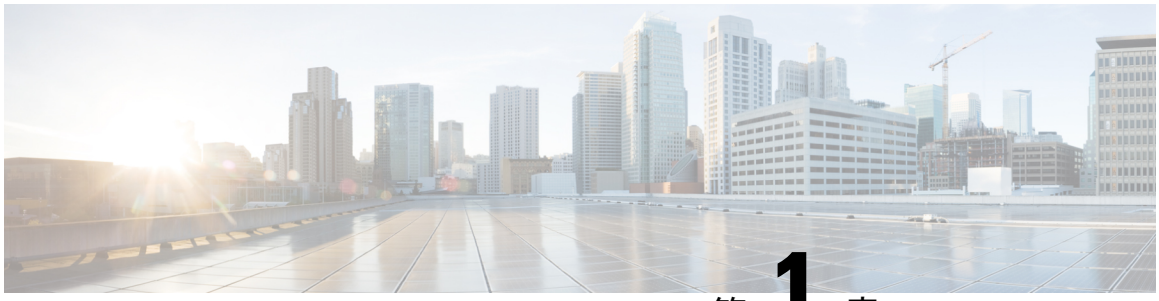
Cisco APIC レイヤ 4 ～レイヤ 7 サービス リリース 4.0(1) 導入ガイド

初版：2018 年 10 月 24 日

シスコシステムズ合同会社
〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー
http://www.cisco.com/jp
お問い合わせ先：シスコ コンタクトセンター
0120-092-255 （フリーコール、携帯・PHS含む）
電話受付時間：平日 10:00～12:00、13:00～17:00
http://www.cisco.com/jp/go/contactcenter/

目 次



第 1 章

新機能および変更された機能に関する情報

新機能および変更された機能に関する情報

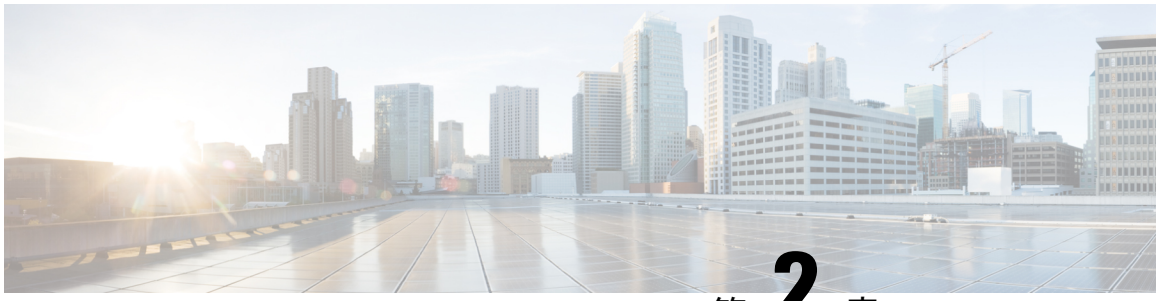
次の表は、この最新リリースまでのガイドでの主な変更点の概要を示したものです。ただし、このリリースまでのこのガイドの変更点や新機能の中には一部、この表に記載されていないものもあります。

表 *1 : Cisco APIC* リリース *4.0(1)* の新機能と動作変更

機能	説明	参照先
リモート リーフ設定の PBR トラッキング	リモート リーフ設定で PBR トラッキングを機能させるには、システム レベルのグローバル GIPo を有効にする必要があります。	GUI を使用してリモート リーフのグローバル GIPo を構成する （111 ページ）を参照してください。
PBR の復元力のあるハッシュ	PBR の復元力のあるハッシュがリモート リーフ設定でサポートされるようになりました。	ポリシー ベースのリダイレクトの修復性のあるハッシュ （93 ページ）を参照してください。
サービス グラフでの EPG 内契約のサポート	シングル ノード、ワンアーム PBR、およびシングル ノード コピー サービスの EPG 内契約を使用したサービス グラフの作成がサポートされるようになりました。	GUI を使用したエンドポイント グループへのサービス グラフ テンプレートの適用 （243 ページ）を参照してください。

新機能および変更された機能に関する情報

機能	説明	参照先
サービス グラフでの優先グループのサポート	サービス グラフによって作成された EPG を優先契約グループに含めることができます。	GUI を使用したエンドポイント グループへのサービス グラフ テンプレートの適用 （243 ページ）を参照してください。
多層アプリケーション プロファイル ウィザードの L3 宛先（VIP）	コネクタの L3 トラフィックを多層アプリケーション プロファイル ウィザードで終端できるようになりました。	GUI を使用した多階層アプリケーション プロファイルの作成 （221 ページ）を参照してください。
多層サービス グラフ ウィザードのコンシューマおよびプロバイダー L3 アウト	多層アプリケーション クイック スタートで、コンシューマおよびプロバイダー L3 アウトとしてデバイスを指定できるようになりました。	GUI を使用した多階層アプリケーション プロファイルの作成 （221 ページ）を参照してください。
サービス VM オーケストレーション	サービス仮想マシン（VM）オーケストレーションは、Cisco Application Policy Infrastructure Controller（Cisco APIC）でのサービス VM の作成と管理を容易にするポリシーベースの機能です。	サービス VM オーケストレーション （25 ページ）を参照してください



第 2 章

概要

アプリケーションセントリックインフラストラクチャのレイヤ 4 ～ 7 サービスの導入について

従来の方法を使用する場合、サービスをネットワークに挿入すると、手間がかかって複雑な VLAN（レイヤ 2）または仮想ルーティングおよび転送（VRF）インスタンス（レイヤ 3）スティッチングを、ネットワーク要素およびサービス アプライアンスの間で実行する必要があります。この従来のモデルでは、アプリケーションに対する新規サービスを配備するのに数日から数週間かかります。サービスには柔軟性が少なく、操作エラーはより頻繁に発生し、トラブルシューティングはより困難です。アプリケーションが使用されなくなる場合、ファイアウォール ルールなどのサービス デバイス設定の削除は困難になります。ロードに基づいたサービスのスケール アウト/スケール ダウンを実行することもできません。

VLAN および仮想ルーティングおよび転送（VRF）スティッチングは従来のサービス挿入モデルによってサポートされますが、Application Policy Infrastructure Controller（APIC）はポリシー制御の中心点として機能する一方でサービス挿入を自動化できます。APIC ポリシーは、ネットワーク ファブリックとサービス アプライアンスの両方を管理します。APIC は、トラフィックがサービスを通って流れるように、ネットワークを自動的に設定できます。APIC は、アプリケーション要件に従ってサービスを自動的に設定することもでき、それにより組織はサービス挿入を自動化し、従来のサービス挿入の複雑な技術の管理に伴う課題を排除できます。

開始する前に次の APIC オブジェクトを設定する必要があります。

- レイヤ 4 ～ 7 サービスを提供/消費するテナント
- テナントのネットワーク外部のレイヤ 3
- 最低でも 1 個のブリッジ ドメイン

- アプリケーション プロファイル
- 物理ドメインまたは VMM ドメイン

  VMM ドメインについて、VMM ドメインのクレデンシャルを設定し、vCenter/vShield コントローラ プロファイルを設定します。
- カプセル化ブロック範囲を持つ VLAN プール
- 最低でも 1 個の契約
- 最低でも 1 個の EPG

次のタスクを実行し、レイヤ 4 ～ 7 サービスを展開します。

1. **デバイス パッケージ** をインポートします。

   プロバイダーの管理者のみがデバイス パッケージをインポートできます。
2. デバイスおよび論理インターフェイスを登録します。

   また、このタスクでは、具象デバイスと具象インターフェイスを登録し、具象デバイス パラメータを設定します。
3. **論理デバイス** を作成します。
4. デバイス パラメータを設定します。
5. オプション。ASA ファイアウォール サービスを設定する場合は、デバイスのトランキングを有効にします。
6. **デバイス選択ポリシー** を設定します。
7. **サービス グラフ テンプレート** を設定します。
   1. アプリケーション プロファイルからのデフォルトのサービス グラフ テンプレートのパラメータを選択します。
   2. 必要に応じた追加のサービス グラフ テンプレートのパラメータを設定します。
8. 契約のサービス グラフ テンプレートを添付します。
9. 必要な場合は、追加の設定パラメータを設定します。

（注） 仮想アプライアンスは、VLAN を使用して VMware ESX サーバとリーフ ノード間にトランスポートとして導入できますが、ハイパーバイザとして導入する場合は VMware ESX のみが使用できます。

GUI を使用したレイヤ 4 ～ レイヤ 7 サービスの設定

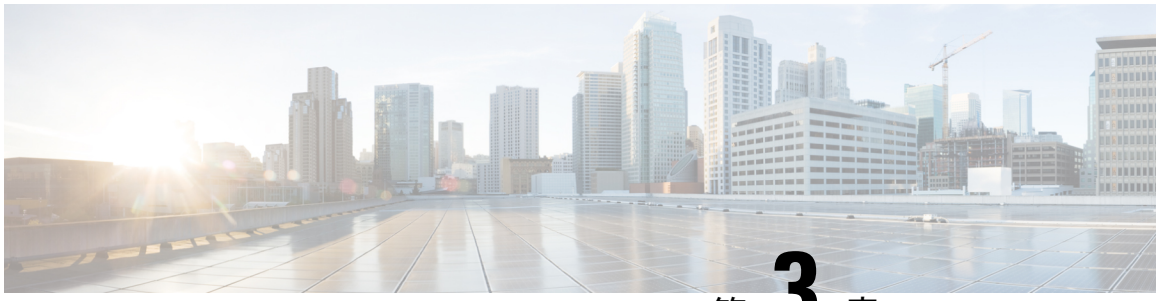
GUI を使用して、Application Policy Infrastructure Controller（APIC）にレイヤ 4 ～レイヤ 7 サービスを設定できます。

サービスおよびサービス グラフ テンプレートを設定する手順については、GUI の使用方法 （237 ページ）を参照してください。

サービス グラフ テンプレートについて

Cisco Application Centric Infrastructure（ACI）では、特定のタイプのファイアウォールとそれに続く特定のモデルおよびバージョンのロードバランサといった一連のメタデバイスを定義できます。これは、サービス グラフ テンプレート呼ばれ、また、抽象グラフとも呼ばれます。抽象サービス グラフ テンプレートがコントラクトによって参照されると、サービス グラフ テンプレートはファブリック内に存在するファイアウォールやロードバランサなどの具象デバイスにマッピングすることでインスタンス化されます。マッピングは「コンテキスト」の概念で発生します。「デバイス コンテキスト」とは、ACI がサービス グラフ テンプレートにマッピングできるファイアウォールおよびロードバランサを特定できるようにするためのマッピング設定です。もう 1 つの重要な概念は、具象デバイスのクラスタを表す「論理デバイス」です。サービス グラフ テンプレートのレンダリングは、コントラクトによって定義されるパスに挿入可能な適切な論理デバイスの識別に基づいています。

ACI はサービスをアプリケーションの重要部分と見なします。必要とされるすべてのサービスが、Cisco Application Policy Infrastructure Controller（APIC）から ACI ファブリックでインスタンス化されるサービス グラフとして扱われます。ユーザは、アプリケーションに対してサービスを定義し、サービス グラフテンプレートはアプリケーションが必要とする一連のネットワークまたはサービス機能を識別します。グラフを APIC に設定すると、APIC はサービス グラフ テンプレートで指定されたサービス機能要件に基づいてサービスを自動的に設定します。さらに APIC は、サービス グラフ テンプレートで指定されたサービス機能のニーズに応じてネットワークを自動的に設定しますが、これによってサービス デバイスでの変更が必要になることはありません。



第 3 章

デバイス パッケージのインポート

デバイス パッケージについて

Application Policy Infrastructure Controller（APIC）は、サービス デバイスの設定およびモニタリングにデバイス パッケージを必要とします。APIC にサービスの機能を追加するには、デバイス パッケージを使用します。デバイス パッケージは、単一クラスのサービス デバイスを管理し、デバイスとその機能に関する情報を APIC に提供します。デバイス パッケージは次の項目を含む `zip` ファイルです。

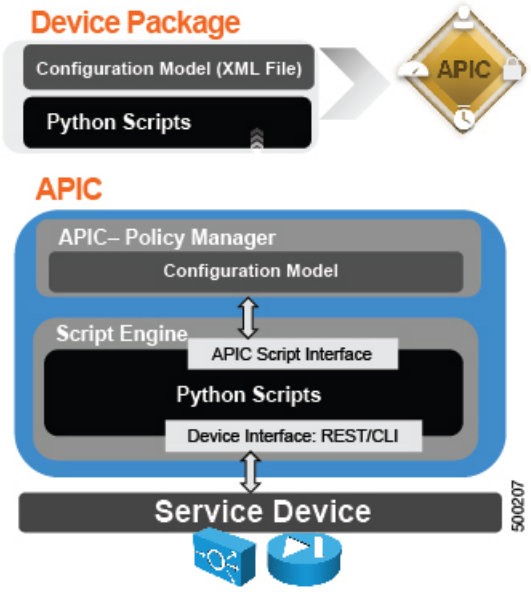
デバイス仕様	次を定義する XML ファイル：
	• デバイス プロパティ：
	  • [Model]：デバイスのモデル。
	  • [Vendor]：デバイスのベンダー。
	  • [Version]：デバイスのソフトウェア バージョン。
	• ロード バランシング、コンテンツ切り替え、および SSL 終端などの、デバイスによって提供される機能。
	• 各機能のインターフェイスおよびネットワーク接続情報。
	• デバイス設定パラメータ。
	• 各機能の設定パラメータ。

デバイス スクリプト	APIC とデバイスのやりとりに使用される Python スクリプト。APIC イベントは、デバイス スクリプトで定義した機能呼び出しにマッピングされます。デバイス パッケージには、複数のデバイス スクリプトを含めることができます。デバイス スクリプトは、REST、SSH、または、同様のメカニズムを使用して、デバイスと連携できます。
機能プロファイル	ベンダーによって指定されたデフォルト値を持つ機能パラメータ。これらのデフォルト値を使用するように機能を設定できます。
デバイス レベル設定パラメータ	デバイスに必要なパラメータを指定するコンフィギュレーション ファイル。この設定は、デバイスを使用している 1 つ以上のグラフで共有できます。

デバイス パッケージを作成できます。または、デバイス ベンダーか Cisco によって提供されるものを使用できます。

次の図では、デバイス パッケージと APIC の関係について説明します:

図 *1 :* デバイス パッケージと、*APIC*

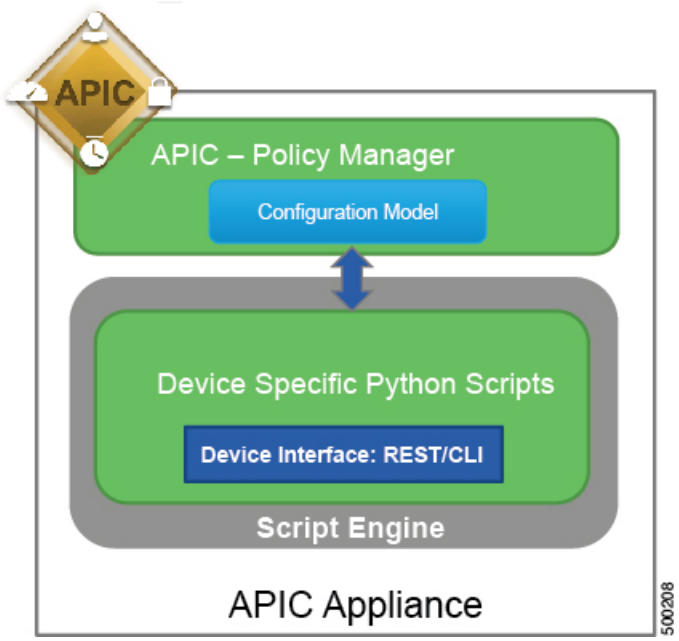


デバイスのスクリプトでの機能は、次のカテゴリに分類されます。

- デバイス/インフラストラクチャ — デバイス レベルの設定とモニタリングを行うため
- サービス イベント — デバイス上でサーバのロード バランサまたはセキュア ソケット レイヤなどの機能を設定するため
- エンドポイント/ネットワーク イベント — エンドポイントとネットワークの接続/接続解除イベントを処理するため

APIC は、デバイス パッケージで提供されたデバイス構成モデルを使用して、デバイス スクリプトに適切な構成を渡します。デバイス スクリプト ハンドラは、REST または CLI インターフェイスを使用してデバイスと連解します。

図 *2:* デバイス スクリプトがサービス デバイスと連携する方法



デバイス パッケージにより、管理者は次のサービスの管理を自動化することができます。

- デバイスの接続と切断
- エンドポイントの接続と切断
- サービス グラフのレンダリング
- ヘルス モニタリング
- アラーム、通知、ロギング
- カウンタ

デバイス パッケージとデバイス パッケージを作成する方法の詳細については、『*Cisco APIC Layer 4 to Layer 7 Device Package Development Guide*』を参照してください。

REST API を使用したデバイス パッケージのインストール

デバイス パッケージは HTTP または HTTPS POST を使用してインストールできます。

デバイス パッケージをインストールします。

- HTTP が Application Policy Infrastructure Controller（APIC）で有効になっている場合の POST の URL は次のとおりです。

  ```
  http://10.10.10.10/ppi/node/mo/.xml
  ```

- HTTPS が APIC で有効になっている場合の POST の URL は次のとおりです。

  ```
  https://10.10.10.10/ppi/node/mo/.xml
  ```

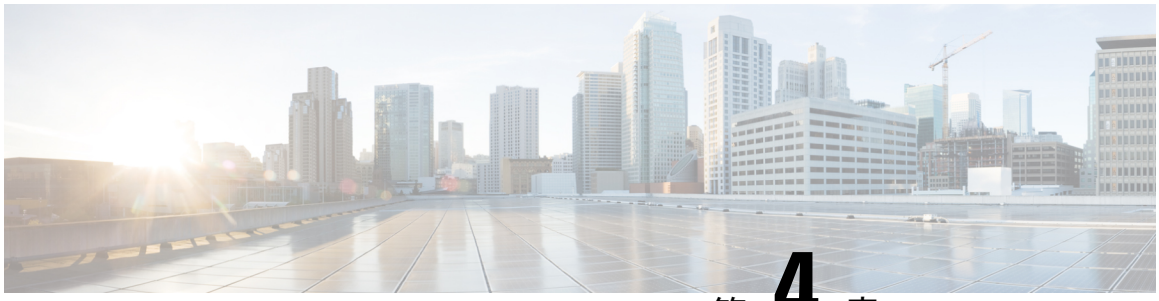
メッセージには有効なセッション Cookie が必要です。

POST の本文にはアップロードされるデバイス パッケージを含める必要があります。POST で許可されるのは 1 つのパッケージだけです。

GUI を使用したデバイス パッケージのインポート

デバイス パッケージは GUI を使用してインポートできます。

デバイス パッケージのインポート手順については、GUI の使用方法（237 ページ）を参照してください。



第 4 章

論理デバイスの定義

デバイス クラスタについて

デバイス クラスタ（別名論理デバイス）は、単一のデバイスとして機能する 1 つ以上の具象デバイスです。デバイス クラスタには、そのデバイス クラスタのインターフェイス情報を説明するクラスタ（論理）インターフェイスがあります。サービス グラフ テンプレートのレンダリング時に、機能ノードコネクタはクラスタ（論理）インターフェイスに関連付けられます。Application Policy Infrastructure Controller（APIC）は、サービス グラフ テンプレートのインスタンス化およびレンダリング時に機能ノード コネクタにネットワーク リソース（VLAN または Virtual Extensible Local Area Network（VXLAN））を割り当て、クラスタ（論理）インターフェイスにネットワーク リソースをプログラミングします。

サービス グラフ テンプレートは、管理者が定義するデバイス選択ポリシー（論理デバイス コンテキストと呼ばれます）に基づく特定のデバイスを使用します。

管理者は、アクティブ/スタンバイ モードで最大 2 つの具象デバイスをセットアップできます。

デバイス クラスタをセットアップするには、次のタスクを実行する必要があります。

1. ファブリックに具象デバイスを接続します。

2. デバイス クラスタに管理 IP アドレスを割り当てます。
3. デバイス クラスタを APIC に登録します。APIC は、デバイス パッケージに含まれるデバイス仕様を使ってデバイスを検証します。

（注） APIC は、2 つのデバイスのクラスタに IP アドレスが重複して割り当てられているかどうかを検証しません。APICは、2つのデバイスのクラスタが同じ管理IPアドレスを持っている場合、不適切なデバイスのクラスタをプロビジョニングすることがあります。デバイス クラスタで IP アドレスが重複している場合には、いずれかのデバイスの IP アドレスの設定を削除し、管理 IP アドレスの設定のためにプロビジョニングされた IP アドレスが重複していないことを確認してください。

管理対象デバイス クラスタについて

デバイス クラスタは管理対象デバイス クラスタとして設定できます。管理対象モードでは、Application Policy Infrastructure Controller（APIC）が APIC 管理者によって APIC に提供された設定を使用し、グラフのインスタンス化時にデバイスをプログラミングします。管理対象デバイス クラスタの場合、APIC がそのデバイス クラスタのデバイスを管理するためにはデバイス パッケージが必要です。

デフォルトでは、デバイス クラスタは管理対象デバイス クラスタとして設定されます。

デバイス クラスタが管理対象として設定されている場合は次の設定が必要です。

- デバイス パッケージ
- 論理デバイス（vnsLDevViP）とデバイス（CDev）の接続情報（管理 IP アドレス、クレデンシャル、およびインバンド接続情報）
- サポートされる機能タイプ（go-through、go-to）に関する情報
- コンテキスト認識に関する情報（シングル コンテキストかマルチコンテキスト）

APIC はデバイス クラスタおよびデバイスのトポロジ情報（論理インターフェイスと具象インターフェイス）を把握する必要があります。この情報は、APIC がリーフ上で適切なポートをプログラミングするために必要です。また、APIC はこの情報をトラブルシューティング ウィザードのために使用することもあります。さらに、APIC はカプセル化の割り当てに使用する `DomP` との関係も把握する必要があります。

非管理対象デバイス クラスタについて

デバイス クラスタは非管理対象デバイス クラスタとして設定できます。非管理対象デバイス クラスタの場合、Application Policy Infrastructure Controller（APIC）はサービス グラフにネットワーク リソースのみを割り当て、グラフのインスタンス化時にファブリック側のみでプログラミングします。これは、デバイス クラスタ内のデバイスをプログラミングする既存のオーケストレータまたは dev-op がすでに環境にある場合に便利です。また、サービス アプライアンス

のデバイス パッケージが使用できない場合もあります。非管理モードでは、APIC がデバイス パッケージなしでサービス デバイスと連携できます。

APIC はデバイス クラスタおよびデバイスのトポロジ情報（論理インターフェイスと具象インターフェイス）を把握する必要があります。この情報は、APIC がリーフ上で適切なポートをプログラミングするために必要です。また、APIC はこの情報をトラブルシューティング ウィザードのために使用することもあります。さらに、APIC はカプセル化の割り当てに使用する `DomP` との関係も把握する必要があります。

具象デバイスについて

具象デバイスとしては、物理デバイスまたはバーチャル デバイスがあり得ますl。具象デバイスには、具象インターフェイスがあります。具象デバイスが論理デバイスに追加されると、具象インターフェイスが論理インターフェイスにマッピングされます。サービス グラフ テンプレートのインスタンス化時に、VLAN および VXLAN は、論理インターフェイスとの関連付けに基づいた具象インターフェイス上でプログラミングされます。

トランキングの概要

レイヤ 4 ～ レイヤ 7 仮想 ASA デバイスのトランキングを有効にでき、これはトランク ポート グループを使用してエンドポイント グループのトラフィックを集約します。トランキングを使用せず、仮想サービス デバイスには各インターフェイスに 1 個の VLAN のみ所有し、最大 10 個のサービス グラフを所有できます。トランキングが有効にしている状態では、仮想サービス デバイスはサービス グラフの数を無制限に設定できます。

トランク ポート グループについての詳細は、『*Cisco ACI Virtualization Guide*』を参照してください。

トランキングは、仮想 ASA デバイスでのみサポートされます。ASA デバイス パッケージは、バージョン 1.2.7.8 以降である必要があります。

GUI を使用したレイヤ 4 ～ レイヤ 7 デバイスの作成

レイヤ 4 ～ レイヤ 7 デバイスを作成する際には、物理デバイスまたは仮想マシンのいずれかに接続することができます。接続先のタイプによって、フィールドが若干異なります。物理デバイスに接続する場合は、物理インターフェイスを指定します。仮想マシンに接続する場合は、VMM ドメイン、仮想マシン、および仮想インターフェイスを指定します。さらに、不明モデルを選択することで、接続を手動で設定することもできます。

（注） ロード バランサであるレイヤ 4 ～ レイヤ 7 デバイスを設定する場合には、context aware パラメータは使用しません。context aware パラメータには `single context` というデフォルト値がありますが、これは無視されます。

始める前に

- テナントを作成しておく必要があります。

ステップ 1 メニュー バーで、**[Tenants]** > **[All Tenants]** の順に選択します。

ステップ 2 [Work] ペインで、テナントの名前をダブルクリックします。

ステップ 3 [Navigation] ウィンドウで、**Tenant** ***tenant_name*** > **Services** > **L4-L7** > **Devices** を選択します。

ステップ 4 [Work] ペインで、**[Actions]** > **[Create L4-L7 Devices]** の順に選択します。

ステップ 5 [Create L4-L7 Devices] ダイアログボックスで、[General] セクションの次のフィールドに入力します。

名前	説明
[Managed] チェックボックス	管理対象デバイスを作成する場合はこのチェックボックスをオンにします。非管理対象デバイスを作成する場合はこのチェックボックスをオフにします。
[Name] フィールド	デバイスの名前を入力します。
[Service Type] ドロップダウン リスト	サービス タイプを選択します。
[Device Type] ボタン	デバイス タイプを選択します。
[Physical Domain] ドロップダウン リストまたは [VMM Domain] ドロップダウン リスト	物理ドメインまたは VMM ドメインを選択します。
スイッチング モード （Cisco ACI Virtual Edgeのみ）	Cisco ACI Virtual Edge仮想ドメインでは、次のモードのいずれかを選択します: • **AVE** — トラフィックは Cisco ACI Virtual Edge を通してスイッチングされます。 • **native** — トラフィックは VMware DVS を通してスイッチングされます。
View ラジオボタンを表示します。	デバイスのビューを選択します。ビューとしては、次のものがあり得ます: • **Single Node** — ただ 1 つのノード • **HA Node** — 高可用性ノード (2 つのノード) • **Cluster** — 3 以上のノード
Device Package ドロップダウン リスト	（管理対象デバイスの場合のみ）使用するベンダー提供のパッケージを選択します。

名前	説明
Model ドロップダウン リスト	（管理対象デバイスの場合のみ）デバイスのモデルを選択します。

ステップ 6 （管理対象デバイスのみ）[Connectivity] セクションで、次のフィールドに入力します。

名前	説明
[APIC to Device Management Connectivity] オプション ボタン	接続のタイプを選択します。ファブリックの外側のデバイスに接続するときは [Out-of-Band] を選択し、ファブリックを通じてデバイスに接続するときは [In-Band] を選択します。

ステップ 7 （管理対象デバイスのみ）[Credentials] セクションで、次のフィールドに入力します。

名前	説明
[User Name] フィールド	ユーザ名を入力します。
[Password] フィールド	パスワードを入力します。
[Confirm Password] フィールド	もう一度パスワードを入力します。

ステップ 8 [Device 1] セクションで、次のフィールドに入力します。

名前	説明
[Management IP Address] フィールド	（管理対象デバイスの場合のみ）接続するデバイスの管理 IP アドレスを入力します。
[Management Port] フィールドとドロップダウン リスト	（管理対象デバイスの場合のみ）管理ポートを入力するか、またはドロップダウン リストから値を選択します。
[VM] ドロップダウン リスト	（仮想デバイス タイプの場合のみ）仮想マシンを選択します。
[Chassis] ドロップダウン リスト	（管理対象デバイスの場合のみ）シャーシを選択します。

ステップ 9 [Device Interfaces] テーブルで、[+] ボタンをクリックしてインターフェイスを追加し、次のフィールドに入力します。

名前	説明
[Name] ドロップダウン リスト	インターフェイス名を選択します。
[VNIC] ドロップダウン リスト	（仮想デバイス タイプの場合のみ）vNIC を選択します。

名前	説明
[Path] ドロップダウン リスト	インターフェイスの接続先のポート、ポート チャネル、またはバーチャル ポート チャネルを選択します。

ステップ 10 [Update] をクリックします。

ステップ 11 （HA クラスタの場合のみ）各デバイスのフィールドに入力します。

ステップ 12 [Cluster] セクションのフィールドに入力します。

HA クラスタでは、クラスタのインターフェイスが、クラスタ内の両方の具体デバイスにある対応するインターフェイスにマッピングされていることを確認してください。

ステップ 13 [Next] をクリックします。

使用しているパッケージで実行可能な機能とパラメータのリストが [Device Configuration] ページに表示されます。基本パラメータが表示された [Basic] タブと、デバイス パッケージで使用可能なすべてのパラメータが表示された [All Parameters] タブが表示されます。基本パラメータは [All Parameters] に含まれています。

ステップ 14 [Features] セクションで、使用する一連の機能を選択します。
使用する特定のパッケージと選択した特定の機能に応じて、一連のパラメータは変化します。

ステップ 15 選択した機能のパラメータに対して、次のように値を指定します。

a) 変更するフィールドをダブルクリックします。
b) 表示されたフィールドに必要な情報を入力します。
c) [Update] をクリックします。

ステップ 16 [Finish] をクリックします。

NX OS スタイル CLI を使用したレイヤ 4 ～ レイヤ 7 の作成

レイヤ 4 ～ レイヤ 7 デバイスを作成するときに、物理デバイスまたは仮想マシンのいずれかに接続できます。物理デバイスに接続する場合は、物理インターフェイスを指定します。仮想マシンに接続する場合は、VMM ドメイン、仮想マシン、および仮想インターフェイスを指定します。

（注） ロード バランサであるレイヤ 4 ～ レイヤ 7 デバイスを設定する場合、[コンテキスト認識] パラメータは使用されません。[コンテキスト認識] パラメータには、無視可能な シングル コンテキスト のデフォルト値があります。

始める前に

- テナントを作成しておく必要があります。

ステップ 1 コンフィギュレーション モードを開始します。

例：

```
apic1# configure
```

ステップ 2 テナントのコンフィギュレーション モードを開始します。

```
tenant tenant_name
```

例：

```
apic1(config)# tenant t1
```

ステップ 3 レイヤ 4 ～ レイヤ 7 デバイス クラスタを追加します。

```
l4l7 cluster name cluster_name type cluster_type vlan-domain domain_name
  [function function_type] [service service_type]
```

パラメータ	説明
name	デバイス クラスタの名前。
type	デバイス クラスタのタイプ。値は次のとおりです。 • virtual • physical
vlan-domain	VLANの割り当てに使用するドメイン。このドメインは、仮想デバイスの場合はVMMドメイン、物理デバイスの場合は物理ドメインである必要があります。
switching-mode （Cisco ACI Virtual Edgeのみ）	（オプション）次のいずれかのモードを選択します。 • **AVE**：Cisco ACI Virtual Edge を通過するトラフィックのスイッチ。 • **ネイティブ**：VMware DVS を通過するトラフィックのスイッチ。これはデフォルト値です。
機能	（任意）機能タイプ。値は次のとおりです。 • go-to • go-through

パラメータ	説明
service	（任意）サービス タイプ。ADC 固有またはファイアウォール固有のアイコンおよび GUI を表示するために GUI で使用します。値は次のとおりです。 • ADC • FW • OTHERS

例：

物理デバイスの場合は、次のように入力します。

```
apic1(config-tenant)# l4l7 cluster name D1 type physical vlan-domain phys
  function go-through service ADC
```

仮想デバイスの場合は、次のように入力します。

```
apic1(config-tenant)# l4l7 cluster name ADCCluster1 type virtual vlan-domain mininet
```

ステップ 4 1 つ以上のクラスタ デバイスをデバイス クラスタに追加します。

```
cluster-device device_name [vcenter vcenter_name] [vm vm_name]
```

パラメータ	説明
vcenter	（仮想デバイスの場合のみ）仮想デバイスの仮想マシンをホストする VCenter の名前。
vm	（仮想デバイスの場合のみ）仮想デバイスの仮想マシンの名前。

例：

物理デバイスの場合は、次のように入力します。

```
apic1(config-cluster)# cluster-device C1
apic1(config-cluster)# cluster-device C2
```

仮想デバイスの場合は、次のように入力します。

```
apic1(config-cluster)# cluster-device C1 vcenter vcenter1 vm VM1
apic1(config-cluster)# cluster-device C2 vcenter vcenter1 vm VM2
```

ステップ 5 1 つ以上のクラスタ インターフェイスをデバイス クラスタに追加します。

```
cluster-interface interface_name [vlan static_encap]
```

パラメータ	説明
vlan	（仮想デバイスの場合のみ）クラスタ インターフェイスのスタティックなカプセル化。VLAN の値は、1 ～ 4094 とする必要があります。

例：

物理デバイスの場合は、次のように入力します。

```
apic1(config-cluster)# cluster-interface consumer vlan 1001
```

仮想デバイスの場合は、次のように入力します。

```
apic1(config-cluster)# cluster-interface consumer
```

ステップ 6 1 つ以上のメンバーをクラスタ インターフェイスに追加します。

```
member device device_name device-interface interface_name
```

パラメータ	説明
デバイス	**cluster-device** コマンドを使用して、このデバイスにすでに追加されている必要があるクラスタ デバイスの名前。
device-interface	クラスタ デバイス上のインターフェイスの名前。

例：

```
apic1(config-cluster-interface)# member device C1 device-interface 1.1
```

ステップ 7 メンバーにインターフェイスを追加します。

```
interface {ethernet ethernet_port | port-channel port_channel_name [fex fex_ID] |
  vpc vpc_name [fex fex_ID]} leaf leaf_ID
```

インターフェイスではなく vNIC を追加する場合は、このステップをスキップします。

パラメータ	説明
ethernet	（イーサネットまたは FEX イーサネット インターフェイスの場合のみ）クラスタ デバイスがCisco Application Centric Infrastructure（ACI）ファブリックに接続されるリーフ上のイーサネット ポート。FEX イーサネット メンバーを追加する場合は、FEX IDと FEX ポートの両方を次の形式で指定します。 *FEX_ID*/*FEX_port* 次に例を示します。 101/1/23 FEX ID は、クラスタ デバイスがファブリック エクステンダにどこで接続するかを指定します。
port-channel	（ポート チャネルまたは FEX ポート チャネル インターフェイスの場合のみ）クラスタ デバイスが ACI ファブリックに接続されるポート チャネル名。
vpc	（バーチャル ポート チャネルまたは FEX バーチャル ポート チャネル インターフェイスの場合のみ）クラスタ デバイスが ACI ファブリックに接続されるバーチャル ポート チャネル名。
fex	（ポート チャネル、FEX ポート チャネル、バーチャル ポート チャネル、または FEX バーチャル ポートの場合のみ）ポート チャネルまたはバーチャル ポート チャネルの形成に使用するスペース区切りリスト形式の FEX ID。
leaf	クラスタ デバイスがどこで接続するかのスペース区切りリスト内のリーフ ID。

例：

イーサネット インターフェイスの場合は、次のように入力します。

```
apic1(config-member)# interface ethernet 1/23 leaf 101
apic1(config-member)# exit
```

FEX イーサネット インターフェイスの場合は、次のように入力します。

```
apic1(config-member)# interface ethernet 101/1/23 leaf 101
apic1(config-member)# exit
```

ポート チャネル インターフェイスの場合は、次のように入力します。

```
apic1(config-member)# interface port-channel pc1 leaf 101
apic1(config-member)# exit
```

FEX ポート チャネル インターフェイスの場合は、次のように入力します。

```
apic1(config-member)# interface port-channel pc1 leaf 101 fex 101
apic1(config-member)# exit
```

バーチャル ポート チャネル インターフェイスの場合は、次のように入力します。

```
apic1(config-member)# interface vpc vpc1 leaf 101 102
apic1(config-member)# exit
```

FEX バーチャル ポート チャネル インターフェイスの場合は、次のように入力します。

```
apic1(config-member)# interface vpc vpc1 leaf 101 102 fex 101 102
apic1(config-member)# exit
```

ステップ 8 メンバーに vNIC を追加します。

```
vnic "vnic_name"
```

vNIC の代わりにインターフェイスを追加する場合は、前のステップを参照してください。

パラメータ	説明
vnic	クラスタ デバイスの仮想マシンの vNIC アダプタの名前。名前を二重引用符で囲みます。

例：

```
apic1(config-member)# vnic "Network adapter 2"
apic1(config-member)# exit
```

ステップ 9 デバイスの作成が完了したら、コンフィギュレーション モードを終了します。

例：

```
apic1(config-cluster-interface)# exit
apic1(config-cluster)# exit
apic1(config-tenant)# exit
apic1(config)# exit
```

GUI を使用してレイヤ 7 仮想 ASA デバイスにレイヤ 4 でのトランキングを有効化

次の手順では、GUI を使用したレイヤ 7 仮想 ASA デバイスにレイヤ 4 でのトランキングが有効にします。

始める前に

- ASA デバイスの仮想レイヤ 7 にレイヤ 4 に設定した必須。

ステップ 1 メニュー バーで、**[Tenants]** > **[All Tenants]** の順に選択します。
ステップ 2 [Work] ペインで、テナントの名前をダブルクリックします。
ステップ 3 ナビゲーション ペインで、次のように選択します。 **テナント** ***tenant_name*** > **サービス** > **L4 L7** > **デバイス** > ***device_name*** 。
ステップ 4 [Work] ウィンドウで、**Trunking Port** チェック ボックスをオンにします。
ステップ 5 [Submit] をクリックします。

REST Api を使用してレイヤ 7 仮想 ASA デバイスにレイヤ 4 でのトランキングを有効化

次の手順では、REST Api を使用して、レイヤ 7 仮想の ASA デバイスにレイヤ 4 でのトランキングを有効にする例を示します。

始める前に

- ASA デバイスの仮想レイヤ 7 にレイヤ 4 に設定した必須。

名前付きレイヤ 7 デバイスにレイヤ 4 でのトランキングを有効にする `InsiemeCluster` ：

```
<polUni>
    <fvTenant name="tenant1">
        <vnsLDevVip name="InsiemeCluster" devtype="VIRTUAL" trunking="yes">
            ...
            ...
        </vnsLDevVip>
    </fvTenant>
</polUni>
```

