4G CUPS に対する NSO オーケストレーション

機能説明

Cisco Network Service Orchestrator（NSO）ベースの VNF オーケストレーションを使用すると、新たに作成された仮想ネットワーク機能（VNF）デバイス（CP、UP、RCM など）のライフサイクルを管理できます。

Cisco NSO Orchestration for 4G CUPS ソリューションは、次の機能を提供します。

- NSO CLI、Web インターフェイス、または NSO RESTCONF API によるインスタンス化
- インスタンス化が成功した場合の CP、UP、RCM などの VNF デバイスのオンボーディング
- インスタンス化が成功した後の Day-0.5 および Day-1 CUPS 設定のプッシュ
- VNF デバイスのデコミッション

使用例

NSO オーケストレーション ソリューションは、次のユースケースに対応します。

1. 新しい CP、UP、RCM のインスタンス化

   CUPS で新しい 4G ベースの VNF（CP、UP、RCM）をインスタンス化します。CP は、仮想化パケット コア シングル インスタンス（VPC-SI）または仮想化パケットコア分散インスタンス（VPC-DI）にすることができますが、UP は VPC-SI のみにすることができます。

   障害が発生した場合は、ユーザーに通知されます。

2. CP、UP、RCM の終了

   CUPS で 4G ベースの VNF（CP、UP、RCM）を終了します。

   障害が発生した場合は、ユーザーに通知されます。

3. VNF ダッシュボードの現在のステータスの更新

   VNF のダッシュボードに現在のステータスを表示します。

4. CP、UP、RCM の論理グループの設定

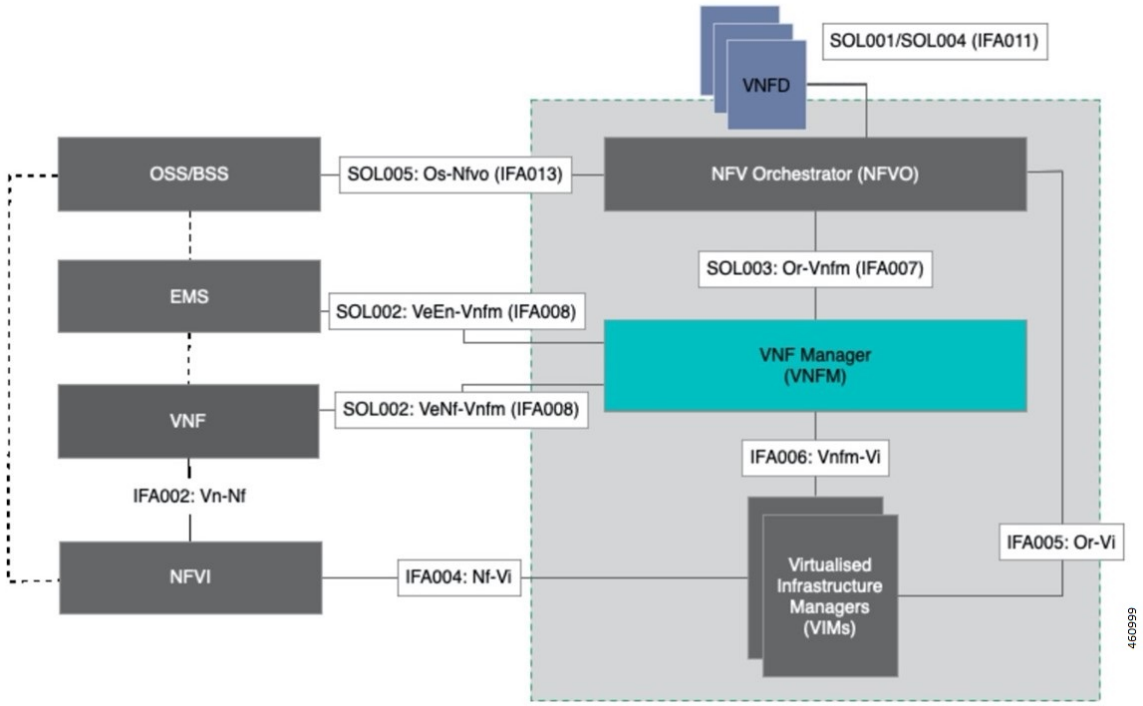
   CP、UP、および RCM をグループ化するように NSO でデバイスグループを設定し、対応する VNF をデバイスグループに追加します。

機能の仕組み

アーキテクチャ

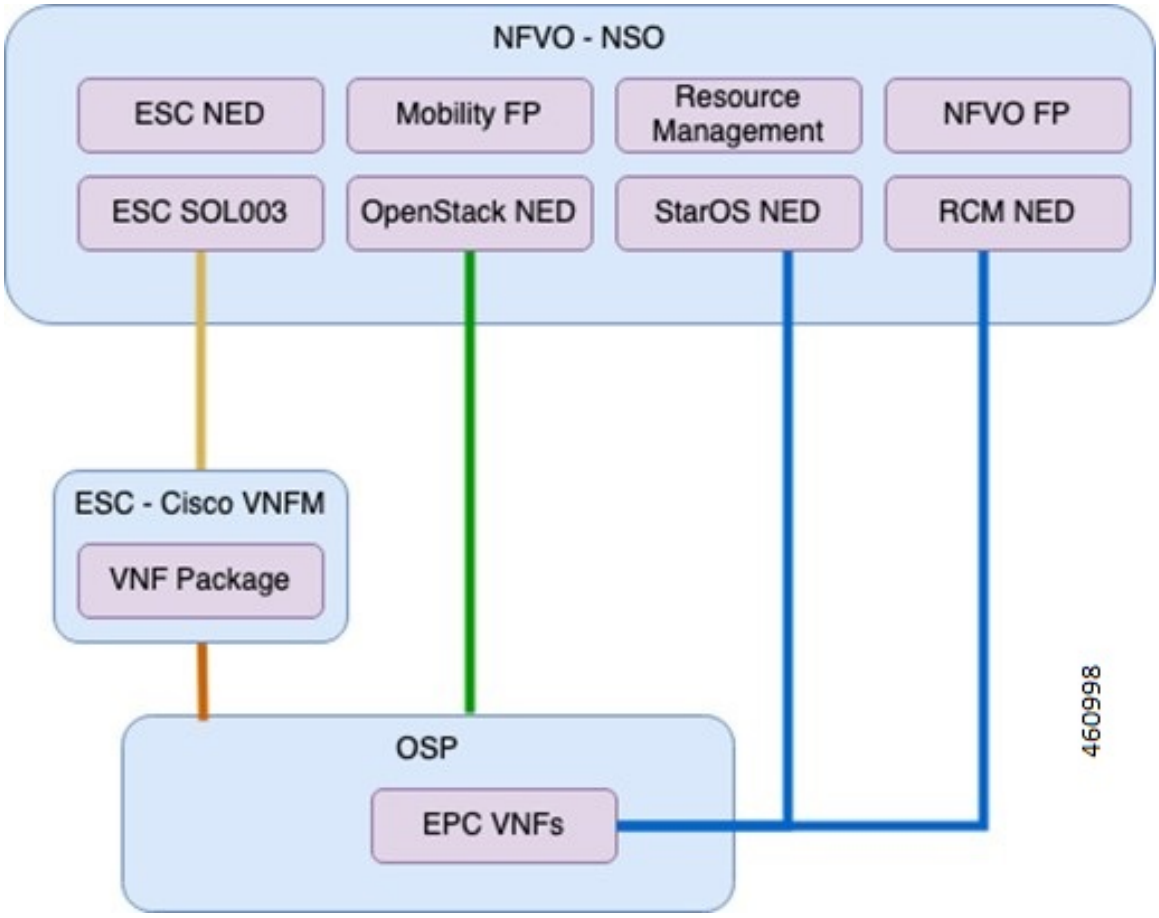
Cisco NSO オーケストレーション エンジン ソフトウェア モジュールは、ネットワーク機能の仮想化オーケストレータ（NFVO）機能を処理します。NFV ソリューションは、次の図に示すように、ETSI NFV 管理およびオーケストレーション（MANO）モデルに従います。

図 *1 : NVF* ソリューション 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。

図 *2: NFV* ソリューションのコンポーネント



コンポーネント

以下に、NSO の重要なコンポーネントの一部を示します。

- **Cisco NFVO 機能パック** ：

  Cisco NFVO 機能パックには、MANO 仕様（SOL006）に準拠した YANG モデルが含まれます。

  Cisco NFVO 機能パックには、VNF マネージャ（VNFM）および OpenStack に MANO 記述子のインスタンス化ロジックを実装する **cisco-etsi-nfvo** のモデルが含まれています。仮想ネットワーク機能（VNF）とネットワークサービス（NS）は、このパッケージの主要なサービスです。ノースバウンドユーザーは、これらのサービスとやり取りして VNF またはネットワークサービスを開始します。

  また、リソース オーケストレーション（RO）機能を含む **cisco-etsi-nfvo-ro** のモデルも含まれます。リソース オーケストレーションは、仮想化インフラストラクチャ マネージャ（VIM）の物理リソースの割り当てを管理します。これらの物理リソースは、VNF または NS によって使用されます。

- **NSO の StarOS NED**：

  StarOS ベースのネットワーク エレメント ドライバ（NED）は、設定をプッシュするために Cisco 4G CUPS VNF と接合します。

- **NSO の RCM NED**：

  RCM ベースの NETCONF NED は、NSO と RCM デバイス間の通信を確立するために使用されます。

- **Cisco ESC SOL003 NED**：

  この NED は、ETSI SOL3 準拠デバイスに使用されます。Elastic Services Controller（ESC）も、SOL3 準拠デバイスとして NSO に追加されます。

- **NFV アプリケーション モビリティ パッケージ**：

  これは、VNF ライフサイクル管理と VNF ダッシュボードの更新を提供するカスタムパッケージです。

プラットフォームおよびソフトウェアの最小要件

NSO オーケストレーションをサポートするために必要なプラットフォームとソフトウェアの最小要件は次のとおりです。

- サポートされる VIM：OpenStack
- サポートされる VNFM：Cisco ESC
- サポートされるオーケストレータ：NSO
- ネットワーク要素：
  - RCM
  - VPC-SI（UP/CP）
  - VPC-DI（CP）

表 *1:* ソフトウェア バージョン

ソフトウェア	最小バージョン
Redhat OpenStack	13（Queens） （注） VMWare や OSP 16 は、サポートおよび検証の対象外です。
Cisco ESC	5.5.0.86
Cisco NSO	6.1.6.1
OpenStack NED	4.2.30

ソフトウェア	最小バージョン
ESC NED	5.10.0.97
StarOS NSO NED	5.52.4
Cisco NFVO FP	4.7.3
Mobility FP	3.5
NSO リソース管理	3.5.2
Cisco NSO HCC	6.0.1

この機能は、次の ETSI MANO 仕様をサポートします。

表 *2: ETSI MANO* の仕様

仕様	サポートされるバージョン	説明
SOL001	v2.5.1	VNF 記述子のフォーマットと構造を定義します
SOL003	v2.4.1	Or-Vnfm 参照ポイント上のすべてのインタラクションを定義します

ネットワークおよびハードウェア要件

ネットワーク要件：

次の表に、NSO および ESC のネットワーク要件を示します。

表 *3: NSO* および *ESC* ネットワーク要件

アプリケーション	[Management IP]	オーケストレーション	HA ペア間の接続
NSO（2 VM + VIP）	3	3	遅延が 30 ミリ秒未満の 100 Mbps の L2 接続
ESC（2 VM + VIP）	3	3	遅延が 30 ミリ秒未満の 100 Mbps の L2 接続

ハードウェア要件

次の表に、最大 250 の VNF をサポートするための NSO および ESC 仮想マシンの仕様を示します。

表 *4 : NSO* および *ESC VM* の仕様

アプリケーション	VM の数	VM CPU コア数	VM RAM	VM ストレージ	VM 接続
NSO	2	8	サポート対象のすべての StarOS デバイス用に 16 GB RAM の基準＋10 MB RAM	100 GB ディスク（SSD を推奨）	10 Gbps ネットワークリンク X 1
ESC	2	4	16 GB	100 GB	

ライセンス

4G CUPS の NSO オーケストレーションは、ライセンス供与されたシスコの機能です。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。

