定しない場合

次に、トランザクションタイプと *mop-type* をペアとして指定しない REST API 要求の例を示します。

```

{
  "mop-automation": {
    "operation-type": "commit",
    "mop-type": "pair",
    "generate-dry-run": "true",
    "save-1-1-config": "true",
    "mop-file-name": [
      {
        "file-name": "up_dayN.txt" ,
        "order": 1,
        "target-devices-list": [
          {
            "target-device-name": "up2-SI"
          }
        ]
      }
    ]
  }
}

```

[mop-type] が「rcm-upf」の MOP の自動化

```

    ]
  }
}
}

```

前述の非同期要求の呼び出しが成功すると、一意のタスク ID とタイムスタンプが返され、mop-automation 要求のステータスを確認するために使用されます。

```

{
  "mobility-mop:output": {
    "task-id": "1a1f62f0-487a-4c8c-bdeb-a760c26925cc",
    "time-stamp": "2021-07-19T11:10:51+0000",
    "time-zone": "Coordinated Universal Time"
  }
}

```

[mop-type] が「rcm-upf」の MOP の自動化

「rcm-upf」という mop-type は、単一またはすべての関連付けられた UPF に設定をプッシュするために使用されます。次の 2 つのシナリオが対象です。

1. 単一の UPF に MOP を適用する場合

UPF デバイスを指定する方法は次のとおりです。

- [target-device-name] に「upf-device」を指定します。
- upf-device に対応する [rcm-vip]、[group]、および [device-id] を指定します。

上記の 2 つの方法では、要求の [only-to-target-devices] を「true」に設定する必要があります。

ペイロードの例：

1. [target-device-name] に「upf-device」を指定します。

```

{
  "mop-automation": {
    "operation-type": "commit",
    "mop-type": "rcm-upf",
    "generate-dry-run": "true",
    "save-config-permanently": "true",
    "only-to-target-devices": "true",
    "mop-file-name": [
      {
        "file-name": "simpleStarOsChange.txt",
        "order": 1,
        "target-devices-list": [
          {
            "target-device-name": "up1-device"
          }
        ]
      }
    ]
  }
}

```

2. upf-device に対応する [rcm-vip]、[group]、および [device-id] を指定します。

```

{
  "mop-automation": {

```



```

        "operation-type": "commit",
        "mop-type": "rcm-upf",
        "generate-dry-run": "true",
        "save-config-permanently": "true",
        "only-to-target-devices": "true",
        "mop-file-name": [
            {
                "file-name": "simpleStarOsChange.txt",
                "order": 1,
                "rcm-vip" : "rcmvip01",
                "group" : "group03",
                "device-id" : "device-id1"
            }
        ]
    }
}

```

2. 関連付けられたすべての UPF デバイスに MOP を適用する場合

UPF デバイスを特定する方法は次のとおりです。

- [target-device-name] にサンプル upf-device を指定します。
- [rcm-vip] と [group] を指定します。

ペイロードの例：

1. [target-device-name] にサンプル upf-device を指定します。

```

{
  "mop-automation": {
    "operation-type": "commit",
    "mop-type": "rcm-upf",
    "generate-dry-run": "true",
    "save-config-permanently": "true",
    "only-to-target-devices": "false",
    "mop-file-name": [
      {
        "file-name": "simpleStarOsChange.txt",
        "order": 1,
        "rcm-vip" : "rcmvip01",
        "group" : "group03"
      }
    ]
  }
}

```

2. [rcm-vip] と [group] を指定します。

```

{
  "mop-automation": {
    "operation-type": "commit",
    "mop-type": "rcm-upf",
    "generate-dry-run": "true",
    "save-config-permanently": "true",
    "only-to-target-devices": "false",
    "mop-file-name": [
      {
        "file-name": "simpleStarOsChange.txt",
        "order": 1,
        "rcm-vip" : "rcmvip01",
        "group" : "group03"
      }
    ]
  }
}

```

```

    ]
  }
}

```

上記の2つの方法では、要求の [only-to-target-devices] を「false」に設定する必要があります。
[rcm-vip] と [group] を使用して UPF デバイスを取得する場合、次のリストに挙げる CDB のデータが使用されます。

- device-id-up-mapping
- up-rcm-mapping
- rcm-upf-mapping

MOP 自動化ステータス

NSO は、ネットワークオペレータによって渡されたタスク ID のデバイスステータスの結果を提供します。

API :

mop-automation-status

要求の詳細

パラメータ	書式	必須	説明
Task-id	文字列	必須	「MOP Automation」API から取得したタスクの一意の識別子。

応答の詳細

パラメータ	書式	必須	説明
task-id	文字列	key	Task ID
task-status	文字列		タスク ステータス
Start-date	文字列		開始日時
End-date	文字列		終了日時
Time-zone	文字列		タイムゾーン
Operation-type	文字列		Commit/Dry-run
Action-type	文字列		Save
devices-list	list		デバイス
Device-name	leafref	key	デバイス名
Start-date	文字列		開始日時

パラメータ	書式	必須	説明
End-date	文字列		終了日時
device-status	leafref		デバイスのステータス (Completed/In-Progress/Failed)
device-state	文字列		Active/Standby/Common /Pair/rcm-upf
files	list		ファイル
file-name	文字列	key	MOP ファイル名
順序	Uint8		MOP が実行された順序
dry-run-mop	文字列		リハーサル出力ファイル の場所
rollback-mop	文字列		ロールバック MOP の場 所
Commit-queue-status	文字列		コミットキューステータ ス
Commit-queue-id	文字列		コミットキュー ID
Error Code	文字列		エラーコード
Error Message	文字列		エラーメッセージ
Error Code	文字列		エラーコード
Error Message	文字列		エラーメッセージ

CLI

NCS CLI を使用した要求の例を以下に示します。

```
mobility-mop:action mop-automation-status task-id  
8d08e359-0bd2-48de-9a34-9192a986a486
```

REST API 要求

以下は、mop-automation のステータスを知るための REST API 要求の例です。

```
POST /restconf/data/mobility-mop:action/mop-automation-status
Host: localhost:8080
Authorization: Basic YWRtaW46YWRtaW4= Content-Type: application/vnd.yang.data+json
cache-control: no-cache

{
  "task-id": "22301071-9a6c-4f27-a0dc-b50c24124806"
}
```

以下に、上記の要求の呼び出し後に生成される応答フォーマットの例を示します。

```
{
  "mobility-mop:output": {
```

```

"task-id": "8d08e359-0bd2-48de-9a34-9192a986a486",
"task-status": "COMPLETED",
"start-date": "2021-09-06T09:08:54+0000",
"end-date": "2021-08-06T09:09:10+0000",
"time-zone": "Coordinated Universal Time",
"operation-type": "commit",
"action-type": "save",
"devices-list": [
  {
    "device-name": "up2-SI",
    "device-status": "COMPLETED",
    "start-date": "2021-08-06T09:08:55+0000",
    "end-date": "2021-08-06T09:08:59+0000",
    "device-state": "active",
    "files": [
      {
        "file-name": "up_dayN.txt",
        "order": "1",
        "dry-run-mop":
"/var/opt/ncs//8d08e359-0bd2-48de-9a34-9192a986a486/up2-SI/up_dayN_commit_2021-08-06T090854+0000.txt",

        "rollback-mop":
"/var/opt/ncs//8d08e359-0bd2-48de-9a34-9192a986a486/up2-SI/up_dayN_rollback_commit_2021-08-06T090854+0000.txt",

        "commit-queue-status": "completed",
        "commit-queue-id": "1628240937590"
      }
    ]
  }
]
}
}
}

```

MOP ロールバック

NSO は、ロールバック要求の入力で指定されたタスク ID、MOP ファイル、およびデバイスのロールバックプロセスを開始します。

これは、MOP で設定された構成ファイルやロールバックされた構成ファイルをロールバックする唯一のオプションです。

この API は、以前に適用された設定をロールバックします。最初に設定を適用した時に作成されたロールバックファイルが使用されます。ロールバックは、ファイルごと、すべてのファイルに対して、デバイスごと、またはすべてのデバイスに対して実行できます。