REST API とともにインポートされたデバイスの使用

次の REST API ではインポートされたデバイスを使用します。

```
<polUni>
  <fvTenant dn="uni/tn-tenant1" name="tenant1">
    <vnsLDevIf ldev="uni/tn-mgmt/lDevVip-ADCCluster1"/>
    <vnsLDevCtx ctrctNameOrLbl="any" graphNameOrLbl="any" nodeNameOrLbl="any">
     <vnsRsLDevCtxToLDev tDn="uni/tn-tenant1/lDevIf-[uni/tn-mgmt/lDevVip-ADCCluster1]"/>

      <vnsLIfCtx connNameOrLbl="inside">
        <vnsRsLIfCtxToLIf
tDn="uni/tn-tenant1/lDevIf-[uni/tn-mgmt/lDevVip-ADCCluster1]/lDevIfLIf-inside"/>
        <fvSubnet ip="10.10.10.10/24"/>
        <vnsRsLIfCtxToBD tDn="uni/tn-tenant1/BD-tenant1BD1"/>
      </vnsLIfCtx>
      <vnsLIfCtx connNameOrLbl="outside">
        <vnsRsLIfCtxToLIf
tDn="uni/tn-tenant1/lDevIf-[uni/tn-mgmt/lDevVip-ADCCluster1]/lDevIfLIf-outside"/>
        <fvSubnet ip="70.70.70.70/24"/>
        <vnsRsLIfCtxToBD tDn="uni/tn-tenant1/BD-tenant1BD4"/>
      </vnsLIfCtx>
    </vnsLDevCtx>
  </fvTenant>
</polUni>
```

NX-OS スタイルの CLI を使用した別のテナントからのデバイスの作成

共有サービスのシナリオでは、別のテナントからデバイスをインポートできます。

ステップ 1 コンフィギュレーション モードを開始します。

例：

```
apic1# configure
```

ステップ 2 テナントのコンフィギュレーション モードを開始します。

```
tenant tenant_name
```

例：

```
apic1(config)# tenant t1
```

ステップ 3 デバイスをインポートします。

```
l4l7 cluster import-from tenant_name device-cluster device_name
```

パラメータ	説明
import-from	デバイスのインポート元のテナントの名前。
device-cluster	指定したテナントからインポートするデバイス クラスタの名前。

例：

```
apic1(config-tenant)# l4l7 cluster import-from common device-cluster d1
apic1(config-import-from)# end
```

GUI を使用したデバイスのインポートの確認

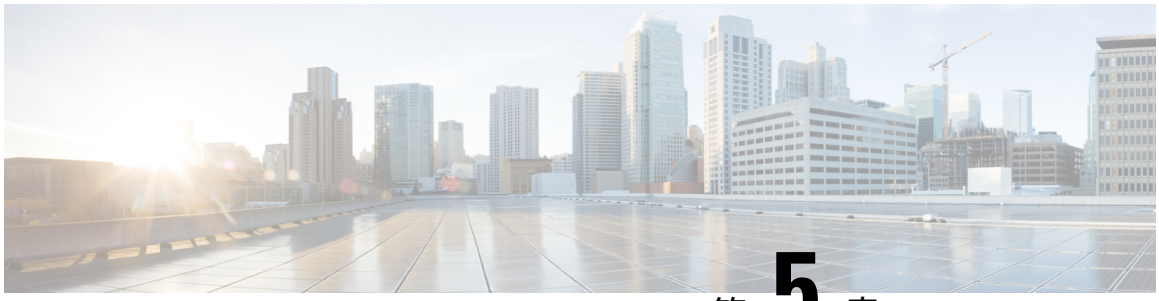
GUI を使用して、デバイスが正常にインポートされたことを確認することができます。

ステップ 1 メニュー バーで、**[Tenants]** > **[All Tenants]** の順に選択します。

ステップ 2 [Work] ペインで、テナントの名前をダブルクリックします。

ステップ 3 [Navigation] ウィンドウで、**Tenant** ***tenant_name*** > **Services** > **L4-L7** > **Imported Devices** > ***device_name*** を選択します。

デバイス情報が [Work] ペインに表示されます。



第 5 章

サービス VM オーケストレーション

サービス VM オーケストレーション

サービス仮想マシン（VM）オーケストレーションは、Cisco Application Policy Infrastructure Controller（APIC）でのサービス VM の作成と管理を容易にするポリシーベースの機能です。サービス VM オーケストレーションは、Cisco APIC 4.0(1) の VMware vCenter 環境向けの新機能です。

以前は、VMware vCenter でサービス VM を作成し、そのサービス VM が属していたデータセンターを定義してデータストアに関連付ける必要がありました。また、管理ネットワークの設定および Cisco APIC への接続も必要でした。ところが、サービス VM オーケストレーションを使用すると、これらのタスクをすべて Cisco APIC で実行できます。

サービス VM オーケストレーションは、具象デバイス（CDev）とも呼ばれるサービス VM の設定プロセスを合理化します。CDevは、論理デバイス（LDev）とも呼ばれるデバイスクラスタにグループ化されます。LDevに適用される設定とポリシーは、LDevに含まれている各CDevに適用されます。

サービス VM オーケストレーションを使用するには、コンフィギュレーション ファイルを作成してアップロードします。次に VM インスタンス化ポリシーを設定してレイヤ 4 ～レイヤ 7 LDev を作成し、LDev に関連付ける CDev を作成します。サービス VM オーケストレーションを設定する前に、サービス VM オーケストレーションの注意事項と制約事項 （26 ページ）を読んで理解してください。

サービス VM オーケストレーション タスクは、Cisco APIC GUI、NX-OS スタイル CLI、または REST API を使用して実行できます。説明については、次の項を参照してください。

- Cisco APIC GUI を使用したサービス VM オーケストレーションの設定 （28 ページ）
- NX-OS スタイル CLI を使用したサービス VM オーケストレーションの設定 （37 ページ）
- REST API を使用したサービス VM オーケストレーションの設定 （38 ページ）

サービス VM オーケストレーションの注意事項と制約事項

サービス VM オーケストレーションを使用する場合は、次の注意事項と制約事項に留意してください。

- サービス VM オーケストレーションは Cisco 適応型セキュリティ仮想アプライアンス（ASAv）および Palo Alto Networks デバイスでのみサポートされます。
- サービス VM オーケストレーションを使用したハイ アベイラビリティ（HA）仮想マシン（VM）の導入は、共有ストレージでのみサポートされます。ローカルデータ ストアではサポートされません。
- 単一サービス VM または HA サービス VM の導入では、Dynamic Host Configuration Protocol（DHCP）IP アドレッシングはサポートされません。
- VMware vCenter で作成されたポート グループまたは VM テンプレートについては、サービス VM オーケストレーションを使用する前に、Cisco Application Policy Infrastructure Controller（APIC）でインベントリを手動で同期する必要があります。設定に関するドキュメントでインベントリの同期をトリガーする方法を確認してください。
- Palo Alto の導入は、デフォルトのユーザ名 **admin** とパスワード **admin** でのみ動作します。
- Palo Alto デバイスを導入すると、「`Script error: force config push is required`」と表示されて Cisco APIC で 10 分間の障害が発生します。このメッセージの原因は Palo Alto デバイスで実行されている内部プロセスです。設定が正常にプッシュされてデバイスが安定すると、障害は解消されます。
- Cisco APIC は、削除および再導入後に Cisco 適応型セキュリティ仮想アプライアンス（ASAv）デバイスに到達できません。この問題は、上流に位置するスイッチで古い MAC アドレスがクリアされていないために発生します。上流に位置するスイッチでサービス VM に使用される IP アドレスの MAC エントリをクリアし、サービス VM オーケストレーションを使用してサービス VM を再導入してください。
- 既存のポリシーを複製する場合は、複製が完了するまで、論理デバイスに関連付けられている VM インスタンス化ポリシーを変更しないでください。

- サービス VM オーケストレーションを使用してサービス VM を導入するには、追加の VMware vCenter 権限を有効にします。『*Cisco ACI Virtualization Guide*』で「Cisco ACI with VMware VDS Integration」の章の「Custom User Account with Minimum VMware vCenter Privileges」を参照してください。

デバイス コンフィギュレーション ファイルの作成

新しいサービス仮想マシン（VM）用にレイヤ 4 ～レイヤ 7 デバイス コンフィギュレーション ファイルを作成する必要があります。コンフィギュレーション ファイルは、Cisco 適応型セキュリティ仮想アプライアンス（ASAv）または Palo Alto Networks デバイスのいずれを使用するかによって異なります。

デバイス コンフィギュレーション ファイルを作成します。

次の例のいずれかをテンプレートとして使用します。

Cisco（ASAv）：

```
VENDOR=CISCO
MODEL=ASA
VERSION=9.9
FILENAME=asav-fixed
CONFIG_START
username $CONFIG_USERNAME password $CONFIG_PASSWORD
passwd $CONFIG_PASSWORD
enable password $CONFIG_PASSWORD
interface management0/0
ip address $CONFIG_IP $CONFIG_SUBNET
nameif management
security-level 100
route management 0.0.0.0 0.0.0.0 $CONFIG_GATEWAY 1
no shutdown
ssh 0.0.0.0 0.0.0.0 Management
ssh timeout 30
ssh version 2
http server enable
http 0.0.0.0 0.0.0.0 management
crypto key generate rsa modulus 1024
aaa authentication ssh console LOCAL
CONFIG_END
```

Palo Alto Networks：

```
VENDOR=PALOALTO
MODEL=PANORAMA
VERSION=8.5
FILENAME=PaloBasicConfig
CONFIG_START
type=static
ip-address=$CONFIG_IP
default-gateway=$CONFIG_GATEWAY
netmask=$CONFIG_SUBNET
vm-auth-key=<add-vmauth-keyhere>
users= $CONFIG_USERNAME
```

```
password= $CONFIG_PASSWORD
CONFIG_END
```

次のタスク

デバイス コンフィギュレーション ファイルを Cisco Application Policy Infrastructure Controller（APIC）にインポートします。このガイドの手順デバイス コンフィギュレーション ファイルのインポート （28 ページ）を参照してください。

デバイスコンフィギュレーションファイルのインポート

デバイス コンフィギュレーション ファイルには、新しいサービス仮想マシン（VM）に必要な設定が含まれています。VM インスタンス化ポリシーを作成する前に、GUI を使用してこのファイルを Cisco Application Policy Infrastructure Controller（APIC）にインポートします。その後、論理デバイス（LDev）とも呼ばれるデバイス クラスタにそのポリシーを適用します。

コンフィギュレーション ファイルのテンプレートについては、デバイス コンフィギュレーション ファイルの作成 （27 ページ）を参照してください。

始める前に

デバイス コンフィギュレーション ファイルを作成済みです。

ステップ 1 Cisco APIC にログインします。

ステップ 2 **[L4-L7 Services]** > **[Packages]** > **[VM Instantiation Files]** に移動します。

ステップ 3 [VM Instantiation Files] を右クリックして [Import Device Configuration File] を選択します。

または、作業ウィンドウの右上にあるハンマーとレンチのアイコンをクリックして [Import Device Configuration File] を選択することもできます。

ステップ 4 [Import Device Configuration File] ダイアログボックスで、デバイス コンフィギュレーション ファイルの保存場所を参照してファイルを選択します。

ステップ 5 [Submit] をクリックします。

Cisco APIC GUI を使用したサービス VM オーケストレーションの設定

Cisco Application Policy Infrastructure Controller（APIC）GUI でいくつかのタスクを実行してサービス VM オーケストレーションを設定できます。

Cisco APIC GUI を使用した VM インスタンス化ポリシーの作成

仮想マシン（VM）インスタンス化ファイルの作成は、サービス仮想マシン（VM）オーケストレーションを使用して Cisco Application Policy Infrastructure Controller でサービス VM を導入および管理するプロセスの最初のタスクです。デバイス クラスタまたは論理デバイス（LDev）用に作成されたポリシーが、LDev に属する具象デバイス（CDev）に適用されます。

始める前に

デバイス コンフィギュレーション ファイルを作成し、Cisco APIC にアップロードできる場所に保存済みである必要があります。本ガイドの「デバイス コンフィギュレーション ファイルの作成とアップロード」の項を参照してください。

ステップ 1 Cisco APIC にログインします。

ステップ 2 **[Tenants]** > **テナント** > **[Policies]** > **[VMM]** > **[VM Instantiation Policies]** に移動します。

ステップ 3 作業ウィンドウの右上隅にあるハンマーとレンチのアイコンをクリックし、[Create VM Instantiation Policy] を選択します。

ステップ 4 [Create VM Instantiation Policy] ダイアログボックスで、次の手順を実行します。

a) [Name] フィールドにポリシーの名前を入力します。

b) [Controller] ドロップダウンリストからコントローラを選択します。

c) [VM Template] ドロップダウンリストで、作成するサービス VM のテンプレートを選択します。

ドロップダウンリストには、コントローラに関連付けられている VM テンプレートが表示されます。

（注） VMware vCenter で作成した VM テンプレートが表示されない場合は、次の手順を実行します。

1. [Controller] ドロップダウンリストの横にある青色のアイコンをクリックします。
2. [Controller Instance] ダイアログボックスの右側にあるレンチとハンマーのアイコンをクリックし、[Trigger Inventory Sync]、[Yes] の順にクリックして同期をトリガーします。
3. [Controller Instance] ダイアログボックスを閉じて [Create VM Instantiation Policy] ダイアログボックスに戻ります。

d) [Host Name] ドロップダウンリストで、サービス VM を導入するホストを選択します。

VMware vSphere 分散リソース スケジューラ（DRS）クラスタまたは個々のホストを選択できます。

e) [Data Store] ドロップダウンリストで、VM ディスクを配置するデータ ストアを選択します。

f) [Device Configuration File] フィールドで、以前に作成したファイルを選択します。

g) [Submit] をクリックします。
作業ウィンドウに VM インスタンス化ポリシーが表示されます。

Cisco APIC GUI を使用してレイヤ 4 ～レイヤ 7 デバイスを作成して VM インスタンス化ポリシーに関連付ける

この手順では、レイヤ 4 ～レイヤ 7 デバイスを作成し、以前に作成した仮想マシン（VM）インスタンス化ポリシーに関連付けます。

レイヤ4～ レイヤ7デバイスを作成する際には、物理デバイスまたは仮想マシンのいずれかに接続することができます。接続先のタイプによって、フィールドが若干異なります。物理デバイスに接続する場合は、物理インターフェイスを指定します。仮想マシンに接続する場合は、VMM ドメイン、仮想マシン、および仮想インターフェイスを指定します。また、不明モデルを選択して接続を手動で設定することもできます。

VM インスタンス化ポリシーに関連付けるレイヤ 4 ～レイヤ 7 デバイスの作成時には、ポリシーの指定および新しいサービス VM の作成も行います。

（注） ロード バランサであるレイヤ 4 ～ レイヤ 7 デバイスを設定する場合には、context aware パラメータは使用しません。context aware パラメータのデフォルト値は single context コンテキストです。無視することができます。

始める前に

- テナントを作成しておく必要があります。
- VM インスタンス化ポリシーを作成済みである必要があります。Cisco APIC GUI を使用した VM インスタンス化ポリシーの作成 （29 ページ）の項を参照してください。

ステップ 1 Cisco Application Policy Infrastructure Controller（APIC）にログインします。

ステップ 2 **[Tenants]** > テナント > **[Services]** > **[L4-L7]** > **[Devices]** に移動します。

ステップ 3 [Devices] を右クリックして [Create L4-L7 Devices] を選択します。

または、作業ウィンドウの右上にあるアクションアイコン（交差したハンマーとレンチ）をクリックし、[Create L4-L7 Devices] を選択することもできます。

ステップ 4 [Create L4-L7 Devices] ダイアログボックスで、[General] セクションの次のフィールドに入力します。

名前	説明
Managed	（オプション）管理対象デバイスを作成する場合はチェックボックスをオンにし、非管理対象デバイスを作成する場合はオフにします。
Name	レイヤ4～レイヤ7デバイスの名前を入力します。

名前	説明
サービス タイプ	ドロップダウンリストからサービスの種類を選択します。次のいずれかの種類を選択できます。 • [ADC]（アプリケーション配信コントローラ） [ADC]は、デフォルトのサービスの種類です。 • [Firewall]：ルーテッドまたはトランスペアレント展開モードを選択します。 • [Other]：その他のモード。 （注） ポリシーベース リダイレクト設定では、サービスの種類として [Firewall] または [ADC] を選択します。
Device Type	[Virtual]（仮想レイヤ 4 ～レイヤ 7 デバイス）を選択します。
VMM ドメイン	ドロップダウンリストから VMM ドメインを選択します。
VM Instantiation Policy	ドロップダウンリストで、前に作成した VM インスタンス化ポリシーを選択します。 選択したポリシーが新しいレイヤ 4 ～レイヤ 7 デバイスに関連付けられます。VMware vCenter で自動的に VM を作成することもできます。
デバイス パッケージ （管理対象デバイスのみ）	使用するデバイス パッケージ（ベンダー提供）をドロップダウンリストから選択します。 Cisco APIC はデバイス パッケージを使用してデバイスと通信し、設定を行います。
Model （管理対象デバイスのみ）	ドロップダウンリストからデバイスのモデルを選択します。
無差別モード	サービス グラフの導入後に生成される Cisco ACI 管理ポート グループで無差別モードを有効にするには、チェックボックスをオンにします。 無差別モードを有効にすると、ポート グループのすべてのトラフィックが無差別ポートに接続されている VM に到達できます。

名前	説明
コンテキスト認識	[Single]（デフォルト）または [Multiple] を選択します。 [Single] を選択した場合、プロバイダー ネットワーク上でホストされた特定のタイプの複数のテナントでデバイスクラスタを共有することはできません。特定のユーザの特定のテナントにデバイス クラスタを提供する必要があります。 [Multiple] を選択した場合は、プロバイダー ネットワーク上でホストする特定のタイプの複数のテナントでデバイスクラスタを共有できます。たとえば、同じデバイスを共有する 2 つのホスティング会社が存在する可能性があります。
機能タイプ	次のオプションを選択できます。 • [GoThrough]（トランスペアレント モード） • [GoTo]（ルーテッド モード）

ステップ 5 （管理対象デバイス）[Connectivity] セクションで次のフィールドに入力します。

名前	説明
APIC to Device Management Connectivity	接続のタイプを選択します。 • ファブリックの外部にあるデバイスに接続するには、[Out-Of-Band] を選択します。 • ファブリックを介してインバンド管理ネットワーク内のデバイスに接続するには、[In-Band] を選択します。

ステップ 6 （管理対象デバイス）[Credentials] セクションで次のフィールドに入力します。

名前	説明
Username	Cisco APIC がレイヤ 4 ～レイヤ 7 デバイスと通信できるように、デバイスのユーザ名を入力します。
Password	Cisco APIC がレイヤ 4 ～レイヤ 7 デバイスと通信できるように、デバイスのパスワードを入力します。

名前	説明
Confirm Password	Cisco APIC がレイヤ 4 ～レイヤ 7 デバイスと通信できるように、デバイスのパスワードを再入力します。

ステップ 7 [Devices] セクションでプラス アイコンをクリックします。

ステップ 8 [Create Device STEP 1] > [Device] ダイアログボックスで次のフィールドに入力し、具象デバイス（CDev）を設定してレイヤ 4 ～レイヤ 7 デバイスに関連付けます。

名前	説明
Name （管理対象デバイスのみ）	新しいサービス VM の CDev 名を入力します。
Management IP	新しいサービス VM の管理ポート IP アドレスを入力します。
Gateway IP	新しいサービス VM のゲートウェイ IP アドレスを入力します。
[Subnet Mask]	新しいサービス VM のサブネットマスクを入力します。
Management Port	新しいサービス VM の管理ポートとして [http] または [https] をドロップダウンリストから選択します。
Management vNIC	ドロップダウンリストから新しいサービス VM の管理 vNIC を選択します。
VM	VMware vCenter に表示される新しいサービス VM の VM 名を入力します。

名前	説明
Host （任意）	ドロップダウンリストから新しいサービス VM のホストを選択します。ホストを選択しない場合は、VMインスタンス化ポリシーで選択されているホストが使用されます。 ポリシーベース リダイレクト（PBR）およびDirect Server Return（DSR）機能の場合は、トポロジに基づいて特定のホストを選択する必要があります。その場合は正しいホストを選択してください。 DSR と PDR の場合は、コンピューティング VM とサービス VM を同じトップオブラック（ToR）スイッチ ペアに置くことはできません。したがって PBR または DSR トポロジのサービス VM を導入するためのホストを選択する必要があります。選択しないと、機能によってサービス VM がコンピューティング VM と同じホストに導入される可能性があります。 Cisco Application Centric Infrastructure Virtual Edge 上で接続するデバイスについては、ハイ アベイラビリティのレイヤ4～レイヤ7デバイスを同じホストに導入することはできません。したがって、プライマリ VM とセカンダリ VM に異なるホストを選択します。
Port Group Name （任意）	ドロップダウンリストで、新しいサービス VM を導入するポート グループを選択します。選択しない場合は、VMテンプレートで使用されているポート グループが使用されます。
HA EPG （任意）	新しいサービス VM のハイ アベイラビリティ（HA）通信用に、HA エンドポイント グループ（EPG）か、vSwitch または分散型仮想スイッチ（DVS）ポート グループをドロップダウンリストから選択します。
HA Network Adapter （任意）	ドロップダウンリストから新しいサービス VM 用の HA ネットワーク アダプタを選択します。
Username	新しいサービス VM のユーザ名を入力します。
Password	新しいサービス VM のパスワードを入力します。
Confirm Password	パスワードを再入力します。

名前	説明
シャーシ （オプション：管理対象デバイスのみ）	ドロップダウンリストからシャーシを選択します。

ステップ 9 [次へ] をクリックします。

ステップ 10 [Create Device STEP 2] > [Interfaces] ダイアログボックスの [Interfaces] セクションで、プラス アイコンをクリックします。

ステップ 11 ダイアログボックスで次のフィールドに入力し、CDev のインターフェイスを設定します。

名前	説明
Name	ドロップダウンリストからレイヤ4～レイヤ7デバイス インターフェイスの名前を選択します。
vNIC （仮想デバイス タイプのみ）	ドロップダウンリストから VM ネットワーク アダプタの名前を選択します。
Path （レイヤ4～レイヤ7デバイスが仮想デバイスの場合は省略可）	インターフェイスを接続するポート、ポート チャネル（PC）、またはバーチャル ポート チャネル（VPC）を選択します。

ステップ 12 [Interfaces] セクションでプラス アイコンをもう一度クリックし、別のインターフェイスを設定します。

ステップ 13 [Update] をクリックします。

ステップ 14

ステップ 15 レイヤ 4 ～レイヤ 7 デバイスにサービス VM をさらに追加するには、ステップ 8 ～ステップ 13 を繰り返します。

ステップ 16 複数のサービス VM を使用する場合は、[Create Device STEP 1] > [Device] ダイアログボックスの [Cluster] セクションで、デバイスごとに次のフィールドに入力します。

HA クラスタでは、クラスタのインターフェイスが、クラスタ内の両方の具体デバイスにある対応するインターフェイスにマッピングされていることを確認してください。

名前	説明
Management IP Address	クラスタの管理 IP アドレスを入力します。
Management Port	クラスタの管理ポートとして [http] または [https] をドロップダウンリストから選択します。
Device Manager （任意）	ドロップダウンリストからクラスタデバイスマネージャを選択します。 デバイスマネージャのみで、シスコアプリケーションセントリック インフラストラクチャ（ACI）ファブリック内の一連のクラスタを設定できます。

名前	説明
[Cluster Interfaces] 領域	次のフィールドに値を入力してレイヤ 4 ～レイヤ 7 デバイスの外部接続を設定します。 • [Type] ドロップダウンリストからクラスタ インターフェイス タイプを選択します。タイプは次のとおりです。 • **failover_link** • **ユーティリティ** • **consumer** • **provider** • **mgmt** • **cluster_ctrl_lk** • **failover-lan** • **consumer and provider** • [Name] ドロップダウンリストからクラスタ インターフェイス名を選択します。 • [Concrete Interfaces] ドロップダウンリストで、関連付けられている具象インターフェイスを選択します。

ステップ 17 [次へ] をクリックします。

使用しているパッケージで使用可能な機能とパラメータのリストが、[Create Device STEP 2] > [Interfaces] ダイアログボックスに表示されます。

ステップ 18 [Create Device STEP 2] > [Interfaces] ダイアログボックスで、次のいずれかの操作を実行します。

状況	結果
単一のサービス VM	ステップ 19 に進みます。
HA ペア内のサービス VM	**1.** [Devices] をクリックします。 **2.** [Basic Parameters] タブまたは [All Parameters] タブをクリックします。 **3.** 作業ウィンドウで、使用するパラメータを選択します。 一連のパラメータは、使用している特定のパッケージと選択した特定の機能によって異なります。 **4.** 選択した機能のパラメータに対して、次のように値を指定します。 **1.** 変更するフィールドをダブルクリックします。

状況	結果
	2. 表示されたフィールドに必要な情報を入力します。 3. [Update] をクリックします。
クラスタ内のサービス VM	1. [Devices] をクリックします。 2. [Basic Parameters] タブまたは [All Parameters] タブをクリックします。 3. 作業ウィンドウで、使用するパラメータを選択します。 一連のパラメータは、使用している特定のパッケージと選択した特定の機能によって異なります。 4. 選択した機能のパラメータに対して、次のように値を指定します。 1. 変更するフィールドをダブルクリックします。 2. 表示されたフィールドに必要な情報を入力します。 3. [Update] をクリックします。

ステップ 19 [Finish] をクリックします。

次のタスク

[Recent Tasks] で、VMware vCenter での新しいサービス VM の作成を確認できます。表示されるまでにしばらく時間がかかることがあります。

NX-OS スタイル CLI を使用したサービス VM オーケストレーションの設定

NX-OS スタイル CLI を使用して、仮想マシン（VM）インスタンス化ポリシーとレイヤ 4 ～レイヤ 7 具象デバイスを作成し、デバイスをインスタンス化ポリシーにマッピングできます。その後、内部および外部インターフェイスを VM ネットワーク アダプタにマッピングできます。

始める前に

デバイス コンフィギュレーション ファイルをインポートし、Cisco Application Policy Infrastructure Controller にアップロードできる場所に保存済みである必要があります。このガイドのデバイス コンフィギュレーション ファイルのインポート （28 ページ）セクションを参照してください。

ステップ 1 VM インスタンス化ポリシーを作成します。

例：

```
APIC1(config-tenant)# inst-pol VMPolName VMMname VcentercontrollerName VMtemplateName ClusterName
datastorename
```

ステップ 2 レイヤ 4 ～レイヤ 7 具象デバイスを作成して VM インスタンス化ポリシーに関連付けます。

例：

```
APIC1(config)# tenant T0
APIC1(config-tenant)# l4l7 cluster name ASA-Single type virtual vlan-domain ASAVMM switching-mode
AVE vm-instantiation-policy ASA-Template-Pol  service FW function go-to context single trunking
disable
```

ステップ 3 内部および外部インターフェイスを VM ネットワーク アダプタにマッピングします。

例：

```
APIC1(config-cluster)# cluster-interface external
APIC1(config-cluster-interface)# member device ASA-Cdev device-interface GigabitEthernet0/0
APIC1(config-member)# vnic "Network adapter 2"
APIC1(config-member)# exit
APIC1(config-cluster)# cluster-interface internal
APIC1(config-cluster-interface)# member device ASA-Cdev device-interface GigabitEthernet0/1
APIC1(config-member)# vnic "Network adapter 3"
APIC1(config-member)# exit
APIC1(config-cluster-interface)# exit
APIC1(config-cluster)#
```

REST API を使用したサービス VM オーケストレーションの設定

REST API を使用してサービス VM オーケストレーションを設定できます。

始める前に

デバイス コンフィギュレーション ファイルをインポートし、Cisco Application Policy Infrastructure Controller にアップロードできる場所に保存済みである必要があります。このガイドのデバイス コンフィギュレーション ファイルのインポート （28 ページ）セクションを参照してください。

サービス VM オーケストレーションを設定します。

例：

```
<vnsLDevVip annotation="" contextAware="single-Context" devtype="VIRTUAL"
dn="uni/tn-T0/lDevVip-NEW-HA-LDEV-20" funcType="GoTo" isCopy="no" managed="yes" mode="legacy-Mode"
 name="NEW-HA-LDEV-20" nameAlias="" packageModel="ASAv" promMode="no" svcType="FW" trunking="no">
                <vnsLIf annotation="" encap="unknown" name="client" nameAlias="">
                                <vnsRsMetaIf annotation="" isConAndProv="no"
```

```
tDn="uni/infra/mDev-CISCO-ASA-1.3/mIfLbl-external"/>
                                <vnsRsCIfAttN annotation=""
tDn="uni/tn-T0/lDevVip-NEW-HA-LDEV-20/cDev-CDEV-HA-S1-NEW/cIf-[GigabitEthernet0/0]"/>
                                <vnsRsCIfAttN annotation=""
tDn="uni/tn-T0/lDevVip-NEW-HA-LDEV-20/cDev-CDEV-HA-P1-NEW/cIf-[GigabitEthernet0/0]"/>
                </vnsLIf>
                <vnsLIf annotation="" encap="unknown" name="server" nameAlias="">
                                <vnsRsMetaIf annotation="" isConAndProv="no"
tDn="uni/infra/mDev-CISCO-ASA-1.3/mIfLbl-internal"/>
                                <vnsRsCIfAttN annotation=""
tDn="uni/tn-T0/lDevVip-NEW-HA-LDEV-20/cDev-CDEV-HA-S1-NEW/cIf-[GigabitEthernet0/1]"/>
                                <vnsRsCIfAttN annotation=""
tDn="uni/tn-T0/lDevVip-NEW-HA-LDEV-20/cDev-CDEV-HA-P1-NEW/cIf-[GigabitEthernet0/1]"/>
                </vnsLIf>
                <vnsRsLDevVipToInstPol annotation="" tDn="uni/tn-T0/svcCont/instPol-HA-POL"/>
              <vnsRsALDevToDomP annotation="" switchingMode="AVE" tDn="uni/vmmp-VMware/dom-mininet"/>

                <vnsCMgmt annotation="" dnsDomain="" gateway="0.0.0.0" host="10.197.146.178"
ipAllocationType="fixed" isInBand="no" name="" nameAlias="" port="443" portGroupName=""
subnetmask="0.0.0.0" vnicName=""/>
                <vnsCDev annotation="" cloneCount="0" devCtxLbl="" host="10.197.146.188"
isCloneOperation="no" isTemplate="no" name="CDEV-HA-S1-NEW" nameAlias=""
vcenterName="orionin103-vcenter1" vmName="ASA-S1-VM-20">
                                <vnsHAPortGroup annotation="" name="" nameAlias=""
portGroupName="10.197.146.188 | VLAN2500-172-25" vnicName="Network adapter 10"/>
                            <vnsDevFolder annotation="" key="FailoverConfig" name="FailoverConfig"
 nameAlias="">
                                                <vnsDevParam annotation="" key="lan_unit"
name="lan_unit" nameAlias="" value="secondary"/>
                                                <vnsDevParam annotation="" key="failover"
name="failover" nameAlias="" value="enable"/>
                                                <vnsDevFolder annotation="" key="mgmt_standby_ip"
name="mgmt_standby_ip" nameAlias="">
                                                                <vnsDevParam annotation=""
key="standby_ip" name="standby_ip" nameAlias="" value="10.197.146.178"/>
                                                </vnsDevFolder>
                                                <vnsDevFolder annotation="" key="polltime"
name="polltime" nameAlias="">
                                                                <vnsDevParam annotation=""
key="interval_value" name="interval_value" nameAlias="" value="1"/>
                                                                <vnsDevParam annotation=""
key="interval_unit" name="interval_unit" nameAlias="" value="second"/>
                                                                <vnsDevParam annotation=""
key="holdtime_value" name="holdtime_value" nameAlias="" value="3"/>
                                                </vnsDevFolder>
                                                <vnsDevFolder annotation=""
key="failover_link_interface" name="failover_link_interface" nameAlias="">
                                                                <vnsDevParam annotation=""
key="use_lan" name="use_lan" nameAlias="" value="fover"/>
                                                                <vnsDevParam annotation=""
key="interface_name" name="interface_name" nameAlias="" value="fover"/>
                                                                <vnsDevParam annotation=""
key="interface" name="interface" nameAlias="" value="GigabitEthernet0/8"/>
                                                </vnsDevFolder>
                                                <vnsDevFolder annotation="" key="failover_ip"
name="failover_ip" nameAlias="">
                                                                <vnsDevParam annotation=""
key="interface_name" name="interface_name" nameAlias="" value="fover"/>
                                                                <vnsDevParam annotation=""
key="active_ip" name="active_ip" nameAlias="" value="172.25.0.178"/>
                                                                <vnsDevParam annotation=""
key="netmask" name="netmask" nameAlias="" value="255.255.0.0"/>
                                                                <vnsDevParam annotation=""
key="standby_ip" name="standby_ip" nameAlias="" value="172.25.0.179"/>
```

```
                                                </vnsDevFolder>
                                                <vnsDevFolder annotation=""
key="failover_lan_interface" name="failover_lan_interface" nameAlias="">
                                                        <vnsDevParam annotation=""
key="interface_name" name="interface_name" nameAlias="" value="fover"/>
                                                        <vnsDevParam annotation=""
key="interface" name="interface" nameAlias="" value="GigabitEthernet0/8"/>
                                                </vnsDevFolder>
                                </vnsDevFolder>
                                <vnsCMgmt annotation="" dnsDomain="" gateway="10.197.146.161"
host="10.197.146.179" ipAllocationType="fixed" isInBand="no" name="" nameAlias="" port="443"
portGroupName="10.197.146.188 | MGMT-955" subnetmask="255.255.255.224" vnicName="Network adapter
1"/>
                                <vnsCIf annotation="" name="GigabitEthernet0/1" nameAlias=""
vnicName="Network adapter 3"/>
                                <vnsCIf annotation="" name="GigabitEthernet0/0" nameAlias=""
vnicName="Network adapter 2"/>
                                <vnsCCredSecret annotation="" name="password" nameAlias=""
value="cisco123!"/>
                              <vnsCCred annotation="" name="username" nameAlias="" value="admin"/>

                </vnsCDev>
                <vnsCDev annotation="" cloneCount="0" devCtxLbl="" host="10.197.146.187"
isCloneOperation="no" isTemplate="no" name="CDEV-HA-P1-NEW" nameAlias=""
vcenterName="orionin103-vcenter1" vmName="ASA-P1-VM-20">
                                <vnsHAPortGroup annotation="" name="" nameAlias=""
portGroupName="10.197.146.187 | VLAN2500-172-25" vnicName="Network adapter 10"/>
                            <vnsDevFolder annotation="" key="FailoverConfig" name="FailoverConfig"
 nameAlias="">
                                                <vnsDevParam annotation="" key="lan_unit"
name="lan_unit" nameAlias="" value="primary"/>
                                                <vnsDevParam annotation="" key="failover"
name="failover" nameAlias="" value="enable"/>
                                                <vnsDevFolder annotation="" key="failover_ip"
name="failover_ip" nameAlias="">
                                                        <vnsDevParam annotation=""
key="interface_name" name="interface_name" nameAlias="" value="fover"/>
                                                        <vnsDevParam annotation=""
key="standby_ip" name="standby_ip" nameAlias="" value="172.25.0.179"/>
                                                        <vnsDevParam annotation=""
key="netmask" name="netmask" nameAlias="" value="255.255.0.0"/>
                                                        <vnsDevParam annotation=""
key="active_ip" name="active_ip" nameAlias="" value="172.25.0.178"/>
                                                </vnsDevFolder>
                                                <vnsDevFolder annotation=""
key="failover_lan_interface" name="failover_lan_interface" nameAlias="">
                                                        <vnsDevParam annotation=""
key="interface_name" name="interface_name" nameAlias="" value="fover"/>
                                                        <vnsDevParam annotation=""
key="interface" name="interface" nameAlias="" value="GigabitEthernet0/8"/>
                                                </vnsDevFolder>
                                                <vnsDevFolder annotation="" key="mgmt_standby_ip"
name="mgmt_standby_ip" nameAlias="">
                                                        <vnsDevParam annotation=""
key="standby_ip" name="standby_ip" nameAlias="" value="10.197.146.179"/>
                                                </vnsDevFolder>
                                                <vnsDevFolder annotation=""
key="failover_link_interface" name="failover_link_interface" nameAlias="">
                                                        <vnsDevParam annotation=""
key="interface_name" name="interface_name" nameAlias="" value="fover"/>
                                                        <vnsDevParam annotation=""
key="use_lan" name="use_lan" nameAlias="" value="fover"/>
                                                        <vnsDevParam annotation=""
key="interface" name="interface" nameAlias="" value="GigabitEthernet0/8"/>
```

```
                                                </vnsDevFolder>
                                                <vnsDevFolder annotation="" key="polltime"
name="polltime" nameAlias="">
                                                                <vnsDevParam annotation=""
key="holdtime_value" name="holdtime_value" nameAlias="" value="3"/>
                                                                <vnsDevParam annotation=""
key="interval_unit" name="interval_unit" nameAlias="" value="second"/>
                                                                <vnsDevParam annotation=""
key="interval_value" name="interval_value" nameAlias="" value="1"/>
                                                </vnsDevFolder>
                                </vnsDevFolder>
                                <vnsCMgmt annotation="" dnsDomain="" gateway="10.197.146.161"
host="10.197.146.178" ipAllocationType="fixed" isInBand="no" name="" nameAlias="" port="443"
portGroupName="10.197.146.187 | MGMT-955" subnetmask="255.255.255.224" vnicName="Network adapter
1"/>
                                <vnsCIf annotation="" name="GigabitEthernet0/1" nameAlias=""
vnicName="Network adapter 3"/>
                                <vnsCIf annotation="" name="GigabitEthernet0/0" nameAlias=""
vnicName="Network adapter 2"/>
                                <vnsCCredSecret annotation="" name="password" nameAlias=""
value="cisco123!"/>a
                              <vnsCCred annotation="" name="username" nameAlias="" value="admin"/>

                </vnsCDev>
                <vnsCCredSecret annotation="" name="password" nameAlias="" value="cisco123!"/>
                <vnsRsMDevAtt annotation="" tDn="uni/infra/mDev-CISCO-ASA-1.3"/>
                <vnsCCred annotation="" name="username" nameAlias="" value="admin"/>
</vnsLDevVip>
```

サービスVMオーケストレーションのトラブルシューティング

ここでは、サービスVMオーケストレーションの既知の問題と制限事項、および問題が発生した場合のトラブルシューティング手順について説明します。

サービス VM テンプレートが VM インスタンス化ポリシーに表示されない

VMware vCenter で作成したサービス VM テンプレートが VM インスタンス化ポリシーに表示されない場合は、次の手順を実行します。

ステップ 1 **vnsInstPol** を使用して Visore を確認し、`vmTemplate` を探します。

vnsInstPol フィールドの値がない場合、または値が null の場合は、次の手順に進みます。

ステップ 2 インベントリの同期をトリガーします。

a) Cisco Application Policy Infrastructure Controller（APIC）で **[Virtual Networking]** > **[Inventory]** に移動し、[VMM Domains] および [VMware] フォルダを展開します。

b) VMM ドメインをクリックします。
c) 中央のペインでコントローラをダブルクリックします。
d) [VMM Controller] ダイアログボックスでハンマーとレンチのドロップダウンリストから [Trigger Inventory Sync] を選択し、プロンプトが表示されたら [Yes] をクリックします。

ステップ 3 仮想マシン（VM）インスタンス化ポリシーを確認します（VMM ドメインにマッピングされているコントローラを選択し、VM テンプレートが存在するかどうかを確認してください）。