コール フロー

この項では、4G CUPS オーケストレーション機能の主要なコールフローについて説明します。

VNF オンボーディング

この項では、VNF オンボーディングフローについて説明します。

図 *3 : VNF* オンボーディング

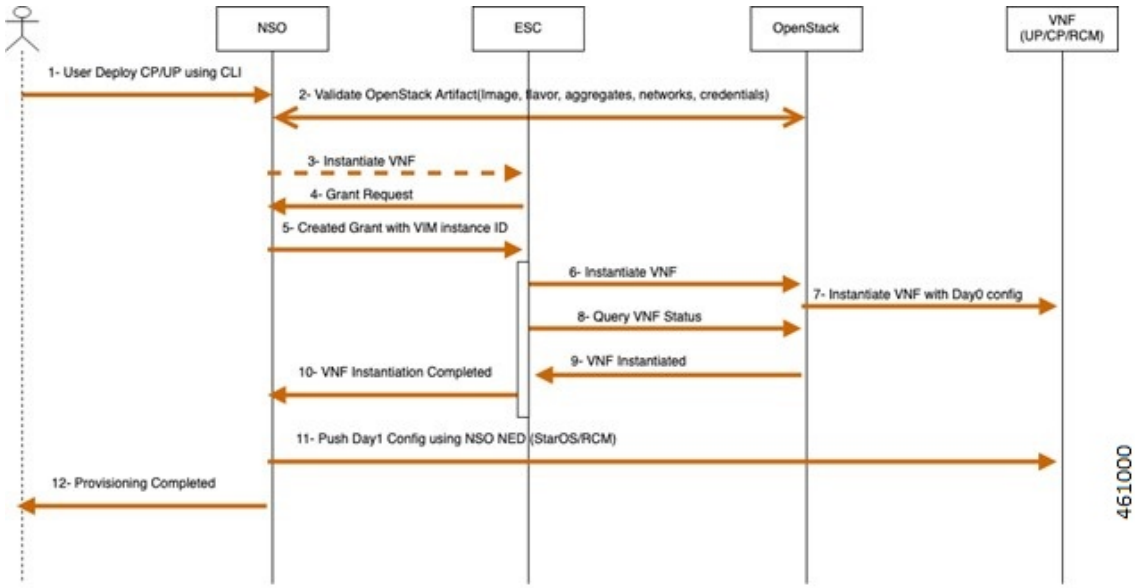


表 *5:* コール フローの説明

ステップ	説明
1	ネットワークオペレータが、NSO CLI を使用して VNF（CP、UP、または RCM）をインスタンス化します。対象には、VNF をホストする VIM ID、および ESC が含まれます。
2	NSO が OpenStack を介してユーザーから提供されたデータを検証します。
3	NSO が ESC で VNF をインスタンス化するために SOL.003 要求を送信します。
4	ESC が NSO に付与要求を送信します。
5	NSO が VIM インスタンス ID を使用してリソース付与メッセージを ESC に送信します。
6	ESC が OpenStack API を使用して VNF をインスタンス化します。
7	OpenStack が VNF を起動します。
8	ESC が OpenStack に VNF ステータスをクエリします。
9	OpenStack が VNF-Up メッセージで応答します。
10	ESC が VNF のインスタンス化について NSO に通知します。
11	NSO が Day-1 設定を VNF にプッシュします。
12	NSO が VNF のプロビジョニングが完了したことをオペレータに通知します。

P2P モジュールのインストール

モビリティ機能パックでは、VNF 展開の一部として P2P モジュールのインストールをサポートします。P2P モジュールは、デバイスのオンボーディング後にインストールされます。P2P モジュールファイルは、VNF を展開する前に NSO にアップロードしておく必要があります。設定可能なパラメータによって、ファイルの場所と、P2P のインストールが必要かどうかを示します。

P2P のインストールが完了すると、MFP 3.4.2 以降のバージョンでは、新しくインスタンス化された VNF の P2P のデフォルト優先順位は「99」になります。MFP 3.4.2 より前のバージョンでは、P2P のデフォルト優先順位は「10」から始まります。「mobility-library」アクションコマンドを使用して P2P の優先順位をアップグレードするには、「付録 *C*」の手順を参照してください。

VNF の終了

ここでは、VNF の終了フローについて説明します。

図 *4 : VNF* の終了フロー

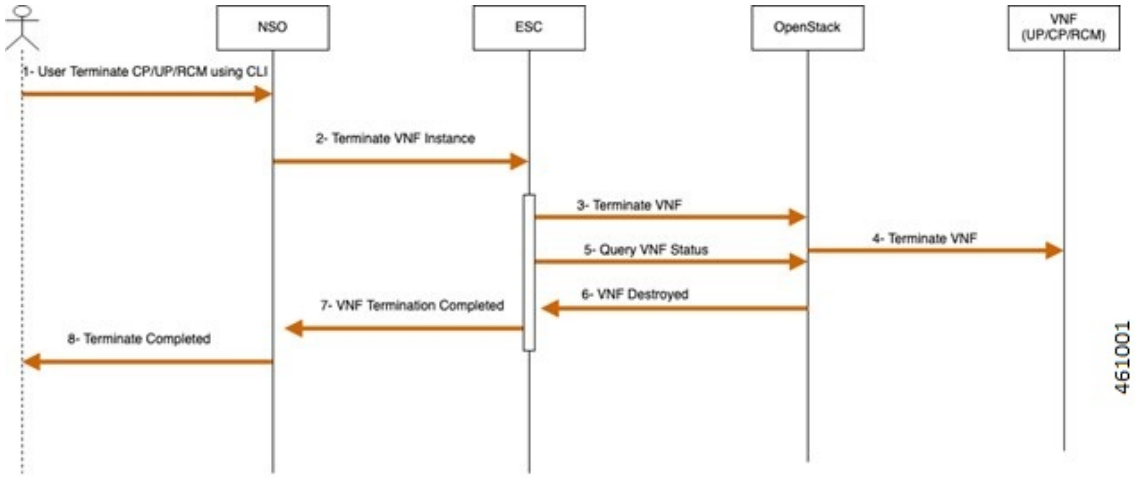


表 *6 :* コール フローの説明

手順	説明
1	オペレータは NSO CLI を使用して VNF（CP、UP、RCM）を終了します。これには、VNF ID、VNF をホストする VIM ID、および ESC が含まれます。
2	NSO は、ESC で VNF を終了するために SOL.003 要求を送信します。
3	ESC は OpenStack API を使用して VNF を終了します。
4	OpenStack は VNF を終了します。
5	ESC は OpenStack に VNF ステータスをクエリします。
6	OpenStack は VNF 破棄メッセージで応答します。
7	ESC は NSO に VNF の終了を通知します。
8	NSO は、VNF の終了が完了したことをオペレータに通知します。

リカバリ

自動修復は現在サポートされていません。

障害状態から以前の状態に回復するには、次の手順を実行します。

- VNFのインスタンス化をキャンセルまたは終了します。システムが元の状態に戻ります。

- VNF 終了プロセスをキャンセルまたは再作成します。システムが元の状態に戻ります。

制限事項

このリリースの 4G CUPS 機能の NSO オーケストレーションには、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。
- 展開を処理する NSO/ESC インスタンスがダウンすると、VNF 展開が失敗する可能性があります。これは、ESC/NSO HA とスタンドアロン ESC/NSO の両方の展開に当てはまります。障害の正確な種類に応じて、オペレータの介入が必要です。展開後に自動設定がプッシュされる場合、展開は成功しても、NSO 障害のタイミングによっては、後続の設定のプッシュが失敗する可能性があります。

NSO パッケージのインストール

NSO オーケストレーション ソリューションは、NED およびその他の NSO パッケージのコレクションを使用します。以下に、各種パッケージとそのロールに関する詳細なリストを示します。これらのパッケージのインストール手順については、該当する NSO バージョンの『*NSO Administration Guide*』[英語] のパッケージに関する章を参照してください。

1. **NSO NED パッケージ**

   ほとんどの NSO NED パッケージは公開され、個別にダウンロードできます。詳しいダウンロード方法については、シスコの担当者にお問い合わせください。

   ncs-6.1-rcm-nc.v21.28.mx_20240415-072244Z.tar.gz：NSO からの RCM デバイス通信用の RCM NETCONF ベース NED

   ncs-6.1.6-cisco-staros-5.52.4.tar.gz：NSO からの StarOS デバイス（SI または DI）通信用の CLI ベース NED

   ncs-6.1.1-etsi-sol003-1.13.18.tar.gz：NSO からの ESC 通信用の ETSI SOL003 ベース NED

   ncs-6.1-openstack-cos-4.2.30.tar.gz：NSO からの Openstack 通信用の Openstack NED

   ncs-6.1.2.1-cisco-etsi-nfvo-4.7.3.tar.gz：NSO からの ESC 通信用の NETCONF ベース NED

   ncs-6.1.2.1-esc-5.10.0.97.tar.gz：NSO からの ESC 通信用の ETSI SOL ベース NED

2. **NSO カスタムパッケージ**

   NSO カスタムパッケージは、モビリティ VNF オーケストレーション用のカスタムビルドパッケージです。これらのパッケージは、モビリティ機能パックの tar アーカイブにバンドルされています。

   Mobility-common.tar.gz：設定およびデバイスメタデータの共通パッケージ

nfv-common.tar.gz：VNF オーケストレーション関連の共通ユーティリティの共通パッケージ

nfv-device-onboarding.tar.gz：NSO デバイスのオンボーディングをサポートするパッケージ

nfv-vim.tar.gz：Openstack 関連の事前チェック機能のパッケージ

nfv-vnf-lcm.tar.gz：VNF のインスタンス化および終了ロジックのパッケージ

mop-common.tar.gz：設定 MOP 関連の共通ユーティリティの共通パッケージ

Mobility-mop.tar.gz：モビリティ MOP 設定プッシュ用パッケージ

3. **オーケストレーションに必要な VNF パッケージ（SOL003/SOL004）**

   特定の VNF のオンボーディングに使用される VNF パッケージです。これらのパッケージはガイドラインとしてのみ提供されます。ほとんどの場合、特定のパッケージは導入環境に合わせてカスタマイズされます。

   VPC-SI-2P-IMAGE-BOOT：SI インスタンス化の参照用 SOL003/SOL004 CSAR パッケージ

   RCM-IMAGE-BOOT：RCM インスタンス化の参照用 SOL003/SOL004 CSAR パッケージ

   VPC-DI-2P-1DI-ENCRYPTVOLBOOT：2 つの CF と 4 つの SF を使用した VPC DI インスタンス化の参照用 SOL003/SOL004 CSAR パッケージ。SF には 2 つのサービスネットワークがあります。

   VPC-DI-2P-1DI-ENCRYPTVOLBOOT-LTD：2 つの CF と 2 つの SF を使用した VPC DI インスタンス化の参照用 SOL003/SOL004 CSAR パッケージ。SF には 2 つのサービスネットワークがあります。

   VPC-DI-2P-1DI-ENCRYPTVOLBOOT-LTD-1S-NETWORK：2 つの CF と 2 つの SF を使用した VPC DI インスタンス化の参照用 SOL003/SOL004 CSAR パッケージ。SF のサービスネットワークは 1 つのみです。

   create-zip.sh：SOL001 定義または Day-0 スクリプトに変更がある場合に、SOL003 パッケージを再構築するシェルスクリプト。

（注） すでにモビリティ機能パックを使用している場合は、「付録*B*：モビリティ機能パック（*MFP*）の一般的なアップグレード手順」の手順を参照してください。

VNF オーケストレーション/展開および自動設定管理

このソリューションには、以下のタスクが含まれます。

- VNF オーケストレーションの設定メタデータの事前入力。
- VNF（CP、UP、RCM）のオーケストレーション/展開
- VNF 展開後の自動デバイスオンボーディング

- 展開後の自動設定プッシュ

VNF オーケストレーションの設定メタデータの事前入力

設定メタデータの事前入力は、自動モードで展開後の設定を NSO からプッシュするために重要です。このデバイスに事前入力されたデータがない場合、NSO は VNF とオンボードを NSO のデバイスとしてインスタンス化します。

設定メタデータの事前入力には次の構造があり、このデータの入力はネットワークスキームとデータセットに基づいています。

```
container
metadata-store {
         list config-metadata {
             key device-name;
             leaf device-name {
                 tailf:info "onboarding device name";
                 type string;
             }
             leaf redundancy_scheme {
                 tailf:info "cluster-topology 1:1, N:M and N+2";
                 type string;
             }
             leaf device-type {
                 tailf:info "Onboarding device type vpc or rcm";
                 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;
                            }
                        }
                    }
                }
            }
        }
    }
```

次の表に、パラメータの説明を示します。

パラメータ	説明
device-name	VNF に対応する NSO デバイスの名前。VNF 名と同じ。
redundancy_scheme	冗長スキームのタイプ。N + 2 はスタンドアロン（冗長性なし）
device-type	VPC（SI および DI の場合）または RCM（RCM の場合）
configuration-type	Day-0.5 は、N:M 冗長性のための特別な設定です。この設定により、UP は RCM に接続できます。この設定は永続的に保存されます。
	Day-1 は設定の大部分を占めます。
	Day-N は通常、動作中の設定に変更されます。NSO オーケストレーション フローには適用されません。
file-name	プッシュされるプライマリ構成ファイル
config-scheme	このパラメータの値は、次のいずれかです。 **Common**：設定は、ロール（アクティブまたはスタンバイ）に関係なく、すべての UP にプッシュされます。 **host-specific**：このスキームは、設定がすべての UP（アクティブまたはスタンバイ）にプッシュされるため、「Common」に似ていますが、設定は「Common」設定の後にのみプッシュされます。そのため、「Common」設定に依存する設定を提供できます。例：コントロール プレーン グループの設定。 **allHostSpecific**：すべてのアクティブ UP のホスト固有の設定の集合が含まれます。設定は、N:M に対するすべてのスタンバイ UP にプッシュされます。 **"Active1", "Active2",... "ActiveN"**：それぞれのアクティブ UP のホスト固有の設定。特定の UP にのみプッシュされます。 **"Active1-rcm", "Active2-rcm", .. "ActiveN-rcm"**：この設定は RCM 形式であり、RCM にプッシュされます。RCM は、スタンバイが特定のアクティブデバイスを引き継ぐ際に設定の否定を実行するためにこのスキームを必要とします。

パラメータ	説明
additional-files	このパラメータは、関連する設定を他のデバイスにプッシュします（たとえば、UP のオンボーディング時に設定を CP にプッシュします）。このパラメータはまだサポートされていません。
attribute-name	このパラメータは、動的置換用の構成ファイル内の属性（変数）を識別し、$attribute_name としてフォーマットされます。
attribute-value	属性の値