- (注) ロールバックが成功するかどうかは、関連する設定がプッシュされてからシステムに加えられた変更の内容によって大きく左右されます。その後の変更によっては、設定のロールバックが意味をなさなくなるほどシステムが変更されている可能性があります。

API :

mop-rollback

要求の詳細

パラメータ	書式	必須	説明
Task-id	文字列	必須	タスク固有の識別子
mop-file-name	リスト	任意	デバイスの MOP ファイル
file-name	文字列	キー	対応するロールバックファイルを識別するために使用される元のファイル名。
target-devices	リスト	任意	デバイスリスト。指定しない場合、元のトランザクションで設定がプッシュされたすべてのデバイスにおいてロールバックが実行されます。
target-device-name	Leafref	キー	
operation-type	文字列	必須	ドライラン/コミット
timeout	数値	オプション。デフォルト値は 600 秒	デバイスがロックされるまで待機する最大秒数。

応答の詳細

パラメータ	書式	必須	説明
Task-id	文字列	必須	タスク固有の識別子 task-id は、別の API を介してロールバックのステータスを確認するために使用します。
Time stamp	文字列	必須	タイムスタンプ
Error Code	文字列	エラーコード	
Error Message	文字列	エラーメッセージ	

CLI

NCS CLI を使用した要求の例を以下に示します。

```
mobility-mop:action mop-rollback task-id
8d08e359-0bd2-48de-9a34-9192a986a486 generate-dry-run true operation-type
```

REST API 要求 : 操作タイプ「commit」の場合

```
commit save-config-permanently true mop-file-name { file-name
up_dayN.txt target-devices-list { target-device-name up2-SI } }
```

REST API 要求 : 操作タイプ「commit」の場合

次に、操作タイプが「commit」の REST API 要求の例を示します。

```
POST /restconf/data/mobility-mop:action/mop-rollback
Host: localhost:8080
Authorization: Basic YWRtaW46YWRtaW4= Content-Type: application/vnd.yang.data+json
cache-control: no-cache
Postman-Token: 1b687031-dc32-41 14-a69f-5984130c36a5
{
  "mop-rollback": {
    "task-id": "0891655c-642b-4ba3-9392-6f05d4e77a63",
    "operation-type": "commit",
    "generate-dry-run": "true",
    "save-config-permanently": "true",
    "mop-file-name": [
      {
        "file-name": "up_dayN.txt",
        "target-devices-list": [
          {
            "target-device-name": "up2-SI"
          }
        ]
      }
    ]
  }
}
```

前述の要求の呼び出しが成功すると、一意のタスク ID とタイムスタンプが返され、mop-rollback 要求のステータスを確認するために使用されます。

```
{
  "mobility-mop:output": {
    "task-id": "8d08e359-0bd2-48de-9a34-9192a986a486",
    "time-stamp": "2021-08-06T09:08:44+0000",
    "time-zone": "Coordinated Universal Time"
  }
}
```

REST API 要求 : 操作タイプ「dry-run」の場合

次に、操作タイプが「dry-run」の REST API 要求の例を示します。

```
POST /restconf/data/mobility-mop:action/mop-rollback
Host: localhost:8080
Authorization: Basic YWRtaW46YWRtaW4= Content-Type:
application/vnd.yang.data+json cache-control: no-cache
Postman-Token: 1b687031-dc32-41 14-a69f-5984130c36a5
{
  "mop-rollback": {
    "task-id": "0891655c-642b-4ba3-9392-6f05d4e77a63",
    "operation-type": "dry-run",
    "generate-dry-run": "true",
    "save-config-permanently": "true",
    "mop-file-name": [
      {
        "file-name": "up_dayN.txt",
        "target-devices-list": [
          {
            "target-device-name": "up2-SI"
          }
        ]
      }
    ]
  }
}
```

```

    }
  ]
}

```

前述の要求の呼び出しが成功すると、一意のタスク ID とタイムスタンプが返され、mop-rollback 要求のステータスを確認するために使用されます。

```

{
  "mobility-mop:output": {
    "task-id": "1a1f62f0-487a-4c8c-bdeb-a760c26925cc",
    "time-stamp": "2021-07-19T11:10:51+0000",
    "time-zone": "Coordinated Universal Time"
  }
}

```

MOP ロールバックのステータス

NSO は、ネットワークオペレータによって渡されたタスク ID のデバイスステータスの結果を提供します。API を使用して、進行中または完了したロールバック操作のステータスをクエリします。

API :

mop-rollback-status

要求の詳細

パラメータ	書式	必須	説明
Task-id	文字列	必須	ロールバック操作におけるタスクの一意の識別子。

応答の詳細

パラメータ	書式	必須	説明
task-id	文字列	key	タスクの一意の識別子
task-status	文字列		タスク ステータス
Start-date	文字列		開始日時
End-date	文字列		終了日時
Time-zone	文字列		タイムゾーン

パラメータ	書式	必須	説明
Operation-type	文字列		Commit/Dry-run
Action-type	文字列		ロールバック (Rollback)
devices-list	list		デバイス
Device-name	leafref	key	デバイス名
Start-date	文字列		開始日時
End-date	文字列		終了日時
device-status	leafref		デバイスのステータス (Completed/In-Progress/Failed)
device-state	文字列		Active/Standby/Common/Pair
files	list		ファイル
file-name	文字列	key	MOP ファイル名
順序	Uint8		MOP が実行された順序
dry-run-mop	文字列		リハーサル出力ファイルの場所
rollback-mop	文字列		ロールバック MOP の場所
Commit-queue-status	文字列		コミットキューステータス
Commit-queue-id	文字列		コミットキュー ID
Error Code	文字列		エラーコード
Error Message	文字列		エラーメッセージ
Error Code	文字列		エラーコード
Error Message	文字列		エラーメッセージ

CLI

NCS CLI を使用した要求の例を以下に示します。

```
mobility-mop:action mop-rollback-status task-id  
fd0fb9ae-8685-420e-9490-0c6858d14148
```

REST API 要求

以下は、mop-rollback のステータスを知るための REST API 要求の例です。

```
POST /restconf/data/mobility-mop:action/mop-rollback-status  
Host: localhost:8080  
Authorization: Basic YWRtaW46YWRtaW4= Content-Type: application/vnd.yang.data+json  
cache-control: no-cache  
Postman-Token: Oe2c4bd3-2dc6-4ddb-aea9-1 1 Occf622da7  
"mop-rollback -status": {  
"task-id": "5733d661-9242-4867-8320-a314da592c93"  
}
```

```
Below is response format generated, post invocation of the above request -  
task-id fd0fb9ae-8685-420e-9490-0c6858d14148  
task-status COMPLETED  
start-date 2021-08-06T09:24:14+0000  
end-date 2021-08-06T09:24:30+0000  
time-zone Coordinated Universal Time  
operation-type commit  
action-type rollback  
devices-list {  
  device-name up2-SI  
  device-status COMPLETED  
  start-date 2021-08-06T09:24:14+0000  
  end-date 2021-08-06T09:24:19+0000  
  device-state active  
  files {  
    file-name up_dayN_rollback_commit_2021-08-06T090854+0000.txt  
    order 1  
    dry-run-mop /var/opt/ncs//fd0fb9ae-8685-420e-9490-0c6858d14148/up2-SI  
/up_dayN_2021-08-06T090854+0000_rollback_commit_2021-08-06T092414+0000.txt  
    rollback-mop /var/opt/ncs//fd0fb9ae-8685-420e-9490-0c6858d14148/up2-SI  
/up_dayN_2021-08-06T090854+0000_commit_2021-08-06T092414+0000.txt  
    commit-queue-status completed  
    commit-queue-id 1628241856973  
  }  
}
```

ドライランおよびリバースドライラン MOP の確認

ドライラン MOP およびリバースドライラン MOP を確認するには、ドライラン MOP およびリバースドライラン MOP の静的データの設定時に提供されたそれぞれのファイルの場所に移動します。

MOP 実行用の構成ファイルへの変数の追加

MOP 自動化パッケージでは、MOP での変数の指定がサポートされているため、MOP が適用されるデバイスに基づいてランタイムに変数が入力されます。たとえば、次の MOP が指定され、デバイス TXPCF003 で実行されたとします。

```
config context local administrator $Host_name password Nsotest123$ exit  
end
```