VMware vCenter で作成したポート グループが CDev に表示されない

VMware vCenter で作成したポート グループが具象デバイス（CDev）に表示されない場合は、次の手順を実行します。

ステップ 1 インベントリの同期をトリガーします。

a) Cisco Application Policy Infrastructure Controller（APIC）で **[Virtual Networking]** > **[Inventory]** に移動し、[VMM Domains] および [VMware] フォルダを展開します。
b) VMM ドメインをクリックします。
c) 中央のペインでコントローラをダブルクリックします。
d) [VMM Controller] ダイアログボックスでハンマーとレンチのドロップダウンリストから [Trigger Inventory Sync] を選択し、プロンプトが表示されたら [Yes] をクリックします。

ステップ 2 ポート グループが表示されるかどうかを確認します。

a) **[Tenants]** > **テナント** > **[Services]** > **[L4-L7]** > **[Devices]** > **デバイス**に移動し、デバイスをクリックします。

ステップ 3 [Concrete Device] 作業ウィンドウで、[Port Group Name] ドロップダウンリストにポート グループが表示されるかどうかを確認します。

サービス VM の IP アドレスに到達できない

サービス仮想マシン（VM）の導入後にサービス仮想マシン（VM）の IP アドレスに到達できない場合は、次の手順を実行します。

ステップ 1 Cisco Application Policy Infrastructure Controller（APIC）でサービス VM の接続性を確認します。

Cisco APIC は、削除および再導入後に Cisco 適応型セキュリティ仮想アプライアンス（ASAv）デバイスに到達できません。この問題は、上流に位置するスイッチで古い MAC アドレスがクリアされていないために発生します。サービス VM に使用される IP アドレスの MAC エントリをクリアしてサービス VM を再導入してください。

ステップ 2 デバイス管理で vSwitch ポート グループを使用している場合は、Cisco APIC と VMware vCenter の間にあるすべての中間スイッチおよびデバイスで、VLAN およびルートの存在を確認します。

Cisco APIC は、サービス VM が正常に導入されたかどうかを確認するために、デバイスの IP アドレスに ping を実行できる必要があります。

ステップ 3 具象デバイス（CDev）の管理インターフェイスに対して、適切なポート グループまたは EPG が選択されていることを確認します。

ステップ 4 サービス VM がアップストリーム ゲートウェイに到達できるように接続性を確認します。

デバイスの状態が Init と表示される

デバイスの状態が init と表示される場合は、次の手順を実行します。

ステップ 1 NX-OS スタイル CLI から、サービス デバイスの到達可能性を確認する ping を実行します。

ステップ 2 サービス デバイスへのログイン クレデンシャルがデバイス設定で指定されたユーザ名とパスワードに一致することを確認します。

ステップ 3 サービス デバイスの仮想 IP アドレスおよびポートが開いていることを確認します。

ステップ 4 Cisco Application Policy Infrastructure Controller（APIC）設定でユーザ名とパスワードが正しいことを確認します。

LIF 設定が無効である

論理デバイスの `lif-invalid-Clf` が原因で論理インターフェイス（LIF）の設定が無効になる F0772 障害が発生した場合は、次の手順を実行します。

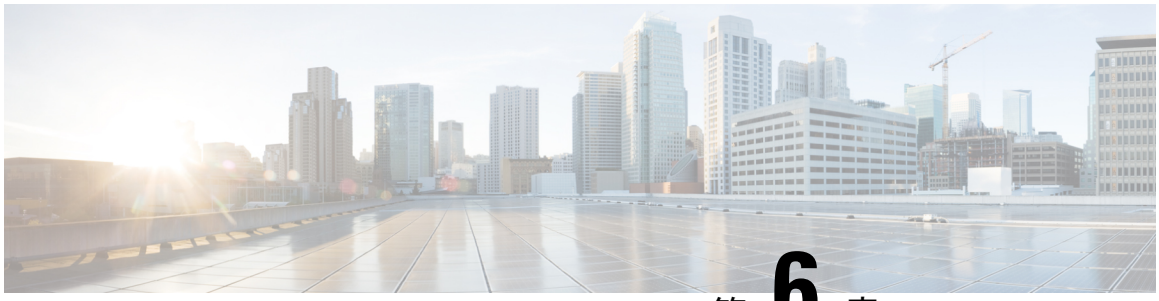
ステップ 1 LIF および具象インターフェイス（ClF）と呼ばれる項目を特定します。

この特定の障害において、LIF は正しくレンダリングされていない要素です。これは、機能ノードが LIF を実際のインターフェイスまたは具象インターフェイスにマッピングして関係を形成する場合に発生します。

F0772 は、次のいずれかの問題を意味します。

- LIF が作成されてない。
- LIF が正しい具象インターフェイスにマッピングされていない。

ステップ 2 レイヤ 4 ～レイヤ 7 デバイスの状態に関するその他の問題については、『*Cisco APIC* レイヤ *4* ～ レイヤ *7* サービス導入ガイド』でトラブルシューティングの情報を参照してください。



第 6 章

デバイスへの接続の設定

デバイスのインバンド管理について

Cisco Application Policy Infrastructure Controller（Cisco APIC）は、Cisco Application Centric Infrastructure（ACI）ファブリックを通過する各テナントのインバンド内のデバイスを管理するメカニズムを提供します。この設定オプションでは、インフラテナントと管理テナント内でのルーティングを可能にするためにデバイスで使用される管理 IP アドレスを必要とすることなく、デバイスの管理接続が実現します。

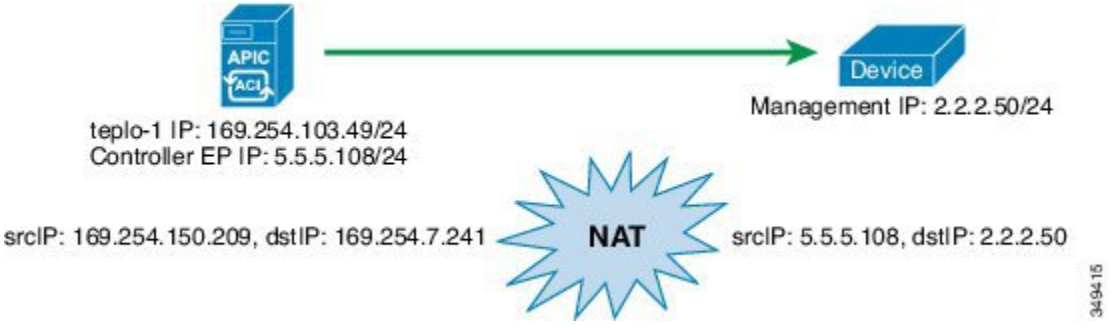
（注） この機能は、Cisco APIC およびファブリック ノードのインバンド管理とは別のものです。デバイスのインバンドを管理するには、ファブリックのインバンド管理は必要ありません。

Cisco APIC とデバイス間のインバンド管理通信は、Cisco APIC に一意の IP アドレスを設定することで可能になります。この IP アドレスはコントローラ エンドポイントと呼ばれています。これらの IP アドレスは実際には Cisco APIC インターフェイスに設定するのではなく、代わりにネットワーク アドレス変換（NAT）と共に使用してデバイスとの管理通信を確立します。Cisco APIC が使用する NAT アドレスは Cisco APIC によって自動的に選択され、169.254.0.0/16 のアドレス範囲内に収まります。

また、各デバイス管理 IP アドレスは変換後の IP アドレスとして Cisco APIC に提示されます。この変換後のアドレスを、マップされたホスト アドレスと呼びます。

次の図に、Cisco APIC とデバイス間のアドレス変換を示します。

図 *3 : Cisco APIC* とデバイスの間のネットワーク アドレス変換



GUI を使用したデバイスのインバンド管理の設定

GUI を使用してデバイスにインバンド管理を設定することができます。

ステップ 1 メニュー バーで、**[Tenants]** > **[All Tenants]** の順に選択します。

ステップ 2 [Work] ペインで、テナントの名前をダブルクリックします。

ステップ 3 [Navigation] ウィンドウで、**Tenant** *tenant_name* > **Services** > **L4-L7** > **Devices** を選択します。

ステップ 4 [Work] ペインで、**[Actions]** > **[Create L4-L7 Devices]** の順に選択します。

ステップ 5 [Create L4-L7 Devices] ダイアログボックスで、下記で指定している項目を除き、必要に応じてフィールドに入力します。

a) [APIC to Device Management Connectivity] オプション ボタンに [In-Band] を選択します。

b) [EPG] ドロップダウン リストで、[Create Management EPG] を選択します。

ステップ 6 [Create Management EPG] ダイアログボックスで、下記で指定している項目を除き、必要に応じてフィールドに入力します。

a) [Application Profile] ドロップダウン リストで、EPG を配置する既存のアプリケーション プロファイルを選択します。必要に応じて新しいアプリケーションプロファイルを作成するには、[Create Application Profile] を選択します。

新しいアプリケーション プロファイルを作成する場合は、[EPG] セクションと [Contracts] セクションは空白のままにします。

b) [Name] フィールドに、管理 EPG の名前を入力します。

c) [Bridge Domain] ドロップダウン リストで、ドメインを選択します。

d) [Domains] で、ドメイン プロファイルを追加します。

e) [Reserved IP addresses for APICs] セクションで、[+] をクリックして新しい IP アドレス プールを作成します。

ステップ 7 [Create IP Address Pool] ダイアログボックスで、すべてのフィールドに入力し、[OK] をクリックします。

IP アドレス プールは、コントローラのエンドポイント アドレスを定義します。プール内の IP アドレスは、デバイスが Application Policy Infrastructure Controller（APIC）の IP アドレスと見なす IP アドレスです。

コントローラのエンドポイントに定義したアドレス範囲が、デバイスに定義した管理IPアドレスと同じサブネットに含まれていない場合は、デバイスにネクストホップ ゲートウェイを提供する管理EPGブリッジドメインの下にサブネットを定義して、コントローラのエンドポイントに到達するようにする必要があります。

ステップ 8 [Create Management EPG] ダイアログボックスで、[Submit] をクリックします。
これで、管理 EPG のドメイン名が設定されました。

ステップ 9 [Create L4-L7 Devices] ダイアログボックスで、デバイスのセットアップを実行します。インターフェイスの設定に管理インターフェイスを必ず含めてください。

GUI を使用したデバイスのインバンド管理のトラブルシューティング

既存のエンドポイント グループ（EPG）をデバイスの管理 EPG として選択した場合は、管理 IP アドレス プールとコントローラ管理ポリシーを手動で追加する必要があります。GUI を使用してこれらを追加することができます。

ステップ 1 メニュー バーで、**[Tenants]** > **[All Tenants]** の順に選択します。

ステップ 2 [Work] ペインで、テナントの名前をダブルクリックします。

ステップ 3 [Navigation] ペインで、***[tenant_name]*** > **[Application Profiles]** > ***[application_profile_name]*** > **[Application EPGs]** > ***[EPG_name]*** > **[L4/L7 IP Address Pool]** の順に選択します。

ステップ 4 [Work] ペインで、**[Actions]** > **[Create Address Pool]** の順に選択します。

ステップ 5 [Create IP Address Pool] ダイアログボックスで、必要に応じてフィールドに入力します。

これで、管理 IP プールが追加されます。

ステップ 6 [Navigation] ウィンドウで、**Tenant *tenant_name*** > **Services** > **L4-L7** > **Inband Management Configuration for L4-L7 devices** を選択します。

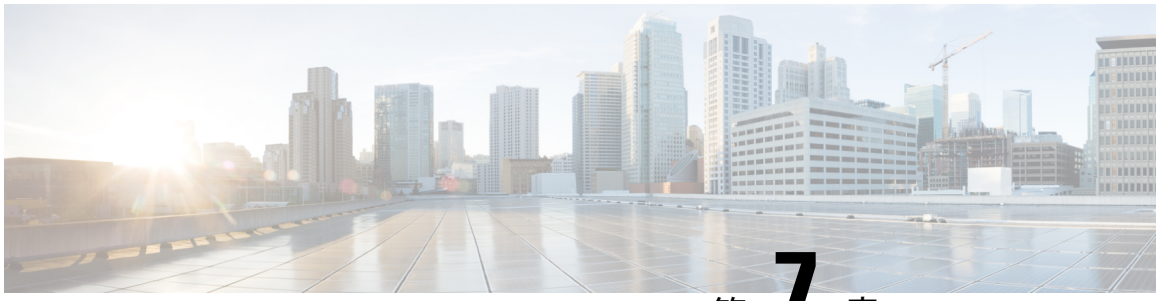
ステップ 7 [Work] ペインの [Controller Management Policies] セクションで、[+] をクリックし、次のようにフィールドに入力します。

a) [Private Networks] ドロップダウン リストで、プライベート ネットワークを選択します。

b) [Address Pool] ドロップダウンで、作成したばかりのプールを選択します。

ステップ 8 [Update] をクリックします。

これで、コントローラ管理ポリシーが追加されます。



第 7 章

グラフをレンダリングするレイヤ 4 ～ レイヤ 7 デバイスの選択

デバイス選択ポリシーについて

デバイスは、コントラクト名、グラフ名、またはグラフ内の機能ノード名に基づいて選択できます。デバイスを作成した後は、デバイスに選択条件ポリシーを提供するデバイス コンテキストを作成できます。

デバイス選択ポリシー（デバイス コンテキストとも呼ばれる）は、サービス グラフ テンプレートのデバイスを選択するためのポリシーを指定します。これにより、管理者は複数のデバイスを持つことができ、それらを異なるサービス グラフ テンプレートに対して使用することができます。たとえば、管理者は、高いパフォーマンス ADC アプライアンスがあるデバイスと、パフォーマンスが低い ADC アプライアンスがある別のデバイスを持つことができます。高いパフォーマンスの ADC デバイス用と低いパフォーマンスの ADC デバイス用の 2 つの異なるデバイス選択ポリシーを使用して、管理者は高いパフォーマンスが必要となるアプリケーションには高いパフォーマンスの ADC デバイスを選択し、低いパフォーマンスが必要なアプリケーションには低いパフォーマンスの ADC デバイスを選択することができます。

GUI を使用したデバイス選択ポリシーの作成

Apply L4-L7 Service Graph Template To EPGs ウィザードを使用せずにサービス グラフ テンプレートを適用した場合には、デバイス選択ポリシー(論理デバイス コンテキストとも呼ばれる)を設定することが必要になる可能性があります。デバイス選択ポリシーは Cisco Application Centric Infrastructure (ACI) に対し、グラフのレンダリングのためにどのファイアウォールやロード バランサを使用するかを指定します。

Apply L4-L7 Service Graph Template To EPGs ウィザードを使用してサービス グラフ テンプレートを適用した場合には、デバイス選択ポリシーは自動的に設定されるので、手動での設定を行う必要はありません。

（注） NX OS スタイルの CLI を使用すると、デバイス選択ポリシーは自動的に設定されますが、同等の NX-OS スタイルの CLI コマンドはありません。

すでに導入されているサービス グラフ テンプレートにコピー デバイスを追加する場合には、コピー サービスのために使用するデバイス選択ポリシーを作成する必要があります。

ステップ 1 メニュー バーで、**[Tenants]** > **[All Tenants]** の順に選択します。

ステップ 2 [Work] ペインで、テナントの名前をダブルクリックします。

ステップ 3 [Navigation] ウィンドウで、**Tenant** *tenant_name* > **Services** > **L4-L7** > **Devices Selection Policies**.

ステップ 4 [Work] ペインで、**[Actions]** > **[Create Logical Device Context]** の順に選択します。

ステップ 5 [Create Logical Device Context] ダイアログボックスで、下記で指定している項目を除き、必要に応じてフィールドに入力します。

a) [Contract Name] ドロップダウンリストで、デバイス選択ポリシーの契約を選択します。デバイスを使用する条件の一部として契約名を使用しない場合は、**any** を選択します。

b) **Graph Name** ドロップダウンリストで、デバイス選択ポリシーのためのグラフを選択します。デバイスを使用する条件の一部としてグラフ名を使用しない場合は、**any** を選択します。

c) **Node Name** ドロップダウンリストで、デバイス選択ポリシーのためのノードを選択します。デバイスを使用する条件の一部としてグノード名を使用しない場合は、[any] を選択します。

ステップ 6 [Cluster Interface Contexts] セクションの [+] をクリックしてクラスタ インターフェイス コンテキストを追加します。

プロパティ	説明
Connector Name	コネクタの名前または論理インターフェイス コンテキストのラベルです。デフォルトは **Any** です。
Logical Interface	論理インターフェイスの識別子です。
ブリッジ ドメイン	物理または仮想ポートのセットで構成される、プライベート レイヤ 2 のブリッジ ドメインです。コピーデバイスの場合には、ブリッジ ドメインは作成しないでください。ブリッジ ドメインは内部的に作成されます。
L3 Network	レイヤ 3 コンテキストの名前です。コピー デバイスの場合には、レイヤ 3 ネットワークは選択しないでください。
L4-L7 Policy based Routing	論理デバイス コンテキストで使用する、ポリシー ベースのリダイレクト ポリシーです。コピー デバイスの場合には、ポリシー ベースのリダイレクト ポリシーは選択しないでください。

プロパティ	説明
Permit Logging	インターフェイス コンテキストの許可ロギングのステータス。デフォルトは [false] です。

ステップ 7 [Create A Cluster Interface Context] ダイアログボックスで次のプロパティを設定します。

プロパティ	説明
Connector Name	コネクタの名前または論理インターフェイス コンテキストのラベルです。デフォルトは **Any** です。
Cluster Interface	ターゲット インターフェイスの一意の名前。 （注） このフィールドは必須です。
ブリッジ ドメイン	ターゲットに関連付けられたネットワークを入力します。 エニーキャストの場合は、ノードに使用するブリッジ ドメインと同じである必要があります。 （注） ターゲットに関連付けられたネットワークは、ブリッジ ドメインまたは L3 ネットワークのいずれかである必要があります。
L3 Network	ターゲットに関連付けられたネットワークを入力します。 （注） ターゲットに関連付けられたネットワークは、ブリッジ ドメインまたは L3 ネットワークのいずれかである必要があります。

プロパティ	説明
L3 Destination (VIP)	この論理インターフェイスがサービス チェーンの L3 トラフィックを終端するかどうかを示します。 このパラメータのデフォルトは有効（オン）です。ただし、論理インターフェイス コンテキストにポリシーベース リダイレクト ポリシーが設定されている場合、この設定は考慮されません。 （注） マルチノードPBRでは、この論理インターフェイスが仮想 IP 外部ネットワークで終端されるロード バランサのコンシューマ構築の場合、このボックスをオンにして、次のフィールド（[L4-L7 Policy Based Redirect]）でリダイレクト ポリシーへの関連付けを削除します。 この論理インターフェイスがロード バランサのプロバイダー構築で、かつSNATを実行している場合は、このボックスをオンにして、次のフィールド（[L4-L7 Policy Based Redirect]）でリダイレクト ポリシーへの関連付けを削除します。
L4-L7 Policy Based Redirect	オプション。ポリシーベース リダイレクト ポリシーを指定するか、[Create L4-L7 Policy Based Redirect] を選択します。 （注） マルチノードPBRでは、この論理インターフェイスが仮想 IP 外部ネットワークで終端されるロード バランサのコンシューマ構築の場合、リダイレクト ポリシー（入力されている場合）への関連付けを削除して、[L3 Destination (VIP)] ボックスをオンにします。
Custom QoS Policy	オプション。カスタムQoSポリシーまたはデフォルト ポリシーを指定するか、[Create Custom QoS Policy] を選択します。

プロパティ	説明
Preferred Contract Group	優先グループポリシーの適用タイプ。有効なタイプは次のとおりです。 • [Include]：このポリシー オプションで設定されたEPGまたはインターフェイスはサブグループに含まれ、サブグループ内で契約なしで通信できます。 • [Exclude]：このポリシー オプションで設定されたEPGまたはインターフェイスはサブグループに含まれず、サブグループ内で契約なしで通信することはできません。
Permit Logging	インターフェイス コンテキストの許可ロギングを有効にします。 デフォルトは無効（false）です。
Subnets	[+] をクリックしてサブネットを追加します。 ゲートウェイアドレス、サブネットのネットワーク可視性（範囲）、プライマリ IP アドレス（優先サブネット）、およびサブネット制御の状態を設定します。
仮想 IP アドレス	このサブネットを L3 仮想宛先に使用する（[L3 Destination (VIP)] がオンになっている）場合は、[+] をクリックして仮想 IP アドレス（VIP）を追加します。

ステップ 8 [OK] をクリックします。

ステップ 9 [Submit] をクリックします。

REST API を使用したデバイス選択ポリシーの設定

REST API を使用してデバイス選択ポリシーを設定することができます。

REST API を使用してデバイス選択ポリシーの作成

次の REST API ではデバイス選択ポリシーを作成します。

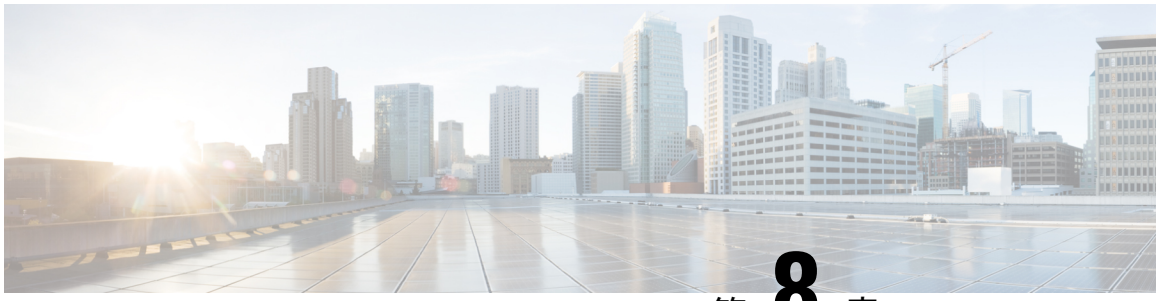
```
<polUni>
    <fvTenant dn="uni/tn-acme" name="acme">
       <vnsLDevCtx ctrctNameOrLbl="webCtrct" graphNameOrLbl="G1" nodeNameOrLbl="Node1">
```

```
            <vnsRsLDevCtxToLDev tDn="uni/tn-acme/lDevVip-ADCCluster1"/>

             <!-- The connector name C4, C5, etc.. should match the
                  Function connector name used in the service graph template -->

             <vnsLIfCtx connNameOrLbl="C4">
                <vnsRsLIfCtxToLIf tDn="uni/tn-acme/lDevVip-ADCCluster1/LIf-ext"/>
            </vnsLIfCtx>
            <vnsLIfCtx connNameOrLbl="C5">
                <vnsRsLIfCtxToLIf tDn="uni/tn-acme/lDevVip-ADCCluster1/LIf-int"/>
            </vnsLIfCtx>
        </vnsLDevCtx>
    </fvTenant>
</polUni>
```

REST API を使用したデバイスでの論理インターフェイスの追加

次の REST API はデバイス内に論理インターフェイスを追加します。

```
<polUni>
    <fvTenant dn="uni/tn-acme" name="acme">
        <vnsLDevVip name="ADCCluster1">

            <!-- The LIF name defined here (such as e.g., ext, or int) should match the

                 vnsRsLIfCtxToLIf 'tDn' defined in LifCtx  -->

            <vnsLIf name="ext">

                <vnsRsMetaIf tDn="uni/infra/mDev-Acme-ADC-1.0/mIfLbl-outside"/>
                <vnsRsCIfAtt tDn="uni/tn-acme/lDevVip-ADCCluster1/cDev-ADC1/cIf-ext"/>
            </vnsLIf>
            <vnsLIf name="int">
                <vnsRsMetaIf tDn="uni/infra/mDev-Acme-ADC-1.0/mIfLbl-inside"/>
                <vnsRsCIfAtt tDn="uni/tn-acme/lDevVip-ADCCluster1/cDev-ADC1/cIf-int"/>
            </vnsLIf>
        </vnsLDevVip>
    </fvTenant>
</polUni>
```



第 8 章

サービス グラフの設定

サービス グラフについて

Cisco Application Centric Infrastructure（ACI）はアプリケーションの重要部分としてサービスを見なします。必要なサービスは、Cisco Application Policy Infrastructure Controller（APIC）からのACIファブリックでインスタンス化されたサービス グラフとして処理されます。ユーザは、アプリケーションに対してサービスを定義し、サービス グラフはアプリケーションが必要とする一連のネットワークまたはサービス機能を識別します。

サービス グラフは、次の要素を使ってネットワークを表します。

- 機能ノード：機能ノードは、トランスフォーム（SSL ターミネーション、VPN ゲートウェイ）、フィルタ（ファイアウォール）、または端末（侵入検知システム）など、トラフィックに適用される機能を表します。サービス グラフ内の 1 つの機能は 1 つ以上のパラメータを必要とし、1 つまたは複数のコネクタを持っている場合があります。
- 端末ノード：端末ノードはサービス グラフからの入出力を有効にします。
- コネクタ：コネクタはノードからの入出力を有効にします。
- 接続：接続によって、ネットワーク経由でトラフィックを転送する方法が決定されます。

グラフが APIC に設定されると、APIC はサービス グラフに明記されたサービス機能の要件に従って、サービスを自動的に設定します。APIC はまた、サービス グラフで指定されるサービス機能のニーズに応じてネットワークを自動的に設定しますが、これによってサービスデバイスでの変更は要求されません。

サービス グラフは、アプリケーションの複数の階層として表され、適切なサービス機能が間に挿入されます。

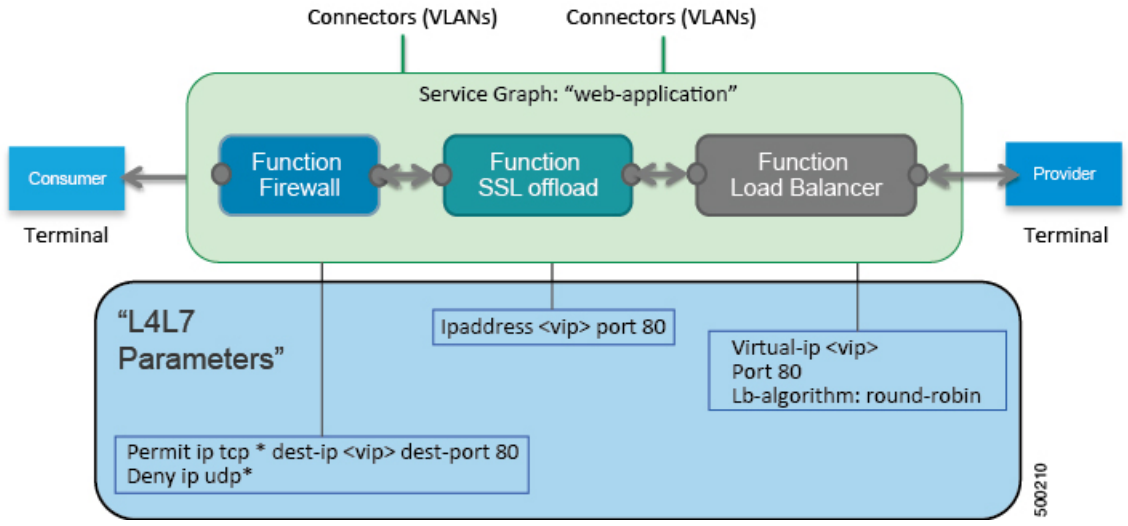
サービスアプライアンス（デバイス）は、グラフ内でサービス機能を実行します。1つ以上のサービスアプライアンスが、グラフに必要なサービスをレンダリングするために必要になることがあります。1 つ以上のサービス機能が単一のサービス デバイスで実行できます。

サービス グラフおよびサービス機能には、次の特性があります。

- エンドポイント グループで送受信されたトラフィックはポリシーに基づいてフィルタリングでき、トラフィックのサブセットはグラフ内の異なるエッジにリダイレクトできます。
- サービス グラフのエッジには方向性があります。
- タップ（ハードウェア ベースのパケット コピー サービス）は、サービス グラフの異なるポイントに接続できます。
- 論理機能は、ポリシーに基づいて適切な（物理または仮想）デバイスでレンダリングできます。
- サービス グラフでは、エッジの分割と結合がサポートされ、管理者は線形サービス チェーンに制限されません。
- トラフィックは、サービスアプライアンスが発信した後にネットワーク内で再度分類できます。
- 論理サービス機能は、要件に応じて、拡張や縮小が可能で、クラスタ モードまたは 1:1 アクティブ/スタンバイ ハイアベイラビリティ モードで展開できます。

次の図は、サービスグラフの導入の例を示しています:

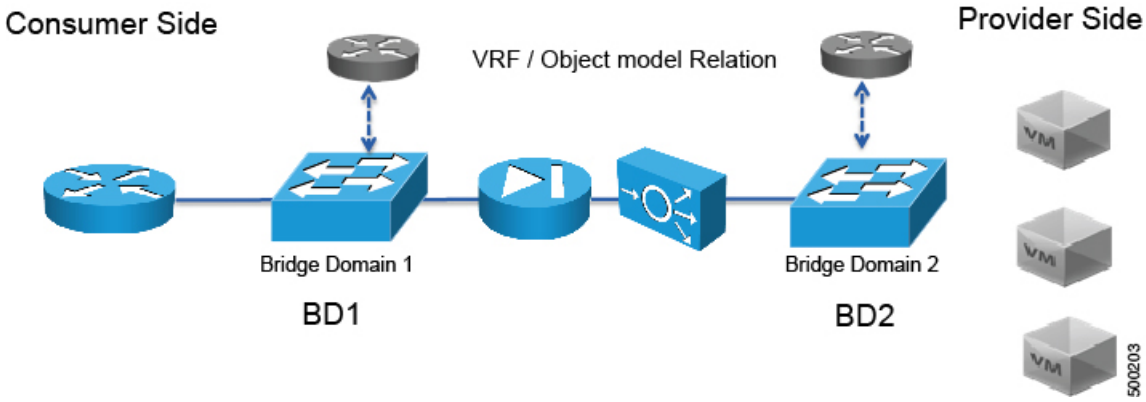
図 *4 :* サービス グラフの展開の例



サービス グラフを使用すると、ASA ファイアウォールなどのサービスを一度インストールして、異なる論理トポロジで何度も展開できます。グラフを導入するたびに、ACI は新しい論理トポロジでの転送を行えるように、ファイアウォールで設定の変更を行います。

サービス グラフを展開する必要がありますブリッジ ドメインと Vrf では、次の図に示すように。

図 *5 :* ブリッジ ドメインおよびサービス グラフの *Vrf*



（注） 使用すると、その他のテナント内のエンドポイント グループに関連付けられているサービス グラフの脚の一部があるかどうか、 **グラフ テンプレートの関連のオブジェクトを削除** GUI で、機能、APIC 以外のテナントからインポートされた契約は削除されませんサービス グラフが存在します。APICもサービス グラフよりも異なるテナントにあるエンドポイント グループ契約のクリーニングはありません。手動で異なるテナントではこれらのオブジェクトを削除する必要があります。

機能ノードについて

機能ノードは、単一のサービス機能を表します。機能ノードには、サービス機能のネットワーク要件を表す機能ノード コネクタがあります。

サービス グラフ内の機能ノードは、1つ以上のパラメータが必要になる場合があります。パラメータは、エンドポイント グループ（EPG）、アプリケーション プロファイル、またはテナント VRF により指定できます。パラメータは、サービス グラフ定義時に割り当てることができます。パラメータ値は、変更がさらに加えられるのを防ぐためにロックできます。

機能ノード コネクタについて

機能ノード コネクタは、サービス グラフに機能ノードを接続し、グラフのコネクタ サブネットに基づいて適切なブリッジ ドメインと接続と関連付けられます。各コネクタは、VLAN または Virtual Extensible LAN（VXLAN）に関連付けられます。コネクタの両側がエンドポイント グループ（EPG）として扱われ、ホワイトリストがスイッチにダウンロードされ、2 つの機能ノード間の通信がイネーブルになります。

サービス グラフ接続について

サービス グラフ接続は、1 つの機能ノードを別の機能ノードに接続します。

端末ノードについて

端末ノードはサービス グラフとコントラクトを接続します。コントラクトに端末ノードを接続することにより、2台のアプリケーション エンドポイント グループ（EPG）間のトラフィックにサービス グラフを挿入できます。接続されると、コントラクトのコンシューマ EPG とプロバイダー EPG 間のトラフィックはサービス グラフにリダイレクトされます。

サービスグラフテンプレートのコンフィギュレーションパラメータについて

サービス グラフテンプレートは、デバイス パッケージによって指定される、コンフィギュレーション パラメータを持つことができます。コンフィギュレーション パラメータは、EPG、アプリケーション プロファイルまたはテナント コンテキストでも指定できます。サービス グラフテンプレート内の機能ノードでは、1 つ以上のコンフィギュレーション パラメータが必要になる場合があります。パラメータ値は変更がさらに加えられるのを防ぐためにロックできます。

サービス グラフ テンプレートを設定してコンフィギュレーション パラメータの値を指定すると、Application Policy Infrastructure Controller（APIC）はデバイス パッケージ内のデバイス スクリプトにパラメータを渡します。デバイス スクリプトは、パラメータ データをデバイスにダウンロードされる設定に変換します。

GUI を使用したサービス グラフ テンプレートの設定

GUI を使用して、サービス グラフ テンプレートを設定することができます。

サービス グラフテンプレートを設定する手順については、GUI の使用方法 （237 ページ）を参照してください。

REST API を使用したサービス グラフテンプレートの作成

次の REST API を使用してサービス グラフ テンプレートを作成することができます。

```
<polUni>
  <fvTenant name="acme">
    <vnsAbsGraph name="G1">
      <vnsAbsTermNodeCon name="Input1">
        <vnsAbsTermConn name="C1">
                </vnsAbsTermConn>
      </vnsAbsTermNodeCon>
      <vnsAbsNode name="Node" funcType="GoTo">
        <vnsRsDefaultScopeToTerm
          tDn="uni/tn-acme/AbsGraph-G1/AbsTermNodeProv-Output1/outtmnl"/>
        <vnsAbsFuncConn name="inside">
          <vnsRsMConnAtt
            tDn="uni/infra/mDev-Insieme-Generic-1.0/mFunc-SubnetFunc/mConn-external"/>
        </vnsAbsFuncConn>
        <vnsAbsFuncConn name="outside">
          <vnsRsMConnAtt
            tDn="uni/infra/mDev-Insieme-Generic-1.0/mFunc-SubnetFunc/mConn-internal"/>
        </vnsAbsFuncConn>
        <vnsAbsDevCfg>
          <vnsAbsFolder key="oneFolder" name="f1">
            <vnsAbsParam key="oneParam" name="p1" value="v1"/>
          </vnsAbsFolder>
```

```
        </vnsAbsDevCfg>
        <vnsAbsFuncCfg>
          <vnsAbsFolder key="folder" name="folder1" devCtxLbl="C1">
            <vnsAbsParam key="param" name="param" value="value"/>
          </vnsAbsFolder>
          <vnsAbsFolder key="folder" name="folder2" devCtxLbl="C2">
            <vnsAbsParam key="param" name="param" value="value"/>
          </vnsAbsFolder>
        </vnsAbsFuncCfg>
        <vnsRsNodeToMFunc tDn="uni/infra/mDev-Insieme-Generic-1.0/mFunc-SubnetFunc"/>
      </vnsAbsNode>
      <vnsAbsTermNodeProv name="Output1">
        <vnsAbsTermConn name="C6">
                </vnsAbsTermConn>
      </vnsAbsTermNodeProv>
      <vnsAbsConnection name="CON1">
        <vnsRsAbsConnectionConns
          tDn="uni/tn-acme/AbsGraph-G1/AbsTermNodeCon-Input1/AbsTConn"/>
        <vnsRsAbsConnectionConns
tDn="uni/tn-acme/AbsGraph-G1/AbsNode-Node/AbsFConn-inside"/>
      </vnsAbsConnection>
      <vnsAbsConnection name="CON3">
        <vnsRsAbsConnectionConns
tDn="uni/tn-acme/AbsGraph-G1/AbsNode-Node/AbsFConn-outside"/>
        <vnsRsAbsConnectionConns
          tDn="uni/tn-acme/AbsGraph-G1/AbsTermNodeProv-Output1/AbsTConn"/>
      </vnsAbsConnection>
    </vnsAbsGraph>
  </fvTenant>
</polUni>
```

NX-OS スタイルの CLI を使用したサービス グラフの設定

NX-OS スタイルの CLI を使用して、サービス グラフを設定することができます。

ステップ 1 コンフィギュレーション モードを開始します。

例：

```
apic1# configure
```

ステップ 2 テナントのコンフィギュレーション モードを開始します。

```
tenant tenant_name
```

例：

```
apic1(config)# tenant t1
```

ステップ 3 サービス グラフを追加します。

```
l4l7 graph graph_name [contract contract_name]
```

パラメータ	説明
グラフ	サービス グラフの名前。

パラメータ	説明
contract	このサービス グラフ インスタンスに関連付けられたコントラクトの名前。サービス グラフ インスタンスを作成する場合にのみ、コントラクトを指定します。インスタンス化せずに（サービス グラフ テンプレートと同様に）簡単にサービス グラフを設定できます。

例：

```
apic1(config-tenant)# l4l7 graph G2 contract C2
```

ステップ 4 サービス グラフにノード（サービス）を追加します。

```
service node_name [device-cluster-tenant tenant_name] [device-cluster device_name] [mode
deployment_mode]
```

パラメータ	説明
service	追加するサービス ノードの名前。
device-cluster-tenant	デバイス クラスタのインポート元のテナント。グラフを設定するテナントと同じテナントにデバイス クラスタがない場合にのみ、このパラメータを指定します。
device-cluster	このサービス ノードに使用するデバイス クラスタの名前。
mode	導入モード。値は次のとおりです。 • ADC_ONE_ARM：ワンアーム モードを指定します。 • ADC_TWO_ARM：ツーアーム モードを指定します。 • FW_ROUTED：ルーテッド（GoTo）モードを指定します。 • FW_TRANS：トランスペアレント（GoThrough）モードを指定します。 • OTHERS：他の導入モードを指定します。 モードを指定しないと、導入モードは使用されません。

例：

次に、ノード N1 をテナント t1 からデバイス クラスタ D4 に追加する例を示します。

```
apic1(config-graph)# service N1 device-cluster-tenant t1 device-cluster D4
```

次に、ノード N1 をテナント t1 からデバイス クラスタ D4 に追加し、ルーテッド導入モードを使用する例を示します。

```
apic1(config-graph)# service N1 device-cluster-tenant t1 device-cluster D4 mode FW_ROUTED
```

ステップ 5 コンシューマ コネクタを追加します。

```
connector connector_type [cluster-interface interface_type]
```

パラメータ	説明
コネクタ	サービス グラフ内のコネクタのタイプ。値は次のとおりです。 • provider • consumer
cluster-interface	デバイス クラスタ インターフェイスのタイプ。値は次のとおりです。 • provider • consumer テナント Common 内のサービス グラフ テンプレートの場合は、このパラメータを指定しないでください。

例：

```
apic1(config-service)# connector consumer cluster-interface consumer
```

ステップ 6 ブリッジ ドメイン情報と、そのブリッジ ドメインが存在するテナントを指定し、コネクタにブリッジ ドメインを設定します。

```
bridge-domain tenant tenant_name name bridge_domain_name
```

パラメータ	説明
テナント	ブリッジ ドメインを所有するテナント。同じテナントまたはテナント Common からのみ、ブリッジを指定できます。たとえば、テナント t1 の場合、テナント t2 からのブリッジ ドメインは指定できません。
name	ブリッジ ドメインの名前。