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

```
container
  config-metadata {
         // config true;
         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;
                     }
                     leaf device-type {
                         tailf:info "Onboarding device type vpc or rcm";
                         type enumeration {
                             enum vpc;
                             enum rcm;
                         }
                     }
                     leaf redundancy_scheme {
                         tailf:info "cluster-topology 1:1, N:M and N+2";
                         type enumeration {
                             enum 1:1;
                             enum N:M;
                             enum RCUPS;
                         }
                     }


                     list configuration-type {
                         key config-type;
                         tailf:info "Configuration type Day0.5, Day1 or DayN";
                         leaf config-type {
                             type enumeration {
                                 enum Day0.5;
                                 enum Day1;
                                 enum DayN;
                             }
                         }
                         list files {
                             key file-name;
                             tailf:info "file name";
                             leaf file-name {
                                 type string;
                             }
                             leaf config-scheme {
                                 type enumeration {
                                     enum common;
```

```
                                    enum host-specific;
                                    enum host-specific-common;
                                }
                            }
                            // 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;
                                    }
                                }


                                }
                            }
                        }
                    }
                    list attributes {
                        key attribute-name;
                        leaf attribute-name {
                            tailf:info "Attribute Name";
                            type string;
                        }
                        leaf attribute-value {
                            tailf:info "Attribute Value";
                            type string;
                        }
                    }
                }
                leaf delete-config-data {
                    type boolean;
                    default false;
                }
            }
            output {
                leaf status {
                    type string;
                }
                leaf message {
                 type string;
                }
            }
        }
    }
```

次の例に示されているように、このアクションは RESTCONF から呼び出せます。

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

メソッド：POST

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":"test",
                 "attribute-value": "gh"
                  },
            "configuration-type":{
                "config-type": "Day0.5",
                "files":{
                    "file-name":"/home/ubuntu/tmo_action/test.txt"
                    },
                "files":{
                    "file-name":"/home/ubuntu/tmo_action/day0.5.txt"
                    }

            }
        }
}
```

結果：

```
{
    "mobility-common:output": {
        "status": "Success
/home/ubuntu/tmo_action/test.txt ==> syntax error: unknown command,Error: on line 3:
kkkl,
/home/ubuntu/tmo_action/day0.5.txt ==> Success"
    }
}
```

次の例に示されているように、このアクションは NCS CLI を使用して呼び出せます。

```
ubuntu@ncs> request config-metadata config-metadata-request config-metadata { device-name
 staros-1 attributes { attribute-name hostname attribute-value TEST } configuration-type
 { config-type Day0.5 files { file-name /home/ubuntu/tmo_action/test.txt } files {
file-name /home/ubuntu/tmo_action/day0.5.txt } } schema 1:1  }
status Success
/home/ubuntu/tmo_action/test.txt ==> syntax error: unknown command,Error: on line 3:
kkkl,
/home/ubuntu/tmo_action/day0.5.txt ==> Success
[ok][2021-07-12 08:05:01]
```

注：

- Config-metadata-request アクションには内部設定バリデータがあります。設定バリデータを使用すると、設定をプッシュする前に、構文や特定のセマンティックエラー（範囲外の値など）を検出できます。設定の検証には、少なくとも NSO でオンボーディングされているデバイス（real-one または NetSim）が必要です。

  設定可能なパラメータは次のとおりです。

  ```
  container
  configurable-parameters {
     leaf config-pre-validation-vpc-device-name {
             type string;
     leaf config-pre-validation-rcm-device-name {
             type string;
     }
  }
  ```

このファイルの設定の検証も任意です。設定を検証しない場合は、設定可能なパラメータを使用してこの機能をオフにできます。設定の検証がオフになっている場合、構成ファイルにエラーがあると設定のプッシュエラーが発生するため、ロールバックする必要があります。

```
container
configurable-parameters {
   leaf config-pre-validation-required {
           type boolean;
           default false;
   }
   }
  }
```

この設定メタデータには、設定可能なすべてのパラメータが含まれています。

デバイスとしての ESC および OpenStack のオンボーディング

ESC のインストールについては、ESC のマニュアルを参照してください。VNF の設定またはオンボーディングとインスタンス化の前に、次の設定手順を実行します。

NFV 用の NSO および ESC 環境のセットアップ

1. ユーザー名とパスワードを使用して ESC ホストに SSH 接続します。

   ```
   ssh esc@<esc-ip>
   ```

2. Sudo ユーザーになります。

   sudo su

3. 次のファイルを編集します。`vi /opt/cisco/esc/esc_database/etsi-production.properties`

4. 以下に示されているように情報を編集し、ファイルを保存します（Spring ユーザーとパスワードは変更しないでください）。適宜NSOの詳細を変更します。通信にはローカルサブネット管理 IP のみを使用し、ESC/NSO 通信間のフローティング IP は使用しないでください。

   ```
   spring.security.user.name=esc
   spring.security.user.password=$1$J7BUBX$Ce4vqA6JcrWCggRpYrPYg1

   security.pam.service=
   server.additionalConnector.port=8253
   server.additionalConnector.key-alias=esc
   server.esc.key-alias=esc

   nfvo.apiRoot=<NSO-IP>:9191
   nfvo.httpScheme=http
   nfvo.userName=<NSO-User-name>
   nfvo.password=<NSO-Password>
   nfvo.authenticationType=BASIC

   server.host=<ESC-Orch-IP>
   http.enabled=true
   https.enabled=false
   certificate.validation=false
   ```

```
spring.datasource.password=${PGSQL_PASSWORD}
spring.flyway.password=${PGSQL_PASSWORD}
```

5. 次に示されているように、**escadm** サービスを再起動します。

```
escadm restart
Stopping esc_service: [OK]
Stopping escadm service: [OK]
Starting escadm service: [OK]
#
```

6. 次に示されているように、escadmの正常性を確認します（数分かかる場合があります）。

```
escadm health
============== ESC =================
vimmanager (pgid 18651) is running
monitor (pgid 18688) is running
mona (pgid 18741) is running
snmp is disabled at startup
etsi (pgid 19316) is running
pgsql (pgid 18944) is running
portal (pgid 19355) is running
confd (pgid 18978) is running
escmanager (pgid 19131) is running
=====================================
ESC HEALTH PASSED
```

7. NSO にログインし、環境に応じて設定を変更し、ファイルに保存します。

```
<config xmlns="http://tail-f.com/ns/config/1.0">
  <nfv xmlns="urn:etsi:nfv:yang:etsi-nfv-descriptors">
  <settings xmlns="http://cisco.com/ns/nso/cfp/cisco-etsi-nfvo">
    <image-server>
      <ip-address><NSO-IP></ip-address>
      <port>8010</port>
      <document-root>/var/opt/ncs/vnfpackages</document-root>
    </image-server>
    <etsi-sol3>
      <server>
        <ip-address><NSO-IP></ip-address>
        <port>9191</port>
        <use-ssl>false</use-ssl>
        <document-root>/var/opt/ncs</document-root>
        <auth-enabled>true</auth-enabled>
        <auth-types>
          <basic>
            <username><NSO-USERNAME></username>
            <password><NSO-PASSWORD></password>
          </basic>
        </auth-types>
      </server>
      <vnfm-behaviour>
        <vnfm-behaviour-override>
          <id>default-sol3</id>
          <rpc-behaviour>
            <rpc>
              <include>
                <vim-info>false</vim-info>
              </include>
            </rpc>
            <modify>
              <pre>
                <rpc>false</rpc>
              </pre>
```

```
            <post>
              <rpc>true</rpc>
            </post>
          </modify>
        </rpc-behaviour>
        <grant>
          <store-history>false</store-history>
          <heal>
            <authorise-grant>true</authorise-grant>
          </heal>
        </grant>
        <onboarding>
          <store-details>true</store-details>
        </onboarding>
      </vnfm-behaviour-override>
    </vnfm-behaviour>
  </etsi-sol3>
 </settings>
 </nfv>
</config>
```

8. パッケージフォルダ内のパッケージをすべてコンパイルし、パッケージのリロードを実行します。

```
ubuntu@test-nso:/var/opt/ncs/packages$ ncs_cli -C
User ubuntu last logged in 2021-09-23T08:00:34.649202+00:00, to test-nso, from
209.165.200.225 using cli-ssh
ubuntu connected from 209.165.200.225 using ssh on test-nso
ubuntu@ncs# packages reload
```

9. 次に示されているように、ファイルをロードマージします。この手順では、NSO を NFVO として有効にし、9191 ポートで NFVO サービスを実行します。

```
ubuntu@test-nso:~$ vi config.xml
ubuntu@test-nso:~$ ncs_cli -C

User ubuntu last logged in 2021-08-04T09:10:55.819283+00:00, to test-nso, from
209.165.200.226 using cli-ssh
ubuntu connected from 209.165.200.227 using ssh on test-nso
ubuntu@ncs# config
Entering configuration mode terminal
ubuntu@ncs(config)# load merge config.xml
Loading.
1.54 KiB parsed in 0.01 sec (128.38 KiB/sec)
ubuntu@ncs(config)# commit
```

10. NSO ユーザー名を「ncsadmin」グループに追加して NACM ルールを更新します。

```
ubuntu@test-nso:~$ ncs_cli -C

User ubuntu last logged in 2021-08-06T09:56:26.370979+00:00, to test-nso, from
209.165.200.227 using cli-ssh
ubuntu connected from 209.165.200.227 using ssh on test-nso
ubuntu@ncs# config
Entering configuration mode terminal
ubuntu@ncs(config)# nacm groups group ncsadmin user-name ubuntu
ubuntu@ncs(config-group-ncsadmin)# commit
```

11. 必要なパッケージを NSO の標準の場所（通常は/var/opt/ncs/packages）にコピーします。

12. パッケージのリロードを実行し、パッケージのステータスを確認します。すべてのパッケージのステータスが UP である必要があります。

```
ubuntu@test-nso:~$ ncs_cli -C

User ubuntu last logged in 2021-08-06T09:58:39.866838+00:00, to test-nso, from
209.165.200.227 using cli-ssh
ubuntu connected from 209.165.200.227 using ssh on test-nso
ubuntu@ncs# packages reload
ubuntu@ncs# show packages package oper-status

NAME                UP  PROGRAM CODE ERROR  JAVA UNINITIALIZED PYTHON UNINITIALIZED
 *
------------------------------------------------------------------------------
cisco-etsi-nfvo        X      -                     -             -

cisco-rcm-nc-1.0       X      -                     -             -

cisco-staros-cli-5.38  X      -                     -             -

esc                    X      -                     -             -

etsi-sol003-gen-1.13   X      -                     -             -

mobility-common        X      -                     -             -

mop-automation         X      -                     -             -

mop-common             X      -                     -             -

nfv-common             X      -                     -             -
nfv-device-onboarding  X      -                     -             -

nfv-vim                X      -                     -             -

nfv-vnf-lcm            X      -                     -             -

openstack-cos-gen-4.2  X      -                     -             -
```

13. 通知ストリームを設定します。/etc/ncs/ncs.confファイルを更新して、「nfv-events」ストリームを追加します。

```
<ncs-config>
    <event-streams>
      <notifications>
        <stream>
          <name>nfv-events</name>
          <description>Generic netconf notification stream for NFV
events</description>
          <replay-support>true</replay-support>
          <builtin-replay-store>
            <enabled>true</enabled>
            <dir>${NCS_RUN_DIR}/state</dir>
            <max-size>S10M</max-size>
            <max-files>50</max-files>
          </builtin-replay-store>
        </stream>
      </event-streams>
    </notifications>
</ncs-config>
```

14. sudo ユーザーとして NSO を再起動します。

```
/etc/init.d/ncs stop
Stopping ncs (via systemctl):                              [  OK  ]
/etc/init.d/ncs start
Starting ncs (via systemctl):                              [  OK  ]
```

15. デバイスオンボーディング API を介して、NSO のデバイスとして NETCONF、ESC、ETSI SOL003 ESC、および OpenStack をオンボードします。

    1. デバイスとして OpenStack をオンボードします。次に、例を示します。特定の展開に合わせてカスタマイズします。これは、コンフィギュレーション モードで NSO CLI を使用して設定できます。authgroup については、NSO のマニュアルを参照してください。

    ```
    devices device openstack
    address 209.165.200.228
    port 5000
    authgroup openstack
    device-type generic ned-id openstack-cos-gen-4.2
    ```

    2. ESC ETSI インターフェイスをデバイスとしてオンボードします。次に、例を示します。特定の展開に合わせてカスタマイズします。

    ```
    devices device esc-etsi
    address 209.165.200.229
    port 8250
    authgroup esc-etsi
    device-type generic ned-id etsi-sol003-gen-1.13
    ```

    3. ESC ネイティブ NETCONF インターフェイスをデバイスとしてオンボードします。次に、例を示します。特定の展開に合わせてカスタマイズします。

    ```
    devices device esc-netconf
    address 209.165.200.229
    ssh host-key ssh-rsa
    key-data "AAAAB3NzaC1yc2EAAAADAQABAAABAQDYwNCaa3ghJtnJSvn/
    aSPjCuoMKmssZds+J5d9JCOS\n3h3V/fCtJwiH7qMgMXnNc0LEr1fZhxQ4kg5o/
    IafmoYD7N+w/ECqWEp68sjeN+AftiZ9J74D\n+/KDonffgBCHxIVEo0XHYlojrtmpg/
    EH9/N3fQgoSzEhGItGG4uMaAzbWr1pO8AApOPlPi4r\nciL4Qemi6u4i/
    HGFr8MqQp5qcMFd8O30OlB1q1vKn9sq/9sL6EzqyUd2lMounDglEQYMgi8J\
    nyG6upsOFuvhiYRC9qfHML45quyepsJdVi2Li2QwUJLa89EDh148RlhLTJs4s2iAwBGNdvLdK\ntzLu2VGyWKqH"
    !
    authgroup esc-netconf
    device-type netconf ned-id esc
    ```

16. 次に示されているように、さまざまなデバイスについてデバイスの追加ステータスを追跡します。

```
ubuntu@test-nso:~$ ncs_cli -C

User ubuntu last logged in 2021-08-06T10:09:23.550686+00:00, to test-nso, from
209.165.200.227 using cli-ssh
ubuntu connected from 209.165.200.227 using ssh on test-nso
ubuntu@ncs# show vnf-status instances esc-netconf

INSTANCE ID  TIMESTAMP   FUNCTION TYPE OPERATION  STATUS   STATUS MESSAGE
------------------------------------------------------------------------------------------
esc-netconf  2021-07-21 *    -          init   success   Device Onboarding
initialized
             2021-07-21 *    -          init   success   Device Onboarding
initialized
```