ホスト名は、次のアクションコールを使用して設定できます。

```
config-metadata config-metadata-request config-metadata { device-name
up2-SI device-type vpc attributes { attribute-name Host_name
attribute-value TXPCF003} scheme 1:1 }
```

生成される dry-run MOP は次のとおりです。

```
config context local administrator TXPCF003 password Nsotest123$ exit
end
```

N:M 冗長性での UP 設定のプッシュとリカバリ

N:M シナリオでは、RCM が各 UP のロール（アクティブとスタンバイ）を決定します。任意の M 個のスタンバイインスタンスが任意の N 個のアクティブインスタンスを引き継げる必要があるため、プッシュされる設定は異なり、動的に変化します。これは、すべての設定を UP に永続的に保存できるわけではないことも意味します。

RCM は、UP 起動や UP スイッチオーバーなどの関連イベントに対して NETCONF 通知を発行します。NSO は NETCONF 通知をリッスンし、必要に応じて必要な設定を適用します。

UP の設定は以下の論理コンポーネントで構成されます。

- Day-0 設定：これは主に、UP の管理インターフェイスを到達可能にするための基本設定です。この設定は、VNFM による UP 展開時にプッシュされ、再起動後も変わらないと想定されています。



(注) 必要な

rcm-configmgr CLI

コマンドは、NSO ベースの設定のプッシュを機能させるために、Day-0 設定の一部として UP で設定する必要があります。このコマンドは、RCM がソリューションで使用されているかどうかに関係なく必要です。このコマンドを設定しない場合、ECS 設定のプッシュは非表示になります。

- Day-0.5 設定：UP が RCM に接続できるようにする設定です。この設定は、UP 展開の直後（NSO が VM を展開している場合）に自動的に、または手動で MOP の設定のプッシュを実行して、Day-0 設定とともにプッシュしたり、NSO とは別にプッシュしたりできます。この設定は、再起動後も永続的に保存されると想定されています。
- 共通設定：これは、アクティブかスタンバイかに関係なく、すべての UP に共通の設定で、ECS と APN の設定のみ該当します。この設定は、NSO に事前入力する必要があります。NSO は、RCM から通知を受信すると、この設定をプッシュします。この設定は、起動設定の一部として永続的に保存されるのではなく、各 UP でファイルとしてローカルに保存され、再起動のたびに NSO によって自動的に再適用されます。

- **ホスト固有の設定**：これは、アクティブな各 UP に固有の設定で、主に各種サービスの IP アドレスです。各アクティブ UP には、そのアクティブインスタンスに固有の設定がプッシュされます。各スタンバイインスタンスには、すべてのアクティブ UP に関する結合されたホスト固有の設定がプッシュされます。この設定は、NSO に事前入力されている必要があります。NSO は、RCM からの通知に基づいて、必要に応じてこの設定を各 UP にプッシュします。
- **ホスト固有の設定：RCM コピー**：これは各 UP のホスト固有の設定ですが、RCM 互換形式でフォーマットされています。この設定は RCM にプッシュする必要があります。RCM は設定にはほとんど関係しませんが、UP スイッチオーバー時における設定の否定の実行には関係します。設定の否定とは、特定のアクティブ UP を引き継ぐスタンバイ UP から、他のすべてのアクティブ UP の設定を削除することを意味します。たとえば、3:1 のシナリオで、Active3 UP がダウンしたとします。スタンバイには、Active1、Active2、および Active3 のホスト固有の設定があり、スタンバイが Active3 を引き継ぐため、RCM はスイッチオーバーの一環としてそのスタンバイから Active1 と Active2 の設定を無効にします。

NSO での NETCONF 通知サブスクリプション

RCM から送信されたすべての通知は、NSO によってキャプチャされます。NSO は通知をフィルタリングし、RCM 関連の通知を処理します。

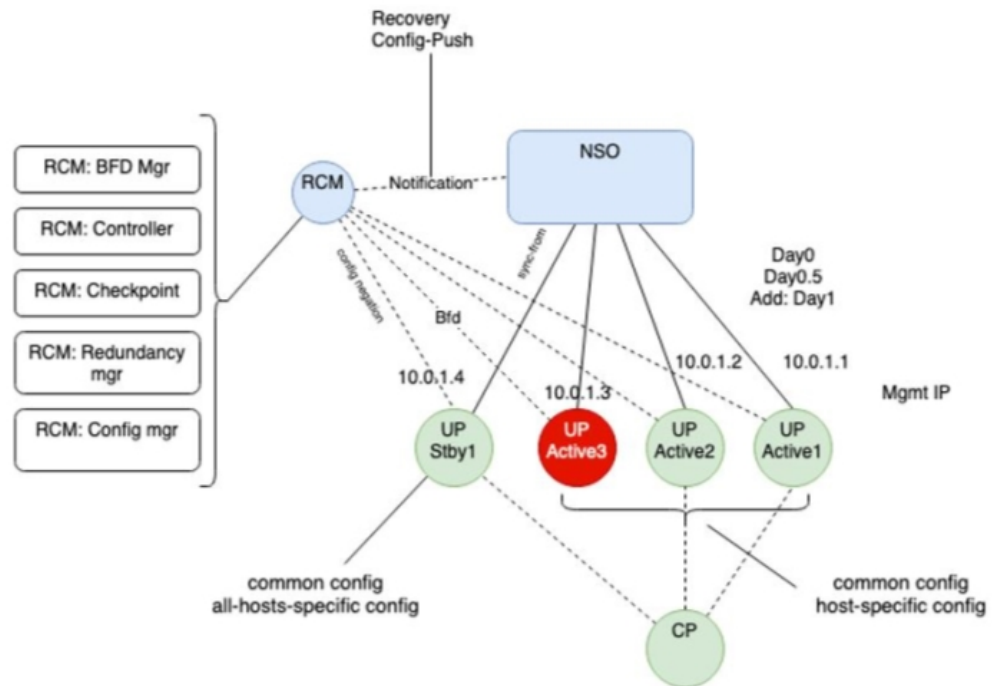
次の表では、処理可能な RCM UP 通知のタイプと、それらの通知が NSO によってどのように処理されるかについて説明します。

	リカバリ	設定のプッシュ	
	UP リカバリ	UP リブート	新規 UP
アクティブ UP	NSO でデバイスメタデータを更新します。	ホスト固有の設定をプッシュします。 共通設定がデバイスにない場合は、共通設定ファイルを再プッシュします。 NSO でデバイスメタデータを更新します。	
スタンバイ UP	該当なし	ホスト固有の設定をすべてプッシュします。 共通設定がデバイスにない場合は、共通設定ファイルを再プッシュします。 NSO でデバイスメタデータを更新します。	

RCM UP リカバリ通知の処理

UP 障害が発生した場合、RCM は BFD マネージャを介して障害を検出し、NSO が受信した通知をプッシュします。RCM は UP のスイッチオーバーを処理して、選択されたスタンバイ UP をアクティブ UP にします。スタンバイ UP はすでに必要なすべての設定を備えているため、スタンバイ UP がスイッチオーバーするためのこの設定管理プロセスにそれほど時間はかかりません。