例：

```
apic1(config-connector)# bridge-domain tenant t1 name bd2
```

ステップ 7 （任意） コネクタの Direct Server Return（DSR）仮想 IP アドレス（VIP）を設定します。

```
dsr-vip ip_address
```

DSR VIP を指定した場合、Application Policy Infrastructure Controller（APIC）は VIP を取得しません。

パラメータ	説明
dsr-vip	コネクタの DSR の仮想 IP アドレス。

例：

```
apic1(config-connector)# dsr-vip 192.168.10.100
```

ステップ 8 コンシューマとプロバイダーに対する接続を設定して、サービス グラフ コンフィギュレーション モードを終了します。

```
connection connection_name {terminal terminal_type service node_name connector connector_type} |
  {intra_service service1 node_name connector1 connector_type service2 node_name connector2
```

```
connector_type}
exit
```

パラメータ	説明
connection	接続の名前。
terminal	サービス ノードを端末に接続します。端末のタイプを指定します。値は次のとおりです。 • provider • consumer
service service1 service2	追加するサービス ノードの名前。service は terminal でのみ使用し、service1 と service2 は、intra_service でのみ使用します。
コネクタ connector1 connector2	コネクタのタイプ。値は次のとおりです。 • provider • consumer connector は terminal でのみ使用し、connector1 と connector2 は intra_service でのみ使用します。
intra_service	別のノードにサービス ノードを接続します。

例：

次に、単一ノード グラフの接続を設定する例を示します。

```
apic1(config-graph)# connection CON1 terminal consumer service N1 connector consumer
apic1(config-graph)# connection CON2 terminal provider service N2 connector provider
apic1(config-graph)# exit
```

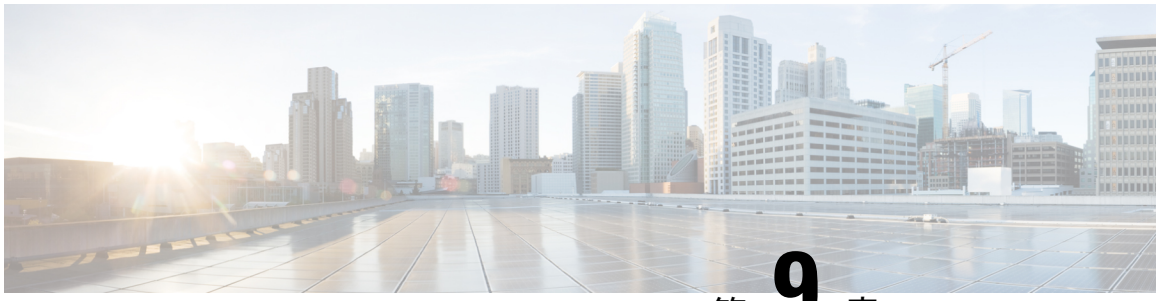
次に、2 ノード グラフの接続を設定する例を示します。

```
apic1(config-graph)# connection CON1 terminal consumer service N1 connector consumer
apic1(config-graph)# connection CON2 intra_service service1 N1 connector1 provider service2 N2
connector2 consumer
apic1(config-graph)# connection CON3  terminal provider service N2 connector provider
apic1(config-graph)# exit
```

ステップ 9 コンフィギュレーション モードを終了します。

例：

```
apic1(config-tenant)# exit
apic1(config)# exit
```



第 9 章

ルート ピアリングの設定

ルート ピアリングについて

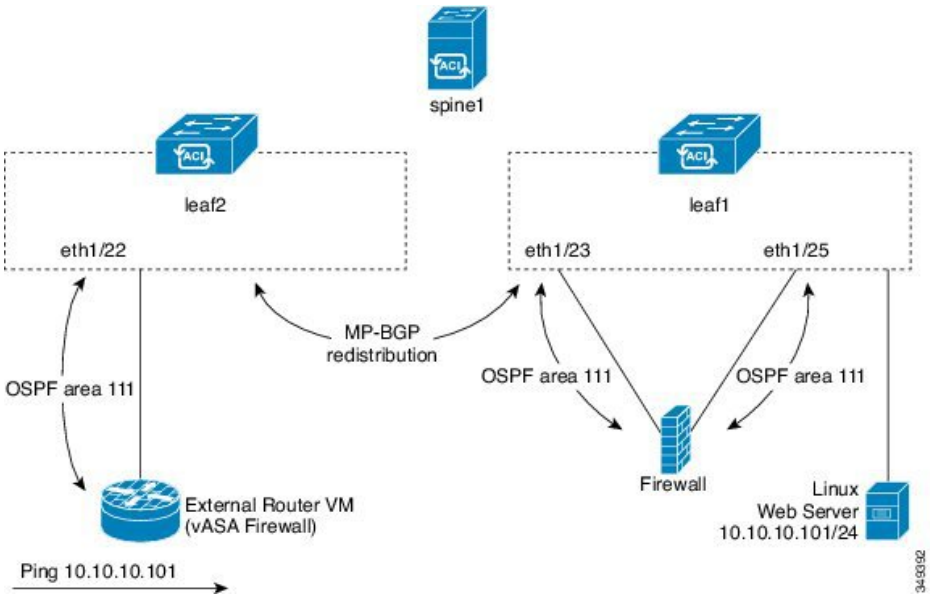
ルート ピアリングは、トランジットの使用例としてより一般的なCisco Application Centric Infrastructure（ACI）ファブリックの特殊ケースで、ルート ピアリングによってACI ファブリックが Open Shortest Path First（OSPF）プロトコルまたは Border Gateway Protocol（BGP）プロトコルのトランジット ドメインとして機能できるようになります。ルート ピアリングの一般的な使用例はルート ヘルス インジェクションであり、サーバのロード バランシング仮想 IP が OSPF または内部 BGP（iBGP）を使用して、ACI ファブリック外にあるクライアントにアドバタイズされます。デバイスが接続されている ACI リーフ スイッチとピアリングしたり、ルートを交換したりできるように、ルート ピアリングを使用して OSPF ピアリングや BGP ピアリングをサーバ デバイス上に設定したりすることができます。

次のプロトコルは、ルート ピアリングをサポートしています。

- OSPF
- OSPFv3
- iBGPv4
- iBGPv6
- スタティック ルート

次の図に、ルート ピアリングの一般的な導入方法を示します。

図 *6:* 一般的なルート ピアリング トポロジ



図に示すように、ルート ピアリングを設定してサービス グラフを導入することによって、Web サーバのパブリック IP アドレスがファイアウォールを介して外部ルータにアドバタイズされます。ファイアウォールの各レッグに OSPF ルーティング ポリシーを導入する必要があります。通常、これを行うには、`l3extOut` ポリシーを導入します。これにより、Web サーバの到達可能性情報がファイアウォールを介してボーダー リーフ スイッチと外部ルータに OSPF でアドバタイズされるようになります。

ファブリック内のリーフ スイッチ間のルート配布は Multi-Protocol Border Gateway Protocol（MP-BGP）により内部的に実行されます。

ルート ピアリング トポロジのより詳しい例については、ルート ピアリングのエンドツーエンド フロー （75 ページ） を参照してください。

`l3extOut` ポリシーの設定の詳細については、『*Cisco Application Centric Infrastructure Fundamentals Guide*』を参照してください。

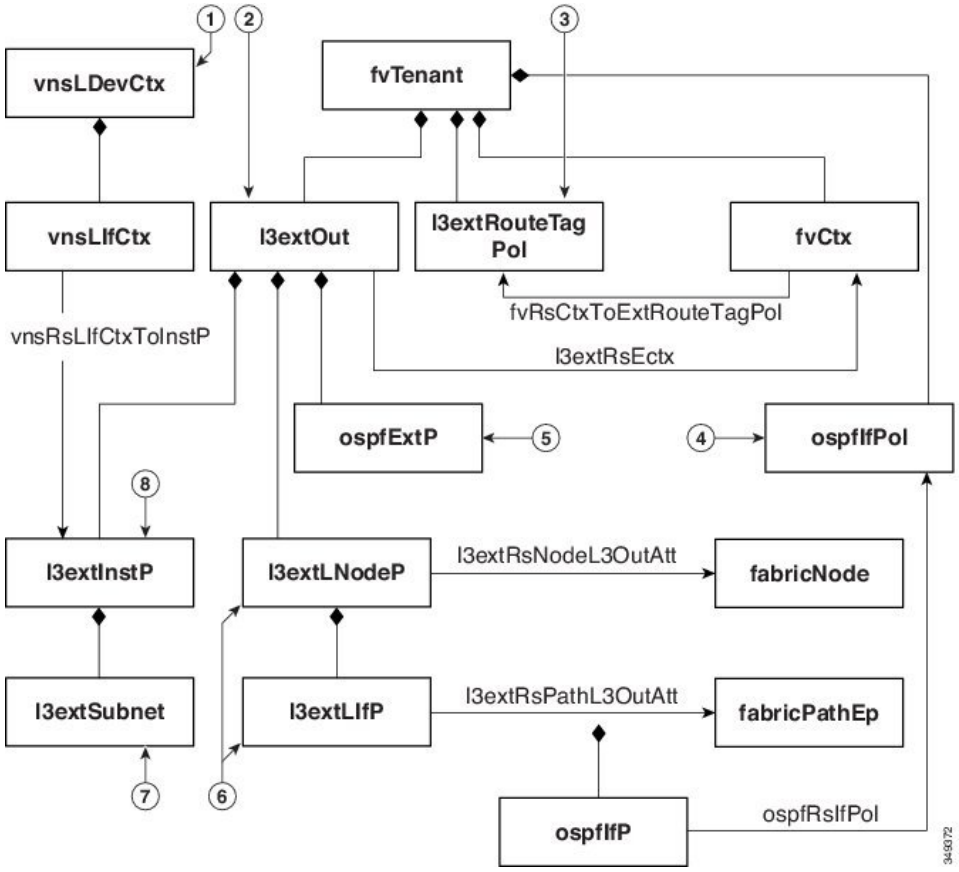
（注） ポイントツーポイントの非ブロードキャスト モードは、Adaptive Security Appliance （ASA）ではサポートされていません。Application Policy Infrastructure Controller（APIC）からポイントツーポイントの非ブロードキャスト モード設定を削除する必要があります（存在する場合）。

Open Shortest Path First ポリシー

ルート ピアリングを設定するには、最初に 1 つ以上の `l3extOut` ポリシーを作成し、サービス デバイスを接続するファブリック リーフ ノードに導入します。これらの `l3extOut` ポリシー

で、ファブリック リーフで有効にする必要がある Open Shortest Path First（OSPF）のパラメータを指定します。これらのポリシーは外部通信に使用される `l3extOut` ポリシーとよく似ています。次の図に、ルート ピアリング オブジェクトの関係を示します。

図 *7: OSPF* ルート ピアリング オブジェクトの関係



1. `vnsLDevCtx`：デバイス選択ポリシー。
2. `l3extOut`：1 つのエリアのすべての OSPF ポリシーが含まれます。
3. `l3extRouteTagPol`：ルート ピアリングで必要な各コンテキストには OSPF ループを回避するための一意のルート タグが必要です。1 つのレッグから取得される OSPF ルートは、ルート タグが異なっていない限り、他のレッグでは取得されません。
4. `ospfIfPol`：インターフェイスごとの OSPF ポリシー。
5. `ospfExtP`：エリア ポリシーごとの OSPF。
6. `l3extLNodeP/l3extLIfP`：この `l3extOut` を導入するノードまたはポート。
7. `l3extSubnet`：ファブリックに対してエクスポートまたはインポートするサブネット。
8. `l3extInstP`：プレフィックス ベースの EPG。

次に、l3extOut の 2 つの 例（OspfExternal と OspfInternal）を示します。これらのポリシーは、図 6 : 一般的なルート ピアリング トポロジ （66 ページ） のファイアウォール デバイスの外部レッグと内部レッグに導入されます。l3extOut ポリシーは、ファブリック リーフがトラフィックを分類する方法と、サービスデバイスに対してルートをインポートまたはエクスポートする方法も制御する 1 つ以上のプレフィックス ベースの EPG（l3extInstP）を指定します。l3extOut ポリシーには、そのポリシーの下で指定される OSPF のエリアごとのポリシー（ospfExtP）と 1 つ以上の OSPF インターフェイス ポリシー（ospfIfPol）が含まれています。

次に、値「100」で設定される area-Id を持つ OSPF エリアの例を示します。

```
<ospfExtP areaId="100" areaType="regular" areaCtrl="redistribute"/>
```

エリア タイプは「regular」に設定し、エリア制御属性は「redistribute」に設定します。

OSPF インターフェイス ポリシーで、1 つ以上の OSPF インターフェイス タイマーを指定します。

```
<ospfIfPol name="ospfIfPol" ctrl="mtu-ignore" nwT="bcast" xmitDelay="1" helloIntvl="10"
  deadIntvl="40" status="created,modified"/>
```

デフォルト タイマーが正常であれば、このポリシーを指定する必要はありません。このポリシーでは、特定のタイマーをデフォルト値から変更し、次の関係を使用することによって、1 つ以上のインターフェイスに関連付けることができます。

```
<l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/25]" ifInstT="ext-svi"
  encap="vlan-3844" addr="30.30.30.100/28" mtu="1500"/>
```

l3extRsPathL3OutAtt の関係の属性は次のとおりです。

- ifInstT：論理インターフェイス タイプ。通常は「ext-svi」。
- encap：このインターフェイスを作成するときは VLAN カプセル化を指定する必要があります。カプセル化はサービス デバイスにプッシュされます。
- addr：この l3extOut を導入するファブリック リーフで作成された SVI インターフェイスの IP アドレス。

次のポリシーで、l3extOut ポリシーをどこに導入するかを制御します。

```
<l3extNodeP name="bLeaf-101">
  <l3extRsNodeL30utAtt tDn="topology/pod-1/node-101" rtrId="180.0.0.11"/>
  <l3extLIfP name="portIf">
    <l3extRsPathL30utAtt tDn="topology/pod-1/paths-101/pathep-teth1/251"
      ifInstT="ext-svi" encap="vlan-3844" addr="30.30.30.100/28" mtu="1500"/>
    <ospfIfP authKey="tecom" authType="md5" authKeyId='1'>
      <ospfRslfPol tnOspfIfPolName="ospfIfPol"/>
    </ospflfP>
  </l3extLIfP>
</l3extLNodeP>
```

l3extOut ポリシーは、サービス デバイスが接続されているリーフ ポートと同じものに導入する必要があります。

scope=import-security 属性は次を実行します。

- データ プレーン内のトラフィックのフローを制御する
- このルートをアドバタイズする外部デバイスへのディレクティブとして機能する

（注） ルート ピアリングを正しく動作させるには、l3extRsPathL3OutAtt の関係が、デバイスを表す vnsCDev の下の RsCIfPathAtt の関係と同じファブリックの宛先を指している必要があります。

OspfExternal ポリシー

OspfInternal ポリシー

仮想サービス

```
<polUni>
  <fvTenant name="common">
    <fvCtx name="commonctx">
      <fvRsCtxToExtRouteTagPol tnL3extRouteTagPolName="myTagPol"/>
    </fvCtx>
    <l3extRouteTagPol tag="212" name="myTagPol"/>
    <l3extOut name="OspfExternal" status="created,modified">
      <l3extLNodeP name="bLeaf-101">
        <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="180.0.0.8/28"/>
        <l3extLIfP name="portIf">
          <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/23]"
            ifInstT="ext-svi" encap="vlan-3843" addr="40.40.40.100/28" mtu="1500"/>
          <ospfIfP authKey="tecom" authType="md5" authKeyId='1'>
            <ospfRsIfPol tnOspfIfPolName="ospfIfPol"/>
          </ospfIfP>
        </l3extLIfP>
      </l3extLNodeP>
      <ospfExtP areaId="100" areaType="regular" areaCtrl="redistribute"/>
      <l3extInstP name="ExtInstP">
        <l3extSubnet ip="40.40.40.100/28" scope="import-security"/>
        <l3extSubnet ip="10.10.10.0/24" scope="import-security"/>
      </l3extInstP>
      <l3extRsEctx tnFvCtxName="commonctx"/>
    </l3extOut>
   <ospfIfPol name="ospfIfPol" ctrl="mtu-ignore" nwT="bcast" xmitDelay="1" helloIntvl="10"

      deadIntvl="40" status="created,modified"/>
  </fvTenant>
</polUni>
```

```
<polUni>
  <fvTenant name="tenant1">
    <l3extRouteTagPol tag="213" name="myTagPol"/>
    <fvCtx name="tenant1ctx1">
      <fvRsCtxToExtRouteTagPol tnL3extRouteTagPolName="myTagPol"/>
    </fvCtx>
    <l3extOut name="OspfInternal" status="created,modified">
      <l3extLNodeP name="bLeaf-101">
        <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="180.0.0.11"/>
        <l3extLIfP name="portIf">
          <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/25]"
            ifInstT="ext-svi" encap="vlan-3844" addr="30.30.30.100/28" mtu="1500"/>
          <ospfIfP authKey="tecom" authType="md5" authKeyId='1'>
            <ospfRsIfPol tnOspfIfPolName="ospfIfPol"/>
          </ospfIfP>
        </l3extLIfP>
      </l3extLNodeP>
      <ospfExtP areaId="100" areaType="regular" areaCtrl="redistribute"/>
```

```
      <l3extInstP name="IntInstP">
        <l3extSubnet ip="30.30.30.100/28" scope="import-security"/>
        <l3extSubnet ip="20.20.20.0/24" scope="import-security"/>
      </l3extInstP>
      <l3extRsEctx tnFvCtxName="tenant1ctx1"/>
    </l3extOut>
   <ospfIfPol name="ospfIfPol" ctrl="mtu-ignore" nwT="bcast" xmitDelay="1" helloIntvl="10"

      deadIntvl="40" status="created,modified"/>
  </fvTenant>
</polUni>
```

`OspfExternalInstP` ポリシーは、プレフィクスの 40.40.40.100/28 と 10.10.10.0/24 をプレフィクス ベースのエンドポイントのアソシエーションに使用する必要があることを指定します。また、このポリシーは、プレフィックスの 20.20.20.0/24 をサービス デバイスにエクスポートするようにファブリックに指示します。

```
      <l3extInstP name="OspfExternalInstP">
        <l3extSubnet ip="40.40.40.100/28" scope="import-security"/>
        <l3extSubnet ip="10.10.10.0/24" scope="import-security"/>
        <l3extSubnet ip="20.20.20.0/24" scope="export"/>
      </l3extInstP>
```

`bleaf-101` ポリシーは、この `l3extOut` ポリシーを導入する場所を制御します。

```
      <l3extLNodeP name="bLeaf-101">
        <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="180.0.0.8/28"/>
        <l3extLIfP name="portIf">
          <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/23]"
            ifInstT="ext-svi" encap="vlan-3843" addr="40.40.40.100/28" mtu="1500"/>
          <!-- <ospfIfP authKey="tecom" authType="md5" authKeyId='1'> -->
          <ospfIfP>
            <ospfRsIfPol tnOspfIfPolName="ospfIfPol"/>
          </ospfIfP>
        </l3extLIfP>
      </l3extLNodeP>
```

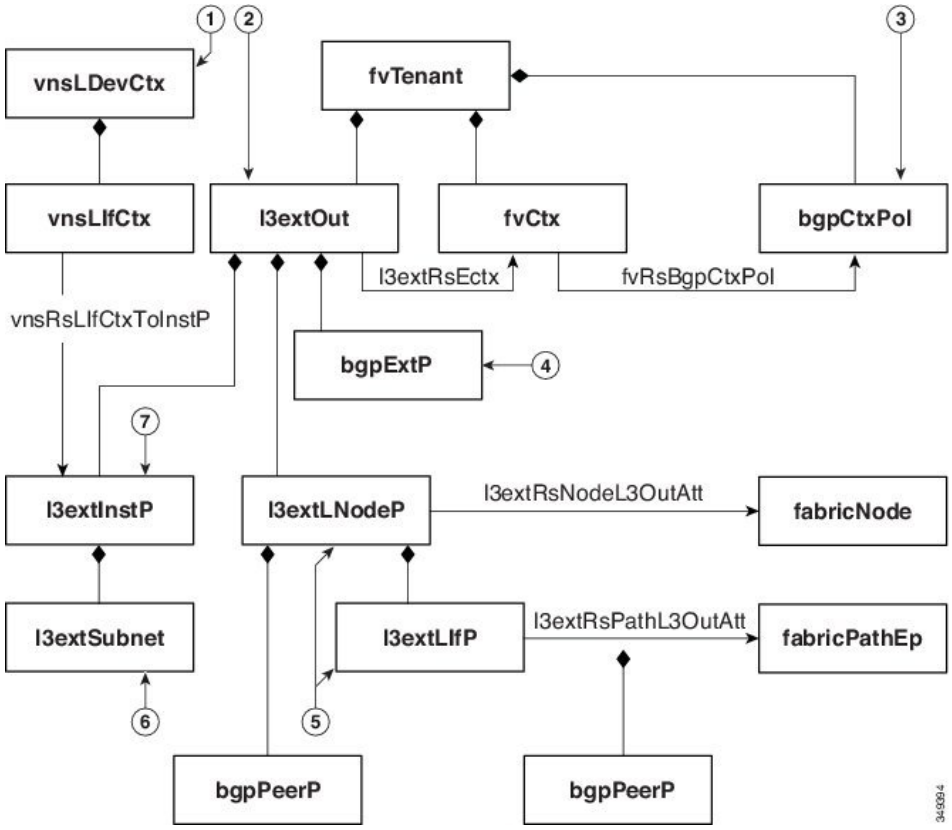
仮想サービスはルート ピアリングとともに導入できますが、`vnsCIf` オブジェクトでの `l3extRsPathL3OutAtt` 検証は実行されません。このデータ パスは、`l3extOut` オブジェクトが仮想サービス データが接続されている正しいリーフに導入されている場合にのみ動作します。

Border Gateway Protocol ポリシー

内部 Border Gateway Protocol（iBGP）を使用してデバイスの外部インターフェイスにルート ピアリングを設定し、内部インターフェイスに静的ルートを設定できます。追加設定なしにデバイスの内部インターフェイスと外部インターフェイスの両方に iBGP を設定することはできません。これは、インターフェイスが異なる自律システムに存在する必要があり、相互自律システム再配布ポリシーをプッシュ ダウンしないためです。

次の図に、ルート ピアリング オブジェクトの関係を示します。

図 *8 : iBGP* ルート ピアリング オブジェクトの関係



1. `vnsLDevCtx`：デバイス選択ポリシー。
2. `l3extOut`：単一の自律システム用のすべての BGP ポリシーが含まれます。
3. `bgpCtxPol`：コンテキスト単位の BGP タイマー。
4. `bgpExtP`：ASN ポリシー単位の BGP。
5. `l3extLIfP/l3extLNodeP`：これらのエンドポイント グループ（EPG）を導入するノードまたはポートを制御します。
6. `l3extSubnet`：ファブリックからのエクスポートするサブネットとファブリックにインポートするサブネット。
7. `l3extInstP`：プレフィックス ベースの EPG。

次のポリシーは、外部インターフェイスに iBGPv4/v6 を設定します。

```
<polUni>
  <fvTenant name="common">
    <fvCtx name="commonctx">
      <fvRsBgpCtxPol tnBgpCtxPolName="timer-3-9"/>
      <fvRsCtxToExtRouteTagPol tnL3extRouteTagPolName="myTagPol"/>
    </fvCtx>
    <l3extRouteTagPol tag="212" name="myTagPol"/>
   <bgpCtxPol grCtrl="helper" holdIntvl="9" kaIntvl="3" name="timer-3-9" staleIntvl="30"/>
```

```
    <l3extOut name="BgpExternal" status="created,modified">
      <l3extLNodeP name="bLeaf-101">
        <!-- <bgpPeerP addr="40.40.40.102/32 "ctrl="send-com"/> -->
        <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="180.0.0.8/28>"
          <l3extLoopBackIfP addr="50.50.50.100/32"/>
        </l3extRsNodeL3OutAtt>
        <l3extLIfP name="portIf">
          <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/23]"
            ifInstT="ext-svi" encap="vlan-3843" addr="40.40.40.100/28 "mtu="1500">
            <bgpPeerP addr="40.40.40.102/32 "ctrl="send-com"/>
          </l3extRsPathL3OutAtt>
        </l3extLIfP>
      </l3extLNodeP>
      <bgpExtP/>
      <l3extInstP name="ExtInstP">
        <l3extSubnet ip="40.40.40.100/28 "scope="import-security"/>
        <l3extSubnet ip="10.10.10.0/24 "scope="import-security"/>
        <l3extSubnet ip="20.20.20.0/24 "scope="export-rtctrl"/>
      </l3extInstP>
      <l3extRsEctx tnFvCtxName="commonctx"/>
    </l3extOut>
  </fvTenant>
</polUni>
```

iBGP ピアは、物理インターフェイス レベルまたはループバック レベルで設定できます。次に、物理インターフェイス レベルで設定された iBGP ピアの例を示します。

```
<l3extLIfP name="portIf">
          <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/23]"
            ifInstT="ext-svi" encap="vlan-3843" addr="40.40.40.100/28 "mtu="1500">
            <bgpPeerP addr="40.40.40.102/32 "ctrl="send-com"/>
          </l3extRsPathL3OutAtt>
</l3extLIfP>
```

この場合、ファブリック上で実行する iBGP プロセスはスイッチ仮想インターフェイス（SVI）IP アドレス 40.40.40.100/28 を使用して、ネイバーとピアリングします。ネイバーは、IP アドレス 40.40.40.102/32 のサービス デバイスです。

次に、iBGP ピアの定義が論理ノード レベル（`l3extLNodeP` の下）に移動され、ループバック インターフェイスが作成されている例を示します。

```
      <l3extLNodeP name="bLeaf-101">
        <bgpPeerP addr="40.40.40.102/32 "ctrl="send-com"/>
        <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="180.0.0.8/28>"
          <l3extLoopBackIfP addr="50.50.50.100/32"/>
        </l3extRsNodeL3OutAtt>
        <l3extLIfP name="portIf">
          <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/23]"
            ifInstT="ext-svi" encap="vlan-3843" addr="40.40.40.100/28 "mtu="1500">
          </l3extRsPathL3OutAtt>
        </l3extLIfP>
      </l3extLNodeP>
```

この例では、iBGP プロセスはループバック アドレスを使用してネイバーとピアリングします。ループバックが設定されていない場合は、ファブリックは rtrId で指定された IP アドレスを使用してネイバーとピアリングします。

この場合、デバイスには SVI に到達するルートが必要です。通常、これは、IP アドレス 50.50.50.0 が IP アドレス 40.40.40.100 から到達できる場合は、次の ASA の例に示すようにグラフ パラメータを使用して設定します。

```
<vnsAbsFolder name="ExtRouteCfg" key="StaticRoute">
              <vnsAbsFolder name="route1" key="route">
                <vnsAbsParam name="network" key="network" value="50.50.50.0"/>
                <vnsAbsParam name="netmask" key="netmask" value="255.255.255.0"/>
                <vnsAbsParam name="gateway" key="gateway" value="40.40.40.100"/>
              </vnsAbsFolder>
              <vnsAbsFolder name="route2" key="ipv6_route">
                <vnsAbsParam name="prefix" key="prefix" value="2005::/64"/>
                <vnsAbsParam name="gateway" key="gateway" value="2004::2828:2866"/>
              </vnsAbsFolder>
</vnsAbsFolder>
```

次に、デバイスの内部インターフェイス用にファブリック上で静的ルートを設定する例を示します。

```
<polUni>
  <fvTenant name="tenant11">
    <l3extOut name="StaticInternal" status="created,modified">
      <l3extLNodeP name="bLeaf-201">
        <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="180.0.0.11>"
          <ipRouteP ip="20.20.20.0/24>"
            <ipNexthopP nhAddr="30.30.30.102/32"/>
          </ipRouteP>
        </l3extRsNodeL3OutAtt>
        <l3extLIfP name="portIf">
          <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/25]"
            ifInstT="ext-svi" encap="vlan-3844" addr="30.30.30.100/28 "mtu="1500"/>
        </l3extLIfP>
      </l3extLNodeP>
      <l3extInstP name="IntInstP">
        <l3extSubnet ip="20.20.20.0/24 "scope="import-security"/>
      </l3extInstP>
      <l3extRsEctx tnFvCtxName="tenant1ctx1"/>
    </l3extOut>
  </fvTenant>
</polUni>
```

クラスタ用の L3extOut ポリシーの選択

特定の `l3extOut` ポリシーを、選択ポリシー `vnsLIfCtx` を使用して論理デバイスのインターフェイスに関連付けることができます。次に、これを実現する例を示します。

```
<vnsLDevCtx ctrctNameOrLbl="webCtrct1" graphNameOrLbl="WebGraph" nodeNameOrLbl="FW">
  <vnsRsLDevCtxToLDev tDn="uni/tn-tenant1/lDevVip-Firewall"/>
  <vnsRsLDevCtxToRtrCfg tnVnsRtrCfgName="FwRtrCfg"/>
  <vnsLIfCtx connNameOrLbl="internal">
    <vnsRsLIfCtxToInstP tDn="uni/tn-tenant1/out-OspfInternal/instP-IntInstP"
      status="created,modified"/>
    <vnsRsLIfCtxToLIf tDn="uni/tn-tenant1/lDevVip-Firewall/lIf-internal"/>
  </vnsLIfCtx>
  <vnsLIfCtx connNameOrLbl="external">
    <vnsRsLIfCtxToInstP tDn="uni/tn-common/out-OspfExternal/instP-ExtInstP"
      status="created,modified"/>
    <vnsRsLIfCtxToLIf tDn="uni/tn-tenant1/lDevVip-Firewall/lIf-external"/>
  </vnsLIfCtx>
</vnsLDevCtx>
```

`vnsRsLIfCtxToInstP` の関係を使用して、サービス デバイスのこのレッグと関連付ける特定のプレフィックス ベースの EPG（`l3extInstP`）を選択します。この関係に、`redistribute` プロトコル再配布プロパティを指定できます。`redistribute` プロパティのデフォルト値は「ospf,bgp」

です。redistributeをデフォルト値のままにすると、各レッグで設定されているルーティング プロトコルがApplication Policy Infrastructure Controller（APIC）によって自動検出され、適切な再配布設定にプッシュされます。自動設定は、常にInterior Gateway Protocol（OSPF）から外部ゲートウェイ プロトコル（BGP）に再配布します。

静的または接続済みといった特定の再配布設定を使用する場合は、それらの設定をこの関係に追加します。たとえば、redistribute="ospf,bgp,static" は、自動検出設定とredistribute-static をサービス デバイスにプッシュします。

このプロパティをデフォルト値を含まない特定の値（たとえば、redistribute="ospf,static,connected"）に設定すると、それらの設定がそのままサービス デバイスにプッシュされます。これは、APIC によって選択されたデフォルト値を上書きする場合に役に立ちます。

（注） この関係はl3extOut自体でなく、EPG（l3extInstP）を指します。これは、l3extOutポリシーにはこのようなEPGが複数存在する可能性があり、別のデバイス選択ポリシーがそれらのEPGを指していることがあるためです。これにより、さまざまなサービス グラフによってインポートまたはエクスポートされるプレフィックスを細かく制御できます。

vnsRsLDevCtxToRtrCfg関係を使用して、このデバイス セレクタに対して特定のvnsRtrCfgポリシーが選択されます。vnsRtrCfgポリシーは、Open Shortest Path First（OSPF）や内部ボーダーゲートウェイ プロトコル（IBGP）などのルーティング プロトコルで使用するルータ ID を指定するために必要です。これらのポリシーはユーザが指定する必要があります。このルータ ID はデバイスに送信されます。

次のコードで、vnsRtrCfgポリシーの例を示します。

```
<vnsRtrCfg name="FwRtrCfg" rtrId="180.0.0.10”/>
```

関連付けられた具象デバイスにはvnsRsCIfPathAttオブジェクトが必要です。このオブジェクトでは、デバイスを同じファブリック リーフに導入します（下記参照）。

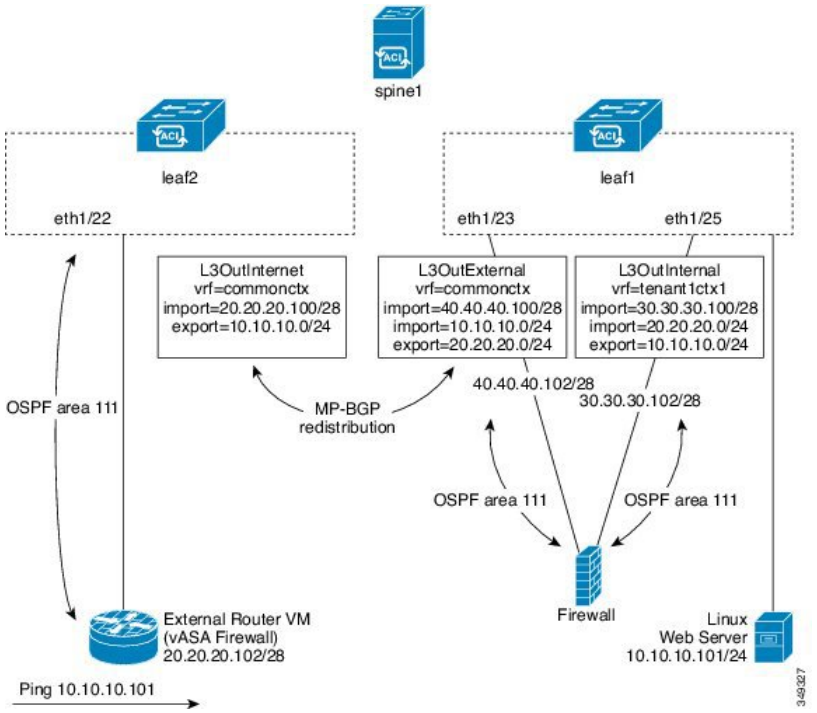
```
<vnsCDev name="ASA">
  <vnsCIf name="Gig0/0">
    <vnsRsCIfPathAtt tDn="topology/pod-1/paths-101/pathep-[eth1/23]"/>
  </vnsCIf>
  <vnsCIf name="Gig0/1">
    <vnsRsCIfPathAtt tDn="topology/pod-1/paths-101/pathep-[eth1/25]"/>
  </vnsCIf>
  <vnsCMgmt name="devMgmt" host="{{asaIp}}" port="443"/>
  <vnsCCred name="username" value="admin"/>
  <vnsCCredSecret name="password" value="insieme"/>
</vnsCDev>
```

（注） ルートピアリングを設定した場合は、vnsLIfCtxセレクタにブリッジ ドメインを設定する必要がありません。ブリッジ ドメインの関係（vnsRsLIfCtxToBD）とl3extInstPの関係（vnsRsLIfCtxToInstP）の両方を設定しようとすると、エラーになります。

ルート ピアリングのエンドツーエンド フロー

次の図に、ルート ピアリングがエンドツーエンドでどのように動作するかを示します。

図 *9*: ルート ピアリングのエンドツーエンド フロー



この図には、ルート ピアリングを使用して Linux Web サーバの IP アドレスが外部ルータにアドバタイズされる、単一スパイン スイッチ トポロジである 2 台のリーフ スイッチの例が示されています。Linux Web サーバは IP アドレス 10.10.10.101 にあり、`leaf1` に接続する ESX サーバ上でホストされています。通常のブリッジドメインベースのエンドポイント グループ（EPG）が導入されており、Web サーバから発信されるトラフィックを表しています。

2 アームのルーティング可能なファイアウォールから構成され、両方のアームを `leaf1` に接続したサービス グラフを導入します。ファイアウォール デバイスでは、Virtual Routing and Forwarding (VRF) 分割が行われています。つまり、ファイアウォールの各アームが異なる VRF のリーフ（コンテキスト）に接続されています。VRF 分割は、トラフィックがリーフ スイッチによって短絡されるのではなく、サービスデバイスを通じて確実にルーティングされるようにするために必要です。外部トラフィックは `leaf2` に導入されている `l3extOut`（`L3OutInternet`）で表されます。このシナリオでは、`leaf2` をファブリックの境界リーフ スイッチと見なすことができます。`L3OutInternet` と Web サーバ EPG 間にコントラクトを導入できます。このコントラクトは、ファイアウォール デバイスを含むサービス グラフに関連付けられます。

Web サーバ ルートを外部にパブリッシュするには、2 つの `l3extOut`（`L3OutExternal` と `L3OutInternal`）を、サービス デバイスを接続するリーフ スイッチ ポートに展開します。その結果、Open Shortest Path First (OSPF) ピアリング セッションが、両方のコンテキスト（`commonctx` と `tenant1ctx1`）のリーフ スイッチとファイアウォール間で確立されます。これらの `l3extOut`

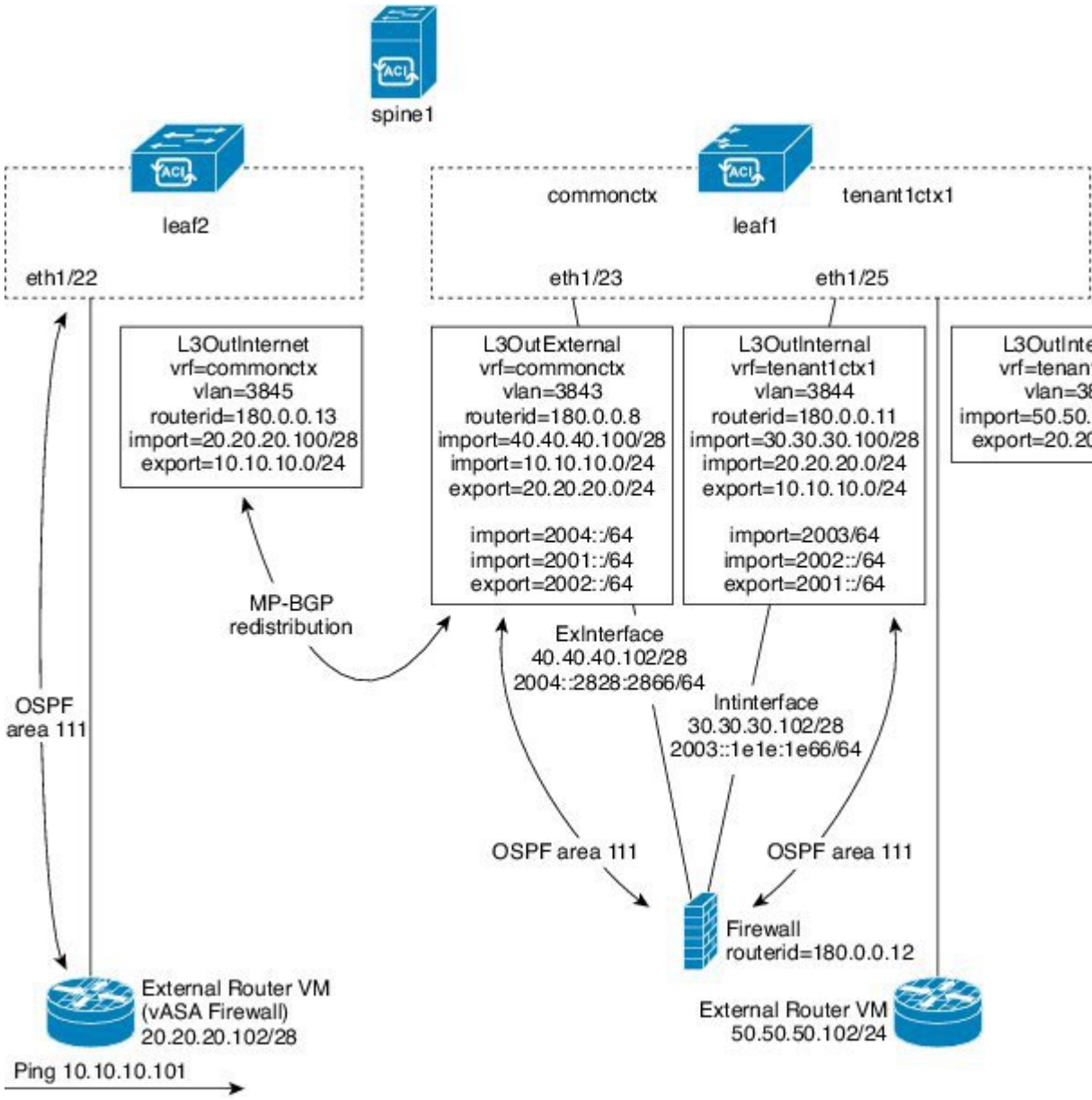
のexport属性が境界リーフ スイッチへのルーティング情報のアドバタイズ方法を制御します。ルートはマルチプロトコルBorder Gateway Protocol (MP-BGP) の再配布を使用して、ファブリック リーフ スイッチの間で内部的に交換されます。