```
              2021-07-21 *    - fetch-ssh-keys  success   fetch-ssh-keys was
successful
              2021-07-21 *    -         connect success   connect was successful
              2021-07-21 *    -       sync-from success   sync-from was successful
             2021-07-21 *    - device-config   success   Subscribed to ESC Netconf
    notification escEvent Stream
              2021-07-21 *    -          ready  success   Device Successfully
onboarded
```

VNF のインスタンス化の前提条件

VNF 展開要求を送信する前に、次の設定の変更を行います。

1. **設定パラメータ**

   必要に応じて、次の設定パラメータを設定します。

   - configurable-parameters device-ping-sleeptimesec 30（デフォルト値は 30 秒）
   - configurable-parameters device-ping-retries 150（デフォルト値は 30）。RCM の場合は、より大きな値（150 など）に設定します。
   - configurable-parameters p2p-required true（デフォルト値は false）
   - configurable-parameters p2p-soFile-path /var/opt/ncs/patch_libp2p-2.64.1418.so.tgz

2. **設定メタデータの事前入力**

   Config-metadata を設定する場合、デバイス名は VNF インスタンス名と同じにする必要があります。

   次の例に示されているように、このアクションは RESTCONF から呼び出せます。

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

   メソッド：POST

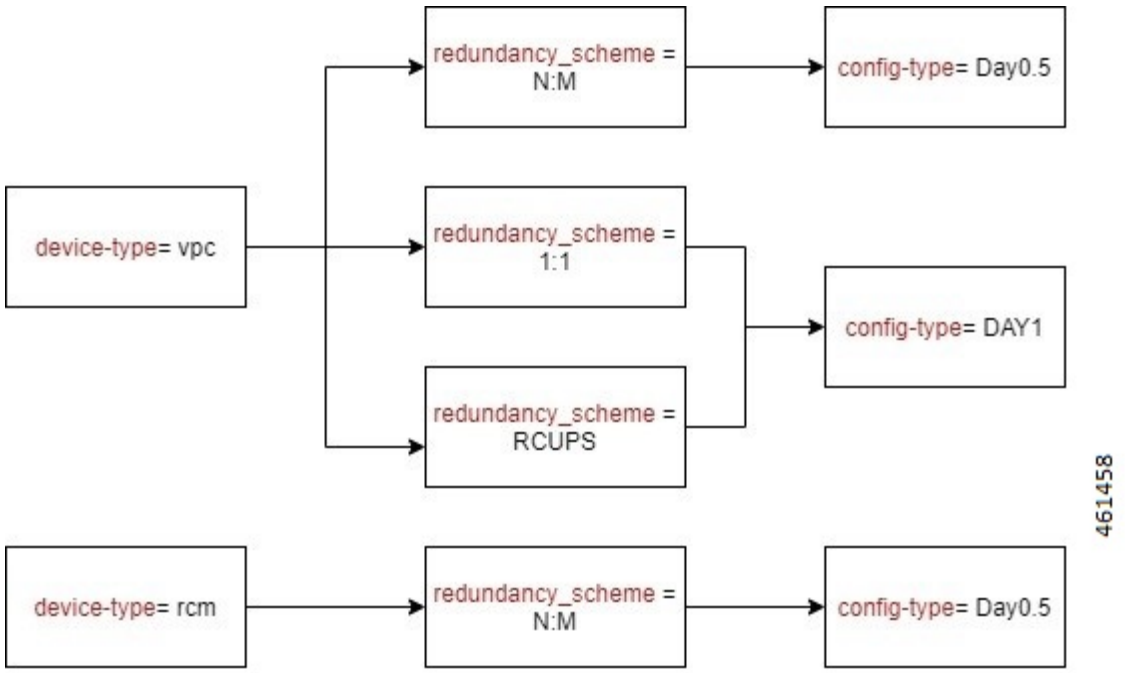
   コンテンツタイプ：application/yang-data+json

   サンプル ペイロード：

```
{
        "config-metadata": {
            "device-name": "test2",
    "schema" : "1:1",
            "attributes":{
                "attribute-name":"test",
                 "attribute-value": "gh"
                  },
            "configuration-type":{
                "config-type": "Day1",
                "files":{
                    "file-name":"/home/ubuntu/tmo_action/test.txt"
                    },
                "files":{
                    "file-name":"/home/ubuntu/tmo_action/day0.5.txt"
                    }
```

```
                }
            }
        }
```

設定メタデータを事前入力する際は、次の図に示されている基準に従ってください。



VNF のインスタンス化

VNF は設定時にインスタンス化されます。したがって、VNF をインスタンス化するには、VNF 設定を NSO にロードする必要があります。VNF には、SOL006 VNFD への参照が含まれます。また、Openstack テナントネットワークや IP アドレスなどの VIM アーティファクトへの参照もあります。VNF の YANG 定義の詳細については、付録 A：VNF の YANG の定義を参照してください。

VNF のインスタンス化には、次のようにさまざまなコンポーネントが関係します。

- TOSCA VNF パッケージとしてパッケージ化された ETSI SOL001 VNFD テンプレート
- VNF パッケージと同じ名前または ID を持つ ETSI SOL006 VNFD
- NSO 独自の VNF インスタンス

モビリティ機能パックには、いくつかのサンプル VNF パッケージが付属しており、対応する SOL006 VNFD も含まれています。これらのサンプルはベースとして使用できますが、展開に合わせた追加のカスタマイズが必要です。以下に、VNF の設定例を示します。

```
{
   "nfv-vnf-lcm:nfv-vnf":[
```

```
{
   "network-function-type":"VPC-SI",
   "name":"test026",
   "vnfd":"VPC-SI-2P-IMAGE-BOOT",
   "instantiation-level":"default",
   "deployment-flavor":"default",
   "mgmt-user-name":"admin",
   "mgmt-password":"Csco@123",
   "host-name":"vpc-si",
   "domain-name":"cisco.com",
   "ntp-server":"209.165.201.1",
   "name-server":"209.165.201.2",
   "location":{
      "vim":{
         "name":"openstack",
         "project":"test",
         "zone-id":"nova"
      },
      "vnfm":"esc-etsi"
   },
   "network":[
      {
         "type":"VIM_NETWORK_MANAGEMENT",
         "extent":"external",
         "name":"test-mgmt",
         "subnet-name":"test-mgmt-subnet"
      },
      {
         "type":"VIM_NETWORK_ORCHESTRATION",
         "extent":"external",
         "name":"test-orch",
         "subnet-name":"test-orch-subnet"
      },
      {
         "type":"VIM_NETWORK_SERVICE_1",
         "extent":"external",
         "name":"service1",
         "subnet-name":"service1"
      },
      {
         "type":"VIM_NETWORK_SERVICE_2",
         "extent":"external",
         "name":"service2",
         "subnet-name":"service2"
      }
   ],
   "unit":[
      {
         "type":"VPC-SI",
         "image":"core-si-21.23",
         "flavor":"core-si",
         "connection-point":[
            {
               "name":"nic0",
               "ip-address":[
                  {
                     "id":0,
                     "fixed-address":[
                        "209.165.201.3"
                     ]
                  }
               ],
               "security-group":[
                  "default"
```

```
                        ],
                        "network-type":"VIM_NETWORK_ORCHESTRATION"
                     },
                     {
                        "name":"nic1",
                        "ip-address":[
                           {
                              "id":0,
                              "fixed-address":[
                                 "209.165.201.4"
                              ]
                           }
                        ],
                        "security-group":[
                           "default"
                        ],
                        "network-type":"VIM_NETWORK_MANAGEMENT"
                     },
                     {
                        "name":"nic2",
                        "ip-address":[
                           {
                              "id":0,
                              "fixed-address":[
                                 "209.165.201.5"
                              ]
                           }
                        ],
                        "security-group":[
                           "default"
                        ],
                        "network-type":"VIM_NETWORK_SERVICE_1"
                     },
                     {
                        "name":"nic3",
                        "ip-address":[
                           {
                              "id":0,
                              "fixed-address":[
                                 "209.165.201.6"
                              ]
                           }
                        ],
                        "security-group":[
                           "default"
                        ],
                        "network-type":"VIM_NETWORK_SERVICE_2"
                     }
                  ]
               }
            ],
            "extra-parameters":[
               {
                  "name":"BOOTUP_TIME",
                  "value":"100"
               },
               {
                  "name":"LICENSE_KEY",
                  "value":"\"VER=1|DOI=1624646484|DOE=1640457684|ISS=3|NUM=212017|
CMT=SWIFT_License|LSG=5000000|LEC=10000000|LGT=5000000|FIS=Y|FR4=Y|FTC=Y|FSR=Y|
FPM=Y|FID=Y|FI6=Y|FLI=Y|FFA=Y|FCA=Y|FTP=Y|FTA=Y|FDR=Y|FDC=Y|FGR=Y|FAA=Y|FDQ=Y|
FEL=Y|BEP=Y|FAI=Y|FCP=Y|LCF=5000000|LPP=5000000|LSF=5000000|FLS=Y|FSG=Y|
LGW=5000000|HIL=XT2|LSB=5000000|LMM=5000000|FIB=Y|FND=Y|FAP=Y|FRE=Y|FHE=Y|
FUO=Y|FUR=Y|FOP=Y|FRB=Y|FCF=Y|FVO=Y|FST=Y|FSI=Y|FRV=Y|F6D=Y|F13=Y|FIM=Y|
```

```
FLP=Y|FSE=Y|FMF=Y|FEE=Y|FHH=Y|FIT=Y|FSB=Y|FDS=Y|LSE=5000000|FLR=Y|FLG=Y|
FMC=Y|FOC=Y|FOS=Y|FIR=Y|FNE=Y|FGD=Y|LIP=5000000|FOE=Y|FAU=Y|FEG=Y|FL2=Y|
FSH=Y|FLF=Y|FSP=Y|FNI=Y|FCI=Y|FME=Y|FCN=Y|FUB=Y|FSF=Y|FGO=Y|FPE=Y|FWI=Y|
FAC=Y|FIE=Y|FSM=Y|FAG=Y|FNQ=Y|FEW=Y|FAR=Y|FOX=Y|FPW=Y|FAM=Y|FGX=Y|FWT=Y|
FUA=Y|LDT=5000000|LEX=5000000|LVL=5000000|LQP=5000000|LMP=5000000|
LCU=10000000|LUU=10000000|FXS=Y|FLC=Y|FRT=Y|FSX=Y|FBS=Y|FRD=Y|FXM=Y|
LTO=10000000|FNS=Y|LNS=5000000|SIG=MC0CFBge/
0TZha2Ta7c1L5CLOL2tgDIDAhUAhIKwZxXEJJpr9Xk5buNyzZStrNM\""
            }
         ]
      }
   ]
}
```

以下に、RCM VNF のインスタンス化の例をもう 1 例示します。

```
{
    "nfv-vnf-lcm:nfv-vnf": [
      {
        "network-function-type": "RCM",
        "name": "RCM-ahhashem-sol003-78",
        "vnfd": "RCM-IMAGE-BOOT",
        "instantiation-level": "default",
        "deployment-flavor": "default",
        "mgmt-user-name": "luser",
        "mgmt-password": "$8$40/jVMTHJY+Jrd7mZiwqdrKEIz6Kc5Pt2Qvnwi0/65g=;”
        "host-name": "rcm",
        "domain-name": "cisco.com",
        "ntp-server": "209.165.201.1",
        "name-server": "209.165.201.1",
        "location": {
          "vim": {
            "name": "openstack",
            "project": "ahhashem",
            "zone-id": "nova"
          },
          "vnfm": "esc-etsi"
        },
        "network": [
          {
            "type": "VIM_NETWORK_MANAGEMENT",
            "name": "ahhashem-mgmt",
            "extent": "external",
            "subnet-name": "ahhashem-mgmt-subnet"
          },
          {
            "type": "VIM_NETWORK_ORCHESTRATION",
            "name": "ahhashem-orch",
            "extent": "external",
            "subnet-name": "ahhashem-orch-subnet"
          },
          {
            "type": "VIM_NETWORK_SERVICE_1",
            "name": "service1",
            "extent": "external",
            "subnet-name": "service1"
          } ,
          {
            "type": "VIM_NETWORK_SERVICE_2",
            "name": "service2",
            "extent": "external",
            "subnet-name": "service2"
          }
        ],
        "unit": [
```

```
    {
      "type": "RCM",
      "image": "core-rcm-21.23",
      "flavor": "mkal-rcm-hugepages",
      "connection-point": [
        {
          "name": "nic0",
          "ip-address":{
              "id": 1,
              "fixed-address": ["209.165.201.7"]
          },
          "security-group": ["default"],
          "network-type": "VIM_NETWORK_ORCHESTRATION"
        },
        {
          "name": "nic1",
          "ip-address":{
              "id": 1,
              "fixed-address":["209.165.201.8"]
          } ,
          "security-group": ["default"],
          "network-type": "VIM_NETWORK_MANAGEMENT"
        },
        {
          "name": "nic2",
          "ip-address": {
              "id": 1,
              "fixed-address": ["209.165.201.9"]
          },
          "security-group": ["default"],
          "network-type": "VIM_NETWORK_SERVICE_1"
        } ,
        {
          "name": "nic3",
          "ip-address": {
              "id": 1,
              "fixed-address": ["209.165.201.10"]
          },
          "security-group": ["default"],
          "network-type": "VIM_NETWORK_SERVICE_2"
        }