この図は、RCM による UP 通知の処理を示しています。

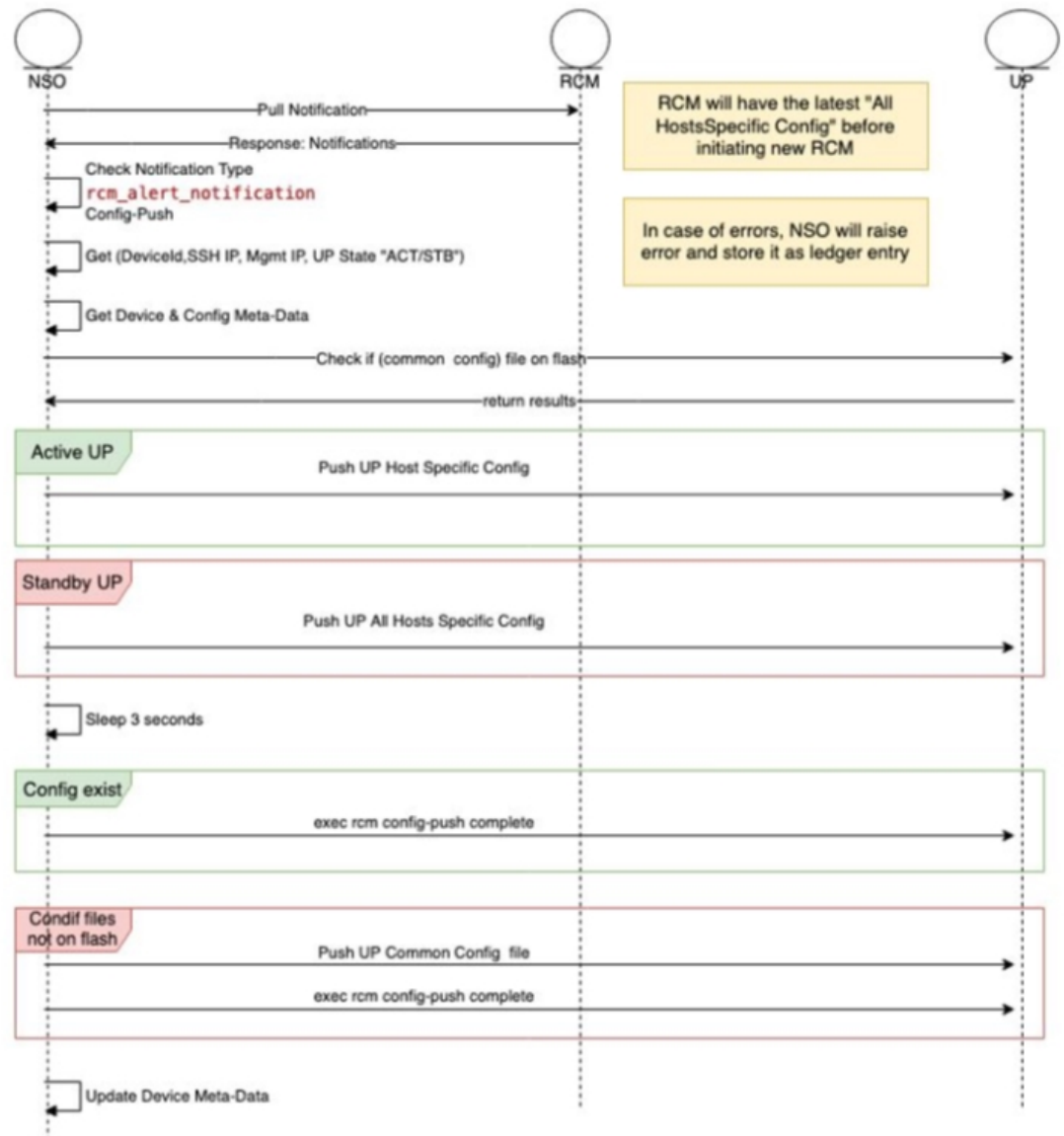


461481

RCM UP 設定プッシュ通知

RCM は、起動中の新しい UP、またはリカバリのためにリポート中の既存の UP があると、設定のプッシュ通知を生成します。

次の図は、RCM による UP 設定のプッシュ通知の処理を示しています。



461482

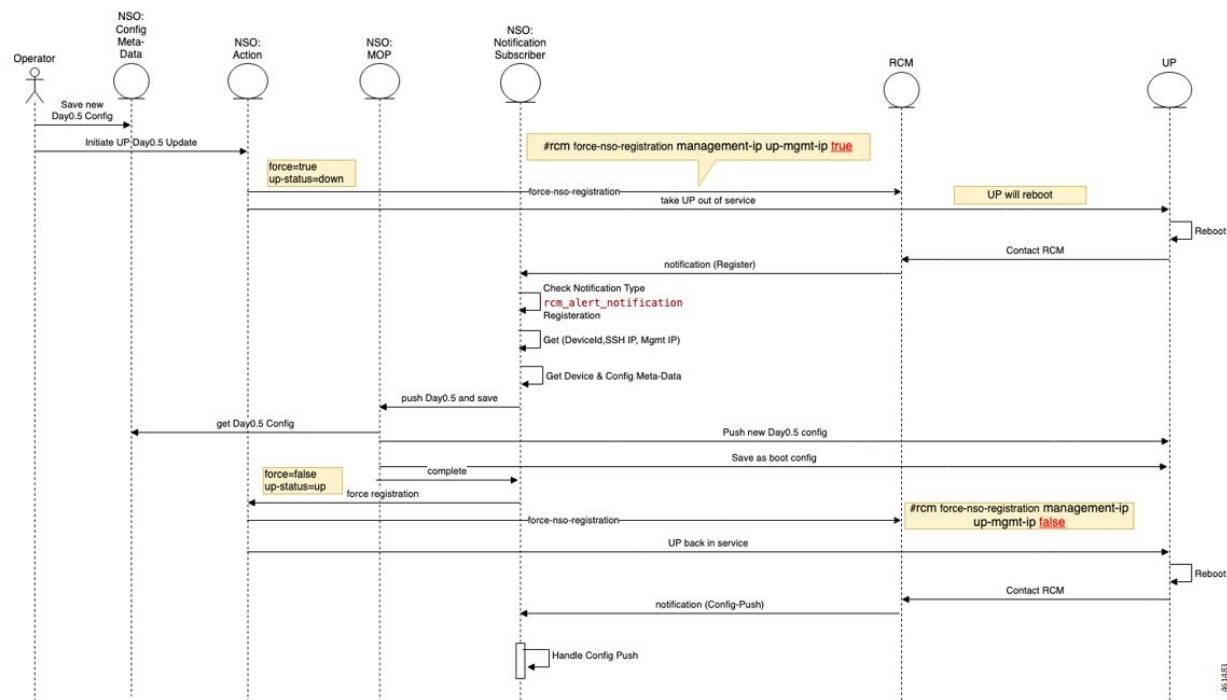
NSO は、UP 設定のプッシュ通知があるたびに、次の手順を実行します。

1. アラートステータスが [config-push] になるまで待機します。
2. デバイスの詳細（デバイス名、SSH IP、管理 IP、UP ステータス）を取得します。
3. NSO からデバイスメタデータを取得します。
4. UP フラッシュに共通ファイルが存在するかを確認します。
5. UP の状態がアクティブの場合、NSO はモビリティ MOP を使用して UP ホストに固有の構成ファイルをプッシュします。
6. UP 状態がスタンバイの場合、NSO はモビリティ MOP を使用して UP のすべてのホストに固有の構成ファイルをプッシュします。

7. 3 秒間スリープします。
8. 共通の構成ファイルが存在する場合、NSO は UP で `live-status` コマンドを実行してそれを適用します。
9. 共通の構成ファイルが存在しない場合、NSO はモビリティ MOP を使用して共通の構成ファイルを UP にプッシュし、UP で `live-status` コマンドを実行してこれを適用します。
10. NSO のデバイスメタデータを更新します。

UP Day-0.5 の更新

UP の Day-0.5 設定を変更するには、UP をリポートする必要があります。このため、変更中は UP はダウン状態となります。RCM は、特定の UP がリポートされると必ず強制的に通知を送信するコマンドを使用して、このユースケースをサポートします。この通知がトリガーとなって、NSO は新しい Day-0.5 設定をプッシュします。



1. NSO config-metadata で UP の Day-0.5 設定を更新する必要があります。
2. UP デバイス名と管理 IP アドレスを指定して、UP の Day-0.5 変更アクションを開始する必要があります。
3. NSO アクションは、UP のリポート時に NSO 登録を強制する RCM コマンドを実行します。

```
rcm force-nso-registration management-ip MGMT-IP true
```