最終的に、別の OSPF セッションを使用して Web サーバ ルートが外部ルータ（IP アドレス 20.20.20.102）にアドバタイズされます。これにより、静的ルートを手動で設定することなく、外部ルータから Web サーバを ping できるようになります。

Cisco Application Centric Infrastructure トランジット ルーティング ドメインとして機能するファブリック

Cisco Application Centric Infrastructure（ACI）ファブリックをトランジット ルーティング ドメインとして導入できるので、ACIの受渡しポイント（POD）が他のPOD間のトランジット ルーティング ドメインとして機能している場合に便利です。次の図に、2つの境界リーフ スイッチへの 2 つの 外部 `l3extOut`（`L3OutInternet` と `L3OutInternet2`）の展開を示します。これらの `3extOut` 間には関連付けられているコントラクトがあり、そのコントラクトはファイアウォール サービス デバイスを含む単一ノードのサービス グラフに適用されています。

図 *10 : ACI* トランジット ルーティング ドメインとして機能するファブリック



2 つの追加 `l3extOut` は、ファイアウォール デバイスの外部レッグと内部レッグに導入され、それらの間に Open Shortest Path First（OSPF）ピアリング セッションを確立します。インポート セキュリティ制御（`import-security` 属性）を適切に設定することで、境界リーフ スイッチへの ACI ファブリックの通過を許可するルートを制御できます。

GUI を使用したルート ピアリングの設定

ルート ピアリングを設定するには、次のタスクを実行する必要があります。

1. デバイスとCisco Application Centric Infrastructure（ACI）ファブリック間のカプセル化VLANに使用するスタティック VLAN プールを作成します。

   GUI を使用したスタティック VLAN プールの作成 （78 ページ）を参照してください。

2. デバイスの場所（リーフ ノード/パス）と VLAN プールを結びつける外部ルーテッド ドメインを作成します。

   GUI を使用した外部ルーテッド ドメインの作成 （78 ページ）を参照してください。

3. ルート ピアリングで ACI ファブリックのルーティング設定を指定するために使用する外部ルーテッド ネットワークを作成します。

   GUI を使用した外部ルーテッド ネットワークの作成 （79 ページ）を参照してください。

4. デバイスで使用するルータ ID を指定する新しいルータ設定を作成します。

   GUI を使用したルータ設定の作成 （82 ページ）を参照してください。

5. サービス グラフのアソシエーションを作成します。これには、外部ルーテッドネットワーク ポリシーおよびルータ設定とデバイス選択ポリシーの関連付けが含まれます。

   GUI を使用したサービス グラフ アソシエーションの作成 （82 ページ）を参照してください。

GUI を使用したスタティック VLAN プールの作成

外部ルーテッド ネットワーク設定を作成する前に、デバイスとファブリック間のカプセル化 VLAN に使用するスタティック VLAN プールを作成する必要があります。

ステップ 1 メニュー バーで、**[Fabric]** > **[Access Policies]** の順に選択します。

ステップ 2 [Navigation] ペインで、**[Pools]** > **[VLAN]** の順に選択します。

ステップ 3 [Work] ペインで、**[Actions]** > **[Create VLAN Pool]** の順に選択します。

ステップ 4 **[Create VLAN Pool]** ダイアログボックスで、下記で指定している項目を除き、必要に応じてフィールドに入力します。

a) **[Allocation Mode]** オプション ボタンでは **[Static Allocation]** を選択します。

b) **[Encap Blocks]** セクションでは、**[+]** をクリックします。

ステップ 5 **[Create Ranges]** ダイアログボックスで、一意の VLAN 範囲を入力し、**[OK]** をクリックします。

ステップ 6 **[Create VLAN Pool]** ダイアログボックスで、**[Submit]** をクリックします。

GUI を使用した外部ルーテッド ドメインの作成

デバイスの場所 (リーフ ノード/パス) とルート ピアリング用に作成するスタティック VLAN プールを結びつける外部ルーテッド ドメインを作成する必要があります。

ステップ 1 メニュー バーで、**[FABRIC]** > **[Access Policies]** の順に選択します。

ステップ 2 [Navigation] ペインで、**[Switch Policies]** を右クリックし、**[Configure Interface, PC and VPC]** を選択します。

ステップ 3 **[Configure Interface, PC, and VPC]** ダイアログボックスで、Application Policy Infrastructure Controller (APIC) に接続されるスイッチ ポートを設定し、次の操作を実行します。

a) スイッチ図の横にある大きい **[+]** アイコンをクリックし、新しいプロファイルを作成して VLAN を APIC 用に設定します。

b) **[Switches]** フィールドのドロップダウン リストから、APIC を接続するスイッチのチェックボックスをオンにします

c) **[Switch Profile Name]** フィールドに、プロファイルの名前を入力します。

d) **[+]** アイコンをクリックして、ポートを設定します。

e) **[Interface Type]** 領域で、**[Individual]** オプション ボタンが選択されていることを確認します。

f) **[Interfaces]** フィールドで、APIC が接続されるポートを入力します。

g) **[Interface Selector Name]** フィールドに、ポート プロファイルの名前を入力します。

h) **[Interface Policy Group]** フィールドで、**[Create One]** オプション ボタンをクリックします。

i) **[Attached Device Type]** ドロップダウン リストで、**[External Routed Devices]** を選択します。

j) **[Domain]** オプション ボタンでは、**[Create One]** オプション ボタンをクリックします。

k) **[Domain Name]** フィールドに、ドメイン名を入力します

l) VLAN プールを前に作成していた場合は、**[VLAN]** オプション ボタンとして、**[Choose One]** オプション ボタンをクリックします。その他の場合は、**[Create One]** オプション ボタンをクリックします。

既存の VLAN プールを選択する場合は、**[VLAN Pool]** ドロップダウン リストで、VLAN プールを選択します。

VLAN プールを作成する場合は、**[VLAN Range]** フィールドに VLAN 範囲を入力します。

m) **[Save]** をクリックし、**[Save]** をもう一度クリックします。

n) [Submit] をクリックします。

GUI を使用した外部ルーテッド ネットワークの作成

外部ルーテッド ネットワークは、ルート ピアリングでCisco Application Centric Infrastructure（ACI）ファブリックのルーティング設定を指定します。

ステップ 1 メニュー バーで、**[Tenants]** > **[All Tenants]** の順に選択します。

ステップ 2 [Work] ペインで、テナントの名前をダブルクリックします。

ステップ 3 [Navigation] ペインで、***[tenant_name]*** > **[Networking]** > **[External Routed Networks]** を選択します。

ステップ 4 [Work] ペインで、**[Actions]** > **[Create Routed Outside]** を選択します。

ステップ 5 [Create Routed Outside] ダイアログボックスで、下記で指定している項目を除き、必要に応じてフィールドに入力します。

a) ダイナミック ルーティングの場合は、[BGP] チェックボックスまたは [OSPF] チェックボックスをオンにします。

Open Shortest Path Firs （OSPF ）の場合は、追加の OSPF 固有のフィールドに入力します。

b) [Private Network] ドロップダウン リストで、デバイスがルートを交換するプライベート ネットワークを選択します。

c) [External Routed Domain] ドロップダウン リストで、ルート ピアリング用に作成した外部ルーテッド ドメインを選択します。

d) [Nodes and Interfaces Protocol Profiles] セクションで、[+] をクリックします。

ステップ 6 [Create Node Profile] ダイアログボックスで、下記で指定している項目を除き、必要に応じてフィールドに入力します。

a) [Nodes] セクションで、[+] をクリックします。

ステップ 7 [Select Node] ダイアログボックスで、下記に指定している項目を除き、必要に応じてフィールドに入力します。

a) [Node ID] ドロップダウン リストで、デバイスを接続するノード ID を選択します。

- 物理デバイスの場合は、物理デバイスをファブリックに接続するノードの ID にする必要があります。
- 仮想デバイスの場合は、仮想マシンをホストしているサーバが接続するノードの ID にする必要があります。

b) [Router ID] フィールドに、ACI ファブリックがルーティング プロトコル プロセスで使用するルータ ID を入力します。

c) ACI ファブリックとデバイスの間でスタティック ルーティングを使用する場合は、[Static Routes] セクションで [+] をクリックします。それ以外の場合は、ステップ 10 （80 ページ） に進みます。

ステップ 8 [Create Static Route] ダイアログボックスで、下記で指定している項目を除き、必要に応じてフィールドに入力します。

a) [Prefix] セクションには、静的ルートのプレフィックスを入力します。

b) [Next Hop Addresses] セクションでは、[+] をクリックします。

c) 静的ルートのネクスト ホップ IP アドレスを入力します。

d) [Update] をクリックします。

ステップ 9 [OK] をクリックします。

ステップ 10 [Select Node] ダイアログボックスで、[OK] をクリックします。

ステップ 11 ダイナミック ルーティング プロトコルとしてデバイスで BGP を使用する場合は、[BGP Peer Connectivity Profiles] セクションで、[+] をクリックします。それ以外の場合は、ステップ 14 （81 ページ） に進みます。

ステップ 12 [Create Peer Connectivity Profile] ダイアログボックスで、下記で指定している項目を除き、必要に応じてフィールドに入力します。

a) [Peer Address] フィールドで、BGP セッションを確立するデバイスの IP アドレスであるピア アドレスを入力します。

ステップ 13 [Create Peer Connectivity Profile] ダイアログボックスで、[OK] をクリックします。

ステップ 14 [Interface Profiles] セクションで、[+] をクリックします。

ステップ 15 [Create Interface Profile] ダイアログボックスで、必要に応じてフィールドに入力します。

a) ダイナミック ルーティング プロトコルとして OSPF を使用する場合は、OSPF プロファイル情報を入力します。

ステップ 16 [Interface] セクションでは、[SVI] タブを選択します。

ステップ 17 [Interface] セクションで、[+] をクリックします。

ステップ 18 [Select SVI Interface] ダイアログボックスで、下記に指定している項目を除き、必要に応じてフィールドに入力します。

a) [Path Type] オプション ボタンでは、デバイスのファブリックへの接続方法と一致するタイプを選択します。

b) [Path] ドロップダウン リストで、デバイスをファブリックに接続するパスを選択します。

- 物理デバイスの場合は、物理デバイスをファブリックに接続するパスです。
- 仮想デバイスの場合は、仮想マシンをホストしているサーバを接続するパスです。

c) [Encap] フィールドで、カプセル化 VLAN を指定します。

d) [IP Address] フィールドで、ファブリック SVI インターフェイスで使用する IP アドレスを指定します。

e) [MTU (bytes)] フィールドで、最大伝送ユニット サイズをバイト単位で指定します。

デフォルト値の「inherit」の場合、ACI ではデフォルト値の「9000」が使用され、リモート デバイスでは通常はデフォルト値の「1500」が使用されます。異なる MTU 値を指定すると、ACI とリモート デバイス間のピアリングで問題が発生する可能性があります。リモート デバイスの MTU 値を「1500」に設定した場合は、リモート デバイスの `L3Out` オブジェクトの MTU 値を「9000」に設定して ACI の MTU 値と一致させます。

ステップ 19 [OK] をクリックします。

ステップ 20 [Create Interface Profile] ダイアログボックスで、[OK] をクリックします。

ステップ 21 [Create Node Profile] ダイアログボックスで、[OK] をクリックします。

ステップ 22 [Create Routed Outside] ダイアログボックスで、[Next] をクリックします。

ステップ 23 [External EPG Networks] セクションで、[+] をクリックします。

ステップ 24 [Create External Network] ダイアログボックスで、下記で指定している項目を除き、必要に応じてフィールドに入力します。

a) [Subnet] セクションで、[+] をクリックします。

ステップ 25 [Create Subnet] ダイアログボックスで、下記で指定している項目を除き、必要に応じてフィールドに入力します。

a) [IP Address] フィールドに IP アドレスまたはサブネット マスクを入力します。

サブネット マスクは、従来のルーティング プロトコル設定で定義するネットワーク ステートメントと同等です。

ステップ 26 [OK] をクリックします。

ステップ 27 （任意） 必要に応じて、さらにサブネットを作成します。

ステップ 28 **[Create External Network]** ダイアログボックスで、**[OK]** をクリックします。

ステップ 29 **[Create Routed Outside]** ダイアログボックスで、**[Finish]** をクリックします。

GUI を使用したルータ設定の作成

ルーティング プロトコル設定の一部として、デバイスで使用するルータ ID を指定する必要があります。

ステップ 1 メニュー バーで、**[Tenants]** > **[All Tenants]** の順に選択します。

ステップ 2 [Work] ペインで、テナントの名前をダブルクリックします。

ステップ 3 [Navigation] ペインで、**テナント名** > **[Services]** > **[L4-L7]** > **[Router configurations]** を選択します。

ステップ 4 [Work] ペインの [Router Configurations] テーブルで、[+] をクリックします。

ステップ 5 デバイスでルータ ID として使用する IP アドレスを入力します。

ステップ 6 [Update] をクリックします。

GUI を使用したサービス グラフ アソシエーションの作成

サービス グラフのアソシエーションを作成する必要があります。これには、外部ルーテッド ネットワーク ポリシーおよびルータ設定とデバイス選択ポリシーの関連付けが含まれます。

ステップ 1 メニュー バーで、**[Tenants]** > **[All Tenants]** の順に選択します。

ステップ 2 [Work] ペインで、テナントの名前をダブルクリックします。

ステップ 3 [ナビゲーション] ペインで、**[Tenant]** ***[tenant_name]*** > **[Services]** > **[L4-L7]** > **[Device Selection Policies]** > ***[device_selection_policy]*** を選択します。

ステップ 4 [Navigation] ペインで、**テナント名** > **[L4-L7 Services]** > **[Device Selection Policies]** > **デバイス選択ポリシー**を選択します。デバイス選択ポリシーは、Cisco Application Centric Infrastructure（ACI）ファブリックでルート ピアリングを実行する際に使用するデバイス選択ポリシーです。

ステップ 5 [Work] ペインの [properties] セクションにある **[Router Config]** ドロップダウン リストで、ルーティング ピアリング用に作成したルータ設定を選択します。

ステップ 6 [Navigation] ペインで、選択したデバイス選択ポリシーを展開し、ACI ファブリックとピアリングするインターフェイスを選択します。

ステップ 7 [Work] ペインの [properties] セクションにある [Associated Network] オプション ボタンで、[L3 External Network] を選択します。

ステップ 8 **[L3 External Network]** ドロップダウン リストで、ルート ピアリング用に作成した外部ルーテッド ネットワークを選択します。

次のように変更されます。

- 外部ルーテッド ネットワークと関連付けたインターフェイスのカプセル化 VLAN が、外部ルーテッド ネットワーク インターフェイス プロファイルの一部として設定した VLAN と一致するようにプログラミングされる
- 外部ルーテッド ネットワーク インターフェイスとルーティング プロトコル設定がルーフスイッチにプッシュされる
- ルーティング プロトコル設定がデバイス パッケージを使用してデバイスにプッシュされる

NX-OS スタイルの CLI を使用したルート ピアリングの設定

ここでは、ルート ピアリングを設定する NX OS スタイルの CLI のコマンドの例を示します。

ステップ 1 コンフィギュレーション モードを開始します。

例：

```
apic1# configure
```

ステップ 2 テナントのコンフィギュレーション モードを開始します。

例：

```
apic1(config)# tenant 101
```

ステップ 3 サービス グラフを追加し、それをコントラクトと関連付けます。

例：

```
apic1(config-tenant)# l4l7 graph g1 contract c1
```

ステップ 4 デバイス クラスタに関連付けるノード（サービス）を追加します。

例：

```
apic1(config-graph)# service ASA_FW device-cluster-tenant 101 device-cluster ASA_FW1
```

ステップ 5 サービス機能で、コンシューマ コネクタとプロバイダー クラスタ インターフェイスを設定します。

例：

```
apic1(config-service)# connector consumer cluster-interface provider
```

ステップ 6 クラスタ インターフェイスで、サービス デバイスでのルート ピアリングで使用するレイヤ 3 Outside（l3extOut）とエンドポイント グループ（l3extInstP）を指定し、コネクタのコンフィギュレーション モードを終了します。

例：

```
apic1(config-connector)# l4l7-peer tenant 101 out l101 epg e101 redistribute bgp
apic1(config-connector)# exit
```

ステップ 7 プロバイダー コネクタとコンシューマのクラスタ インターフェイスにステップ 5 とステップ 6 を繰り返します。

例：

```
apic1(config-service)# connector provider cluster-interface consumer
apic1(config-connector)# l4l7-peer tenant 101 out l101 epg e101 redistribute bgp
apic1(config-connector)# exit
```

ステップ 8 （任意） コネクタからエンドポイント グループの関連付けを解除する場合は、**no l4l7-peer** コマンドを使用します。

例：

```
apic1(config-connector)# no l4l7-peer tenant 101 out l101 epg e101 redistribute bgp
```

ステップ 9 ルータ設定ポリシーをテナントに作成し、ピア レイヤ 4 ～ レイヤ 7 デバイスにルータ ID を指定し、コンフィギュレーション モードに戻ります。

例：

```
apic1(config)# tenant 102
apic1(config-tenant)# rtr-cfg bgp1
apic1(config-router)# router-id 1.2.3.5
apic1(config-router)# exit
```

ステップ 10 ルータ設定ポリシーを特定のサービス デバイスに関連付け、テナント コンフィギュレーション モードに戻ります。

例：

```
apic1(config-tenant)# l4l7 graph g2 contract c2 subject http
apic1(config-graph)# service ASA_FW device-cluster-tenant 102 device-cluster ASA_FW2
apic1(config-service)# rtr-cfg bgp1
apic1(config-service)# exit
apic1(config-graph)# exit
```

ステップ 11 レイヤ 3 Outside をリーフ インターフェイスおよび VRF に関連付けます。

例：

```
apic1(config-tenant)# external-l3 epg e101 l3out l101
apic1(config-tenant-l3ext-epg)# vrf member v101
apic1(config-tenant-l3ext-epg)# match ip 101.101.1.0/24
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant 101 vrf v101 l3out l101
apic1(config-leaf-vrf)# ip route 101.101.1.0/24 99.1.1.2
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# interface ethernet 1/10
apic1(config-leaf-if)# vrf member tenant 101 vrf v101 l3out l101
apic1(config-leaf-if)# vlan-domain member dom101
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# ip address 99.1.1.1/24
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

ルーティング プロトコル（BGP、OSPF）やルート マップなど、名前付きモードを使用したレイヤ 3 外部接続（レイヤ 3 Outside）の詳細な設定については、『*Cisco APIC NX-OS Style CLI Command Reference*』ドキュメントを参照してください。

（注） CLI での外部レイヤ 3 設定は、2 つのモード（基本モードと名前付きモード）で使用できます。特定のテナントまたは VRF では、すべての外部レイヤ 3 設定にこれらのモードの 1 つのみを使用します。ルート ピアリングは名前付きモードでのみサポートされています。

ルート ピアリングのトラブルシューティング

Cisco Application Centric Infrastructure（ACI）ファブリックにルート ピアリングまたはデータ トラフィックの問題がある場合に、その問題をトラブルシーティングするために ACI ファブリック リーフ スイッチ上で実行できるコマンドがいくつかあります。

次の表に、ファブリック リーフ スイッチのスイッチ シェルで実行できるトラブルシューティング コマンドを示しますスイッチ。

コマンド	説明
show ip route vrf all	動的に取得したルートを含む特定のコンテキストのすべてのルートを表示します。
show ip ospf neighbor vrf all	隣接デバイスとの Open Shortest Path First（OSPF）ピアリング セッションを表示します。
show ip ospf vrf all	各コンテキスト内のランタイム OSPF 設定を表示します。
show ip ospf traffic vrf all	Virtual Routing and Forwarding（VRF）の各コンテキストの OSPF トラフィックを確認します。
show system internal policymgr stats	特定のリーフ スイッチのコントラクト フィルタ ルールを表示し、ルールのパケット ヒット カウントを確認します。

次の表に、**vsh_lc** シェルで実行できるトラブルシューティング コマンドを示します。

コマンド	説明
show system internal aclqos prefix	特定のリーフ スイッチの IPv4 プレフィックス アソシエーション ルールとルールのトラフィック ヒット カウントを確認します。

シェル コマンドに加えて、トラブルシューティングに役立つ次の点を確認できます。

- デバイスの健全性カウント
- 特定のテナントの下のすべてのエラーと `NwIssues`

CLI を使用したリーフ スイッチのルート ピアリング機能の確認

ファブリック リーフ上でスイッチ シェル コマンドを使用して、リーフ スイッチ設定とルート ピアリング機能を確認することができます。

ステップ 1 デバイスが接続されているファブリック リーフ スイッチで、SVI インターフェイスが設定されていることを確認します。

```
fab2-leaf3# show ip interface vrf user1:global
IP Interface Status for VRF "user1:global"
vlan30, Interface status: protocol-up/link-up/admin-up, iod: 134,
  IP address: 1.1.1.1, IP subnet: 1.1.1.0/30
  IP broadcast address: 255.255.255.255
  IP primary address route-preference: 1, tag: 0
lo3, Interface status: protocol-up/link-up/admin-up, iod: 133,
  IP address: 10.10.10.1, IP subnet: 10.10.10.1/32
  IP broadcast address: 255.255.255.255
  IP primary address route-preference: 1, tag: 0

fab2-leaf3#
```

インターフェイス vlan30 には SVI インターフェイス設定が含まれており、インターフェイス lo3 には外部ルーテッド ネットワーク設定に指定されているルータ ID が含まれています。

ステップ 2 ファブリック リーフ スイッチの Open Shortest Path First（OSPF）の設定を確認します。

```
fab2-leaf3# show ip ospf vrf user1:global

 Routing Process default with ID 10.10.10.1 VRF user1:global
 Stateful High Availability enabled
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Table-map using route-map exp-ctx-2949120-deny-external-tag
 Redistributing External Routes from
   static route-map exp-ctx-st-2949120
   bgp route-map exp-ctx-proto-2949120
   eigrp route-map exp-ctx-proto-2949120
 Maximum number of non self-generated LSA allowed 100000
    (feature configured but inactive)
    Current number of non self-generated LSA 1
    Threshold for warning message 75%
    Ignore-time 5 minutes, reset-time 10 minutes
    Ignore-count allowed 5, current ignore-count 0
 Administrative distance 110
 Reference Bandwidth is 40000 Mbps
 SPF throttling delay time of 200.000 msecs,
   SPF throttling hold time of 1000.000 msecs,
   SPF throttling maximum wait time of 5000.000 msecs
 LSA throttling start time of 0.000 msecs,
   LSA throttling hold interval of 5000.000 msecs,
   LSA throttling maximum wait time of 5000.000 msecs
 Minimum LSA arrival 1000.000 msec
```

```
 LSA group pacing timer 10 secs
 Maximum paths to destination 8
 Number of external LSAs 0, checksum sum 0x0
 Number of opaque AS LSAs 0, checksum sum 0x0
 Number of areas is 1, 1 normal, 0 stub, 0 nssa
 Number of active areas is 1, 1 normal, 0 stub, 0 nssa
   Area (0.0.0.200)
        Area has existed for 00:17:55
        Interfaces in this area: 1 Active interfaces: 1
        Passive interfaces: 0  Loopback interfaces: 0
        SPF calculation has run 4 times
         Last SPF ran for 0.000273s
        Area ranges are
        Area-filter in 'exp-ctx-proto-2949120'
        Number of LSAs: 3, checksum sum 0x0
fab2-leaf3#
```

ステップ 3 ファブリック リーフ スイッチの OSPF ネイバーの関係を確認します。

```
fab2-leaf3# show ip ospf neighbors vrf user1:global
 OSPF Process ID default VRF user1:global
 Total number of neighbors: 1
 Neighbor ID     Pri State            Up Time  Address         Interface
 10.10.10.2        1 FULL/BDR         00:03:02 1.1.1.2         Vlan30
fab2-leaf3#
```

ステップ 4 ルートがファブリック リーフ スイッチによって取得されることを確認します。

```
fab2-leaf3# show ip route vrf user1:global
IP Route Table for VRF "user1:global"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

1.1.1.0/30, ubest/mbest: 1/0, attached, direct
    *via 1.1.1.1, vlan30, [1/0], 00:26:50, direct
1.1.1.1/32, ubest/mbest: 1/0, attached
    *via 1.1.1.1, vlan30, [1/0], 00:26:50, local, local
2.2.2.0/24, ubest/mbest: 1/0
    *via 1.1.1.2, vlan30, [110/20], 00:06:19, ospf-default, type-2
10.10.10.1/32, ubest/mbest: 2/0, attached, direct
    *via 10.10.10.1, lo3, [1/0], 00:26:50, local, local
    *via 10.10.10.1, lo3, [1/0], 00:26:50, direct
10.122.254.0/24, ubest/mbest: 1/0
    *via 1.1.1.2, vlan30, [110/20], 00:06:19, ospf-default, type-2
fab2-leaf3#
```

ステップ 5 OSPF がデバイス（この例では Cisco ASAv）に設定されていることを確認します。

```
ciscoasa# show running-config
: Saved
:
: Serial Number: 9AGRM5NBEXG
: Hardware:   ASAv, 2048 MB RAM, CPU Xeon 5500 series 2133 MHz
:
ASA Version 9.3(1)
!
hostname ciscoasa
enable password 8Ry2YjIyt7RRXU24 encrypted
names
!
interface GigabitEthernet0/0
 nameif internalIf
```

```
 security-level 100
 ip address 2.2.2.1 255.255.255.0
!
interface GigabitEthernet0/1
 nameif externalIf
 security-level 50
 ip address 1.1.1.2 255.255.255.252
!
<<..>>
router ospf 1
 router-id 10.10.10.2
 network 1.1.1.0 255.255.255.252 area 200
 area 200
 log-adj-changes
 redistribute connected
 redistribute static
!
```

第 10 章

ポリシー ベース リダイレクトの設定

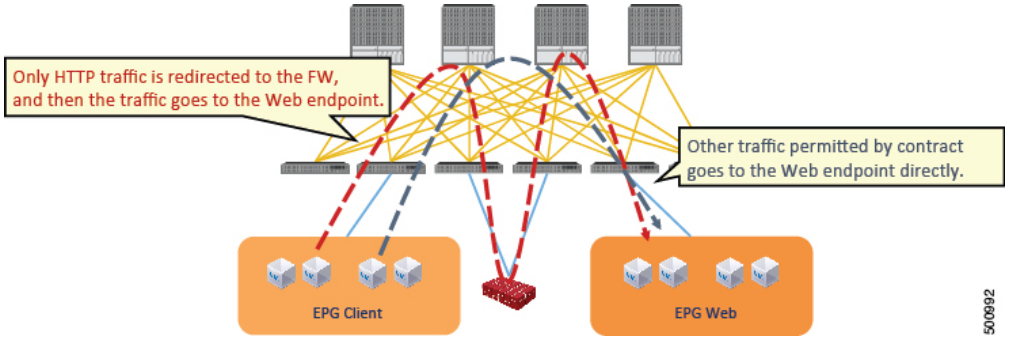
ポリシーベースのリダイレクトについて

Cisco Application Centric Infrastructure(ACI) ポリシーベース リダイレクト（PBR）により、レイヤ 4 ーレイヤ 7 パッケージなしで、ファイアウォールやロード バランサなどのサービス アプライアンスを管理対象ノードまたは非管理対象ノードとしてプロビジョニングできます。一般的な使用例としては、プールしてアプリケーション プロファイルに合わせて調整すること、また容易にスケーリングすることができ、サービス停止の問題が少ないサービス アプライアンスのプロビジョニングがあります。PBR により、プロビジョニングするコンシューマおよびプロバイダー エンドポイント グループをすべて同じ仮想ルーティングおよび転送（VRF）インスタンスに含めることで、サービス アプライアンスの展開をシンプル化できます。PBR の導入は、ルート リダイレクト ポリシーおよびクラスタのリダイレクト ポリシーの設定と、ルーティングとクラスタ リダイレクト ポリシーを使用するサービス グラフ テンプレートの作成から構成されます。サービス グラフ テンプレートを展開した後は、サービス グラフ プロバイダーの

エンドポイントグループを利用するためにエンドポイントグループを有効にすることにより、サービスアプライアンスを使用します。これは、vzAnyを使用することにより、さらに簡素化し、自動化できます。パフォーマンスの要件が、専用のサービスアプライアンスをプロビジョニングするかどうかを決定するものとなるのに対し、PBRを使用すれば、仮想サービス アプライアンスの展開も容易になります。

次の図は、ファイアウォールへのトラフィックに固有の、リダイレクトの使用例を示しています:

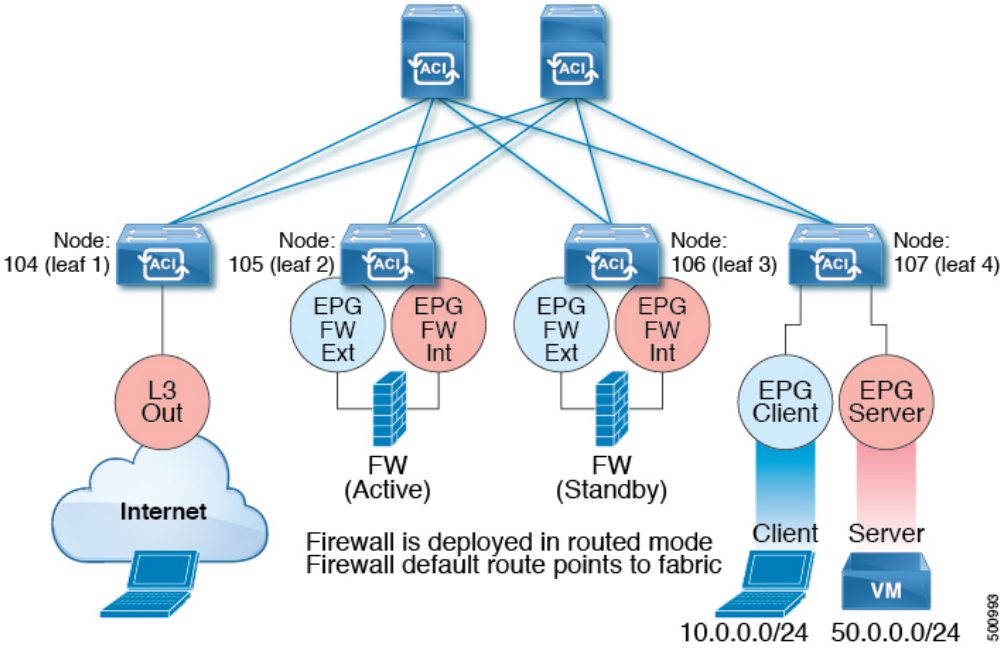
図 *11* : 使用例: ファイアウォール特有のトラフィックのリダイレクト



この使用例では、2つの情報カテゴリを作成する必要があります。最初の情報カテゴリはHTTPトラフィックを許可します。その後このトラフィックはファイアウォールにリダイレクトされます。トラフィックはファイアウォールを通過してから、Webエンドポイントに送られます。2番目の情報カテゴリはすべてのトラフィックを許可します。これは最初の情報カテゴリではリダイレクトされなかったトラフィックをキャプチャします。トラフィックはそのままWebエンドポイントに送られます。

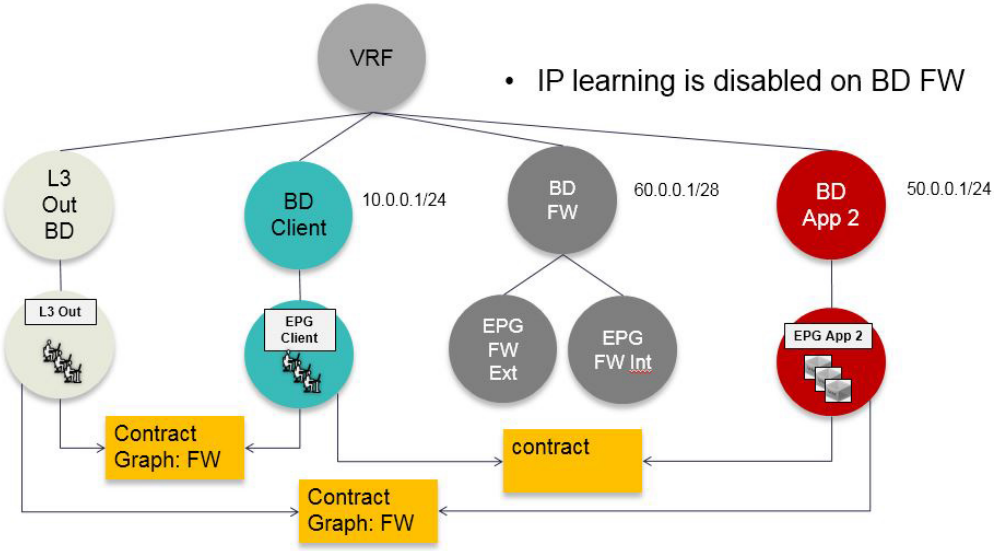
次の図は、ACI PBR物理トポロジのサンプルを示しています:

図 *12* : サンプルの *ACI PBR* 物理トポロジ



次の図は、ACI PBR 論理トポロジのサンプルを示しています:

図 *13* : サンプルの *ACI PBR* 論理トポロジ



これらの例はシンプルな導入ですが、 ACI PBR は、ファイアウォールやサーバのロード バランサなどのような、複数のサービスのために物理および仮想サービス アプライアンスの両方を混在させたものにスケールアップすることを可能にします。

複数ノード ポリシー ベースのリダイレクトについて

複数ノード ポリシー ベースのリダイレクトは、1 つのサービス チェーンで最大 3 つのノードをサポートすることにより、PBR を強化します。どのサービス ノードのコネクタがトラフィックの終端になるかは設定することができ、この設定に基づいて、サービス チェーンの送信元および宛先クラス ID が決定されます。複数のノード PBR 機能では、ポリシー ベースのリダイレクトはサービス ノード コネクタのコンシューマ側、プロバイダ側、またはその両方で有効にすることができます。これは、転送方向にも、または逆方向にも設定できます。サービス ノードのコネクタで PBR ポリシーを設定した場合、そのコネクタがトラフィックを終端することはありません。

対称ポリシー ベースのリダイレクトについて

対称ポリシー ベースのリダイレクト (PBR) 構成により、サービス アプライアンスのプールをプロビジョニングできるため、コンシューマとプロバイダーのエンドポイント グループ トラフィックがポリシー ベースになります。トラフィックは、送信元および宛先 IP 等価コスト マルチパス ルーティング (ECMP) プレフィックス ハッシュに応じて、プール内のサービス ノードの 1 つにリダイレクトされます。

（注） 対称 PBR 構成には 9300-EX ハードウェアが必要です。

対称 PBR REST のサンプルの例を以下に示します。

```
Under fvTenant svcCont

<vnsSvcRedirectPol name="LoadBalancer_pool">
          <vnsRedirectDest name="lb1" ip="1.1.1.1" mac="00:00:11:22:33:44"/>
          <vnsRedirectDest name="lb2" ip="2.2.2.2" mac="00:de:ad:be:ef:01"/>
          <vnsRedirectDest name="lb3" ip="3.3.3.3" mac="00:de:ad:be:ef:02"/>
</vnsSvcRedirectPol>

<vnsLIfCtx name="external">
          <vnsRsSvcRedirectPol tnVnsSvcRedirectPolName="LoadBalancer_pool"/>
          <vnsRsLIfCtxToBD tDn="uni/tn-solar/bd-fwBD">
</vnsLIfCtx>

<vnsAbsNode name="FW" routingMode="redirect">
```

対称 PBR NX-OS スタイルの CLI コマンドの例を次に示します。

テナント スコープの下の次のコマンドは、サービス リダイレクト ポリシーを作成します。

```
apic1(config-tenant)# svcredir-pol fw-external
apic1(svcredir-pol)# redir-dest 2.2.2.2 00:11:22:33:44:56
```

次のコマンドは PBR を有効にします。

```
apic1(config-tenant)# l4l7 graph FWOnly contract default
apic1(config-graph)# service FW svcredir enable
```

次のコマンドは、デバイス選択ポリシー コネクタの下にリダイレクト ポリシーを設定します。