      ]
    }
  ],
  "extra-parameters": [
    {
      "name": "VIM_VM_NAME",
      "value": "RCM-ahhashem-sol003-78"
    },
    {
      "name": "HOST_NAME",
      "value": "rcm"
    },
      {
        "name": "NIC0_TYPE",
        "value": "virtual"
      },
      {
        "name": "NIC1_TYPE",
        "value": "virtual"
      },
      {
        "name": "NIC2_TYPE",
```

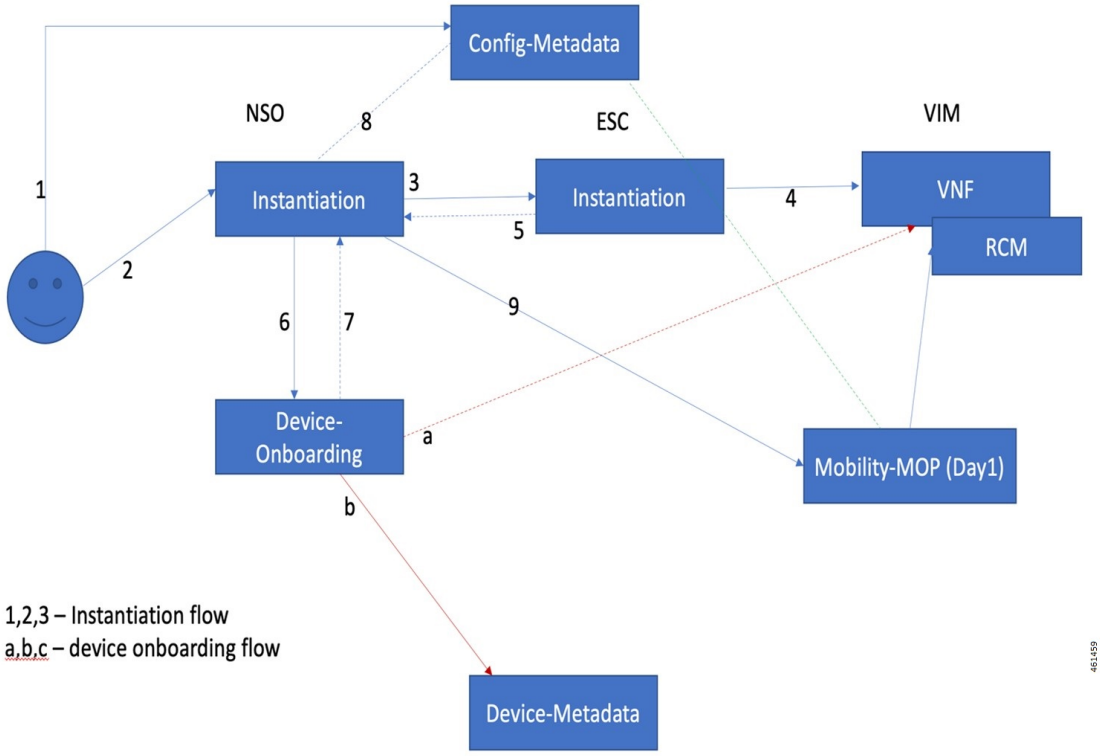
```
                "value": "direct"
              },
              {
                "name": "NIC3_TYPE",
                "value": "direct"
              },
              {
                "name": "MGMT_USER_NAME",
                "value": "luser"
              },
              {
                "name": "MGMT_PASSWORD_ROUND4096",
                "value": "$6$rounds=4096$P2wdTbEBO0LHmHi$OwbVEIarMbt
Qxbu5Us5kW0n0MOWp3QN9eVRX7WjvLm4xTJvFpl6vHez3XkKm39XJJ7dGRRIsZqXfcZRjQBA7E."
              },
              {
                  "name": "SERVICE_INTERFACE_IP_1",
                  "value": "209.165.201.9"
              },
              {
                  "name": "SERVICE_INTERFACE_IP_2",
                  "value": "209.165.201.11"
              },
              {
                "name": "NTP_SERVER",
                "value": ["209.165.201.12","209.165.201.13","209.165.201.14"]
              }

          ]
        }
      ]
    }
```

VNF のインスタンス化 - コンポーネントのインタラクションとフロー

次の図は、エンドツーエンドのインスタンス化の自動化について、その全体のフローを示しています。

図 *5 : VNF* のインスタンス化のインタラクション



手順の詳細：

1. ネットワークオペレータは、名前、タイプ、ダイナミック属性、構成ファイルなど、VNF のインスタンス化に必要なすべての詳細を把握しています。構成ファイルを NSO ファイルシステムに配置し、自動化のための NSO 設定 DB に詳細を登録します。

   このステップには、以下のタスクが含まれます。

   - ネットワークオペレータは、構成ファイルを NSO ファイルシステムに Secure Copy（SCP）でコピーします。このコピー先は NFS であるか、NSO HA 環境で複製されている必要があります。
   - すべての属性値ペア、ダイナミック置換値、Day-0.5、または Day-1 設定を登録します。
   - 構成ファイルの検証を有効にし、テスト支援デバイスの詳細を入力します。
   - 再検証フラグが「true」に設定されている場合、設定メタデータアクションがすべての構成ファイルを内部的に検証します。検証が行われない場合は、設定の適用中に失敗します。

2. ネットワークオペレータは、すべての詳細を含む VNF のインスタンス化用ペイロードを準備します。次に、ペイロードを呼び出してインスタンスを作成します。基本的な検証が行われ、命令が処理されます。

このステップには、以下のタスクが含まれます。

- 命令を呼び出す前に、パスワードの長さ、イメージ、フレーバー、ネットワークの存在などの入力情報を OpenStack で検証します。

3. NSO が内部で命令を処理し、ESC VNF のインスタンス化命令を準備します。

   このステップには、以下のタスクが含まれます。

   - サービスの NSO フットプリントを作成します。
   - CSAR を検証します。
   - SOL3/SOL4 入力を使用して ESC VNF のインスタンス化命令を呼び出します。
   - ESC 通知（ETSI と NETCONF の両方）のリッスンを開始します。

4. ESC が SOL3/SOL4 の入力検証を実行し、VIM で命令を作成します。

   このステップには、以下のタスクが含まれます。

   - ESC が VNF のインスタンス化を呼び出します。
   - VNF の呼び出しが成功すると、VNF をモニターするモノモニターが作成されます。
   - ETSI および NETCONF 通知を通じて更新を NSO に返します（成功、失敗いずれの場合も）。

5. ESC は、ETSI または NETCONF 通知を通じて、進行状況に関する定期的な更新を NSO に返します。

   このステップには、以下のタスクが含まれます。

   - ESC は、進行状況に関する ETSI および NETCONF 通知を継続的に送信します。
   - ETSI 通知は、展開～初期化、処理、および完了通知で構成されます。
   - NETCONF 通知は、VM ステータスに関するより詳細な情報を提供します。
   - 失敗すると、適切なエラーメッセージが表示されます。

6. ESC から VNF のインスタンス化完了メッセージを受信すると、NSO は NSO デバイスとしてオンボーディングします。

   このステップには、以下のタスクが含まれます。

   - インスタンス化ロジックが入力ペイロードから詳細を取得し、デバイスのオンボーディングロジックを呼び出します。
   - NSO がデバイスから fetch-ssh-host-key を実行します。
   - NSO が接続チェックを実行します。
   - NSO が sync-from を実行します。
   - NSO がデバイスで「show version」などの事後チェックコマンドを実行します。

- NSO がデバイスを NSO デバイスツリーに追加します。

7. NSO のインスタンス化ロジックは、デバイスの追加が完了するまで待機します。

   このステップには、以下のタスクが含まれます。

   - NSO がデバイスのオンボーディングプロセスが完了したかどうかを確認します。
   - デバイスのオンボーディングが失敗した場合、NSO は実行を停止します。

8. NSO のインスタンス化ロジックは、事前に入力された設定メタデータを読み取り、プッシュされる設定を解釈します。

   このステップには、以下のタスクが含まれます。

   - 事前に入力された設定メタデータを読み取り、device-name に基づいて Day-0.5 または Day-1 構成ファイルを解釈します（デバイス名は VNF 名に基づきます）。
   - RCM ベースの N:M スキームの場合、Day-0.5 がプッシュされます。
   - 1:1 の場合、Day-1 がプッシュされます。
   - 不足している情報がある場合は、インスタンス化が完了し、処理が停止します。

9. NSO は、設定メタデータから構成ファイルを取得し、モビリティ MOP 入力フォーマットを作成し、設定のプッシュ用 MOP を呼び出します。

   このステップには、以下のタスクが含まれます。

   - モビリティ MOP を呼び出し、task-id を取得します。
   - 定期的に task-id のステータスを確認します。
   - 1:1 の CP または UP ペア（MOP 経由）の場合、デバイスのフラッシュに設定を永続的に保存します。
   - 完了ステータスが vnf-status 元帳で更新されます。

VNF のインスタンス化ステータスの確認

vnf-status コマンドを定期的に使用して、VNF インスタンス化のステータスを確認できます。

失敗、処理中、または完了に関連するメッセージはすべて、ステータスメッセージに追加されます。

```
show vnf-status instances vnf-instance-name
INSTANCE ID TIMESTAMP TYPE OPERATION STATUS STATUS MESSAGE
------------------------------------------------------------------------
<VNF-Name> <Time-Stamp> <type> <function> <status> <message-if-any>
```

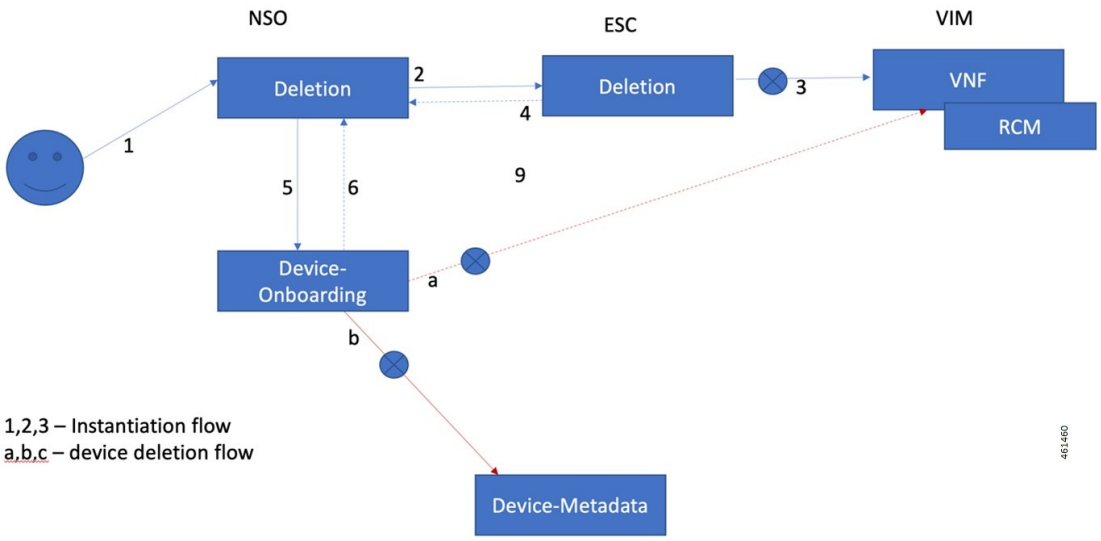
VNF ダッシュボード

VNF のインスタンス化手順と VNF の現在のステータスは、NSO ベースのダッシュボードに表示されます。

VNF の削除

次のフロー図は、エンドツーエンドの削除の自動化に関する完全なフローを示しています。

図 *6 : VNF* 削除の連携動作



詳細な手順：

1. ネットワークオペレータが、実行中または失敗状態になっている既存インスタンスのデコミッションまたは削除を決定します。

   このステップには、以下のタスクが含まれます。

   - ネットワークオペレータが、削除タイプを含む VNF 名を提供します。
   - NSO が VNF の存在を検証します。

2. NSO が VNF インスタンスのステータスを確認し、NSO 側で失敗したインスタンスがある場合、ESC を呼び出して VIM から削除するか、ロールバックを実行します。

   このステップには、以下のタスクが含まれます。

   - NSO が、ESC にインスタンスをプッシュするか、ロールバックを実行する（NSO 内でインスタンスが失敗した場合）かを決定します。
   - NSO が ESC への非同期要求を行い、通知が来るまで待機します。

3. ESC がクリーンアップを実行し、VNF モニターを削除します。
4. ESC が NSO への ETSI/NETCONF 通知を生成します。
5. NSO が ESC 通知を処理し、次の処理を実行します。

- インスタンスを削除するためにデバイス オンボーディング パッケージを呼び出します。
- 「nfv-vnf-inventory」からエントリを削除します。

6. デバイス オンボーディング パッケージによって NSO からデバイスが削除され、VNF 元帳でステータスが更新されます。

VNF の削除ステータスの確認

vnf-status コマンドを使用して、VNF の削除ステータスを確認できます。

失敗、処理中、または完了に関連するメッセージがあれば、ステータスメッセージに追加されます。