4. NSO アクションによって UP はダウン状態になります。このとき、次の 2 つのシナリオのいずれかとなります。
 1. スタンバイ UP : NSO アクションは UP でリロードコマンドを実行します。
 2. アクティブ UP : NSO アクションは、UP の `management-ip` とスタンバイ UP の `management-ip` を指定して、RCM で計画的スイッチオーバーコマンドを実行します。スタンバイ UP の `management-ip` は、UP グループの NSO デバイスメタデータから取得できます。
5. UP のリポート後 (アクティブ UP またはスタンバイ UP) 、RCM は [Registration] タイプの通知を生成し、これを NSO が受信します。
6. NSO RCM サブスクライバは、登録通知を受信すると、新しい MOP プロセスを開始して、対象となる UP の Day-0.5 設定をプッシュします。
7. NSO RCM サブスクライバは、継続して MOP ステータスを収集します。MOP が完了すると、NSO は UP で次のコマンドを実行します。


```
rcm-config-push-complete
```
8. NSO RCM サブスクライバは、次のコマンドを実行して NSO アクションを呼び出し、`force-notification` コマンドを無効にします。


```
rcm force-nso-registration management-ip MGMT-IP false
```
9. NSO アクションは、`reload` コマンドを実行して UP を起動します。
10. UP がリポートすると、RCM は Config-Push 通知を生成します。この通知は、通常どおり NSO によって処理されます。

設定プッシュの前提条件

設定のプッシュの前提条件を次に示します。

1. UP の場合 :


```
require rcm-configmgr
```

CLI コマンドは、Day-0 設定の一部として事前に設定する必要があります。このコマンドは、N:M、1:1、およびスタンドアロンのシナリオに必要です。設定することで、ECS 設定の適切な動作が有効になります。
2. UP の場合、該当する場合は、CP から、または UP で PFD プッシュを無効にする必要があります。すべての UP 設定は、NSO から直接プッシュされます。これは、N:M、1:1、およびスタンドアロンのシナリオに当てはまります。
3. RCM Ops Center コンフィギュレーション モードの CLI は、次のように設定する必要があります。


```
k8 smf profile rcm-config-ep config-mode NSO  
k8 smf profile rcm-config-ep switchover deployment false
```

4. オーバーライドする必要があるデフォルトの StarOS NED 設定がいくつかあります。
 1. 設定の変更はすべて、デバイスの起動設定 (system.cfg) に自動的に保存されます。N:M 冗長性が使用されている場合、UP の変更の設定はそのロールによって異なるため、自動保存は望ましくないので、次の CLI 設定コマンドを使用してグローバルに無効化する必要があります。

```
devices global-settings ned-settings cisco-staros  
write-memory-setting disabled
```

1:1 またはスタンドアロン展開のみを使用する場合は、この設定をそのまま保持できます。

N:M と 1:1/スタンドアロンを組み合わせる場合は、前述のように設定の保存を無効にしてから、1:1/スタンドアロンに対する手動設定のプッシュで「save-config-permanently」パラメータを融合します。自動設定のプッシュの場合、モビリティ機能パックが必要に応じて自動的に設定を保存します。

2. NED は警告をエラーとして扱い、設定のプッシュを失敗させます。多くの場合、警告は無視できるため、設定のプッシュを続行する必要があります。警告の正規表現を使用して、選択した警告を無視するように NED を設定できます。次に例を示します。

```
devices global-settings ned-settings cisco-staros behaviour  
config-warning-ignore .*not recommended to change the dictionary.*
```

その他の一般的な例は次のとおりです。

```
ned-settings cisco-staros behaviour config-warning-ignore .*About  
to overwrite boot.*
```

```
ned-settings cisco-staros behaviour config-warning-ignore  
.*Standby card not ready.*
```

最後の例は、SI への設定のプッシュに必要です。

5. RCM は SSH IP の概念をサポートしています。SSH IP は、機能を提供している VM に関係なく、特定のアクティブ UP を明確に追跡する方法です。NSO ベースのソリューションでは、SSH IP は使用されませんが、ダミーの SSH IP を設定する必要があります。これは、管理インターフェイスでセカンダリ IP アドレスとして設定されます。この設定におけるエラーを回避するには、Day-0.5 の一部として次の設定を推奨します。

```
configure
```

```
redundancy-configuration-module rcm rcm
```

```
nso-ssh-ip context local interface-name local1 mask 255.255.255.224
```

6. NSO から VNF への読み取りおよび書き込み操作にかかる時間は、遅延に応じ変化する場合があります。遅延時間は、次に示されているように調整可能です。調整は、本当に遅延が原因である読み取りまたは書き込みエラーの問題が発生した場合にのみ行います。通常は、デフォルト設定で十分です。

```
devices global-settings read-timeout 180
```

```
devices global-settings write-timeout 180
```


制限事項と制約事項

このリリースでは、NSO ベースの設定管理機能に次の制限事項があります。

- 実稼働 NSO インスタンスは、一般的な Linux フレーバー（RedHat、Cisco Linux、Ubuntu、CentOS など）でのみ実行できます。
- Day-1 設定のみが RCM 通知で UP にプッシュされます。他の設定はプッシュされません。
- Day-N 設定の変更を後でプッシュする場合は、その変更を Day-1 設定ファイルとマージして、今後 RCM 通知で自動的にプッシュされるようにする必要があります。
- 事前に入力された設定ファイルに変更がある場合、それらは自動的にプッシュされません。すべてのターゲットデバイスに手動でプッシュする必要があります。次の自動プッシュの設定変更のみが考慮されます。

N:M の UP の場合、事前に入力された設定ファイルは、それらを使用する VNF が少なくとも 1 つある場合、NSO（HA ペアとして実行されている場合は両方のインスタンス）で保持する必要があります。

- 設定コマンドはサポートされていません。設定ファイル内の show CLI コマンドはサポートされていません。
- NSO から管理する設定は、対応する NED（CP、UP の場合は StarOS NED、RCM の場合は RCM NED）によって認識される必要があります。現在、すべての StarOS 設定コマンドがサポートされているわけではありません。CUPS フィールド展開で最もよく使用される設定のみがサポートされています。欠落しているコマンドをサポートするには、新しい NED が必要です。
- ネイティブ StarOS CLI のサポートは 100% ではありません。サポートされている CLI コマンドの大部分はネイティブの StarOS CLI 形式で使用できますが、NSO が対応する StarOS CLI のバリエーションのみを受け入れる場合もあります。このような CLI については、[付録 A：互換性のない StarOS ネイティブ コマンド シNTAXS（51 ページ）](#) を参照してください。設定 MOP の「ドライラン」機能を使用すると、設定のプッシュを実行する前に、互換性がないかサポートされていない CLI によるエラーを検出できます。
- 操作中に要求を処理する NSO がダウンした場合、設定のプッシュが失敗する可能性があります。これは、手動と自動の両方の設定プッシュに当てはまります。また、NSO HA 展開とスタンドアロン NSO 展開の両方に当てはまります。障害の正確な種類に応じて、オペレータの介入が必要になる場合があります。
- N:M の冗長性シナリオの Day-0.5 設定変更ワークフローは、このリリースでは完全には機能しません。このリリースでの回避策は次のとおりです。
 1. 冗長グループから UP を削除します（アクティブだった場合は、最初にスタンバイにします）。
 2. UP は Day-0 および現在の Day-0.5 設定で起動します。
 3. UP で Day-0.5 設定を変更し、ブート設定として永続的に保存します。

4. UP を冗長グループに追加します。UP は RCM に登録され、残りの設定は NSO によって自動的にプッシュされます。
- このリリースでは、rcm-upf の MOP タイプを使用してアクティブな課金設定の変更をプッシュする場合、N:M 冗長シナリオでの Day-N 設定のプッシュには、事前の追加手順が必要です。MOP を呼び出す前に、その冗長グループ内のすべての UP でファイル `/flash/mobility_production.cfg` を手動で削除する必要があります。
 - 標準の StarOS CLI NED は、特定の機密設定データをクリアテキストとしてローカルに保存します。NSO では、NACM ルールを使用して機密設定データへのアクセスを制限できます。それでも懸念が残る場合は、シスコのアカウント担当者に連絡して、この機密データをローカルで暗号化する NED バージョンを入手してください。この暗号化は NED と NSO に固有であることに注意してください。StarOS は独自に機密データを暗号化するため、この 2 つの暗号化は個別のものです。NSO がローカルで機密データを暗号化する場合、StarOS デバイスに送信する前に復号します (SSH 経由で送信されるため、転送中に暗号化されますが、StarOS CLI ではクリアテキストとして受信されます)。
 - N:M 冗長性シナリオでは、RCM は SSH IP の概念をサポートします。NSO ベースのソリューションでは、SSH IP を使用しません。ただし、FCS (3.0.0 および 21.25) の場合、ソリューションの SSH IP を指定する必要があります。ルーティング不可能なプライベートアドレスを含む任意のアドレスで十分です。このアドレスは、UP の管理インターフェイスでセカンダリアドレスとして設定されます。また、UP の SSH IP 設定要件に関する注意事項については、[設定プッシュの前提条件 \(47 ページ\)](#) の項を参照してください。