```
apic1(config-service)# connector external
apic1(config-connector)# svcredir-pol tenant solar name fw-external
```

ポリシーベースのリダイレクトとハッシュアルゴリズム

（注） この機能は、APIC リリース 2.2(3x) リリースおよび APIC リリース 3.1 (1) でさせてで使用できます。APIC Release 3.0(x) ではサポートされていません。

Cisco APIC、リリース 2.2(3x) では、ポリシー ベースのリダイレクト機能 (PBR) は、次のハッシュ アルゴリズムをサポートします。

- 送信元 IP アドレス
- 宛先 IP アドレス
- ソース IP アドレス、宛先 IP アドレスおよびプロトコル タイプ (着信も、対称) に基づいてアルゴリズムが以前のリリースでサポートされます。

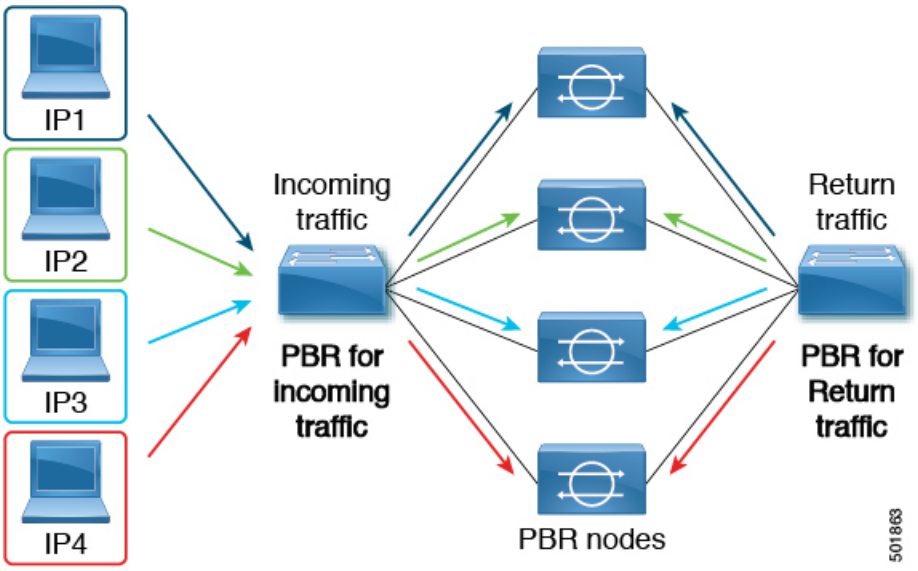
ポリシーベースのリダイレクトの修復性のあるハッシュ

対称 PBR では、着信と戻りユーザ トラフィックは、ECMP グループで同じ PBR ノードを使用します。ただし、PBR ノードのいずれかがダウンするか、障害を起こした場合には、既存のトラフィック フローは別のノードに送られて再ハッシュされます。これは、機能しているノードの既存のトラフィックが、現在の接続情報を持っていない他の PBR ノードに負荷分散のために送られるといったような問題の原因となります。トラフィックがステートフルファイアウォールを通過する場合には、接続がリセットされることにもつながります。

修復性のあるハッシュは、トラフィック フローを物理ノードへマッピングするプロセスで、障害の発生したノードからのフロー以外のトラフィックが再ハッシュされるのを避けられるようにします。障害を起こしたノードからのトラフィックは、「バックアップ」ノードに再マッピングされます。「バックアップ」ノード上の既存のトラフィックは移動できません。

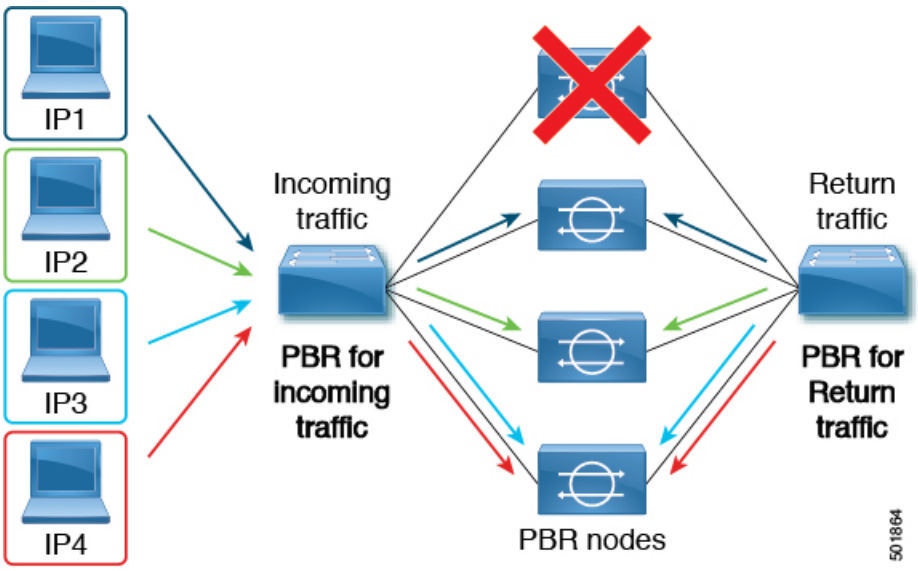
次の図は、着信と戻りユーザ トラフィックが同じ PBR ノードを使用している、対称 PBR の基本的な機能を示しています。

図 *14* : 対称 *PBR*



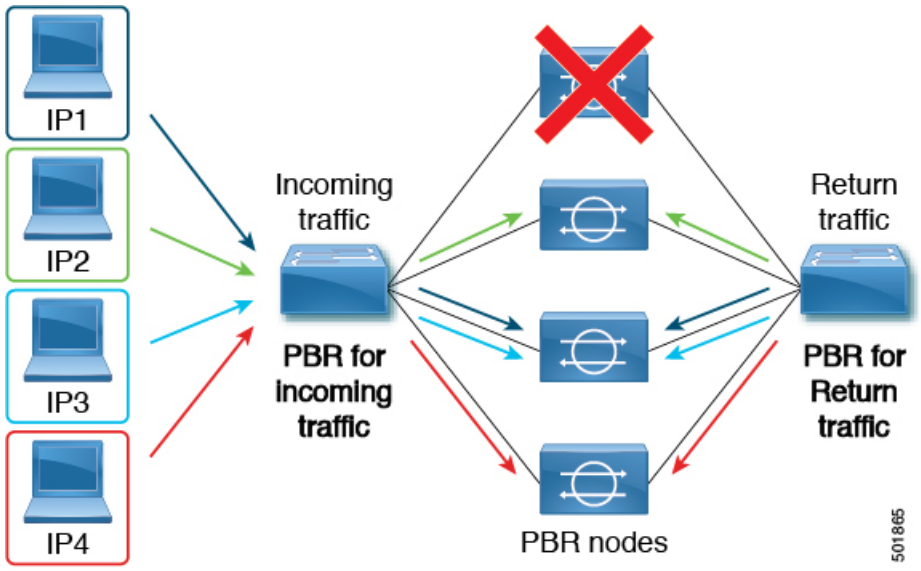
次の画像は、PBR ノードのいずれかが無効か、障害が発生したときに何が生じるかを示しています。IP1 のトラフィックは隣のノードへ再ハッシュされ、IP2 および IP3 のトラフィックがもう 1 つの PBR ノードに負荷分散されます。このことは、前述のように、他の PBR ノードが IP2 および IP3 トラフィックの現在の接続情報を持っていない場合、接続の中断や遅延という問題につながることがあり得ます。

図 *15* : 修復性のあるハッシュがない場合の無効化された/障害の発生した *PBR* ノード



最後の図は、修復性のあるハッシュが有効になっている場合に、この同じ使用例がどのように対処されるかを示しています。無効化された/障害の発生したノードからのユーザ トラフィックだけが移動されます。その他のすべてのユーザ トラフィックは、それぞれの PBR ノードに残ります。

図 *16* : 修復性のあるハッシュがある場合の無効化された/障害の発生した *PBR* ノード



ノードがサービス可能状態に戻ると、障害の発生したノードからアクティブなノードに再ハッシュされたトラフィック フローは、再度アクティブ化されたノードに戻ります。

（注） ECMP グループの PBR ノードを追加または削除すると、すべてのトラフィック フローが再ハッシュされる原因となることがあります。

L4 ～ L7 のポリシー ベース リダイレクトで復元力のあるハッシュを有効にする

始める前に

このタスクでは、L4-7 ポリシー ベースのリダイレクト ポリシーが作成されたことを前提としています。

ステップ 1 メニュー バーで、**Tenants** > **All Tenants** の順に選択します。

ステップ 2 作業ウィンドウで、テナントの名前をダブルクリックします。

ステップ 3 ナビゲーション ウィンドウで、**Tenant** ***tenant_name*** > **Policies** > **Protocol** > **L4-L7 Policy Based Redirect** > ***L4-L7_PBR_policy_name*** を選択します。

ステップ 4 Work ペインで、**Resilient Hashing Enabled** チェック ボックスをオンにします。

ステップ 5 [Submit] をクリックします。

コンシューマとプロバイダブリッジドメイン内のサービス ノードへの PBR によるサポート

Cisco APIC 3.1(1) リリース以降、コンシューマやプロバイダを含むブリッジ ドメイン (BD) は、サービス ノードもサポートするようになりました。したがって今後は、別の PBR ブリッジ ドメインをプロビジョニングする必要はありません。

Cisco Nexus 9300-EX と 9300-FX プラットフォームのリーフ スイッチは、この機能をサポートします。

ポリシーベースのリダイレクトを設定する際の注意事項と制約事項

ポリシーベースのリダイレクトを行うサービス ノードを計画する際には、次の注意事項と制約事項に従ってください:

- Cold Standby のアクティブ/スタンバイ導入では、サービス ノードにアクティブな導入の MAC アドレスを設定します。Cold Standby のアクティブ/スタンバイ導入では、アクティブ ノードがダウンすると、スタンバイ ノードがアクティブ ノードの MAC アドレスを引き継ぎます。
- ネクスト ホップ サービス ノードの IP アドレスと仮想 MAC アドレスを指定する必要があります。
- ポリシー ベースのリダイレクト ブリッジ ドメインでは、エンドポイント データプレーンの学習を無効にする必要があります。
- 別のブリッジ ドメイン サービスにアプライアンスをプロビジョニングします。Cisco Application Policy Infrastructure Controller（Cisco APIC）リリース 3.1(x) 以降、別のブリッジ ドメインでのサービス アプライアンスのプロビジョニングは必須ではなくなりました。そのためには、Cisco Nexus 9300-EX と 9300-FX プラットフォームのリーフ スイッチが必要です。
- Cisco APIC リリース 3.1 ソフトウェアからダウングレードすると、内部コードが、ポリシーベースのリダイレクト ブリッジ ドメインがコンシューマまたはプロバイダと同じブリッジ ドメインを使用しているかのチェックを行います。その場合にはダウングレード中にエラーが出されます。そのような設定は Cisco APIC の以前のバージョンではサポートされないからです。
- サービス アプライアンス、送信元、およびブリッジ ドメインは、同じ VRF に存在できます。
- Cisco N9K-93128TX、N9K-9396PX、N9K-9396TX、N9K-9372PX、および N9K-9372TX スイッチでは、サービス アプライアンスを、送信元または宛先のいずれかのエンドポイント

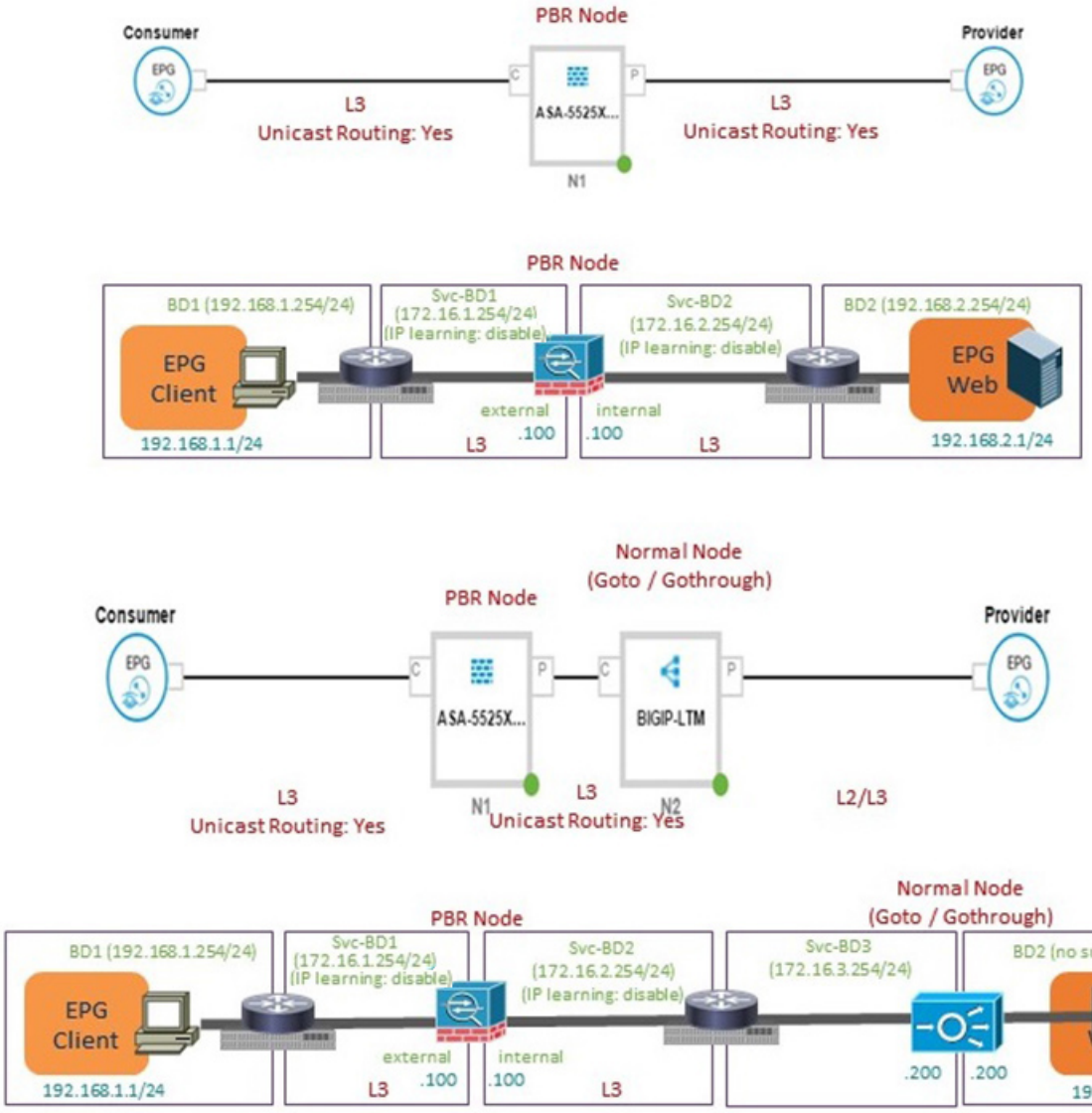
グループと同じリーフ スイッチに配置することはできません。Cisco N9K-C93180YC-EX および N9K-93108TC-EX スイッチでは、サービス アプライアンスを、送信元または宛先のいずれかのエンドポイント グループと同じリーフ スイッチに配置することができます。

- サービス アプライアンスは、通常のブリッジ ドメインにのみ配置できます。
- サービス アプライアンスのプロバイダのエンドポイント グループによって提供される契約は `allow-all` に設定できますが、トラフィックを Cisco Application Centric Infrastructure (Cisco ACI) ファブリックでルーティングすることはできません。
- Cisco APIC リリース 3.1(1) 以降では、Cisco Nexus 9300-EX と 9300-FX プラットフォームのリーフ スイッチを使用する場合、ポリシー ベースのリダイレクト ブリッジ ドメインでエンドポイント データプレーン学習を無効にする必要はありません。サービス グラフの導入時には、ポリシーベースのリダイレクト ノード EPG の場合にのみ、エンドポイント データプレーンの学習は自動的に無効にされます。非 EX および非 FX プラットフォーム リーフ スイッチを使用する場合は、ポリシー ベースのリダイレクト ブリッジ ドメインでエンドポイント データプレーンの学習を無効にする必要があります。
- 複数のノードのポリシー ベースのリダイレクト (複数ノード PBR):
  - ポリシー ベース ルーティングを設定できるサービス チェーンでは、最大 3 つのノードをサポートしています。
  - ロード バランサの複数ノード PBR L3 宛先についての注意事項:
    - L3 宛先のアップグレード: L3 Destination (VIP) パラメータは、アップグレード後にはデフォルトで有効になります。このことで問題は発生しません。PBR ポリシーは特定のサービス ノードで設定されていたわけではなく (3.2(1) より前)、ノード コネクタが L3 宛先として扱われており、新しい Cisco APIC バージョンでも引き続き同様だからです。
    - トラフィックは、必ずしもコンシューマ/プロバイダを宛先とする必要はありません。
    - 転送方向では、トラフィックはロード バランサを宛先とします。
    - 逆方向では、SNAT が有効になっている場合、トラフィックの宛先はロード バランサの内部レッグになります。
    - 両方向では、論理インターフェイス コンテキストの L3 宛先 (VIP) を有効にします (チェックします)。
    - 両方向で L3 宛先 (VIP) を有効にする (チェックする) と、内部側で設定された PBR ポリシーにより、ロード バランサ内部で SNAT から 非 SNAT への切り替えを行うことができます。
    - SNAT が無効の場合:
      - 逆方向トラフィックは、ロードバランサの内部レッグではなく、コンシューマを宛先とします (内部レッグで PBR ポリシーが有効にされている)

- PBR ポリシーが適用されるため、この状況では L3 宛先 (VIP) は適用されません。

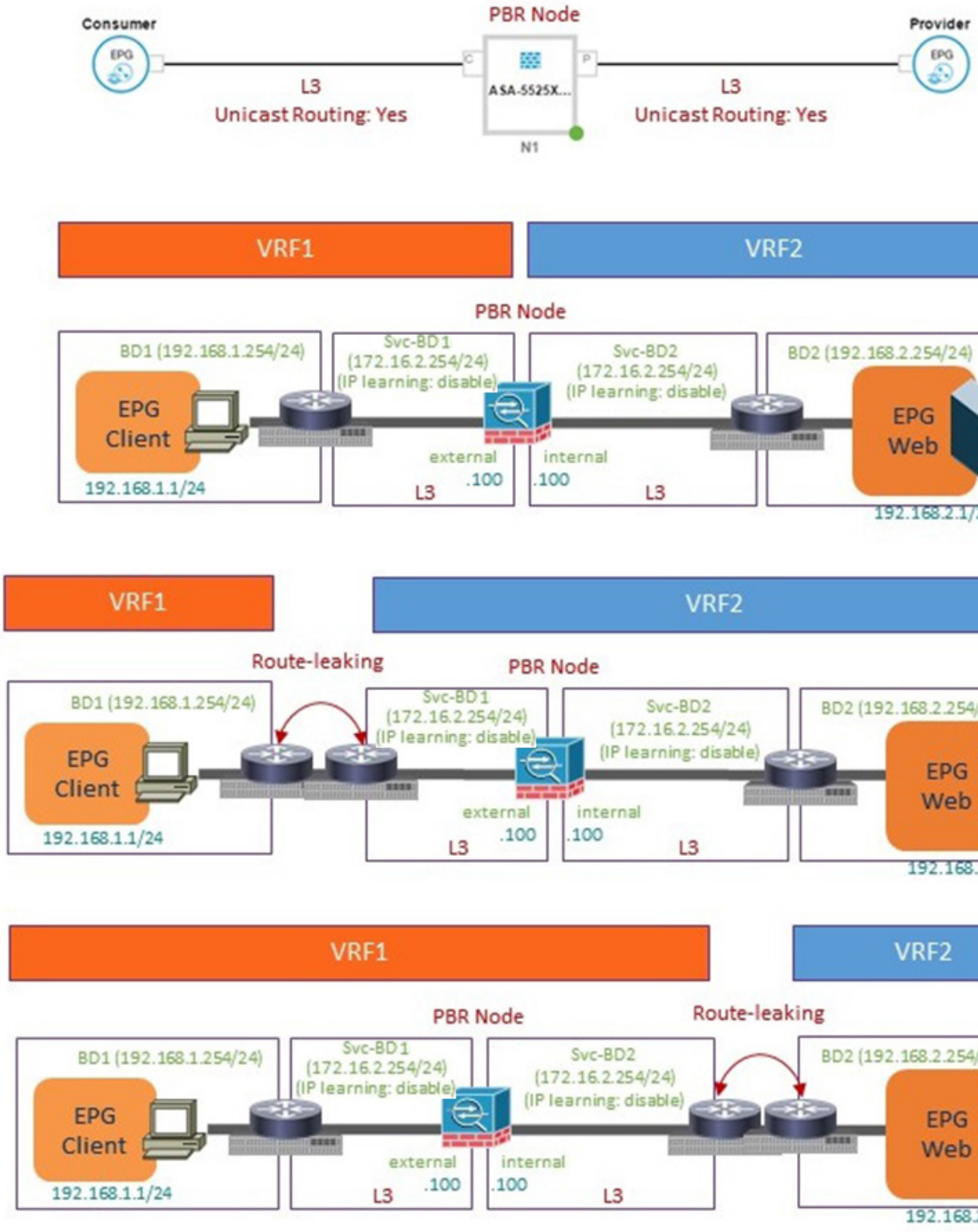
- マルチキャストおよびブロードキャスト トラフィック リダイレクションはサポートされていません。
- 透過的なサービスへのリダイレクションはサポートされていません。
- リダイレクト ポリシーの宛先を別のグループに変更した場合、Cisco APIC は変更に対してエラーを発生し、ポリシーの動作状態は無効になります。ポリシーを再度有効にするには、エラーをクリアする必要があります。
- 同じ VRF インスタンス内でサポートされているポリシーベースのリダイレクトの設定には、次のものが含まれます:

図 *17* : 同じ *VRF* インスタンス内でサポートされるポリシー ベースのリダイレクトの設定



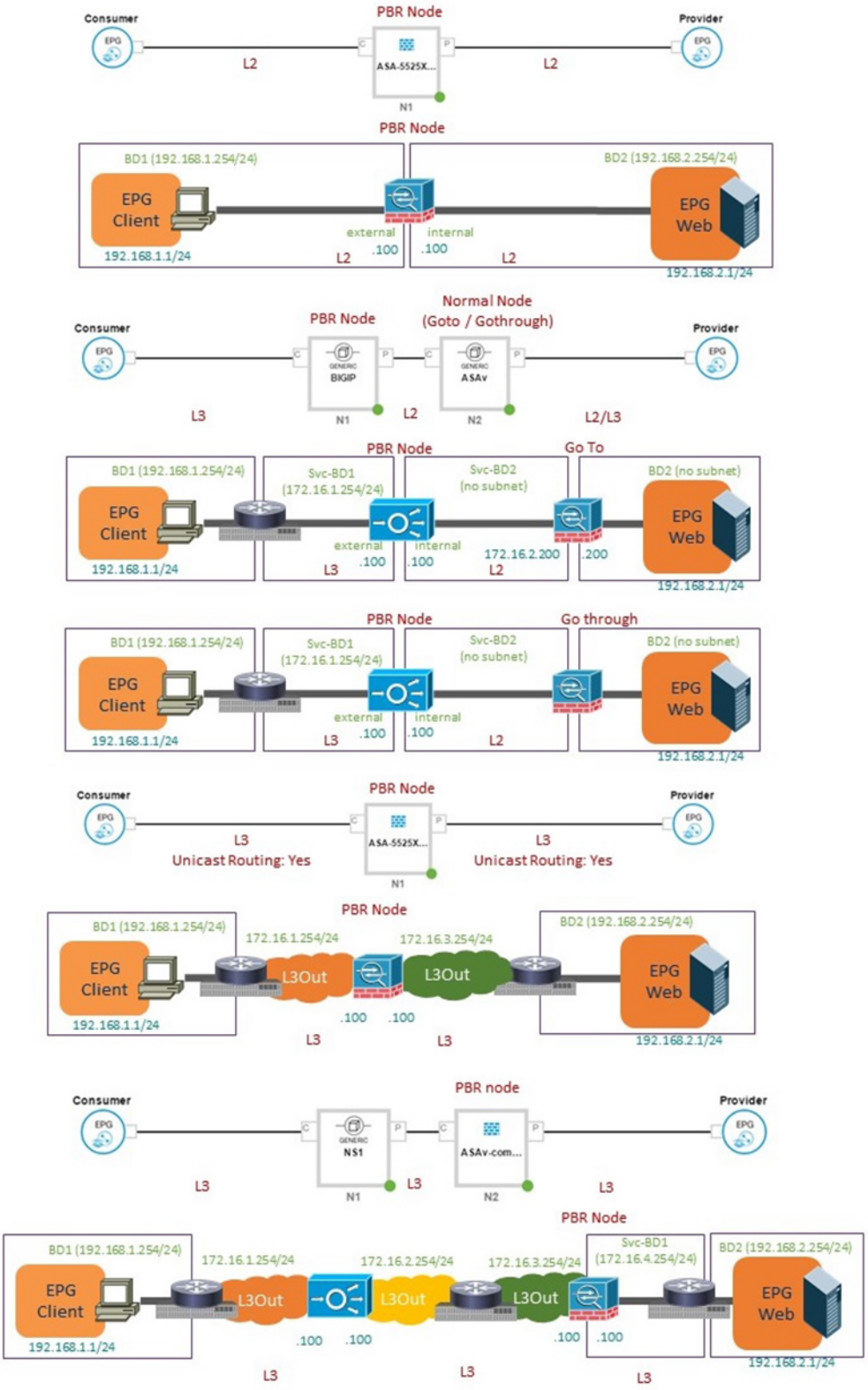
- 別の VRF インスタンス内でサポートされるポリシーベースのリダイレクトの設定には、次のものが含まれます:

図 *18* : 別の *VRF* インスタンス内でサポートされるポリシーベースのリダイレクトの設定



- サポートされていないポリシー ベースのリダイレクト設定は次のとおりです:

図 *19* : サポートされていないポリシー ベースのリダイレクト設定



GUI を使用したポリシー ベース リダイレクトの設定

次の手順では、GUI を使用してポリシー ベース リダイレクト (PBR) を設定します。

（注） ポリシー ベースのリダイレクトの機能は、GUI では「policy-based routing」と呼ばれます。

ステップ 1 メニュー バーで、**[Tenants]** > **[All Tenants]** の順に選択します。

ステップ 2 [Work] ペインで、テナントの名前をダブルクリックします。

ステップ 3 [Navigation] ウィンドウで、**Tenant** *tenant_name* > **Services** > **L4-L7** > **Devices** を選択します。

ステップ 4 作業ウィンドウで、 **Actions** > **Create L4-L7 Devices** を選択します。

ステップ 5 **Create L4-L7 Devices** ダイアログボックスで、必要に応じてフィールドに入力します。

General セクションの **Service Type** は、**Firewall** または **ADC** にできます。

ステップ 6 ナビゲーション ウィンドウで、**Tenant** *tenant_name* > **Services** > **L4-L7** > **Service Graph Templates** を選択します。

ステップ 7 作業ウィンドウで、**Action** > **Create L4-L7 Service Graph Template** を選択します。

ステップ 8 **Create L4-L7 Service Graph Template** ダイアログボックスで、次の操作を実行します:

a) **Graph Name** フィール小渡に、サービス グラフ テンプレートの名前を入力します。

b) **Graph Type** ラジオ ボタンで、**Create A New Graph** をクリックします。

c) **Device Clusters** ペインで作成したデバイスを、コンシューマ エンドポイント グループとプロバイダ エンドポイント グループの間にドラッグ アンド ドロップします。これで、サービス ノードが作成されます。

APIC リリース 3.2(1) においては、オプションとしてステップ c PBR を繰り返すことで、PBR をサポートするには、最大 3 つのサービス ノードのデバイスを含めることができます。

d) デバイスのサービスの種類に基づいて、以下を選択します:

ファイアウォールの場合には、**Routed** を選択して、次の手順を続けます。

ADC の場合には、**One-Arm** または **Two-Arm** を選択して、次の手順を続けます。

e) **Profile** ドロップダウンリストで、デバイスに適した機能プロファイルを選択します。プロファイルが存在しない場合は、 GUI を使用した機能プロファイルの作成の手順に従って作成します。

f) **Route Redirect** チェックボックスをオンにします。

g) [Submit] をクリックします。

新しいサービス グラフ テンプレートが [Service Graph Templates] テーブルに表示されます。

ステップ 9 ナビゲーション ウィンドウで、**Tenant** *tenant_name* > **Policies** > **Protocol** > **L4-L7 Policy Based Redirect** を選択します。

ステップ 10 作業ウィンドウで、**Action** > **Create L4-L7 Policy Based Redirect** を選択します。

ステップ 11 **Create L4-L7 Policy Based Redirect** ダイアログボックスで、必要に応じてフィールドに入力します。このポリシーベースのリダイレクト ポリシーは、コンシューマ コネクタ用のものです。

ステップ 12 プロバイダ コネクタ用には、別のポリシー ベースのリダイレクト ポリシーを作成します。

ステップ 13 ナビゲーション ウィンドウで、**Tenant** ***tenant_name*** **> Services > L4-L7 > Service Graph Templates >** ***service_graph_template_name*** を選択します。

作成したサービス グラフ テンプレートを選択します。

ステップ 14 サービス グラフ テンプレートを右クリックして、**Apply L4-L7 Service Graph Template** を選択します。

ステップ 15 **Apply L4-L7 Service Graph Template to EPGs** ダイアログボックスで、次の操作を実行します:

a) **Consumer EPG/External Network** ドロップダウンリストで、コンシューマ エンドポイント グループを選択します。

b) **Provider EPG/External Network** ドロップダウンリストで、プロバイダ エンドポイント グループを選択します。

c) **Contract** オプション ボタンの **Create A New Contract** をクリックします。

d) **Contract Name** フィールドに、契約の名前を入力します。

e) **No Filter (Allow All Traffic)** チェック ボックスはオンにしないでください。

f) **Filter Entries** テーブルで **+** をクリックしてエントリを追加します。

g) 新しいフィルタ エントリで、名前として [IP] を入力し、**IP** を **Ether Type** として選択して、**Update** をクリックします。

h) **Next** をクリックします。

i) コンシューマ コネクタの **BD** ドロップダウンリストで、コンシューマ エンドポイント グループに接続している外部ブリッジ ドメインを選択します。ブリッジ ドメインでは、**Enable Dataplane Learning** チェックボックスをオフにする必要があります。

j) コンシューマ コネクタの **Redirect Policy** ドロップダウンリストで、コンシューマ コネクタ用に作成したリダイレクト ポリシーを選択します。

k) コンシューマ コネクタの **Cluster Interface** ドロップダウンリストで、コンシューマ クラスタ インターフェイスを選択します。

l) プロバイダ コネクタの **BD** ドロップダウンリストで、コンシューマ エンドポイント グループに接続している内部ブリッジ ドメインを選択します。ブリッジ ドメインでは、**Enable Dataplane Learning** チェックボックスをオフにする必要があります。

m) プロバイダ コネクタの **Redirect Policy** ドロップダウンリストで、プロバイダ コネクタ用に作成したリダイレクト ポリシーを選択します。

n) プロバイダ コネクタの **Cluster Interface** ドロップダウンリストで、プロバイダ クラスタ インターフェイスを選択します。

o) **Next** をクリックします。

p) パラメータをデバイスでの必要に合わせて設定します。

q) **Finish** をクリックします。

NX-OS スタイルの CLI を使用したポリシー ベース リダイレクトの設定

この手順のコマンド例にはには、ルート リダイレクト、クラスタのリダイレクト、およびグラフの導入が含まれます。デバイスはテナント T1 の下に作成されます。デバイスは管理対象モードの Cisco ASA 仮想デバイスになります。アンマネージド モードのデバイスだけが CLI で設定できます。

ステップ 1 デバイス クラスタを作成します。

例：

```
l417 cluster name ifav-asa-vm-ha type virtual vlan-domain ACIVswitch service FW function go-to
    cluster-device Device2 vcenter ifav108-vcenter vm "ASAv_HA1"
    cluster-device Device1 vcenter ifav108-vcenter vm "ASAv_HA"
    cluster-interface provider
      member device Device1 device-interface GigabitEthernet0/1
        interface ethernet 1/45 leaf 102
        vnic "Network adapter 3"
        exit
      member device Device2 device-interface GigabitEthernet0/1
        interface ethernet 1/45 leaf 102
        vnic "Network adapter 3"
        exit
      exit
    cluster-interface failover_link
      member device Device1 device-interface GigabitEthernet0/8
        interface ethernet 1/45 leaf 102
        vnic "Network adapter 10"
        exit
      member device Device2 device-interface GigabitEthernet0/8
        interface ethernet 1/45 leaf 102
        vnic "Network adapter 10"
        exit
      exit
    cluster-interface consumer
      member device Device1 device-interface GigabitEthernet0/0
        interface ethernet 1/45 leaf 102
        vnic "Network adapter 2"
        exit
      member device Device2 device-interface GigabitEthernet0/0
        interface ethernet 1/45 leaf 102
        vnic "Network adapter 2"
        exit
      exit
    exit
exit
```

ステップ 2 テナント PBRv6_ASA_HA_Mode の下に、PBR サービス グラフ インスタンスを展開します。

例：

```
tenant PBRv6_ASA_HA_Mode
  access-list Contract_PBRv6_ASA_HA_Mode_Filter
    match ip
    exit
```

ステップ 3 フィルタが IP プロトコルに一致する PBR 用の契約を作成します。情報カテゴリの下で、レイヤ 4 ～ レイヤ 7 サービス グラフ名を指定します。

サービス アプライアンスのプロバイダ エンドポイント グループによって提供される契約は、`allow-all` 設定では構成できません。

例：

```
contract Contract_PBRv6_ASA_HA_Mode
  scope tenant
  subject Subject
    access-group Contract_PBRv6_ASA_HA_Mode_Filter both
    l4l7 graph PBRv6_ASA_HA_Mode_Graph
    exit
  exit
vrf context CTX1
  exit
vrf context CTX2
  exit
```

ステップ 4 クライアントとサーバのエンドポイント グループ用にブリッジ ドメインを作成します。クライアントとサーバの両方が同じ VRF インスタンスに属します。

例：

```
bridge-domain BD1
  arp flooding
  l2-unknown-unicast flood
  vrf member CTX1
  exit
bridge-domain BD2
  arp flooding
  l2-unknown-unicast flood
  vrf member CTX1
  exit
```

ステップ 5 ファイアウォールの内部および外部レッグ用には、別のブリッジ ドメインを作成します。

PBR では、リモート リーフ スイッチの送信元 VTEP の学習が無効になっている必要があります。これは、**no ip learning** コマンドで行います。

例：

```
bridge-domain External-BD3
  arp flooding
  no ip learning
  l2-unknown-unicast flood
  vrf member CTX1
  exit
bridge-domain Internal-BD4
  arp flooding
  no ip learning
  l2-unknown-unicast flood
  vrf member CTX1
  exit
```

ステップ 6 アプリケーションプロファイルを作成し、エンドポイント グループを指定します。

例：

```
application AP1
  epg ClientEPG
    bridge-domain member BD1
```

```
        contract consumer Contract_PBRv6_ASA_HA_Mode
        exit
      epg ServerEPG
        bridge-domain member BD2
        contract provider Contract_PBRv6_ASA_HA_Mode
        exit
      exit
```

ステップ 7 ブリッジ ドメインのデフォルト ゲートウェイを指定します。

例：

```
    interface bridge-domain BD1
      ipv6 address 89:1:1:1::64/64
      exit
    interface bridge-domain BD2
      ipv6 address 99:1:1:1::64/64
      exit

 interface bridge-domain External-BD3
      ipv6 address 10:1:1:1::64/64
      exit
    interface bridge-domain Internal-BD4
      ipv6 address 20:1:1:1::64/64
      exit
```

ステップ 8 テナント T1 からデバイスをインポートします。

例：

```
l4l7 cluster import-from T1 device-cluster ifav-asa-vm-ha
```

ステップ 9 サービス リダイレクト ポリシーを使用してサービス グラフを作成します。

例：

```
l4l7 graph PBRv6_ASA_HA_Mode_Graph contract Contract_PBRv6_ASA_HA_Mode
      service N2 device-cluster-tenant T1 device-cluster ifav-asa-vm-ha mode FW_ROUTED svcredir
enable
      connector consumer cluster-interface consumer_PBRv6
          bridge-domain tenant PBRv6_ASA_HA_Mode name External-BD3
          svcredir-pol tenant PBRv6_ASA_HA_Mode name External_leg
          exit
      connector provider cluster-interface provider_PBRv6
          bridge-domain tenant PBRv6_ASA_HA_Mode name Internal-BD4
          svcredir-pol tenant PBRv6_ASA_HA_Mode name Internal_leg
          exit
        exit
    connection C1 terminal consumer service N2 connector consumer
    connection C2 terminal provider service N2 connector provider
    exit
```

ステップ 10 外部および内部レッグのサービス リダイレクトのポリシーを作成します。IPv6 アドレスは次の例で使用されます。同じコマンドを使用して IPv4 アドレスを指定することもできます。

例：

```
svcredir-pol Internal_leg
    redir-dest 20:1:1:1::1 00:00:AB:CD:00:11
    exit
svcredir-pol External_leg
    redir-dest 10:1:1:1::1 00:00:AB:CD:00:09
```

```
    exit
exit
```

NX-OS スタイルの CLI を使用したポリシー ベースのリダイレクト設定を確認する

ポリシー ベースのリダイレクトを設定した後は、NX-OS スタイル CLI を使用して設定を確認できます。

ステップ 1 テナントの実行設定を表示します。

例：

```
apic1# show running-config tenant PBRv6_ASA_HA_Mode svcredir-pol
# Command: show running-config tenant PBRv6_ASA_HA_Mode svcredir-pol
# Time: Wed May 25 00:57:22 2016
  tenant PBRv6_ASA_HA_Mode
    svcredir-pol Internal_leg
      redir-dest 20:1:1:1::1/32 00:00:AB:CD:00:11
      exit
    svcredir-pol External_leg
      redir-dest 10:1:1:1::1/32 00:00:AB:CD:00:09
      exit
    exit
```

ステップ 2 テナントとそのサービス グラフの実行設定を表示します。

例：

```
apic1# show running-config tenant PBRv6_ASA_HA_Mode l4l7 graph PBRv6_ASA_HA_Mode_Graph
# Command: show running-config tenant PBRv6_ASA_HA_Mode l4l7 graph PBRv6_ASA_HA_Mode_Graph
# Time: Wed May 25 00:55:09 2016
  tenant PBRv6_ASA_HA_Mode
    l4l7 graph PBRv6_ASA_HA_Mode_Graph contract Contract_PBRv6_ASA_HA_Mode
      service N2 device-cluster-tenant T1 device-cluster ifav-asa-vm-ha mode FW_ROUTED svcredir
enable
        connector consumer cluster-interface consumer_PBRv6

          bridge-domain tenant PBRv6_ASA_HA_Mode name External-BD3

          svcredir-pol tenant PBRv6_ASA_HA_Mode name External_leg

          exit

        connector provider cluster-interface provider_PBRv6

          bridge-domain tenant PBRv6_ASA_HA_Mode name Internal-BD4
          svcredir-pol tenant PBRv6_ASA_HA_Mode name Internal_leg
          exit
        exit
      connection C1 terminal consumer service N2 connector consumer
      connection C2 terminal provider service N2 connector provider
      exit
    exit
```

ステップ 3 サービス グラフ設定を表示します。

例：

```
apic1# show l4l7-graph graph PBRv6_ASA_HA_Mode_Graph
Graph           : PBRv6_ASA_HA_Mode-PBRv6_ASA_HA_Mode_Graph
Graph Instances : 1

Consumer EPg     : PBRv6_ASA_HA_Mode-ClientEPG
Provider EPg     : PBRv6_ASA_HA_Mode-ServerEPG
Contract Name    : PBRv6_ASA_HA_Mode-Contract_PBRv6_ASA_HA_Mode
Config status    : applied
Service Redirect : enabled