```
show vnf-status instances vnf-instance-name
INSTANCE ID TIMESTAMP TYPE OPERATION STATUS STATUS MESSAGE
------------------------------------------------------------------------
<VNF-Name> <Time-Stamp> <type> <function> <status> <message-if-any>
```

設定メタデータの削除

これは手動の手順であり、NSOアクションを使用して設定メタデータを削除する必要があります。設定メタデータを保持しても影響はありません。

NSO ファイルシステムの構成ファイルの削除

NSO ファイルシステムから構成ファイルを手動で削除する必要があります。構成ファイルのデータを保持しても影響はありません。

自動化プロセス：VNF の展開、オンボーディング、および設定のプッシュ

自動化プロセスには、次のセクションが含まれます。

入力ペイロードを使用した VNF のインスタンス化

必要な変更を加えた後、入力ペイロードを使用してインスタンス化要求を送信します。VNFインスタンス化の自動化プロセスが開始されます。

入力ペイロードのサンプルについては、VNF のインスタンス化の項を参照してください。

NSO でのデバイスとしての VNF のオンボーディング

インスタンス化が成功すると、VNF は NSO のデバイスとしてオンボーディングされます。デバイス名は VNF 名と同じになります。

VPC デバイスへの P2P モジュールのインストール

「device-type」が VPC で、configurable-parameters の「p2p-required」が「true」に設定され、「p2p-soFile-path」が定義されている場合、P2P ファイルをデバイスのフラッシュディレクトリにコピーしてから P2P モジュールをアップグレードします。

P2P モジュールがデバイスにインストールされます。

オンボーディングされたデバイスへの設定のプッシュ

次に、自動設定のプッシュ中に使用される静的パラメータを示します。

- operation-type：Commit
- mop-type：Common
- save-config-permanently：デフォルトは false で、デバイスタイプが「vpc」の場合は true に設定されます。

構成ファイルを使用した設定のプッシュが完了すると、タスク ID が生成されます。タスク ID を使用して設定のプッシュのステータスをチェックし、ステータスに基づいて元帳エントリが更新されます。

（注） NSO は RCM で設定の監査を実行しません。NSO は設定のプッシュ中に RCM が再起動した場合、再起動の完了時に設定を再プッシュしないため、設定を手動で再プッシュする必要があります。NSO は、設定のプッシュの失敗についてオペレータに警告します。RCM に正常にプッシュされた設定は、その RCM を再起動しても保持されます。

付録 A：VNF の YANG の定義

ここでは、VNF の YANG 定義の例を示します。

```
module nfv-vnf-lcm {
  namespace "http://com/cisco/cx/servicepack/nfv/vnflcm";
  prefix nfv-vnf-lcm;

  import ietf-inet-types { prefix inet; }
  import tailf-common { prefix tailf; }
  import tailf-ncs { prefix ncs; }
  import nfv-common { prefix nfv-common; }
  import tailf-kicker { prefix kicker; }
  include nfv-vnf-lcm-nano {
    revision-date 2020-02-14;
  }

  organization "Cisco-AS";

  contact "Cisco AS";

  description "Generic NFV VNF LCM service package";
```

```
revision 2020-10-22 {
  description "Active Inventory and LCM Auto/on-demand heal support";
}

revision 2020-07-01 {
  description "Re-branded per new naming convention";
}

revision 2020-02-14 {
  description "First version, ready for testing";
}

notification vnf-lcm {
  description "Notification about Network Function Operation";
  uses nfv-common:network-function-notification;
}

notification vnf-alarm {
  description "VNF alarms";
  uses nfv-common:vnf-alarm;
}

container nfv-vnf-inventory {
  tailf:info "CDB model to persist the VNFs, associated project, VIM and the
      VM details";
  config false;
  tailf:cdb-oper {
    tailf:persistent true;
  }

  list vnf {
    tailf:info "VNFs with associated VMs and status";
    key name;
    leaf name {
      tailf:info "VNF Name";
      type string;
    }
    leaf vnfd {
      type string;
      tailf:info "Associated VNFD name";
    }
    leaf project {
      type string;
      tailf:info "Associated vim tenant/project";
    }
    leaf vim {
      type string;
      tailf:info "Associated VIM";
    }
    leaf status {
      type string;
      tailf:info "Overall VNF status";
    }
    list vm {
      tailf:info "Associated VMs and the status";
      key name;
      leaf name {
        type string;
        tailf:info "VM name";
      }
      leaf type {
        type string;
        tailf:info "VM Type";
      }
```

```
      leaf flavor {
        type string;
        tailf:info "VIM flavor that is used to deploy the VM";
      }
      leaf host {
        type string;
        tailf:info "Compute host where the VM has been deployed";
      }
      list connection-point {
        key nic-id;
        leaf nic-id {
          type uint8;
          tailf:info "NIC id of the connection point";
        }
        leaf ip-address {
          type inet:ip-address;
          tailf:info "IP address of the connection point";
        }
      }
      leaf status {
        type string;
        tailf:info "VM status";
      }
    }

    leaf netconf-notification-done {
        tailf:hidden nfv-internal;
        type empty;
    }
  }
}

list nfv-vnf {
  description "Generic RFS model for VNF LCM";

  key "network-function-type name";

  leaf network-function-type {
    tailf:info "virtual network function type";
    type enumeration {
      enum "VPC-SI";
      enum "VPC-DI";
      enum "CSR1KV";
      enum "GENERIC";
      enum "VCU";
      enum "VDU";
      enum "EMS";
      enum "RCM";
    }
  }

  leaf name {
    tailf:info "Unique service id";
    type string;
  }

  leaf vnfd {
    mandatory true;
    type string;
    tailf:info "VNFD to use for this type of Network Function that has to be
          onboarded on the target VIM.";
  }

  uses ncs:service-data;
```

```
  ncs:servicepoint nfv-vnf-lcm;
  uses ncs:nano-plan-data;

  tailf:action heal {
    tailf:info "Heal VNF";
    tailf:actionpoint nfv-lcm-heal-ap;
    input {
    }
    output {
      uses nfv-common:standard-action-response;
    }
  }

 tailf:action start {
           tailf:info "Start VNF";
           tailf:actionpoint nfv-lcm-start-ap;
           input {
           }
           output {
               uses nfv-common:standard-action-response;
           }
       }

 tailf:action stop {
           tailf:info "Stop VNF";
           tailf:actionpoint nfv-lcm-stop-ap;
           input {
           }
           output {
               uses nfv-common:standard-action-response;
           }
       }

tailf:action scale {
           tailf:info "Scale-In VNF";
           tailf:actionpoint nfv-lcm-scale-ap;
           input {
               leaf scale-type {
                   mandatory true;
                   tailf:info "SCALE IN or OUT";
                   type enumeration {
                       enum "OUT";
                       enum "IN";
                   }
               }

               leaf no-of-instances {
                   tailf:info "Number of scale IN or OUT instances. Default is 1";
                   type uint32;
                   default 1;
               }

               leaf vdu-type {
                   mandatory true;
                   tailf:info "vdu-type as CF/SF/VPC-SI etc";
                   type string;
               }

           }
           output {
               uses nfv-common:standard-action-response;
           }
       }
```

```
    tailf:action retry {
          tailf:info "Stop VNF";
          tailf:actionpoint nfv-lcm-retry-ap;
          input {
          }
          output {
              uses nfv-common:standard-action-response;
          }
      }

  leaf instantiation-level {
    type string;
    default "default";
    tailf:info "Instantiation level defined in VNFD to use. This will determine
          the number of VMs/VDUs to be deployed.";
  }

  leaf deployment-flavor {
    type string;
    default "default";
    tailf:info "Deployment flavor defined in the VNFD to use. Describes a specific
          deployment version of a VNF with specific requirements for capacity
          and performance.";
  }
  leaf mgmt-user-name {
    type nfv-common:identifier;
    description " Management login username specific to this VNF. Default values
          can be configured per VNF type.";
  }

  leaf mgmt-password {
    tailf:suppress-echo "true";
    type tailf:aes-cfb-128-encrypted-string;
    description "Management login password specific to this VNF.";
  }

  leaf host-name {
    type inet:domain-name;
    description "Hostname to use to communicate with this network function";
  }

  leaf domain-name {
    type inet:domain-name;
    description "Domain name used to construct Fully Qualified Domain Name by
          concatenating with VM hostname: <hostname>.<domain>";
  }

  leaf ntp-server {
    description "NTP server to use for VNFs deployed in this data center";
    type inet:host;
  }

  leaf name-server {
    type inet:ip-address;
    description "Name server";
  }

  container location {
    container vim {
      leaf name {
        description "NFVI this Network Function is deployed on.";
        type leafref {
          path "/ncs:devices/ncs:device/ncs:name";
        }
```

```
      //must "/ncs:devices/ncs:device[ncs:name=current()]/ncs:platform/ncs:name
      //          = 'Openstack'" {
      //  error-message "Please select Openstack devices only";
      //}
    }
    leaf project {
      type nfv-common:identifier;
      description "VIM project used to instantiate VNFs";
      mandatory true;
    }
    leaf zone-id {
      type string;
      default "nova";
      description "VIM zone id";
    }
    //TODO might need to support user domain and project domain
  }
  leaf vnfm {
    mandatory true;
    type leafref {
      path "/ncs:devices/ncs:device/ncs:name";
    }
    //must "/ncs:devices/ncs:device[ncs:name=current()]/ncs:platform/ncs:name
    //        = 'ETSI SOL'" {
    //  error-message "Please select ETSI-SOL VNFM devices only";
    //}
    description "ESC VNFM onboarded";
  }
}
list network {
  key type;
  leaf type {
    type nfv-common:identifier;
  }
  leaf name {
    type nfv-common:identifier;
    mandatory true;
  }
  leaf extent {
    type nfv-common:network-extent;
  }
  leaf subnet-name {
    when "../extent='external'";
    type nfv-common:identifier;
    mandatory true;
  }
}

list unit {
  description "Virtual Deployment Unit, a single VM.";
  key type;

  leaf type {
    description "VDU type as defined in the VNFD of this Network Function.";
    type nfv-common:identifier;
  }
  leaf image {
    type string;
    description " Image to use for this type of Network Function. Must have been
            be onboarded on the target VIM.";
  }
  leaf flavor {
    mandatory true;
    type string;
```

```
          description " Flavor to use for this type of Network Function. Must have been
                  onboarded on the target VIM.";
        }
        list storage-volume {
          key id;
          description "Out of band Storage volumes to use for this network function";
          leaf id {
            type string;
          }
          leaf volume-name {
            type string;
            description "Storage Volume to use for this type of Network function";
          }
        }
        list connection-point {
          key name;
          description " Network connection point such as a network interface card, as
                  defined in the descriptor.";
          leaf name {
            mandatory true;
            type nfv-common:identifier;
          }

          list ip-address {
              key id;
              ordered-by user;
              leaf id {
               type uint8;
                  tailf:info "IP Address ID for connection points";
              }
              leaf-list fixed-address {
                  ordered-by user;
                  description " IP address(es) to assign this network interface for both
scaled and non-scaled VNF's. Both IPv4 and
                              IPv6 is possible to allow for dual-stack cases if this VNF
requires
                              it for Internet access.";
                  type inet:ip-address;
               }
          }

          list vip {
            key address;
            ordered-by user;
            description " Virtual IP address(es) to assign this network interface. Both
                      IPv4 and IPv6 is possible to allow for dual-stack cases if this
                      VNF requires it for Internet access. Setting this will populate
                      allowed-address-pair list in the CVIM";

            leaf address {
              type inet:ip-address;
            }
            leaf netmask {
              type inet:ip-address;
              mandatory true;
            }
          }
          leaf-list security-group {
            type nfv-common:identifier;
            description "Security group(s) to apply to this network interface.";
          }
          leaf network-type {
            type leafref {
              path "../../../network/type";
```

```
        }
        description "Network used for this connection-point.";
      }
    }
  }
  list extra-parameters {
    description "VNF instance specific additional parameters defined in the VNFD.
         This will override the values configured in the VNFD";
    key name;
    leaf name {
      type string {
        pattern "[A-Za-z0-9_]+";
      }
    }
    leaf value {
      type string;
    }
  }
}

 list nfv-retry-vnfs {
      tailf:info "Retry VNF's to tweak the notifications";
      config false;
      tailf:cdb-oper {
          tailf:persistent true;
      }
      tailf:hidden nfv-internal;

      key name;
      leaf name {
          tailf:info "VNF Name";
          type string;
      }
  }
}
```

付録 B：モビリティ機能パック（MFP）の一般的なアップグレード手順

この付録では、次の手続きについて説明します。

NSO 5.7.5.1-MFP 3.4.1 から NSO 5.8.10-MFP 3.4.2 へのアップグレード

NSO 5.7.5.1-MFP 3.4.1 から NSO 5.8.10-MFP 3.4.2 にアップグレードするには、次の手順を使用します。この場合、MFP バージョンのアップグレードと同時に NSO バージョンのアップグレードを行います。

1. NSO 5.8.10 インストール bin ファイルを */tmp* フォルダにコピーし、NSO をバージョン 5.8.10 にアップグレードします。
2. */opt/ncs* で新しい NSO バージョン 5.8.10 へのシンボリックリンクを設定します。

3. MFP 3.4.2 のパッケージと NED をコピーし、*/var/opt/ncs/packages* フォルダの中身と置き換えます。
4. **start-with-package-reload** オプションを使って NSO を再起動します。これで、MFP 3.4.1 が 3.4.2 にアップグレードされ、NSO が NSO 5.7.5.1 から 5.8.10 にアップグレードされます。

以下に、NSO 5.7.5.1-MFP 3.4.1 から NSO 5.8.10-MFP 3.4.2 にアップグレードするための詳しい手順を示します。

（注） アップグレードが完了していない場合は、後でリカバリが必要になった場合に備えて、必ずバックアップを作成しておくことを推奨します。

データをバックアップするには、次の設定を使用します。

```
$ sudo su
# source /etc/profile.d/ncs.sh
# /etc/init.d/ncs stop
# ncs-backup
# exit
$
```

1. NSO 5.7.5.1 で MFP 3.4.1 を実行します。

```
root@test-nso:/var/opt/ncs# ncs --version
5.7.5.1

root@ncs# show packages package package-version
                         PACKAGE
NAME                     VERSION
----------------------------------
cisco-etsi-nfvo          4.7.2
cisco-rcm-nc-1.6         1.6
cisco-staros-cli-5.43    5.43.4
esc                      5.7.0.73
etsi-sol003-gen-1.13     1.13.16
mobility-common          3.4.1
mobility-rcm-subscriber  3.4.1
mop-automation           3.4.1
mop-common               3.4.1
nfv-common               3.4.1
nfv-device-onboarding    3.4.1
nfv-vim                  3.4.1
nfv-vnf-lcm              3.4.1
openstack-cos-gen-4.2    4.2.26

root@ncs# show packages package oper-status
packages package cisco-etsi-nfvo
 oper-status up
packages package cisco-rcm-nc-1.6
 oper-status up
packages package cisco-staros-cli-5.43
 oper-status up
packages package esc
 oper-status up
packages package etsi-sol003-gen-1.13
 oper-status up
packages package mobility-common
 oper-status up
```