トラブルシューティング

障害対応では、次のオプションを使用できます。

1. タスク ID のダッシュボード出力を使用して、詳細を確認できます。次に例を示します。

```
mobility-mop:action mop-automation-status task-id 12d5fc33-2f9e-44e3-81e3-14043d4ee39d
```
2. 障害が発生した場合、一部のアラームが発生する可能性があります。これらを確認するには、

```
show alarms
```

 CLI コマンドを使用します。
3. 詳細なログは、`/var/log/ncs/ncs-java-vm.log` ファイルで確認できますが、これは開発者のデバッグ向けです。

付録 A : 互換性のない StarOS ネイティブコマンドシンタックス

ここでは、NED ではすでにサポートされているものの（下に太字で表示するもの）、StarOS ネイティブコマンドシンタックスとの互換性がないコマンドを示します。

モード	コマンド	説明
context xxx/ggsn-service yyy	ip qos-dscp qci 9 af31 gtpc af41	「ip qos-dscp qci 9 af31」と「ip qos-dscp gtpc af41」を個別に受け付けます。デバイスにプッシュするときは、両方合わせてプッシュします。
context xxx/apn yyy	apn fnetcoriolis default max-contexts default cc-roaming default ipv6 address alloc-method exit	「no max-contexts」、「no cc-roaming」、および「no ipv6 address alloc-method」を受け付けて、デバイスに対して「default xxx」を生成します。
context xxx/crypto map yyy (ikev2-pv4)/payload zzz match ipv4	crypto map ipsec_tunnel ikev2-ipv4 keepalive interval 4 timeout 1 num-retry 4 payload mypayload match ipv4 default lifetime exit	「no lifetime」を受け付けて、デバイスに対して「default lifetime」を生成します。
active-charging service xxx/credit-control group yyy/	credit-control group cc-m2mpt quota validity-time 600 diameter reauth-blacklisted-content content-based-rar default diameter send-ccru on-rar always default diameter mscc-per-ccr-update exit	「no xxx」を受け付けて、デバイスに対して「default xxx」を生成します。

モード	コマンド	説明
context xxx/ikev2-ikesa transform-set yyy	ikev2-ikesa transform-set transformset_li default encryption default group default hmac default lifetime default prf exit	「no xxx」を受け付けて、デバイスに対して「default xxx」を生成します。
global config mode	end および #exit	rload はこれらのコマンドを受け付けません。
global config mode	snmp trap enable	相当するコマンド : no snmp trap suppress
active-charging service xxx/ruledef yyy または「!」を含むすべてのコマンド	active-charging service ecs ruledef rd-webredirect-apn-sl2sfr bearer 3gpp imsi !range imsi-pool imsi-NOREDIRECT exit	二重引用符で囲まれているか、バックスラッシュでエスケープされていない限り、rload は「!」記号を認識しません。
active-charging service xxx/ruledef yyy %NN を含むすべての URL	ruledef rd1 www url = http://itunes.apple.com/WebObjects/MZStore.woa/wa/viewSoftware%3f id=362713555& exit	%3f は「\」でエスケープする必要があります。

モード	コマンド	説明
context xxx/gtpp group yyy	gtpp group sgw no gtpp attribute node-id no gtpp attribute losdv no gtpp trigger time-limit no gtpp trigger tariff-time-change no gtpp trigger serving-node-change-limit no gtpp trigger inter-plmn-sgsn-change no gtpp trigger qos-change no gtpp trigger rat-change no gtpp trigger ms-timezone-change no gtpp trigger uli-change	これらのコマンドは、NED ではデフォルト設定として処理されますが、StarOS では処理されません。
context xxx/ims-auth-service yyy	p-cscf table 1 row-precedence 1 ipv4-address 209.165.200.227 secondary ipv4-address 209.165.200.229	ハイライトしたキーワードは、セカンダリ IP アドレスに対してのみ ipv4-address に変更する必要があります。
context xxx/ims-auth-service yyy	default signaling-flag	代わりに「no signalling-flag」を使用します。
context xxx/ims-auth-service yyy	default traffic-policy general-pdp-context no-matching-gates direction downlink	代わりに、「no traffic-policy general-pdp-context no-matching-gates direction downlink」を使用します。
context xxx/ims-auth-service yyy	p-cscf table 1 row-precedence 1 ipv6-address 2001:860:ffff:feb6::a secondary ipv6-address 2001:860:ffff:feb4::9	ハイライトしたキーワードは、セカンダリ IP アドレスに対してのみ ipv6-address に変更する必要があります。
context xxx/hexdump-module	default file rotation volume time-stamp monitor- subscriber-file-name	相当するコマンド : no file rotation volume no file time-samp no file monitor-subscriber-file-name
context xxx/hexdump-module	default file rotation volume	相当するコマンド : no file rotation volume

モード	コマンド	説明
context xxx/hexdump-module	default file time-stamp	相当するコマンド : no file time-stamp
context xxx/hexdump-module	default subscriber-file-name	相当するコマンド : no subscriber-file-name
context xxx/hexdump-module	default hexdump transfer-mode	相当するコマンド : no hexdump transfer-mode
context xxx/hexdump-module	default hexdump push-interval	相当するコマンド : no hexdump push-interval
context xxx/edr-module active-charging-service	default cdr transfer-mode push via transfer-mode	相当するコマンド : no cdr transfer-mode push
context xxx/session-event-module	default event transfer-mode push via transfer-mode	相当するコマンド : no event transfer-mode push
context xxx/router bgp NNN	context gy router bgp 64650 neighbor 209.165.200.226 remote-as 15557 no neighbor 209.165.200.226 capability graceful-restart	StarOS にはデフォルト設定として、グレースフルリスタート機能があります。NED は、これとは逆の方法でこれらのコマンドを処理します。 (注) この CLI は、StarOS CLI NED バージョン 5.50 以降でサポートされています。

付録 B : RCM を使用した N:M 展開の設定例

ホスト固有の設定 : UP

次に、2つのアクティブ UP のホスト固有の設定例を示します。設定はそれぞれの UP にプッシュされます。

最初のアクティブ UP

```
config
context EPC2
interface loop1_up1 loopback
```

```
ip address 209.165.200.225 255.255.255.224

interface loop2_up1 loopback
ip address 209.165.200.226 255.255.255.224

interface loop3_up1 loopback
ip address 209.165.200.227 255.255.255.224

interface loop4_up1 loopback
ip address 209.165.200.228 255.255.255.224

interface loop5_up1 loopback
ip address 209.165.200.229 255.255.255.224

exit
exit

context EPC2
sx-service sx_up1
instance-type userplane
bind ipv4-address 209.165.200.229
exit

exit
exit

context EPC2
gtpu-service pgw-gtpu_up1
bind ipv4-address 209.165.200.226
exit
gtpu-service saegw-sxu_up1
bind ipv4-address 209.165.200.227
exit
gtpu-service sgw-engress-gtpu_up1
bind ipv4-address 209.165.200.228
exit
gtpu-service sgw-ingress-gtpu_up1
bind ipv4-address 209.165.200.225

exit
exit
config
context EPC2
user-plane-service up_up1
associate gtpu-service pgw-gtpu_up1 pgw-ingress
associate gtpu-service sgw-ingress-gtpu_up1 sgw-ingress
associate gtpu-service sgw-engress-gtpu_up1 sgw-egress
associate gtpu-service saegw-sxu_up1 cp-tunnel
associate sx-service sx_up1
associate fast-path service
associate control-plane-group g1
exit

exit
exit
```