Function Node Name : N2
 Connector   Encap       Bridge-Domain  Device Interface      Service Redirect Policy
 ----------  ----------  ----------     --------------------  -----------------------
 consumer    vlan-241    PBRv6_ASA_HA_  consumer_PBRv6        External_leg
                         Mode-
                         External-BD3
 provider    vlan-105    PBRv6_ASA_HA_  provider_PBRv6        Internal_leg
                         Mode-
                         Internal-BD4
```

ポリシー ベースのリダイレクトとサービス ノードのトラッキング

（注） この機能は、APIC リリース 2.2(3x) リリースおよび APIC リリース 3.1 (1) でさせてで使用できます。APIC Release 3.0(x) ではサポートされていません。

Cisco APIC、リリース 2.2(3x) とポリシーベースのリダイレクトとサービス ノードの追跡 (PBR) のサポートが機能します。

宛先ノードのサポート デュアル IP スタックをリダイレクトします。したがって、IPv4 と IPv6 の両方のアドレスは、同時に設定できます。

スイッチは、トラッキング PBR をサポートするのに Cisco IP SLA モニタリング機能を内部的に使用します。トラッキング機能では、サービス ノードに到達できない場合に、リダイレクト宛先ノードがマークされます。トラッキング機能は、サービス ノードの接続を再開するかどうかリダイレクト宛先ノードを示します。サービス ノードがマークダウンときに送信または、トラフィックのハッシュを使用できません。代わりに、トラフィックを送信またはリダイレクト宛先ノードのクラスタ内の異なるサービス ノードにハッシュがされます。

一方向のトラフィックのブラック holing を避けるためには、リダイレクト正常性ポリシー サービス ノードの入力と出力をリダイレクト宛先ノードを関連付けることができます。これによりを入力または出力のいずれかのリダイレクト宛先ノードがダウンしている場合、その他のリダイレクト宛先ノードもマークダウンされます。したがって、入力と出力トラフィックの両方のは、リダイレクト宛先ノードのクラスタ内の異なるサービス ノードにハッシュを取得します。

しきい値設定

サービス ノードを追跡するため PBR ポリシーを設定するとき、次のしきい値の設定を使用できます。

- しきい値の有効化または無効化：しきい値が有効になっているとき、最小および最大のしきい値のパーセンテージを指定します。リダイレクト先グループを完全に無効にして、リダイレクトを防止したい場合は、有効になっているしきい値は必須です。リダイレクトがないときに、トラフィックがコンシューマとプロバイダ間で直接送信されます。
- 最小しきい値：指定した最小しきい値のパーセンテージ。トラフィックが最小パーセンテージを下回る場合、リダイレクトではなくパケットが許可されます。デフォルト値は 0 です
- 最大しきい値：指定された最大しきい値のパーセンテージ。最小しきい値に達すると、操作状態に戻すため最大パーセンテージに最初に到達する必要があります。デフォルト値は 0 です

例として、ポリシーに 3 つのリダイレクト先があると仮定してみましょう。最小しきい値が 70% に指定されており、最大しきい値が 80% に指定されています。3 つのリダイレクト先ポリシーのいずれかがダウンすると、 1/3、つまり最小しきい値以下の 33% 可用性パーセンテージが下がります。その結果、リダイレクト先グループの最小しきい値のパーセンテージがダウンし、トラフィックがリダイレクトではなく許可の取得を開始します。同じ例で続けると、最大しきい値が 80% の場合、リダイレクト ポリシー先グループを操作状態に戻すため、最大しきい値のパーセンテージ以上のパーセンテージに最初に達する必要があります。

ポリシー ベース リダイレクトとトラッキング サービス ノードについての注意事項と制約事項

PBR トラッキングおよびサービス ノードを利用するときに、これらの注意事項と制約事項に従います。

- リリース 4.0(1) 以降では、システムレベルのグローバル GIPo が有効になっている場合に限り、リモート リーフ設定で PBR トラッキングがサポートされます。「*GUI* を使用してリモート リーフのグローバル *GIPo* を構成する」を参照してください。
- リリース 4.0(1) 以降では、リモート リーフ設定で PBR の復元力のあるハッシュがサポートされています。
- マルチポッド ファブリック設定はサポートされています。マルチサイト セットアップはサポートされていません。
- コンシューマとプロバイダ Epg のレイヤ 3 Out はサポートされます。
- リダイレクト宛先ノードの追跡では、TCP または ICMP プロトコル タイプが使用されます。

- ポリシー ベース リダイレクトでサポートされる追跡可能 IP アドレスの最大数は、リーフ スイッチで 100、ACI ファブリックでは 200 です。
- ACI ファブリックでのグラフ インスタンスの最大数は、ファブリックあたり 1000 です。
- グラフ インスタンスの最大数は、デバイスあたり 100 です。
- PBR を設定できるサービス ノードの最大数は、ポリシーあたり 40 です。
- 1 つのサービス チェーンでサポートされるサービス ノードの最大数は 3 です。
- PBR トラッキングでは、共有サービスがサポートされています。
- 許可アクションまたは拒否アクションはサポートされています。

PBR を設定し、GUI を使用してサービス ノードのトラッキング

ステップ 1 メニュー バーで [Tenant] > **テナント名**をクリックします。[Navigation] ペインで、[Policies] > [Protocol] > [L4-L7 Policy Based Redirect] をクリックします。

ステップ 2 右クリックして **L4 ~ L7 ポリシー ベースのリダイレクト** をクリックします **作成 L4 ~ L7 ポリシー ベースのリダイレクト** 。

ステップ 3 **Create L4–L7 Policy Based Redirect** ダイアログボックスで、次の操作を実行します:

a) **Name** フィールドに PBR ポリシーの名前を入力します。

b) ダイアログボックスでは、ハッシュ アルゴリズムの、IP SLA モニタリング ポリシー、およびその他の必要な値を設定する適切な設定を選択します。

c) しきい値の設定フィールドでは、必要に応じて設定を指定し、必要な場合。

d) [Destinations] を展開して [Create Destination of Redirected Traffic] を表示します。

e) **リダイレクト トラフィックの宛先の作成** ダイアログボックスなどの適切な詳細を入力します **IP** アドレス、および **MAC アドレス** フィールド。

IP アドレスおよび 2 番目の IP アドレス（IPv4 アドレス/IPv6 アドレス）を指定できるフィールドが表示されます。

（注） このフィールドは必須ではありません。L4-L7 デバイスに複数の IP アドレスがあり、ACI でそれらの両方を確認する必要がある場合に使用します。

[IP] と [Second IP] の両方のパラメータを設定した場合、PBR 宛先が「UP」とマーキングされるには、両方がアップ状態である必要があります。

f) **ヘルス グループのリダイレクト** フィールドで、既存のヘルス グループに関連付けるまたは必要に応じて、新しいヘルス グループを作成します。[OK] をクリックします。

g) **Create L4–L7 Policy Based Redirect** ダイアログボックスで **Submit** をクリックします。

L4 L7 ポリシー ベースのリダイレクトとサービス ノードのトラッキング L4 L7 PBR ポリシーおよびリダイレクト宛先グループを追跡するための設定にリダイレクト ヘルス グループ ポリシーが有効になっているバインディングの後に設定されます。

GUI を使用したインポート ポリシーの設定

ステップ 1 メニュー バーで、**Tenant** > **Tenant_name** をクリックします。**Navigation** ウィンドウで、**Networking** > **Protocol Policies** > **L4-L7 Redirect Health Groups** をクリックします。

ステップ 2 **L4 –L7 Redirect Health Groups** を右クリックし、**Create L4–L7 Redirect Health Group** をクリックします。

ステップ 3 **Create L4–L7 Redirect Health Group** ダイアログボックスで、次の操作を実行します。

a) **Name** フィールドに、 リダイレクト正常性ポリシーの名前を入力します。

b) 適切であれば、**Description** フィールドに追加の情報を入力し、**Submit** をクリックします。

L4 ～ L7 リダイレクト正常性ポリシーが設定されます。

GUI を使用した IP SLA モニタリング ポリシーの設定

ステップ 1 メニューバーで、**Tenant**>**Tenant_name** をクリックします。**Navigation** ウィンドウで、**Policies**>**Protocol**>**IP SLA Monitoring Policies** をクリックします。

ステップ 2 **IP SLA Monitoring Policies** を右クリックして、**Create IP SLA Monitoring Policy** をクリックします。

ステップ 3 **Create IP SLA Monitoring Policy** ダイアログボックスで、次の操作を実行します:

a) **Name** フィールドに、 IP SLA モニタリング ポリシーの名前を入力します。

b) **SLA Frequency** フィールドに、インターバル プローブ時間を秒単位で入力します。最小のインターバル時間は 1 秒です。

c) **SLA Type** フィールドで、SLAタイプを選択します。[Submit] をクリックします。

（注） 現在のところ、**SLA Type** としては、tcp だけがサポートされています。

SLA タイプとしては TCP または ICMP が可能です。ICMP がデフォルト値です。

これで IP SLA モニタリング ポリシーが設定されます。

GUI を使用してリモート リーフのグローバル GIPo を構成する

このタスクを実行すると、リモート リーフ設定で PBR トラッキングを機能させることができます。

（注） リモート リーフで PBR トラッキングを機能させるには、この設定を行う必要があります。この設定を行わないと、メイン データセンターが到達可能でも、リモート リーフで PBR トラッキングは機能しません。

ステップ 1 メニュー バーで、**[System]** > **[System Settings]** の順にクリックします。

ステップ 2 [System Settings] ナビゲーション ウィンドウで [System Global GIPo] をクリックします。

ステップ 3 [System Global GIPo Policy] 作業ウィンドウで [Enabled] をクリックします。

ステップ 4 [Policy Usage Warning] ダイアログで、GIPo ポリシーを使用する可能性があるノードとポリシーを確認し、必要に応じて [Submit Changes] をクリックします。

REST API を使用したサービス ノードのトラッキングのサポートをする PBR の設定

トラッキング サービス ノードをサポートする PBR を設定します。

例：

```
<polUni>
    <fvTenant name="coke" >
    <fvIPSLAMonitoringPol name="tcp_Freq60_Pol1" slaType="tcp" slaFrequency="60" slaPort="2222" />
    <vnsSvcCont>
        <vnsRedirectHealthGroup name="fwService1"/>
          <vnsSvcRedirectPol name="fwExt" hashingAlgorithm="sip" thresholdEnable="yes"
minThresholdPercent="20" maxThresholdPercent="80">
            <vnsRedirectDest ip="40.40.40.100" mac="00:00:00:00:00:01">
              <vnsRsRedirectHealthGroup tDn="uni/tn-coke/svcCont/redirectHealthGroup-fwService1"/>

            </vnsRedirectDest>
            <vnsRsIPSLAMonitoringPol tDn="uni/tn-coke/ipslaMonitoringPol-tcp_Freq60_Pol1"/>
          </vnsSvcRedirectPol>
         <vnsSvcRedirectPol name="fwInt" hashingAlgorithm="sip" thresholdEnable="yes"
minThresholdPercent="20" maxThresholdPercent="80">
             <vnsRedirectDest ip="30.30.30.100" mac="00:00:00:00:00:02">
              <vnsRsRedirectHealthGroup tDn="uni/tn-coke/svcCont/redirectHealthGroup-fwService1"/>

             </vnsRedirectDest>
             <vnsRsIPSLAMonitoringPol tDn="uni/tn-coke/ipslaMonitoringPol-tcp_Freq60_Pol1"/>
          </vnsSvcRedirectPol>
        </vnsSvcCont>
    </fvTenant>
</polUni>
```

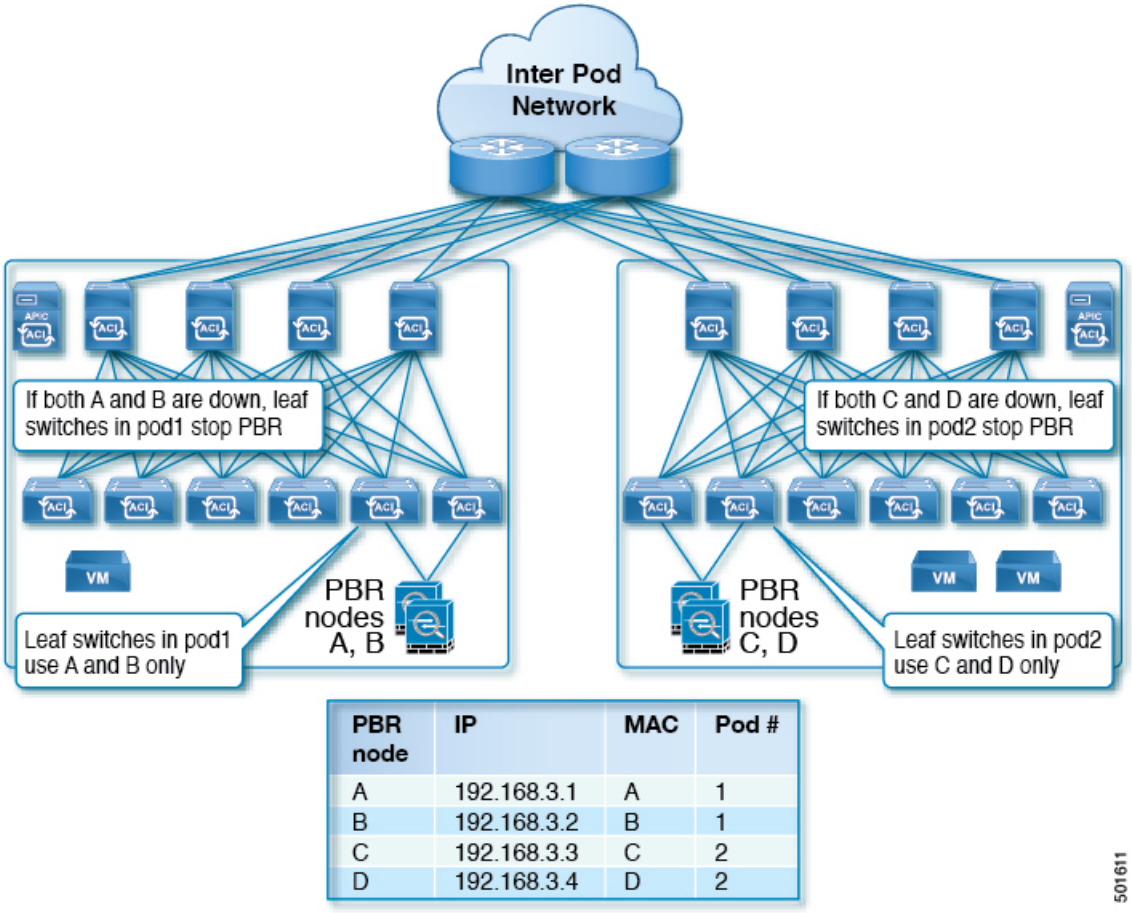
ベースリダイレクトの場所に対応したポリシーについて

ロケーション対応ポリシー ベースのリダイレクト (PBR) はサポートされています。この機能は、multipod 設定シナリオに役立ちます。ここでは、ポッド認識サポートされ、優先ローカル PBR ノードを指定できます。ロケーション対応のリダイレクトを有効にすると、ポッド Id が指定されて、レイヤ 4 ~ レイヤ 7 PBR ポリシー内のすべてのリダイレクト宛先はポッド認識必

要があります。リダイレクト宛先は、特定のポッドにあるリーフ スイッチでのみプログラムされます。

次の図は、2 個のポッドの例を表示します。ポッド 1 で PBR ノード A と B、C と D PBR ノードがポッド 2 では。ポッド 1 のリーフ スイッチが A、B、PBR ノードを使用する prefer し、ポッド 2 のリーフ スイッチ C と D で PBR ノードの使用場所に対応した PBR 設定を有効にするとPBR ノード A と B ポッド 1 では、ダウンは、[ポッド 1 のリーフ スイッチと開始 PBR ノード C と D を使用するには同様に、PBR ノード C と D ポッド 2 では、ダウンが、ポッド 2 のリーフ スイッチと開始 PBR ノード A および B を使用するには

図 *20 : 2* 個のポッドのロケーション対応 *PBR* 設定の例



PBR node	IP	MAC	Pod #
A	192.168.3.1	A	1
B	192.168.3.2	B	1
C	192.168.3.3	C	2
D	192.168.3.4	D	2

ロケーション認識型 PBR の注意事項

ロケーション認識型 PBR を活用する際はこれらの注意事項に従ってください。

- Cisco Nexus 9300（Cisco Nexus 9300 EX および 9300 FX を除く）プラットフォーム スイッチは、ロケーション認識型 PBR 機能をサポートしていません。
- GOLF ホスト アドバタイズメントと北南ファイアウォール連携にロケーション認識型 PBR を使用します。

GUI を使用したロケーション認識型 PBR の設定

この機能を有効にするための 2 つの項目をプログラムする必要があります。ポッド ID 認識リダイレクトを有効にし、特定のポッドにあるリーフ スイッチで、リダイレクト宛先をプログラムして、優先 PBR ノードにポッド ID を関連付けます。

ステップ 1 メニュー バーで [Tenant] > **テナント名**をクリックします。[Navigation] ペインで、[Policies] > [Protocol] > [L4-L7 Policy Based Redirect] をクリックします。

ステップ 2 右クリックして **L4 ~ L7 ポリシー ベースのリダイレクト** をクリックします **作成 L4 ~ L7 ポリシー ベースのリダイレクト** 。

ステップ 3 **Create L4–L7 Policy Based Redirect** ダイアログボックスで、次の操作を実行します:

a) **Name** フィールドに PBR ポリシーの名前を入力します。

b) **[ポッド ID 認識リダイレクトの有効化]** チェック ボックスをオンにします。

c) ダイアログ ボックスでハッシュ アルゴリズム、IP SLA モニタリング ポリシー、およびその他の必要な値を構成するため、適切な設定を選択します。

d) しきい値の設定フィールドでは、必要に応じて設定を指定し、必要な場合。

e) [Destinations] を展開して [Create Destination of Redirected Traffic] を表示します。

f) **リダイレクト トラフィックの宛先の作成** ダイアログボックスなどの適切な詳細を入力します **IP** アドレス、および **MAC アドレス** フィールド。

IP アドレスと 2 番目の IP アドレスのフィールドでは、IPv4 アドレスと IPv6 あれドレスを指定できます。

g) **[ポッド ID]** フィールドに、ポッド ID 値を入力します。

h) **[リダイレクト ヘルス グループ]** フィールドで、既存のヘルス グループに関連付けるか、適切であれば、新しいヘルス グループを作成します。[OK] をクリックします。

必要に応じて別のポッド ID にリダイレクトされたトラフィックの他の宛先を作成します。

i) **Create L4–L7 Policy Based Redirect** ダイアログボックスで **Submit** をクリックします。

L4-L7 ロケーション認識型 PBR が設定されています。

REST API を使用して設定の場所に対応した PBR

2 つ設定する必要があります項目の場所に対応した PBR を有効にして、プログラムが特定のポッドにあるリーフ スイッチ内の送信先をリダイレクトします。次の例の場所に対応した PBR を有効にするよう設定されている属性が: programLocalPodOnly と podId 。

ロケーション対応 PBR を設定します。

例：

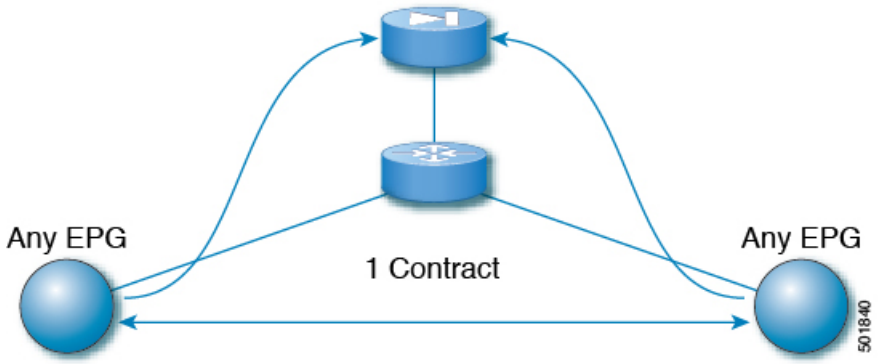
```
<polUni>
```

```
    <fvTenant name="coke" >
    <fvIPSLAMonitoringPol name="icmp_Freq60_Pol1" slaType="icmp" slaFrequency="60"/>
    <vnsSvcCont>
        <vnsRedirectHealthGroup name="fwService1"/>
          <vnsSvcRedirectPol name="fwExt" hashingAlgorithm="sip" thresholdEnable="yes"
minThresholdPercent="20" maxThresholdPercent="80" programLocalPodOnly="yes">
            <vnsRedirectDest ip="40.40.40.100" mac="00:00:00:00:00:01" podId="2">
              <vnsRsRedirectHealthGroup tDn="uni/tn-coke/svcCont/redirectHealthGroup-fwService1"/>

            </vnsRedirectDest>
            <vnsRsIPSLAMonitoringPol tDn="uni/tn-coke/ipslaMonitoringPol-icmp_Freq60_Pol1"/>
          </vnsSvcRedirectPol>
          <vnsSvcRedirectPol name="fwInt" hashingAlgorithm="dip" thresholdEnable="yes"
minThresholdPercent="20" maxThresholdPercent="80">
            <vnsRedirectDest ip="30.30.30.100" mac="00:00:00:00:00:02">
              <vnsRsRedirectHealthGroup tDn="uni/tn-coke/svcCont/redirectHealthGroup-fwService1"/>

            </vnsRedirectDest>
            <vnsRsIPSLAMonitoringPol tDn="uni/tn-coke/ipslaMonitoringPol-icmp_Freq60_Pol1"/>
          </vnsSvcRedirectPol>
        </vnsSvcCont>
    </fvTenant>
</polUni>
```

同じVRFインスタンス内のすべてのEPG-EPGにトラフィックをリダイレクトするには、ポリシーベースのリダイレクトとサービス グラフ

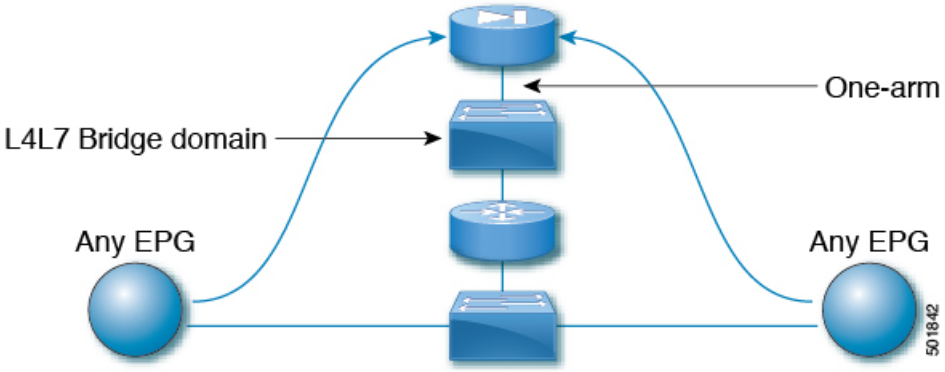
設定できる Cisco Application Centric Infrastructure (Cisco ACI) サービス グラフ リダイレクト vzAny と vzAny の設定によって、デバイスはすべてのエンドポイントを表す構築をレイヤ 7 にレイヤ 4 で同じ:VRF インスタンス内の他のエンドポイント グループをすべてのエンドポイント グループからのすべてのトラフィックを転送するには。同じVRFインスタンスでグループ。vzAny は「any EPG」と呼ばれることがあります。

図 *21 : vzAny* トポロジ



同じ VRF インスタンスの下にある任意のエンドポイント グループ ペア間のトラフィックは、ファイアウォールなどのレイヤ 4 からレイヤ 7 デバイスにリダイレクトできます。また、同じブリッジ ドメイン内のトラフィックをファイアウォールにリダイレクトすることもできます。ファイアウォールは、次の図に示すように、任意の一対のエンドポイント グループ間のトラフィックをフィルタリングできます。

図 *22*: 任意の *EPG* ペア間のトラフィックをフィルタリングするファイアウォール

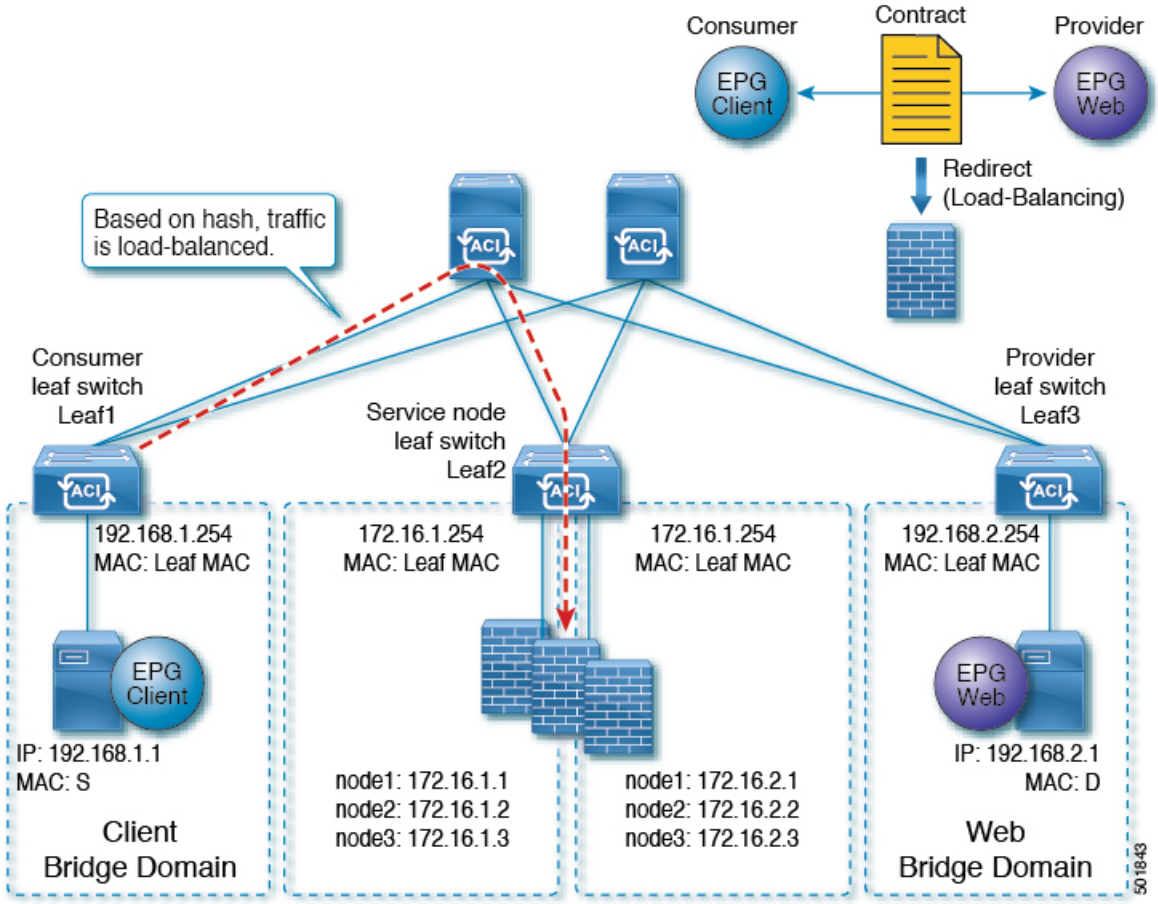


この機能の 1 つの使用例は、Cisco ACI をデフォルト ゲートウェイとして使用することですが、ファイアウォールを通るトラフィックをフィルタリングすることもそうです。vzAny とポリシー ベースのリダイレクト ポリシーにより、セキュリティ管理者は ACL ルールを管理し、ネットワーク管理者はルーティングとスイッチングを管理します。この設定の利点には、エンドポイント トラッキング、ARP インスペクションによるファースト ホップ セキュリティ、IP アドレス ソース ガードなどの Cisco Application Policy Infrastructure Controller (Cisco APIC) ツールを使用できることが含まれます。

ポリシー ベースのリダイレクト ポリシーを使用してサービス グラフを適用すると、次の機能も有効になります。

- ファイアウォール クラスタリング
- ファイアウォールの健全性追跡
- 位置認識リダイレクション

図 *23 :* ファイアウォール クラスタリング



Cisco APIC 3.2 のリリースより前に、vzAny を契約のコンシューマとして使用することができました。Cisco APIC 3.2 のリリースから、vzAny を契約のプロバイダとして使用することもできます。この拡張により、以下の構成が可能になります。

- プロバイダとしての vzAny、コンシューマとしての vzAny (ワンアームのみのポリシーベースのリダイレクト)
- プロバイダとしての vzAny、およびコンシューマとしての通常のエンドポイント グループ (ポリシーベースのリダイレクトおよび非ポリシーベースのリダイレクトの場合)

vzAny を使用してトラフィックをリダイレクトするポリシー ベースのリダイレクト ポリシーを使用してサービス グラフを適用した後、2 つのサーバ間のデータ バックアップ トラフィックなどのトラフィックがファイアウォールをバイパスするようにする場合には、エンドポイント グループ間でより具体的な契約を作成することができます。たとえば、2 つのエンドポイント グループは、特定のポート上でトラフィックを相互に直接送信できます。より具体的なルールは、「任意のEPGから任意のEPGへ」リダイレクト ルールに優先します。

同じ VRF インターフェイス内のすべての EPG 間トラフィックをリダイレクトするために、ポリシーベースのリダイレクトポリシーをサービス グラフとともに設定する際の注意事項と制約事項

次の注意事項と制約事項は、同じ VRF インターフェイス内のすべての EPG 間トラフィックをリダイレクトするために、ポリシーベースのリダイレクト ポリシーをサービス グラフとともに設定する際に適用されます。

- レイヤ 4 ～ 7 デバイスと vzAny は、同じ VRF インスタンスに属している必要があります。
- レイヤ 4 ～ 7 デバイスはワンアーム モードで展開する必要があります。
- 複数ノードのサービス グラフで設定された vzAny も機能する可能性はありますが、この設定は試験されておらず、サポートされません。自身のリスクにおいて使用してください。
- レイヤ 4 ～ 7 デバイスは、アンマネージド モードでのみ展開できます。
- VRF リーキングと組み合わせた使用は、実装されていません。VRF インスタンスの vzAny に、他の VRF インスタンスの vzAny の契約の提供または利用を行わせることはできません。
- 異なるテナントのエンドポイント グループと vzAny の間で契約を設定することは、 VRF インスタンスがテナント **Common** にある場合のように、同じ VRF に属している限りにおいて可能です。
- マルチポッド環境では、 vzAny をプロバイダおよびコンシューマとして使用できます。
- Cisco ACI マルチサイト環境では、vzAny をサイト間でのプロバイダおよびコンシューマとして使用することはできません。

同じ VRF インターフェイス内のすべての EPG 間トラフィックをリダイレクトするために、ポリシーベースのリダイレクトポリシーをサービス グラフとともに設定する

次の手順では、同じ VRF インスタンス内のすべての EPG-EPG にトラフィックをリダイレクトするサービス グラフでポリシー ベースのリダイレクト ポリシーで設定します。

ステップ 1 レイヤ 4 レイヤ 7 デバイスへの接続を割り当てるはサービス ブリッジ ドメインを作成します。

ブリッジ ドメインの作成については、*Cisco APIC* ベーシック コンフィギュレーション ガイド を参照してください。

ステップ 1 > メイン 画面。

a) **VRF** ドロップダウンリスト、エンドポイントのグループが含まれている:VRF インスタンスを選択します。
b) **転送** ドロップダウンリスト、選択した場合 **カスタム** 、次に、 **L2 不明なユニキャスト** ドロップダウンリストを選択できます **フラッド** 必要かどうか。

ステップ 2 > L3 設定 画面。

a) チェックがあることを確認します **ユニキャスト ルーティング** チェック ボックス。
b) **サブネット** テーブルで、サブネットを作成します。

ゲートウェイ IP アドレスは、レイヤ 7 デバイス インターフェイスをレイヤ 4 に与えるは IP アドレスと同じサブネット内にする必要があります。

c) チェックを外し、 **エンドポイント データ ラーニング** チェック ボックス。

ステップ 2 リダイレクト ポリシーを作成します。

a) **[Navigation]** ウィンドウで、**[Tenanttenant_name] > [Networking] > [Policies] > [Protocol] > [L4-L7 Policy Based Redirect]** を選択します。
b) 右クリックして **L4 L7 ポリシー ベースのリダイレクト**] を選択します **作成 L4 L7 ポリシー ベースのリダイレクト** 。
c) [Name] フィールドにポリシーの名前を入力します。
d) **宛先** テーブルで、をクリックして + 。
e) **リダイレクト トラフィックの宛先の作成** ダイアログ ボックスで、次の情報を入力します。

- **IP** : IP アドレスを入力レイヤ 7 デバイスにレイヤ 4 に割り当てるされます。ブリッジ ドメインに支えられている IP アドレスと同じサブネットの IP アドレスがあります。
- **MAC** : レイヤ 7 デバイスにレイヤ 4 に割り当てるが MAC アドレスを入力します。レイヤ 7 デバイスにレイヤ 4 のフェールオーバー時にも有効な MAC アドレスを使用する必要があります。たとえば、ASA ファイアウォール時これと呼ばれる、「仮想 mac です。」

f) その他の適切な値を入力し、クリックして **OK** 。
g) **作成 L4 L7 ポリシー ベースのリダイレクト** ダイアログ ボックスで、他の適切な値を入力し、クリックして **Submit** 。

ステップ 3 1 つの具体的なインターフェイスを 1 つの論理インターフェイス レイヤ 7 デバイスにレイヤ 4 を作成します。

レイヤ 7 デバイスにレイヤ 4 の作成についてを参照してください。 GUI を使用したレイヤ 4 ～ レイヤ 7 デバイスの作成 （13 ページ） 。

ステップ 4 ルート リダイレクトを有効になっていると、サービス グラフ テンプレートを作成します。

a) **Navigation** ウィンドウで、**Tenant tenant_name > Services > L4-L7 > Service Graph Template** を選択します。
b) 右クリックして **サービス グラフ テンプレート**] を選択します **サービス グラフ テンプレートの作成** します。
c) **Name** フィールドに、サービス グラフの名前を入力します。
d) 以前を作成していないレイヤ 7 デバイスにレイヤ 4 の場合、 **デバイス クラスタ**] ペインで、デバイスを作成します。

e) ドラッグ アンド ドロップ レイヤ 4 からレイヤ 7 デバイス、 **デバイス クラスタ** され、中間 EPG コンシューマとプロバイダー EPG にウィンドウ。
f) **L4L7** ラジオ ボタンをクリックします **ルーテッド** 。
g) チェック マークを残します、 **リダイレクト ルーティング** チェック ボックス。
h) [Submit] をクリックします。

ステップ 5 サービス グラフ vzAny (AnyEPG) エンドポイント グループに適用されます。

ステップ 1 > 契約 画面。

a) **Navigation** ウィンドウで、**Tenant tenant_name** > **Services** > **L4-L7** > **Service Graph Template** > ***service_graph_name*** を選択します。

service_graph_name は、作成したサービス グラフ テンプレートです。

b) サービス グラフ テンプレートを右クリックし、選択 **L4 L7 サービス グラフ テンプレートの適用** 。
c) **コンシューマ EPG/外部ネットワーク** ドロップダウンリスト、選択、 **AnyEPG** テナントに対応するリスト項目とのこれを使用する VRF インスタンス使用例。

たとえば、テナントは、「tenant1」:VRF インスタンスは「vrf1」で、選択 **tenant1/vrf1/AnyEPG** .

d) **プロバイダー EPG 内部ネットワーク/** ドロップダウンリスト、同じ選択 **AnyEPG** コンシューマ EPG 用に選択したリスト項目。
e) **Contract Name** フィールドに、契約の名前を入力します。
f) [Next] をクリックします。

ステップ 2 > グラフ 画面。

a) 両方の **BD**] ドロップダウン リスト、ステップ 1 で作成したレイヤ 7 サービス ブリッジ ドメインをレイヤ 4 を選択します。
b) 両方の **リダイレクト ポリシー**] ドロップ ダウン リストでは、この使用例用に作成したリダイレクト ポリシーを選択します。
c) コンシューマ コネクタの **クラスタ インターフェイス** ドロップダウンリスト、ステップ 3 で作成したクラスタ インターフェイス (論理インターフェイス) を選択します。
d) プロバイダー コネクタの **クラスタ インターフェイス** ドロップダウンリスト、ステップ 3 で作成した同じクラスタ インターフェイス (論理インターフェイス) を選択します。
e) [Finish] をクリックします。

第 11 章

Direct Server Return の設定

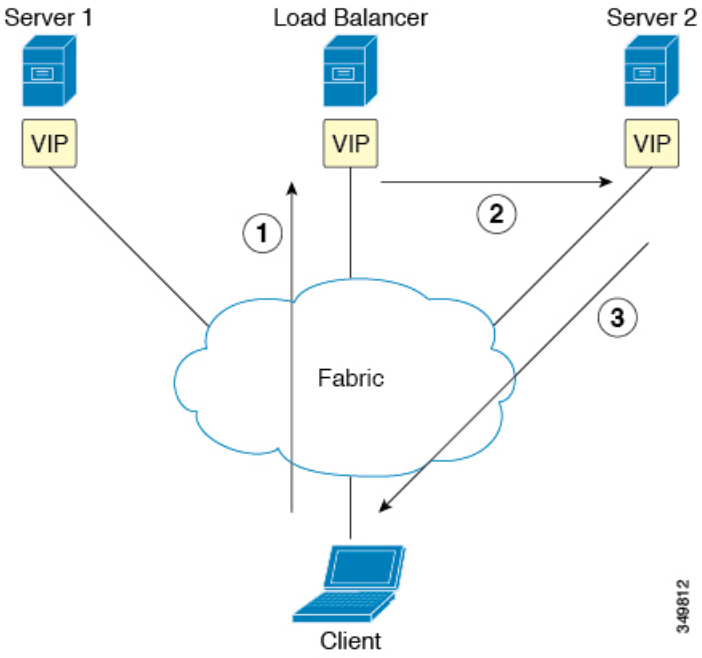
Direct Server Return について

Didrect Server Return 機能により、サーバはロード バランサを通過する必要なく、クライアントに直接応答できます。これにより、サーバからクライアントへのパスにおけるボトルネックが解消されます。従来のロードバランサの導入では、ロードバランサは、クライアントとサーバとの通信のパス（クライアントからサーバへの要求パスとサーバからクライアントへの応答パスの両方）に存在します。クライアントからサーバ方向の要求内のデータの量は比較的少ないものの、サーバからクライアントへの応答トラフィックはかなり大きく、クライアントからサーバへの要求データの約10倍になります。この大量の応答トラフィックがあるパス内のロード バランサがボトルネックになり、通信に悪影響を及ぼします。

Direct Server Return の導入では、ロード バランサとサーバとで仮想 IP アドレスが共有されます。クライアントは、ロード バランサに到達することを目的とした仮想 IP アドレスに常に要求を送信し、また、サーバからクライアントへの直接応答ではこの仮想 IP アドレスを送信元アドレスとして使用します。IP 送信元アドレスのデータパスの取得が有効になっているCisco Application Centric Infrastructure（ACI）は、サーバからクライアントへのトラフィックの仮想 IP アドレスを取得する際に問題を引き起こし、クライアントからロード バランサへの要求トラフィックを途絶させることになります。Direct Server Return の導入を適切に動作させるには、ACI ファブリックは通信中のエンドポイント間の要求と応答のトラフィックを目的の宛先に正しく配信されるようにする必要があります。これには、リーフ上でのデータパス IP アドレスの取得を、クライアントからロード バランサへのトラフィック、ロード バランサからサーバへのトラフィック、およびサーバからクライアントへのトラフィックに割り込みを生じさせないように制御することが必要です。

次の図に、Direct Server Return の導入のデータ パスを示します。

図 *24 : Direct Server Return* の全体的なフロー



1. ロード バランサとすべてのバックエンド サーバが仮想 IP アドレスで設定されています。ロードバランサのみが、この仮想 IP アドレス宛の Address Resolution Protocol（ARP）要求に応答します。クライアント要求のロード バランシング後に、ロード バランサはパケット内の宛先 MAC アドレスを書き換えて、その MAC アドレスをバックエンド サーバの 1 つに転送します。
2. 仮想 IP アドレスはバックエンド サーバ上に設定されますが、ARP が無効になっているため、この仮想 IP アドレス宛の ARP 要求にバックエンド サーバは応答できません。
3. サーバはリターン トラフィックをクライアントに直接送信してロード バランサをバイパスします。

レイヤ 2 の Direct Server Return

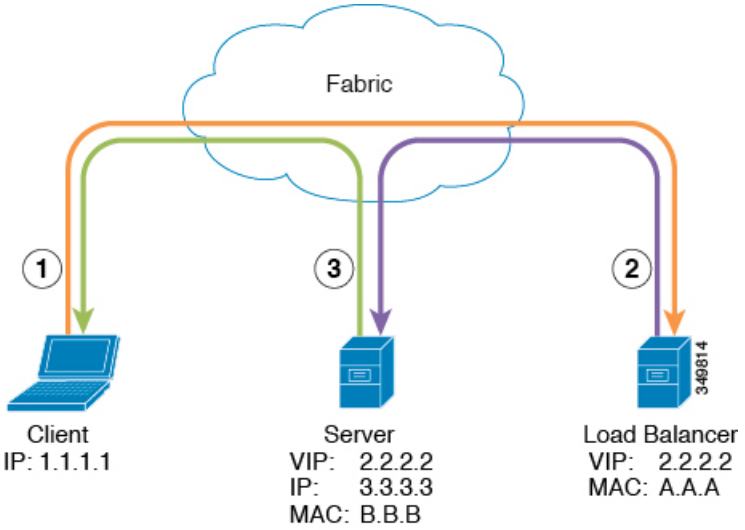
レイヤ 2 の Direct Server Return は一般的な導入または従来型の導入であり、ダイレクト ルーティング、SwitchBack、または nPath とも呼ばれます。この導入では、ロード バランサとサーバで仮想 IP アドレスが共有されます。ロード バランサとサーバはレイヤ 2 隣接である必要があります。レイヤ 2 の Direct Server Return の導入には、次の制限があります。

- サーバ配置の柔軟性が失われる
- クライアントの仮想 IP アドレス要求への Address Resolution Protocol（ARP）応答を抑制するために、追加のサーバ設定が必要になる

- ポート選択はレイヤ 3 で行われ、プロトコルに依存する。ポート選択はレイヤ 2（サーバ通信に対するロード バランサ）で行われない

レイヤ 2 の Direct Server Return の導入には、次のトラフィック フローがあります。

図 *25* : レイヤ *2* の *Direct Server Return* のトラフィック フロー



1. クライアントからロード バランサへ

Source IP Address	1.1.1.1
Destination IP Address	2.2.2.2
宛先 MAC アドレス	A.A.A

2. ロード バランサからサーバへ

Source IP Address	1.1.1.1
Destination IP Address	2.2.2.2
宛先 MAC アドレス	B.B.B

3. サーバからクライアントへ

Source IP Address	2.2.2.2
Destination IP Address	1.1.1.1
宛先 MAC アドレス	デフォルト ゲートウェイの MAC アドレス

でのレイヤ2DirectServerReturnの導入についてCiscoApplicationCentric Infrastructure

次の情報は、Cisco Application Centric Infrastructure（ACI）でのレイヤ 2 Direct Server Return の導入に当てはまります。

- 仮想 IP アドレス（2.2.2.2）は ACI ファブリック内を移動する
  - 同じ送信元仮想 IP アドレス（2.2.2.2）を持つロード バランサからサーバおよびサーバからクライアントへのトラフィック
  - サーバからクライアントへのトラフィックはルーティングされ、トラフィックはファブリック内のゲートウェイ MAC アドレス宛になる
  - サーバからの送信元 IP アドレスのデータ パスの取得はファブリック内の仮想 IP アドレスに移動する
- 異なる送信元から表示されるクライアント IP アドレス（1.1.1.1）についての問題はない
  - クライアント IP アドレスはファブリック内のクライアントとロード バランサの両方からの送信元 IP アドレスとして表示される
  - ロード バランサとサーバは、レイヤ 2 隣接であり、ロード バランサからサーバへのトラフィックはレイヤ 2 に転送される
  - ファブリック内のレイヤ 2 転送トラフィックからのデータパス IP アドレスの取得はない
  - クライアント IP アドレスがファブリック内のロード バランサからの送信元 IP アドレスとして表示された場合も、クライアント IP アドレスは取得されない

Direct Server Return の設定に関する注意事項と制約事項

Direct Server Return を展開する際には、次の注意事項と制約事項に従ってください:

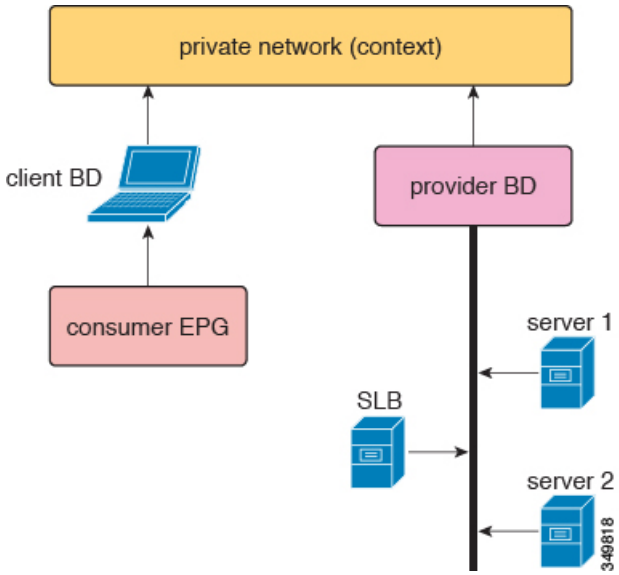
- VRF (VIP が展開される) は、「強制」モードに設定する必要があります。
- VRF は、「入力」の強制を設定する必要があります。
- 共有サービスは、この設定ではサポートされていません。
- EP 移動検出モード: GARP ベースの検出をブリッジ ドメインで有効にする必要があります。
- ユニキャスト ルーティングをブリッジ ドメインで有効にする必要があります。
- VIP がある EPG には、それに関連付けられている契約が必要です (契約はハードウェアの設定を進めます)。

- VRF 下の VZAny 契約は L4 ～ L7 VIP をプログラムしません。契約は EPG 下で引き続き許可されます。
- クライアントから VIP へのトラフィックは、必ずプロキシ スパインを通る必要があります。
- ロード バランサはワンアーム モードにする必要があります。
- サーバとロード バランサ EPG を同じデバイス上に配置するか、ロード バランサ EPG をすべてのサーバ EPG ToR に展開する必要があります。
- サーバ EPG とロード バランサ EPG は、同じブリッジ ドメインにある必要があります。

サポートされている Direct Server Return の設定

次の図に、サポートされている Direct Server Return の設定を示します。

図 *26* : サポートされている *Direct Server Return* の設定



サポートされている設定に次の情報が適用されます。

- サーバ ロード バランサとサーバは同じサブネットとブリッジ ドメインにある
- サーバ ロード バランサは 1 ARM モードで動作する必要があり、サーバ ロード バランサの内部レッグと外部レッグは同じブリッジ ドメインを指している必要がある
- コンシューマ エンドポイント グループとプロバイダー エンドポイント グループは、同じプライベート ネットワークの下にある必要がある。共有サービス設定はサポートされていない

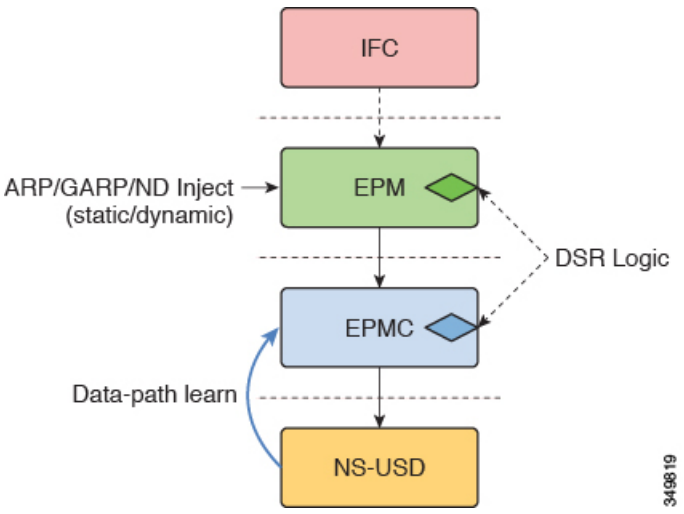
Direct Server Return のアーキテクチャ

ロードバランサとサーバが共有する仮想 IP アドレスをテナントの Virtual Routing and Forwarding（VRF）のファブリック内で静的に設定し、レイヤ 2 Direct Server Return を有効にすることができます。仮想 IP アドレスを静的にすることで、仮想 IP アドレスのデータ パスの取得が阻止されます。スイッチ側のエンドポイント マネージャは、具象モデルの {VRF、VIP、S クラス} タプル形式のポリシー エンジンからの静的設定を処理します。エンドポイント マネージャ（EPM）、エンドポイント マネージャ クライアント（EPMC）および転送シリコンにアーキテクチャは変更されていませんが、一方でこれらのレイヤはすべて変更されており、仮想 IP アドレスの静的設定が許可・維持されています。このアーキテクチャでは、仮想 IPv4 と仮想 IPv6 の両方に対して静的設定が可能です。アドレス解析および初期セットアップ以外は、アーキテクチャ上および設計全体にわたって、仮想 IPv4 と仮想 IPv6 の両方が同じコードパスで処理されます。

Direct Server Return の設計フローと設定フローは EPM/EPMC/USD のフローの一部です。このためのフローは、ノースバウンドからサウスバウンド、つまり、**[policy engine]** > **[EPM]** > **[EPMC]** > **[forwarding silicon]**の場合にのみ存在します。この場合の転送シリコンは North Star です。これは、この目的に対応しているのは North Star のローカル ステーション テーブルに限られることが理由です。同じエンドポイントの作成/変更/削除フローを、追加の静的 IP アドレス エンドポイント フラグと共に使用します。

新しい IP アドレス エンドポイントのすべての取得要求は、静的仮想 IP アドレス エンドポイント チェックを受けます。取得/処理要求がすでに存在する {VRF、VIP、S クラス} タプルに対するものである場合は、Direct Server Return の前処理コードによって変更前処理が行われ、適切なフラグを使用した一般的なエンドポイント処理に戻されます。

図 *27 :* スイッチ側の処理



次のリストに、Direct Server Return の設計ポイントに関する概要を示します。

- すべての仮想 IPv4 および IPv6 の追加/変更/削除設定は、エンドポイント マネージャによって処理される

  - Direct Server Return は、プレフィクスではなく、完全な仮想 IP アドレス（/32、/128）を使用する
  - アドレス ファミリおよびアドレスの最上位レベルのセットアップ以外、コード パスは Direct Server Return 用にマージされる

- EPM と EPMC は完全な（/32 または /128）仮想 IP アドレスを North Star ローカル ステーション テーブルの送信元アドレスにインストールする
  - キー/データは、設定された 3 タプル {VRF、VIP、S クラス} 情報から取得する
  - ローカル ステーション テーブルの送信元アドレスに「静的」としてエントリが挿入される

- EPM は ARP/GARP/ND IP-MAC バインディングと MAC の取得を通じてロード バランサのエンドポイントを検出できる
- EPM と EPMC は、{VRF、VIP} タプルの疑似 North Star データパスの取得を阻止する
- EPM と EPMC では、取得したエントリの S クラスがポリシー エンジンで設定した {VRF、VIP、S クラス} タプルと一致しなければ、IP-MAC バインディング アソシエーションを許可しない。これは、ARP/GARP/ND パスとデータパス取得パスの両方に適用される
- EPM は、COOP へのロード バランサの（ARP/GARP/ND を通じた）検出伝播を代替しない
- エントリの S クラスが一致しない場合、EPM と EPMC は既存の取得済みエントリ（ARP/GARP/ND）を設定時にクリーンアップする
- 仮想 IP アドレスを設定すると、EPM と EPMC は、データパスの取得を通じて作成された同じ {VRF、VIP} タプルの既存のエントリを常にクリーンアップする
- ARP/ND/MAC エージングはこれらの変更に対応していないが、EPM と EPMC が、{VRF、VIP} タプルの設定が削除されない限り、ローカル ステーション テーブルに維持されている静的エントリは削除されない
- この機能を実装する際に、既存のエントリを削除するのではなく、同じエントリ設定をポリシー エンジンから取得する場合、ARP/GARP/ND を通じて取得した {VRF、VIP} タプルを保持するアプローチを取る。これは、既存のエントリの S クラスが設定されたエントリの S クラスと同じである場合に限る。このアプローチにより、エントリの削除が原因で発生するファブリック全体にわたる大規模な変動を回避する
- S クラスとエントリに関連するその他の情報は IP アドレス情報の一部として保持される。つまり、情報はエンドポイント レベルではなく、エンドポイントの IP アドレスレベルで保持される
- {BD、仮想 IP のプレフィックス、S クラス} タプルとポリシーエンジンで設定した {VRF、VIP、S クラス} タプル間に重複がある場合は、{VRF、仮想 IP、S クラス} タプルが優先される

静的なサービス導入のための Direct Server Return の XML POST の例

次に、Direct Server Return の静的なサービス導入の例を示します。