```
packages package mobility-rcm-subscriber
 oper-status up
packages package mop-automation
 oper-status up
packages package mop-common
 oper-status up
packages package nfv-common
 oper-status up
packages package nfv-device-onboarding
 oper-status up
packages package nfv-vim
 oper-status up
packages package nfv-vnf-lcm
 oper-status up
packages package openstack-cos-gen-4.2
 oper-status up
root@ncs#

root@ncs# show devices list
NAME         ADDRESS        DESCRIPTION  NED ID                  ADMIN STATE
-----------------------------------------------------------------------------
esc-etsi     64.1.0.6       -            etsi-sol003-gen-1.13    unlocked
esc-netconf  64.1.0.6       -            esc                     unlocked
openstack    10.225.202.49  -            openstack-cos-gen-4.2   unlocked
root@ncs#
```

2. NSO 5.7.5.1 で MFP 3.4.1 を使用して、テスト用 VNF VPC-SI デバイスをインスタンス化します。

```
root@ncs#
System message at 2023-10-09 07:52:05...
Commit performed by ubuntu via http using rest.
root@ncs#
System message at 2023-10-09 07:52:05...
Commit performed by ubuntu via http using rest.
root@ncs#
System message at 2023-10-09 07:52:07...
Commit performed by ubuntu via http using rest.
root@ncs#
System message at 2023-10-09 07:52:07...
Commit performed by ubuntu via http using rest.
root@ncs#
System message at 2023-10-09 07:52:08...
Commit performed by ubuntu via http using rest.

root@ncs# show vnf-status instances S1-Test-00001 | tab
                                        FUNCTION

INSTANCE ID    TIMESTAMP                TYPE      OPERATION       STATUS      STATUS
 MESSAGE
-------------------------------------------------------------------------------------------
S1-Test-00001  2023-10-09 07:50:55.198  VPC-SI    deploy          init        init

               2023-10-09 07:51:38.595  VPC-SI    deploy          processing
processing
               2023-10-09 07:52:01.639  VPC-SI    deploy          processing
processing
             2023-10-09 07:52:03.997  VPC-SI    deploy         completed   completed

               2023-10-09 07:53:43.293  -         init            success     Device
 Onboarding initialized
               2023-10-09 07:53:43.874  -         fetch-ssh-keys  success
fetch-ssh-keys was successful
             2023-10-09 07:53:45.285  -         connect         success     connect
```

```
 was successful
             2023-10-09 07:53:46.785  -        sync-from      success     sync-from
 was successful
              2023-10-09 07:53:46.964  -         ready          success     Device
 Successfully onboarded
              2023-10-09 07:54:13.305  -         config-read     success     Config
 MetaData is empty or null

root@ncs# show devices list
NAME           ADDRESS        DESCRIPTION  NED ID                 ADMIN STATE
-------------------------------------------------------------------------------
S1-Test-00001  64.1.0.110     -            cisco-staros-cli-5.43  unlocked
esc-etsi       64.1.0.6       -            etsi-sol003-gen-1.13   unlocked
esc-netconf    64.1.0.6       -            esc                    unlocked
openstack      10.225.202.49  -            openstack-cos-gen-4.2  unlocked
root@ncs#
```

3. NSO 5.8.10 インストール bin ファイルを */tmp* フォルダにコピーし、NSO をバージョン 5.8.10 にアップグレードします。

```
root@test-nso:/var/opt/ncs# cd /tmp
root@test-nso:/tmp# ls -lrt
total 397840
-rwxrwxrwx 1 ubuntu   ubuntu   203071802 Nov 18  2022
nso-5.7.5.1.linux.x86_64.installer.bin
drwx------ 3 root     root          4096 Sep 10 03:02
systemd-private-d7c0f02148d447358a1b6b5995f1f339-systemd-resolved.service-O5tL4V
drwx------ 3 root     root          4096 Sep 10 03:02
systemd-private-d7c0f02148d447358a1b6b5995f1f339-systemd-logind.service-Uj4bic
drwx------ 3 root     root          4096 Sep 10 03:02 snap.lxd
drwx------ 2 ubuntu   ubuntu        4096 Sep 12 09:45 ssh-WxVBdtyvgGzB
drwx------ 2 ubuntu   ubuntu        4096 Sep 12 19:28 ssh-kRFako4TgqJp
drwx------ 2 ubuntu   ubuntu        4096 Sep 12 20:25 ssh-wyrZqTmiA4o1
drwx------ 2 ubuntu   ubuntu        4096 Sep 12 20:50 ssh-a10wclKRgSP2
-rwxrwxrwx 1 ubuntu   ubuntu   204258218 Sep 13 05:38
nso-5.8.10.linux.x86_64.installer.bin
drwx------ 2 ubuntu   ubuntu        4096 Sep 13 12:21 ssh-ReWAFnmi3qSl
drwx------ 2 ubuntu   ubuntu        4096 Sep 13 12:54 ssh-dn16O8f1nkaz
drwx------ 2 ubuntu   ubuntu        4096 Sep 20 05:49 ssh-DtgyHvctQ5S0
drwxr-xr-x 2 root     root          4096 Oct  9 07:01 hsperfdata_root
drwxr-xr-x 2 nsoadmin nsoadmin      4096 Oct  9 07:01 hsperfdata_nsoadmin

root@test-nso:/tmp# sh ./nso-5.8.10.linux.x86_64.installer.bin --system-install
--install-dir /opt/ncs --config-dir /etc/ncs --run-dir /var/opt/ncs --log-dir
/var/log/ncs --run-as-user nsoadmin --non-interactive
INFO  Using temporary directory /tmp/ncs_installer.63734 to stage NCS installation
bundle
INFO  Using /opt/ncs/ncs-5.8.10 for static files
INFO  Doing install for running as user nsoadmin
INFO  Unpacked ncs-5.8.10 in /opt/ncs/ncs-5.8.10
INFO  Found and unpacked corresponding DOCUMENTATION_PACKAGE
INFO  Found and unpacked corresponding EXAMPLE_PACKAGE
INFO  Found and unpacked corresponding JAVA_PACKAGE
INFO  Generating default SSH hostkey (this may take some time)
INFO  SSH hostkey generated
INFO  Generating self-signed certificates for HTTPS
INFO  Environment set-up generated in /opt/ncs/ncs-5.8.10/ncsrc
INFO  NSO installation script finished
INFO  Found and unpacked corresponding NETSIM_PACKAGE
cp: cannot stat '/sbin/arping': No such file or directory
WARN  Failed to copy /sbin/arping command - capability not set
INFO  Found ncs.crypto_keys, not migrating
INFO  The following files have been installed with elevated privileges:
  /opt/ncs/ncs-5.8.10/lib/ncs/lib/core/pam/priv/epam: setuid-root
```

```
  /opt/ncs/ncs-5.8.10/lib/ncs/erts/bin/ncs.smp: capability cap_net_bind_service
  /opt/ncs/ncs-5.8.10/lib/ncs/bin/ip: capability cap_net_admin

INFO  NCS installation complete

root@test-nso:/tmp# /etc/init.d/ncs stop
Stopping ncs: .

root@test-nso:/tmp# cd /opt/ncs
root@test-nso:/opt/ncs# ls -lrt
total 24
drwxr-xr-x 17 root     root 4096 Oct  9 06:41 ncs-5.7.5.1
-rw-r--r--  1 root     root    9 Oct  9 06:41 user
-rw-r--r--  1 root     root   80 Oct  9 06:41 installdirs
lrwxrwxrwx  1 root     root   11 Oct  9 06:41 current -> ncs-5.7.5.1
drwxr-xr-x  2 nsoadmin root 4096 Oct  9 06:41 packages
drwxr-xr-x  2 nsoadmin root 4096 Oct  9 06:41 downloads
drwxr-xr-x 17 root     root 4096 Oct  9 09:43 ncs-5.8.10

Set the current NSO to version 5.8.10 using symbolic link

root@test-nso:/opt/ncs# rm -f current
root@test-nso:/opt/ncs# ln -s ncs-5.8.10 current

root@test-nso:/opt/ncs# ls -lrt
total 24
drwxr-xr-x 17 root     root 4096 Oct  9 06:41 ncs-5.7.5.1
-rw-r--r--  1 root     root    9 Oct  9 06:41 user
-rw-r--r--  1 root     root   80 Oct  9 06:41 installdirs
drwxr-xr-x  2 nsoadmin root 4096 Oct  9 06:41 packages
drwxr-xr-x  2 nsoadmin root 4096 Oct  9 06:41 downloads
drwxr-xr-x 17 root     root 4096 Oct  9 09:43 ncs-5.8.10
lrwxrwxrwx  1 root     root   10 Oct  9 09:44 current -> ncs-5.8.10
```

4. */var/opt/ncs/packages* フォルダにある以前の MFP 3.4.1 のパッケージと NED を参照し、新しい MFP 3.4.2 のパッケージと NED に置き換えます。

```
root@test-nso:/opt/ncs# cd /var/opt/ncs/packages/
root@test-nso:/var/opt/ncs/packages# ls -lrt
total 20104
-rw-rw-r-- 1 ubuntu ubuntu 2191794 Jan 25  2023 ncs-5.7.5.1-cisco-rcm-nc-1.6.tar.gz
-rw-rw-r-- 1 ubuntu ubuntu 2694132 Jan 25  2023 ncs-5.7.3-etsi-sol003-1.13.16.tar.gz
-rw-rw-r-- 1 ubuntu ubuntu  655190 Jan 25  2023 ncs-5.7.2.1-esc-5.7.0.73.tar.gz
-rw-rw-r-- 1 ubuntu ubuntu 2685815 Jan 25  2023
ncs-5.7.2.1-cisco-etsi-nfvo-4.7.2.tar.gz
-rw-rw-r-- 1 ubuntu ubuntu 2702317 Jan 25  2023 ncs-5.7.2-openstack-cos-4.2.26.tar.gz
-rw-rw-r-- 1 ubuntu ubuntu 9606799 Jan 25  2023 ncs-5.7.2-cisco-staros-5.43.4.tar.gz
-rwxrwxrwx 1 ubuntu ubuntu     435 Jan 25  2023 compile-all-packages.sh
-rwxrwxrwx 1 ubuntu ubuntu     275 Jan 25  2023 Ha-Mop.sh
drwxrwxr-x 6 ubuntu ubuntu    4096 Oct  9 06:55 nfv-common
drwxrwxr-x 7 ubuntu ubuntu    4096 Oct  9 06:56 nfv-device-onboarding
drwxrwxr-x 8 ubuntu ubuntu    4096 Oct  9 06:56 nfv-vim
drwxrwxr-x 9 ubuntu ubuntu    4096 Oct  9 06:57 nfv-vnf-lcm
drwxrwxr-x 8 ubuntu ubuntu    4096 Oct  9 07:00 mobility-common
drwxrwxr-x 7 ubuntu ubuntu    4096 Oct  9 07:01 mop-common
drwxrwxr-x 8 ubuntu ubuntu    4096 Oct  9 07:01 mobility-mop
drwxrwxr-x 7 ubuntu ubuntu    4096 Oct  9 07:01 mobility-rcm-subscriber

root@test-nso:/var/opt/ncs/packages# rm -rf *
root@test-nso:/var/opt/ncs/packages# ls -lrt
total 0
root@test-nso:/var/opt/ncs/packages#

Copy the MFP 3.4.2 packages along with NEDS:
```

```
root@test-nso:/var/opt/ncs/packages# ls -lrt
total 26328
-rw-rw-r-- 1 ubuntu ubuntu 2191794 Sep 25 05:40 ncs-5.7.5.1-cisco-rcm-nc-1.6.tar.gz
-rw-rw-r-- 1 ubuntu ubuntu 2694132 Sep 25 05:40 ncs-5.7.3-etsi-sol003-1.13.16.tar.gz
-rw-rw-r-- 1 ubuntu ubuntu  655190 Sep 25 05:40 ncs-5.7.2.1-esc-5.7.0.73.tar.gz
-rw-rw-r-- 1 ubuntu ubuntu 2685815 Sep 25 05:40
ncs-5.7.2.1-cisco-etsi-nfvo-4.7.2.tar.gz
-rw-rw-r-- 1 ubuntu ubuntu 2702317 Sep 25 05:40 ncs-5.7.2-openstack-cos-4.2.26.tar.gz
-rw-rw-r-- 1 ubuntu ubuntu 9606799 Sep 25 05:40 ncs-5.7.2-cisco-staros-5.43.4.tar.gz
-rw-rw-r-- 1 ubuntu ubuntu  824211 Sep 25 05:40 nfv-vnf-lcm.tar.gz
-rw-rw-r-- 1 ubuntu ubuntu  307054 Sep 25 05:40 nfv-vim.tar.gz
-rw-rw-r-- 1 ubuntu ubuntu  197449 Sep 25 05:40 nfv-device-onboarding.tar.gz
-rw-rw-r-- 1 ubuntu ubuntu   59217 Sep 25 05:40 nfv-common.tar.gz
-rw-rw-r-- 1 ubuntu ubuntu 3905393 Sep 25 05:40 mop-common.tar.gz
-rw-rw-r-- 1 ubuntu ubuntu  113829 Sep 25 05:40 mobility-rcm-subscriber.tar.gz
-rw-rw-r-- 1 ubuntu ubuntu  243790 Sep 25 05:40 mobility-mop.tar.gz
-rw-rw-r-- 1 ubuntu ubuntu  746045 Sep 25 05:40 mobility-common.tar.gz
```

5. **start-with-package-reload** オプションを使って NSO を再起動します。これで、NSO 5.8.10 で MFP が 3.4.1 から 3.4.2 にアップグレードされます。