2 番目のアクティブ UP

```
config
context EPC2
interface loop1_up2 loopback
ip address 209.165.200.230 255.255.255.224

interface loop2_up2 loopback
```

```

ip address 209.165.200.231 255.255.255.224

interface loop3_up2 loopback
ip address 209.165.200.232 255.255.255.224

interface loop4_up2 loopback
ip address 209.165.200.233 255.255.255.224

interface loop5_up2 loopback
ip address 209.165.200.234 255.255.255.224

exit
exit

context EPC2
sx-service sx_up2
instance-type userplane
bind ipv4-address 209.165.200.234
exit

exit
exit

context EPC2
gtpu-service pgw-gtpu_up2
bind ipv4-address 209.165.200.231
exit
gtpu-service saegw-sxu_up2
bind ipv4-address 209.165.200.232
exit
gtpu-service sgw-engress-gtpu_up2
bind ipv4-address 209.165.200.233
exit
gtpu-service sgw-ingress-gtpu_up2
bind ipv4-address 209.165.200.230

exit
exit

context EPC2
user-plane-service up_up2
associate gtpu-service pgw-gtpu_up2 pgw-ingress
associate gtpu-service sgw-ingress-gtpu_up2 sgw-ingress
associate gtpu-service sgw-engress-gtpu_up2 sgw-egress
associate gtpu-service saegw-sxu_up2 cp-tunnel
associate sx-service sx_up2
associate fast-path service
associate control-plane-group g1
exit

exit
exit

```

ホスト固有の設定 : RCM

RCM のホスト固有の設定例を以下に示します。この設定は RCM にプッシュされます。

最初のアクティブ RCM

```

config
control-plane-group g1

```



```
redundancy-group 1
  host Active1
  peer-node-id ipv4-address 209.165.200.240
  exit
exit
context EPC2
interface-loopback loop1_up1
  redundancy-group 1
  host Active1
  ipv4-address 209.165.200.224/27
  exit
exit
interface-loopback loop2_up1
  redundancy-group 1
  host Active1
  ipv4-address 209.165.201.0/27
  exit
exit
interface-loopback loop3_up1
  redundancy-group 1
  host Active1
  ipv4-address 209.165.202.128/27
  exit
exit
interface-loopback loop4_up1
  redundancy-group 1
  host Active1
  ipv4-address 192.0.2.0/24
  exit
exit
interface-loopback loop5_up1
  redundancy-group 1
  host Active1
  ipv4-address 198.51.100.0/24
  exit
exit
user-plane-service up_up1
  redundancy-group 1
  host Active1
  associate control-plane-group g1
  associate fast-path service
  associate sx-service sx_up1
  associate gtpu-service pgw-gtpu_up1 pgw-ingress
  associate gtpu-service saegw-sxu_up1 cp-tunnel
  associate gtpu-service sgw-engress-gtpu_up1 sgw-egress
  associate gtpu-service sgw-ingress-gtpu_up1 sgw-ingress
  exit
exit
gtpu-service pgw-gtpu_up1
  redundancy-group 1
  host Active1
  bind ipv4-address 209.165.201.0
  exit
exit
gtpu-service saegw-sxu_up1
  redundancy-group 1
```

```

    host Active1
      bind ipv4-address 209.165.202.128
    exit
  exit
exit
gtpu-service sgw-engress-gtpu_up1
  redundancy-group 1
  host Active1
    bind ipv4-address 192.0.2.0
  exit
exit
exit
gtpu-service sgw-ingress-gtpu_up1
  redundancy-group 1
  host Active1
    bind ipv4-address 198.51.100.123
  exit
exit
exit
sx-service sx_up1
  redundancy-group 1
  host Active1
    bind ipv4-address 198.51.100.0
    instance-type userplane
  exit
exit
exit
exit

```

2 番目のアクティブ RCM

```

config
control-plane-group g1
  redundancy-group 1
  host Active2
    peer-node-id ipv4-address 209.165.200.240
  exit
exit
exit
context EPC2
  interface-loopback loop1_up2
    redundancy-group 1
    host Active2
      ipv4-address 209.165.200.224/27
    exit
  exit
  interface-loopback loop2_up2
    redundancy-group 1
    host Active2
      ipv4-address 209.165.201.0/27
    exit
  exit
  interface-loopback loop3_up2
    redundancy-group 1
    host Active2
      ipv4-address 209.165.202.128/27
    exit
  exit
  interface-loopback loop4_up2
    redundancy-group 1
    host Active2

```

```
        ipv4-address 192.0.2.0/24
    exit
    exit
exit
interface-loopback loop5_up2
    redundancy-group 1
    host Active2
    ipv4-address 198.51.100.0/24
    exit
    exit
exit
user-plane-service up_up2
    redundancy-group 1
    host Active2
    associate control-plane-group g1
    associate fast-path service
    associate sx-service sx_up2
    associate gtpu-service pgw-gtpu_up2 pgw-ingress
    associate gtpu-service saegw-sxu_up2 cp-tunnel
    associate gtpu-service sgw-engress-gtpu_up2 sgw-egress
    associate gtpu-service sgw-ingress-gtpu_up2 sgw-ingress
    exit
    exit
exit
gtpu-service pgw-gtpu_up2
    redundancy-group 1
    host Active2
    bind ipv4-address 209.165.201.0
    exit
    exit
exit
gtpu-service saegw-sxu_up2
    redundancy-group 1
    host Active2
    bind ipv4-address 209.165.202.128
    exit
    exit
exit
gtpu-service sgw-engress-gtpu_up2
    redundancy-group 1
    host Active2
    bind ipv4-address 192.0.2.0
    exit
    exit
exit
gtpu-service sgw-ingress-gtpu_up2
    redundancy-group 1
    host Active2
    bind ipv4-address 198.51.100.123
    exit
    exit
exit
sx-service sx_up2
    redundancy-group 1
    host Active2
    bind ipv4-address 198.51.100.0
    instance-type userplane
    exit
    exit
exit
exit
```

共通の設定

以下に共通の設定例を示します。これは、すべてのアクティブ UP とすべてのスタンバイ UP にプッシュされます。

```

config
  active-charging service ACS
  idle-timeout udp 60
  statistics-collection ruledef all
  host-pool IPv6_VoLTE_Phone_Host_7
  ip 209.165.200.224/27
  ip 64:ff9b::d3f6:6b00/120
  ip 2001:e60:6000::/46
  ip 2001:e60:6004::/46
  ip range 209.165.200.225 to 209.165.200.234
  ip range 64:ff9b::e00:4f12 to 64:ff9b::e00:4f14
  ip range 64:ff9b::3d6e:ff52 to 64:ff9b::3d6e:ff59
  ip range 64:ff9b::d3f6:682c to 64:ff9b::d3f6:683e
  exit
  port-map M_learning_Port
  port range 1 to 9500
  port range 10001 to 30000
  exit
  port-map OTM_Advertisement_port
  port 90
  port 9090
  exit
  ruledef ICMP
  ip protocol = 1
  exit
  ruledef ICMPv6
  ip protocol = icmpv6
  exit
  ruledef IPv6_VoLTE_Phone_1
  udp either-port range port-map M_learning_Port
  ip server-ip-address range host-pool IPv6_VoLTE_Phone_Host_7
  exit
  ruledef RD-allTraffic
  ip any-match = TRUE
  exit
  ruledef RD_Charge
  ip server-ip-address = 209.165.201.0/27
  exit
  ruledef catchall
  ip any-match = TRUE
  exit
  ruledef googles
  icmpv6 any-match = TRUE
  exit
  ruledef qcil
  tcp any-match = TRUE
  exit
  ruledef route-ims-ipv6-nextHop
  ip uplink = TRUE
  ip version = ipv6
  exit
  ruledef optIn
  ip any-match = TRUE
  exit
  group-of-ruledefs GoR_FOTA
  add-ruledef priority 1 ruledef FOTA_SAMSUNG
  add-ruledef priority 2 ruledef FOTA_LG
  add-ruledef priority 3 ruledef FOTA_LG_2

```

```
add-ruledef priority 5 ruledef FOTA_PANTECH_2
add-ruledef priority 8 ruledef IOS_OTA_Update
add-ruledef priority 9 ruledef GOTA_google
add-ruledef priority 10 ruledef FOTA_Hybrid_Egg
add-ruledef priority 11 ruledef FOTA_SAMSUNG_2
add-ruledef priority 12 ruledef FOTA_SAMSUNG_3
add-ruledef priority 13 ruledef FOTA_SAMSUNG_4
add-ruledef priority 15 ruledef FOTA_SAMSUNG_5
add-ruledef priority 16 ruledef FOTA_LG_4
add-ruledef priority 17 ruledef FOTA_LG_5
add-ruledef priority 18 ruledef FOTA_HUAWEI_Egg
add-ruledef priority 20 ruledef KTF_DMS_FOTA
add-ruledef priority 21 ruledef FOTA_Nlabs
add-ruledef priority 22 ruledef FOTA_LTE_Beam
add-ruledef priority 23 ruledef FOTA_S_Mobile
add-ruledef priority 24 ruledef FOTA_Giga_Genie
add-ruledef priority 100 ruledef SAMSUNG_SKT_issue
add-ruledef priority 104 ruledef new_FOTA_Pantech
add-ruledef priority 106 ruledef new_FOTA_KTtech
add-ruledef priority 107 ruledef new_IOS_OTA_Log
add-ruledef priority 114 ruledef new_FOTA_LG_3
add-ruledef priority 200 ruledef IoT_FOTA_mexus
add-ruledef priority 201 ruledef IoT_FOTA_acnt
add-ruledef priority 202 ruledef IoT_FOTA_amtel
exit
packet-filter qcil
ip protocol = 1
ip remote-port = 1001
priority 1
exit
packet-filter subscriber-pools
exit
charging-action CA-nothing
content-id 5
exit
charging-action CA_Chargeable_2
content-id 1
billing-action egcdr
exit
charging-action CA_Charge
exit
charging-action DSI
billing-action egcdr
flow action discard
tft packet-filter permit_all
exit
charging-action call
service-identifier 22
billing-action egcdr
cca charging credit
flow action discard
flow limit-for-bandwidth id 4
exit
charging-action catchall
content-id 10
billing-action egcdr
cca charging credit rating-group 10 preemptively-request
exit
charging-action qcil
billing-action egcdr
cca charging credit rating-group 1 preemptively-request
qos-class-identifier 1
tft packet-filter qcil
exit
```

```

bandwidth-policy bw-policy
flow limit-for-bandwidth id 2 group-id 2
flow limit-for-bandwidth id 4 group-id 4
flow limit-for-bandwidth id 10 group-id 12
flow limit-for-bandwidth id 562 group-id 562
group-id 2 direction downlink peak-data-rate 225280 peak-burst-size 2253 violate-action
discard
group-id 4 direction uplink peak-data-rate 450560 peak-burst-size 4506 violate-action
discard
group-id 10 direction uplink peak-data-rate 1153434 peak-burst-size 11534 violate-action
discard
group-id 11 direction uplink peak-data-rate 10000 peak-burst-size 10000 violate-action
discard
exit
rulebase 5G-DF
tcp packets-out-of-order timeout 30000
no retransmissions-counted
edr sn-charge-volume count-dropped-units
bandwidth default-policy bw-policy
exit
rulebase RB-allTraffic
action priority 10 ruledef RD-allTraffic charging-action CA_Charge
egcdr threshold interval 3600
egcdr threshold volume total 4000000
exit
rulebase RB_Charge
action priority 10 ruledef RD_Charge charging-action CA_Charge
exit
rulebase cisco
billing-records egcdr
action priority 12 ruledef catchall charging-action catchall monitoring-key 123
egcdr threshold interval 120
egcdr threshold volume total 1000000
exit
rulebase cisco_dynamic
action priority 11 dynamic-only ruledef qcil charging-action qcil
action priority 10000 ruledef catchall charging-action catchall
egcdr threshold interval 120
egcdr threshold volume total 100000
exit
rulebase P2P
transactional-rule-matching
dynamic-rule order first-if-tied
tethering-detection application ip-ttl value 62
flow end-condition timeout normal-end-signaling session-end charging-edr flow-edr
billing-records egcdr
edr transaction-complete http charging-edr http-edr
flow control-handshaking charge-to-application all-packets
egcdr threshold interval 3600
egcdr threshold volume total 4000000000
no cca quota retry-time
cca diameter requested-service-unit sub-avp volume cc-total-octets 5000
p2p dynamic-flow-detection
no tft-notify-ue-def-bearer
exit
rulebase default
exit
rulebase wap_adult
    transactional-rule-matching
    tcp mss 1320 limit-if-present
    flow end-condition handoff timeout normal-end-signaling session-end charging-edr
flow-edr
    billing-records egcdr radius
    action priority 28 ruledef catchall charging-action CA_Chargeable_2

```

```
    action priority 29 ruledef catchall charging-action CA_Chargeable_2
    edr transaction-complete http charging-edr http-edr
    flow control-handshaking charge-to-application mid-session-packets tear-down-packets

    egcdr threshold volume total 3000000
#exit
service-scheme ssl
exit
credit-control group DCCA_grp1
diameter origin endpoint Gy
diameter peer-select peer minid-Gy
pending-traffic-treatment noquota buffer
pending-traffic-treatment quota-exhausted buffer
pending-traffic-treatment validity-expired pass
exit
credit-control group default
pending-traffic-treatment noquota pass
pending-traffic-treatment quota-exhausted buffer
exit
policy-control charging-rule-base-name active-charging-rulebase
policy-control burst-size auto-readjust duration 3
exit
context ecs
apn cisco.com
ip context-name ecs
exit
apn starent.com
ip context-name ecs
exit
end
```

スタンバイ設定 (Active1 + Active2)

```
config
context EPC2
interface loop1_up1 loopback
ip address 198.51.100.123 255.255.255.224

interface loop2_up1 loopback
ip address 209.165.201.0 255.255.255.224

interface loop3_up1 loopback
ip address 209.165.202.128 255.255.255.224

interface loop4_up1 loopback
ip address 192.0.2.0 255.255.255.224

interface loop5_up1 loopback
ip address 198.51.100.0 255.255.255.224

exit
exit

context EPC2
sx-service sx_up1
instance-type userplane
bind ipv4-address 198.51.100.0
exit

exit
exit

context EPC2
```

```
gtpu-service pgw-gtpu_up1
bind ipv4-address 209.165.201.0
exit
gtpu-service saegw-sxu_up1
bind ipv4-address 209.165.202.128
exit
gtpu-service sgw-engress-gtpu_up1
bind ipv4-address 192.0.2.0
exit
gtpu-service sgw-ingress-gtpu_up1
bind ipv4-address 198.51.100.0

exit
exit

context EPC2
user-plane-service up_up1
associate gtpu-service pgw-gtpu_up1 pgw-ingress
associate gtpu-service sgw-ingress-gtpu_up1 sgw-ingress
associate gtpu-service sgw-engress-gtpu_up1 sgw-egress
associate gtpu-service saegw-sxu_up1 cp-tunnel
associate sx-service sx_up1
associate fast-path service
associate control-plane-group g1
exit

exit
exit

config
context EPC2
interface loop1_up2 loopback
ip address 209.165.200.230 255.255.255.224

interface loop2_up2 loopback
ip address 209.165.200.231 255.255.255.224

interface loop3_up2 loopback
ip address 209.165.200.232 255.255.255.224

interface loop4_up2 loopback
ip address 209.165.200.233 255.255.255.224

interface loop5_up2 loopback
ip address 209.165.200.234 255.255.255.224

exit
exit

context EPC2
sx-service sx_up2
instance-type userplane
bind ipv4-address 209.165.200.234
exit

exit
exit

context EPC2
gtpu-service pgw-gtpu_up2
bind ipv4-address 209.165.200.231
exit
gtpu-service saegw-sxu_up2
bind ipv4-address 209.165.200.232
```



```
exit
gtpu-service sgw-engress-gtpu_up2
bind ipv4-address 209.165.200.233
exit
gtpu-service sgw-ingress-gtpu_up2
bind ipv4-address 209.165.200.230

exit
exit

context EPC2
user-plane-service up_up2
associate gtpu-service pgw-gtpu_up2 pgw-ingress
associate gtpu-service sgw-ingress-gtpu_up2 sgw-ingress
associate gtpu-service sgw-engress-gtpu_up2 sgw-egress
associate gtpu-service saegw-sxu_up2 cp-tunnel
associate sx-service sx_up2
associate fast-path service
associate control-plane-group g1
exit

exit
exit
```


翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。