```
root@test-nso:/var/opt/ncs/packages# source /etc/profile.d/ncs.sh

root@test-nso:/var/opt/ncs/packages# /etc/init.d/ncs start-with-package-reload
Starting ncs: .

root@test-nso:/var/opt/ncs/packages# ncs --version
5.8.10

root@test-nso:/var/opt/ncs/packages# ncs_cli -C

root connected from 127.0.0.1 using console on test-nso
root@ncs# show packages package package-version
                        PACKAGE
NAME                    VERSION
-----------------------------------
cisco-etsi-nfvo         4.7.2
cisco-rcm-nc-1.6        1.6
cisco-staros-cli-5.43   5.43.4
esc                     5.7.0.73
etsi-sol003-gen-1.13    1.13.16
mobility-common         3.4.2
mobility-rcm-subscriber 3.4.2
mop-automation          3.4.2
mop-common              3.4.2
nfv-common              3.4.2
nfv-device-onboarding   3.4.2
nfv-vim                 3.4.2
nfv-vnf-lcm             3.4.2
openstack-cos-gen-4.2   4.2.26

root@ncs# show packages package oper-status
packages package cisco-etsi-nfvo
 oper-status up
packages package cisco-rcm-nc-1.6
 oper-status up
packages package cisco-staros-cli-5.43
 oper-status up
packages package esc
 oper-status up
packages package etsi-sol003-gen-1.13
 oper-status up
packages package mobility-common
```

```
 oper-status up
packages package mobility-rcm-subscriber
 oper-status up
packages package mop-automation
 oper-status up
packages package mop-common
 oper-status up
packages package nfv-common
 oper-status up
packages package nfv-device-onboarding
 oper-status up
packages package nfv-vim
 oper-status up
packages package nfv-vnf-lcm
 oper-status up
packages package openstack-cos-gen-4.2
 oper-status up
root@ncs#

root@ncs# show devices list
NAME           ADDRESS        DESCRIPTION  NED ID                 ADMIN STATE
-------------------------------------------------------------------------------
S1-Test-00001  64.1.0.110     -            cisco-staros-cli-5.43  unlocked
esc-etsi       64.1.0.6       -            etsi-sol003-gen-1.13   unlocked
esc-netconf    64.1.0.6       -            esc                    unlocked
openstack      10.225.202.49  -            openstack-cos-gen-4.2  unlocked
```

6. mop-automation メソッドにより、NSO 5.8.10 で MFP 3.4.2 を使用して、以前の MFP 3.4.1 および NSO 5.7.5.1 でインスタンス化されたテスト用 VNF VPC-SI デバイスに設定をプッシュします。

```
root@test-nso:/var/opt/ncs# cat day1config.cfg
config
port ethernet 1/1
description test-description-1/1-by-mop18oct
no shutdown
exit

root@ncs# mobility-mop:action mop-automation generate-dry-run true operation-type
commit mop-type common mop-file-name { file-name day1config.cfg order 1
target-devices-list { target-device-name S1-Test-00001 } } save-config-permanently
true
task-id 036f5e94-364b-4d5f-a95e-4663fe5ed08a
time-stamp 2023-10-09T10:18:19+0000
time-zone Coordinated Universal Time
root@ncs#

root@ncs# mobility-mop:action mop-automation-status task-id
036f5e94-364b-4d5f-a95e-4663fe5ed08a
task-id 036f5e94-364b-4d5f-a95e-4663fe5ed08a
task-status COMPLETED
start-date 2023-10-09T10:18:19+0000
end-date 2023-10-09T10:18:23+0000
time-zone Coordinated Universal Time
operation-type commit
action-type save
devices-list {
    device-name S1-Test-00001
    device-status COMPLETED
    start-date 2023-10-09T10:18:19+0000
    end-date 2023-10-09T10:18:23+0000
    device-state common
    files {
        file-name day1config.cfg
```

```
        order 1
        dry-run-mop
/var/opt/ncs//036f5e94-364b-4d5f-a95e-4663fe5ed08a/S1-Test-00001/day1config_commit_2023-10-09T101819+0000.cfg

        rollback-mop
/var/opt/ncs//036f5e94-364b-4d5f-a95e-4663fe5ed08a/S1-Test-00001/day1config_rollback_commit_2023-10-09T101819+0000.cfg

        commit-queue-status completed
        commit-queue-id 1696846701998
    }
}

root@test-nso:/var/opt/ncs# cat
/var/opt/ncs//036f5e94-364b-4d5f-a95e-4663fe5ed08a/S1-Test-00001/day1configroot@test-nso:/var/opt/ncs#
 cat
/var/opt/ncs//036f5e94-364b-4d5f-a95e-4663fe5ed08a/S1-Test-00001/day1config_commit_2023-10-09T101819+0000.cfg
config
port ethernet 1/1
 description test-description-1/1-by-mop18oct
exit
end

root@test-nso:/var/opt/ncs# cat
/var/opt/ncs//036f5e94-364b-4d5f-a95e-4663fe5ed08a/S1-Test-00001/day1configroot@test-nso:/var/opt/ncs#
 cat
/var/opt/ncs//036f5e94-364b-4d5f-a95e-4663fe5ed08a/S1-Test-00001/day1config_rollback_commit_2023-10-09T101819+0000.cfg
config
port ethernet 1/1
 no description
exit
end
```

NSO バージョンを変更せずに MFP 3.4.1 から MFP 3.4.2 にアップグレード

NSO バージョンを変更せずに MFP 3.4.1 から MFP 3.4.2 にアップグレードするには、次の手順を使用します。

1. MFP 3.4.2 のパッケージと NED をコピーし、*/var/opt/ncs/packages* フォルダの中身と置き換えます。
2. **ncs_cli** でパッケージのリロードを実行して、MFP バージョンが 3.4.2 にアップグレードされていることを確認します。NSO を再起動します。

付録 C：P2P 優先順位のアップグレード

mobility-library アクションコマンドを使用して P2P 優先順位をアップグレードするには、次の手順を実行します。

1. P2P ファイルの配置とパス設定を含む事前チェックを実行し、その後に VNF のインスタンス化を行います。

```
[cloud-user@qwerty ncs]$ ncs --version
5.8.10

[cloud-user@qwerty ncs]$ ncs_cli -C

User cloud-user last logged in 2023-09-20T03:23:18.655123+00:00, to qwerty, from
```

```
10.65.51.122 using cli-ssh
cloud-user connected from 10.65.51.122 using ssh on qwerty
cloud-user@ncs# show packages package package-version
                         PACKAGE
NAME                     VERSION
-----------------------------------
cisco-etsi-nfvo          4.7.2
cisco-rcm-nc-1.6         1.6
cisco-staros-cli-5.43    5.43.4
esc                      5.7.0.73
etsi-sol003-gen-1.13     1.13.16
mobility-common          3.4.2
mobility-rcm-subscriber  3.4.2
mop-automation           3.4.2
mop-common               3.4.2
nfv-common               3.4.2
nfv-device-onboarding    3.4.2
nfv-vim                  3.4.2
nfv-vnf-lcm              3.4.2
openstack-cos-gen-4.2    4.2.26

cloud-user@ncs# show packages package oper-status
packages package cisco-etsi-nfvo
 oper-status up
packages package cisco-rcm-nc-1.6
 oper-status up
packages package cisco-staros-cli-5.43
 oper-status up
packages package esc
 oper-status up
packages package etsi-sol003-gen-1.13
 oper-status up
packages package mobility-common
 oper-status up
packages package mobility-rcm-subscriber
 oper-status up
packages package mop-automation
 oper-status up
packages package mop-common
 oper-status up
packages package nfv-common
 oper-status up
packages package nfv-device-onboarding
 oper-status up
packages package nfv-vim
 oper-status up
packages package nfv-vnf-lcm
 oper-status up
packages package openstack-cos-gen-4.2
 oper-status up

[cloud-user@qwerty ncs]$ ls -lrt
total 4740
drwxrwxrwx.  2 nsoadmin   root             6 Sep  5 03:36 scripts
drwxrwxrwx.  2 nsoadmin   root             6 Sep  5 03:36 streams
drwxrwxrwx.  2 nsoadmin   root             6 Sep  5 03:36 backups
-rwxrwxrwx.  1 nsoadmin   root          1513 Sep  5 03:36 INSTALLATION-LOG
drwxrwxrwx.  3 nsoadmin   nsoadmin        22 Sep  5 03:37 target
drwxrwxrwx.  7 cloud-user cloud-user     204 Sep  5 03:56 vnfpackages
-rwxrwxrwx.  1 root       root            87 Sep  5 06:26 day1config.cfg
-rwxrwxrwx.  1 root       root            31 Sep  5 06:47 rcm-day1config.cfg

-rwxrwxrwx.  1 root       root       4253395 Sep  8 03:19
patch_libp2p-2.69.0.1534.so.tgz
```

```
-rwxrwxrwx.  1 cloud-user cloud-user     142 Sep 10 14:11 daynconfig.cfg

drwxrwxrwx. 10 nsoadmin   root          4096 Sep 18 02:20 packages
-rwxrwxrwx.  1 nsoadmin   nsoadmin       333 Sep 18 02:31 storedstate

drwxrwxrwx.  2 nsoadmin   root            98 Sep 18 08:59 cdb

drwxrwxrwx.  2 nsoadmin   root         20480 Sep 19 23:23 rollbacks
drwxrwxrwx.  5 nsoadmin   root          4096 Sep 19 23:26 state

cloud-user@ncs# config
Entering configuration mode terminal
cloud-user@ncs(config)# configurable-parameters p2p-required true
cloud-user@ncs(config)# configurable-parameters p2p-soFile-path
/var/opt/ncs/patch_libp2p-2.69.0.1534.so.tgz
cloud-user@ncs(config)# commit
Commit complete.
cloud-user@ncs(config)# exit

cloud-user@ncs# show vnf-status instances UP-Test001-p2p
                                          FUNCTION

INSTANCE ID     TIMESTAMP               TYPE      OPERATION       STATUS      STATUS
 MESSAGE
-------------------------------------------------------------------------------------------------
UP-Test001-p2p  2023-09-19 23:29:42.335  VPC-SI    deploy          init        init

                2023-09-19 23:30:19.377  VPC-SI    deploy          processing
processing
                2023-09-19 23:30:47.948  VPC-SI    deploy          processing
processing
                2023-09-19 23:30:49.269  VPC-SI    deploy          completed
completed
               2023-09-19 23:31:55.555  -         init            success     Device
 Onboarding initialized
                2023-09-19 23:31:56.061  -         fetch-ssh-keys  success
fetch-ssh-keys was successful
              2023-09-19 23:31:57.005  -         connect         success     connect
 was successful
                2023-09-19 23:31:58.353  -         sync-from       success
sync-from was successful
               2023-09-19 23:31:58.523  -         ready           success     Device
 Successfully onboarded
               2023-09-19 23:37:29.386  -         config-read     success     Config
 MetaData is empty or null

[cloud-user@qwerty ncs]$ ssh admin@64.1.0.96
The authenticity of host '64.1.0.96 (64.1.0.96)' can't be established.
RSA key fingerprint is SHA256:TKCql7DQvty52OHp8WzGt0lYKiloAtEmMt1xAMQ23a0.
Are you sure you want to continue connecting (yes/no/[fingerprint])? Csco@123
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '64.1.0.96' (RSA) to the list of known hosts.
Cisco Systems QvPC-SI Intelligent Mobile Gateway
admin@64.1.0.96's password:
Last login: Tue Sep 19 23:32:04 -0400 2023 on pts/1 from 64.1.0.7.

No entry for terminal type "xterm-256color";
using dumb terminal settings.

[local]vpc-si# show module p2p verbose
Module p2p
   Priority  card   version  loaded    location      update/rollback time    status
```

```
       99      1  2.69.1534    2/2    /var/opt/lib   Tue Sep 19 23:32:35 2023
success                                <<<<<<< p2p priority starting from 99 instead
 of 10
        X      1  1.161.656    0/2            /lib                   (never)     N/A

[local]vpc-si#
[local]vpc-si#
[local]vpc-si# exit
Connection to 64.1.0.96 closed.
[cloud-user@qwerty ncs]$
[cloud-user@qwerty ncs]$
[cloud-user@qwerty ncs]$ ncs_cli -C

User cloud-user last logged in 2023-09-20T03:30:52.813071+00:00, to qwerty, from
10.65.51.122 using rest-http
cloud-user connected from 10.65.51.122 using ssh on qwerty
cloud-user@ncs#
cloud-user@ncs#
```

2. P2P 優先順位の実際のアップグレードには、**mobility-library** アクションコマンドを使用します。

```
cloud-user@ncs# mobility-library configure-library library-name p2p device-list {
device-name UP-Test001-p2p }
status success
message Configured Successfully

cloud-user@ncs#
cloud-user@ncs#
cloud-user@ncs# exit
[cloud-user@qwerty ncs]$ ssh admin@64.1.0.96
Cisco Systems QvPC-SI Intelligent Mobile Gateway
admin@64.1.0.96's password:
Last login: Tue Sep 19 23:41:37 -0400 2023 on pts/1 from 64.1.0.7.

No entry for terminal type "xterm-256color";
using dumb terminal settings.

[local]vpc-si# show module p2p verbose
Module p2p
   Priority  card   version  loaded    location      update/rollback time     status
>      98      1  2.69.1534    2/2    /var/opt/lib   Tue Sep 19 23:41:39 2023
success
*      99      1  2.69.1534    2/2    /var/opt/lib                   (never)     N/A
        X      1  1.161.656    0/2            /lib                   (never)     N/A

>  current module priority is 98
*  some modules have not unloaded from the p2p application and are still in use

[local]vpc-si#
```

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。