```
<fvAp name="dev">
      <fvAEPg name="loadbalancer">
        <fvRsDomAtt tDn="uni/phys-{{tenantName}}"/>
        <fvRsBd tnFvBDName="lab"/>
        <fvVip addr="121.0.0.{{net}}" />
        <fvRsPathAtt tDn="topology/pod-1/paths-104/pathep-[eth1/1]" encap="vlan-33" />
        <fvRsProv tnVzBrCPName="loadBalancer"/>
        <fvRsCons tnVzBrCPName="webServer"/>
      </fvAEPg>
        <fvAEPg name="webServer">
          <fvRsDomAtt tDn="uni/phys-{{tenantName}}"/>
            <fvRsBd tnFvBDName="lab"/>
          <fvRsPathAtt tDn="topology/pod-1/paths-101/pathep-[eth1/1]" encap="vlan-34"/>

            <fvRsProv tnVzBrCPName="webServer"/>
        </fvAEPg>
      <fvAEPg name="client">
        <fvRsDomAtt tDn="uni/phys-{{tenantName}}"/>
        <fvRsBd tnFvBDName="lab"/>
        <fvRsPathAtt tDn="topology/pod-1/paths-103/pathep-[eth1/4]" encap="vlan-1114"/>
        <fvRsCons tnVzBrCPName="loadBalancer"/>
      </fvAEPg>
</fvAp>
```

L4-L7 VIP の EPG が展開されているか、コントラクトの方向に関わらず L4-L7 VIP の EPG とのコントラクトを持つ EPG が展開されているすべての top-of-rack スイッチ（ToR）にダイレクト サーバ リターン設定がダウンロードされます。この例では、ダイレクト サーバ リターン仮想 IP アドレス設定が ToR ノード 101、103、104 にダウンロードされます。ノード 104 には設定された L4-L7 VIP のロードバランサ EPG があり、ノード 101 および 103 には Web サーバまたはクライアント EPG があり、ロードバランサ EPG へのコントラクトを有します。

ダウンロードされたダイレクト サーバ リターン設定を持つすべての ToR は、データパスから L4-L7 VIP アドレスを学習せず、その他の EPG から L4-L7 VIP アドレスを学習しません（ARP/GARP/ND の場合でも）。たとえば、L4 - L7 VIP アドレスは、コントロール プレーンを介してロードバランサ EPG からのみ学習します。Web サーバ EPG から誤って L4 L7 VIP を学習してしまうことを防ぐのに役立ちます（たとえば web サーバで ARP を抑制することを忘れた場合）。

静的なサービス導入のための Direct Server Return

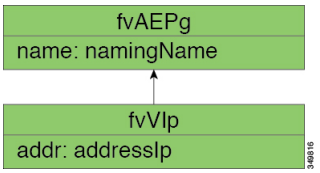
静的なサービス導入モードでは、適切なアプリケーション エンドポイント グループとコントラクトをホップごとに作成することによって、サービス フローを設定します。

静的なサービス導入の論理モデル用の Direct Server Return

アプリケーション エンドポイント グループ（fvAEPg）の下に fvVip オブジェクトを使用することによって、ロード バランサが使用する仮想 IP アドレスを設定できます。

次の図に、静的なサービス導入の論理モデルを示します。

図 *28 :* 静的なサービス導入の論理モデル



サービス グラフを挿入するための Direct Server Return

Cisco Application Centric Infrastructure（ACI）は、ベンダー パッケージとサービス グラフを使用してサービスの挿入を自動化します。このモードでは、サービスデバイスのレッグに対して作成されるエンドポイント グループ（内部および外部エンドポイント グループなど）が、オペレータによる設定を必要とせずに、ACI によって作成されます。

サービス グラフの挿入では、次の XML POST の例に示すように、サービス デバイスの適切な論理インターフェイス コンテキストの下に仮想 IP アドレスを設定する必要があります。

```
<vnsLDevCtx ctrctNameOrLbl="webCtrct"
                        graphNameOrLbl="G1"
                        nodeNameOrLbl="SLB">

    <vnsRsLDevCtxToLDev tDn="uni/tn-coke/lDevVip-InsiemeCluster"/>

    <vnsLIfCtx connNameOrLbl="inside">
        <vnsRsLIfCtxToBD tDn="uni/tn-coke/BD-cokeBD1"/>
        <vnsRsLIfCtxToLIf tDn="uni/tn-coke/lDevVip-InsiemeCluster/lIf-inside"/>
    </vnsLIfCtx>

    <vnsLIfCtx connNameOrLbl="outside">
        <vnsRsLIfCtxToBD tDn="uni/tn-coke/BD-cokeBD1"/>
        <vnsRsLIfCtxToLIf tDn="uni/tn-coke/lDevVip-InsiemeCluster/lIf-outside"/>
        <vnsSvcVip addr="9.9.9.9" />
        <vnsSvcVip addr="11.11.11.11" />
    </vnsLIfCtx>
</vnsLDevCtx>
```

この要求の例では、2 つの仮想 IP アドレス（9.9.9.9 と 11.11.11.11）をサーバ ロード バランサの外部レッグ上に設定します。仮想 IP アドレスの定義は、静的な Direct Server Return 設定と同様に、エンドポイント グループの下ではなく、LIfCtx の下になります。これは、静的サービスの導入の場合とは異なり、サービス グラフの場合は、オペレータにデバイス レッグのエンドポイント グループへの直接アクセス権がないためです。

Direct Server Return 共有レイヤ 4 ～ レイヤ 7 サービスの設定

サービスデバイスを共通のテナントまたは管理テナントに設定した場合、暗黙モデルには若干の違いがあります。vnsEPpInfo の代わりに、サービス仮想 IP アドレスの更新管理対象オブジェクトが vnsREPpInfo の子として作成されます。1 つの vnsSvcEpgCont の管理対象オブジェクトが vnsRsEPpInfo ごとに作成されて複数のテナント間で共有 SvcVip を追跡します。

Direct Server Return 用の Citrix サーバ ロード バランサの設定

次に、Direct Server Return 用に Citrix サーバ ロード バランサを設定する方法の概要を示した手順を説明します。

ステップ 1 バックエンド サーバがパケットを受け入れるようにバックエンド サーバのループバックに仮想 IP アドレスを設定します。

ステップ 2 バックエンド サーバの仮想 IP アドレスに対する Address Resolution Protocol（ARP）応答を無効にします。

ステップ 3 必要に応じて、ロードバランシング仮想サーバにバインドされたサービスのプロキシ ポートを無効にします。プロキシ ポートはデフォルトで無効になっています。

ステップ 4 ロード バランシング仮想サーバの m パラメータを「MAC」に設定します。

ステップ 5 グローバルか、またはサービスごとに USIP モードを有効にします。

ステップ 6 「L3」モード、「USNIP」モード、および「MBF」モードを有効にします。

ステップ 7 バックエンド サーバのルートを直接インターネットに到達できるように設定します。

Direct Server Return 用の Linux サーバの設定

次に、Direct Server Return 用に Linux サーバを設定する方法の概要を示した手順を説明します。

ステップ 1 次のコンテンツを使用し、Centos 内に /etc/sysconfig/network-scripts/ifcfg-lo ファイルを作成して、ループバック インターフェイス上に仮想 IP アドレスを設定します。

```
DEVICE=lo:1
IPADDRESS=10.10.10.99
NETMASK=255.255.255.255
NETWORK=10.10.10.99
BROADCAST=10.10.10.99
ONBOOT=yes
NAME=loopback
```

この例では、10.10.10.99 が仮想 IP アドレスです。

ステップ 2 クライアント要求への応答に使用するサーバ インターフェイスの arp_ignore と arp_announce の値を設定します。

```
echo 1 > /proc/sys/net/ipv4/conf/eth1/arp_ignore
echo 2 > /proc/sys/net/ipv4/conf/eth1/arp_announce
```

この例では、`eth1` がクライアント要求への応答に使用するサーバ インターフェイスです。

ARP の設定の詳細については、次の Linux 仮想サーバの Wiki ページを参照してください。

http://kb.linuxvirtualserver.org/wiki/Using_arp_announce/arp_ignore_to_disable_ARP

第 12 章

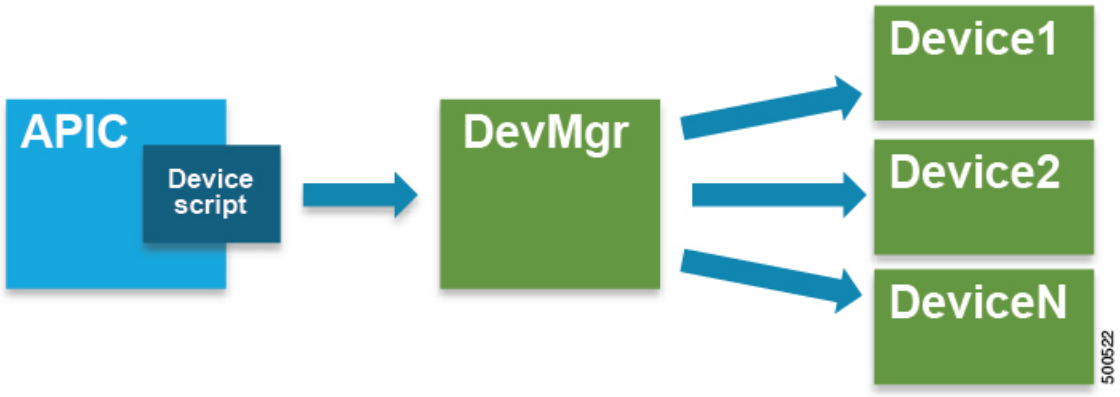
デバイスおよびシャーシ マネージャの設定

デバイス マネージャとシャーシ マネージャについて

デバイス マネージャのみで、Cisco Application Centric Infrastructure（ACI）ファブリック内の一連のクラスタを設定できます。管理状態または動作状態はデバイスのネイティブの GUI に表示されます。デバイス マネージャが個々のデバイスの設定を処理するため、Application Policy Infrastructure Controller（APIC）での設定をシンプル化できます。デバイス マネージャにテンプレートを作成してから、APIC のインスタンス固有の値をデバイス マネージャに入力しますが、必要な値はごくわずかです。

次の図に、クラスタ内で複数のデバイスを制御するデバイス マネージャを示します。

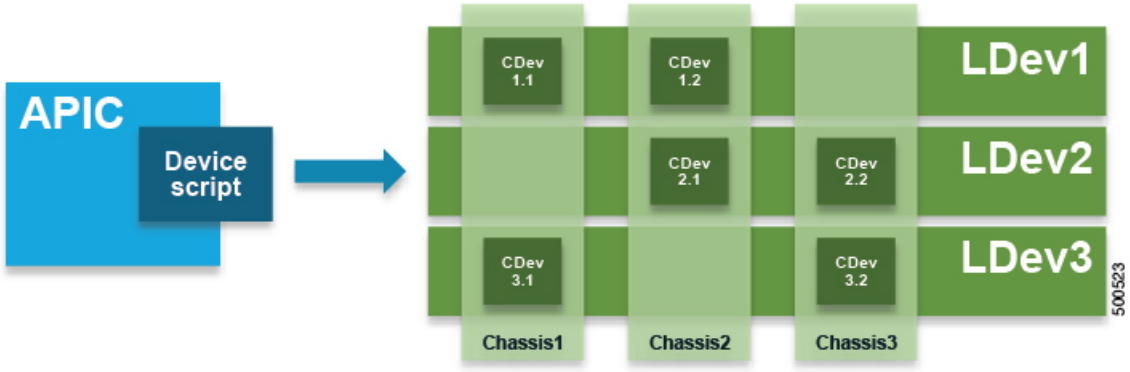
図 *29* : デバイス マネージャでのデバイスの制御



シャーシ マネージャは、処理リソースの物理または仮想「コンテナ」です。シャーシ マネージャは CDev オブジェクトとして表される、いくつかの仮想サービス デバイスをサポートします。シャーシマネージャがネットワーキングを処理し、CDevがプロセスを処理します。シャーシマネージャによって、仮想処理ノードのオンデマンド作成が可能になります。仮想デバイスでは、サービス（特にVLAN）の一部を、仮想マシンではなく、シャーシに適用する必要があります。これを実現するには、シャーシ管理 IP アドレスとクレデンシャルをコールアウトに含める必要があります。

次の図に、処理リソースのコンテナとして機能するシャーシ マネージャを示します。

図 *30* : デバイス マネージャでのデバイスの制御

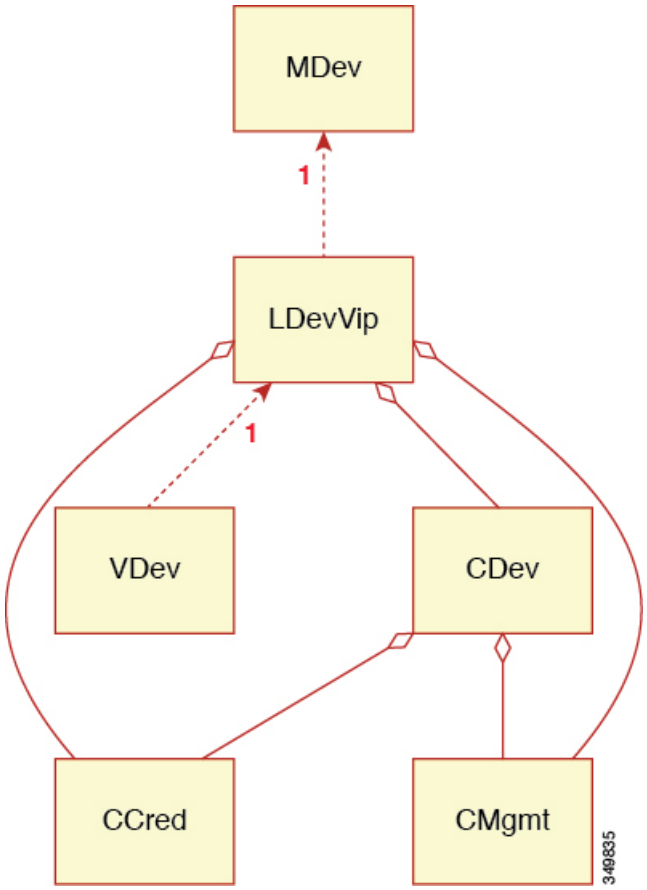


デバイス マネージャまたはシャーシ マネージャを使用せず、サービス デバイスのモデルに次の主要な管理対象オブジェクトを含めます。

- MDev：デバイス タイプ（ベンダー、モデル、バージョン）を表します。
- LDevVIP：クラスタ、つまりCold Standbyを実現するために同一に設定された一連のデバイスを表します。デバイスにアクセスするための CMgmt と CCred が含まれます。
- CDev：物理または仮想のいずれかのクラスタのメンバーを表します。デバイスにアクセスするための CMgmt と CCred が含まれます。
- VDev：サーバ上の仮想マシンと同様のクラスタのコンテキストを表します。

次の図に、CMgmt（管理接続）と CCred（クレデンシャル）が含まれた、主要な管理対象オブジェクトのモデルを示します。

図 *31* : デバイス マネージャまたはシャーシ マネージャを含まない管理対象オブジェクト モデル



CMgmt（ホスト＋ポート）と CCred（ユーザ名＋パスワード）により、スクリプトでデバイスとクラスタにアクセスできます。

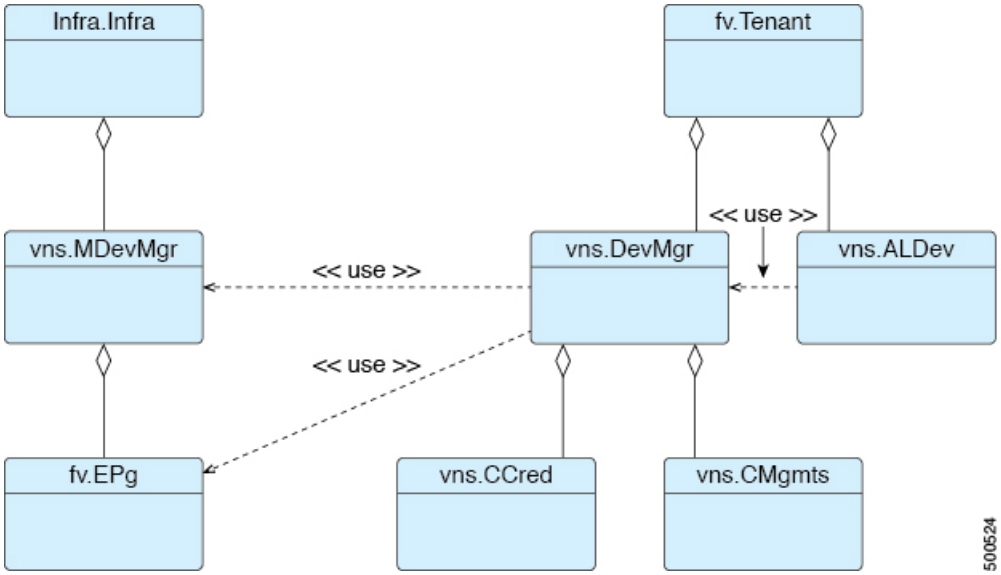
デバイス マネージャとシャーシ マネージャは、集中管理ステーションからのクラスタとデバイスの設定を制御できるようにします。シャーシは並列階層を MDev オブジェクトと ALDev オブジェクトに追加し、特定のシャーシに属しているというタグを CDev オブジェクトに付けることができます。次の管理対象オブジェクトがモデルに追加され、デバイスおよびシャーシマネージャの概念をサポートします。

- MDevMgr：デバイス マネージャのタイプを表します。MDevMgr は、同じベンダーの通常は異なる製品である一連の異なる MDev を管理できます。
- DevMgr：デバイス マネージャを表します。マネージャにアクセスするには、含まれている CMgmt と CCred の管理対象オブジェクトを使用します。各クラスタは 1 つの DevMgr のみと関連付けることができます。
- MChassis：シャーシのタイプを表します。通常、この管理対象デバイスはパッケージに含まれています。

- `Chassis`：シャーシ インスタンスを表します。これには、`CMgmt` と `CCred`[Secret] の管理対象オブジェクトが含まれており、シャーシへの接続を提供します。

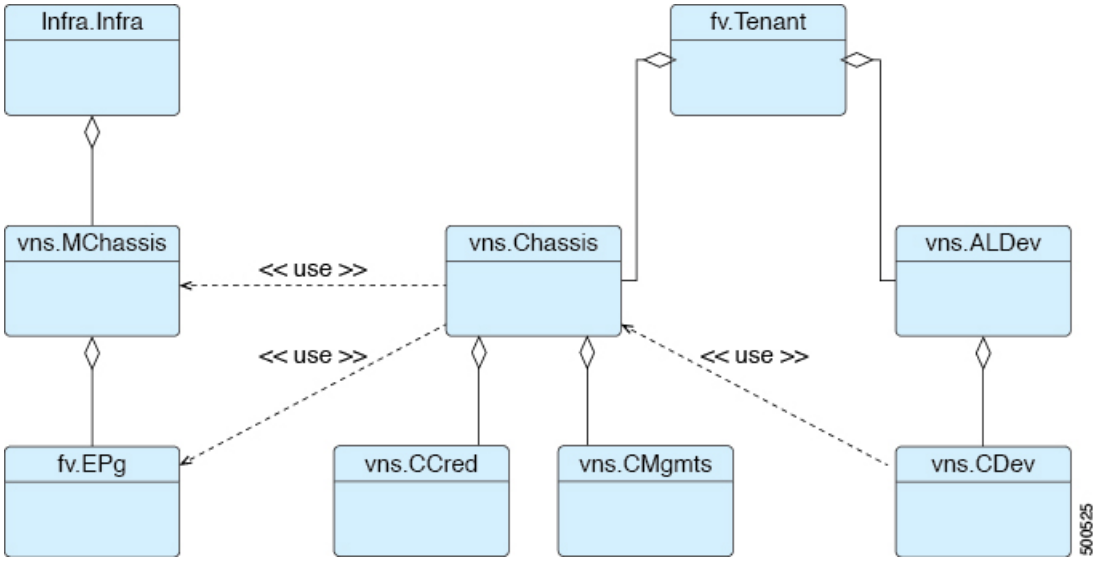
次の図に、デバイス マネージャのオブジェクト モデルを示します。

図 *32*：デバイス マネージャのオブジェクト モデル



次の図に、シャーシ マネージャのオブジェクト モデルを示します。

図 *33*：シャーシ マネージャのオブジェクト モデル



デバイス マネージャとシャーシ マネージャの動作

デバイス マネージャとシャーシ マネージャに関し、次の動作が適用されます。

- DevMgr オブジェクトは必要ありません。LDevVip から DevMgr への関係がない場合、システムはデバイス マネージャを定義せずにコールアウトを実行します。
- Policymgr が健全性チェックを実行し、LDevVip から MDev への関係が LDevVip から DevMgr と MDevMgr を経由して MDev までの 1 つのリレーション パスに一致することを保証します。これに当てはまらない場合はエラーが発生し、それ以降のコールアウトが阻止されます。
- LDevVip から DevMgr、DevMgr から MDevMgr、または MDevMgr から正しい MDev への関係が追加または変更された場合は、クラスタがリセットされ、最初から設定されます。
- Chassis オブジェクトは必要ありません。CDev から Chassis への関係がない場合、システムはシャーシを定義せずにコールアウトを実行します。
- Policymgr が健全性チェックを実行し、CDev から LDevVip を経由して MDev までの関係が、CDev から Chassis と MChassis を経由して MDev までの 1 つのリレーション パスに一致することを保証します。これに当てはまらない場合はエラーが発生し、それ以降のコールアウトが阻止されます。
- CDev から Chassis、Chassis から MChassis、または MChassis から正しい MDev への関係が追加または変更された場合は、クラスタがリセットされ、最初から設定されます。

GUI を使用したデバイス マネージャの作成

GUI を使用して、テナントにデバイス マネージャを作成することができます。

ステップ 1 メニュー バーで、**[Tenants]** > **[All Tenants]** の順に選択します。
ステップ 2 [Work] ペインで、テナントの名前をダブルクリックします。
ステップ 3 ナビゲーション ウィンドウで、 **Tenant *tenant_name*** > **Services** > **L4-L7** > **Device Managers** を選択します。
ステップ 4 [Work] ペインで、**[Actions]** > **[Create Device Manager]** の順に選択します。
ステップ 5 [Create Device Manager] ダイアログボックスで、必要に応じてフィールドに入力します。
ステップ 6 [Submit] をクリックします。

GUI を使用したシャーシの作成

GUI を使用して、テナントにシャーシを作成することができます。

ステップ 1 メニュー バーで、**[Tenants]** > **[All Tenants]** の順に選択します。
ステップ 2 [Work] ペインで、テナントの名前をダブルクリックします。
ステップ 3 ナビゲーション ウィンドウで、**Tenant** *tenant_name* > **Services** > **L4-L7** > **Chassis** を選択します。
ステップ 4 [Work] ペインで、**[Actions]** > **[Create Chassis]** の順に選択します。
ステップ 5 [Create Chassis] ダイアログボックスで、必要に応じてフィールドに入力します。
ステップ 6 [Submit] をクリックします。

デバイス マネージャとシャーシ マネージャの XML の例

以降の項の XML の例は、`Insieme` パッケージがロードされており、`uni/infra/mDev-Insieme-Generic-1.0` 識別名が提供されていることを前提としています。

MDevMgr オブジェクトを作成する XML の例

`MDevMgr` オブジェクトは `MDev` オブジェクトと類似しており、naming プロパティとして vendor、model、および version があります。マネージャでさまざまなタイプのクラスタを管理できる場合は、複数の `MDevMgrToMDev` の関係を作成できます。次に、`MDevMgr` オブジェクトを作成する XML の例を示します。

```
<polUni>
  <infraInfra>
    <vnsMDevMgr
        vendor="Insieme"
        model="DevMgr"
        version="1.0"
        >
      <vnsRsMDevMgrToMDev tDn="uni/infra/mDev-Insieme-Generic-1.0"/>
    </vnsMDevMgr>
  </infraInfra>
</polUni>
```

次に、テナント `tenant1` 内に `MDevMgr` オブジェクトを作成する XML の例を示します。

```
<polUni>
    <fvTenant name="tenant1">
      <vnsDevMgr name="Foo">
        <vnsCMgmts
            host="10.10.11.11"
            port="1234"/>
        <vnsCMgmts
            host="10.10.11.12"
            port="1234"/>
        <vnsCMgmts
            host="10.10.11.13"
            port="1234"/>
        <vnsCCred name="username" value="admin"/>
        <vnsCCredSecret name="password" value="letmein"/>
        <vnsRsDevMgrToMDevMgr tDn="uni/infra/mDevMgr-Insieme-DevMgr-1.0"/>
    </vnsDevMgr>
```

```
  </fvTenant>
</polUni>
```

LDevVip オブジェクトを DevMgr オブジェクトと関連付ける XML の例

LDevVip オブジェクトと DevMgr オブジェクトは、次の XML の例に示すように、LDevVip から DevMgr への関係を作成することによって関連付けます。

```
<polUni>
    <fvTenant name="tenant1">
        <vnsLDevVip name="InsiemeCluster" devtype="VIRTUAL">
            ...
            <vnsRsMDevAtt tDn="uni/infra/mDev-Insieme-Generic-1.0"/>
            <vnsRsALDevToDevMgr tDn="uni/tn-tenant1/devMgr-Foo"/>
            ...
        </vnsLDevVip>
    </fvTenant>
</polUni>
```

MChassis オブジェクトを作成する XML の例

MChassis オブジェクトは MDev オブジェクトと類似しており、naming プロパティとして vendor、model、および version があります。MChassisToMDev の関係によってデバイス タイプが決まります。次に、MChassis オブジェクトを作成する XML の例を示します。

```
<polUni>
  <infraInfra>
    <vnsMChassis vendor="Insieme" model="DevMgr" version="1.0">
      <vnsRsMChassisToMDev tDn="uni/infra/mDev-Insieme-Generic-1.0"/>
    </vnsMChassis>
  </infraInfra>
</polUni>
```

シャーシ オブジェクトを作成する XML の例

次に、Chassis オブジェクトを作成する XML の例を示します。

```
<polUni>
  <fvTenant name="tenant1">
    <vnsChassis name="Foo">
        <vnsCMgmts
            host="10.10.11.11"
            port="1234"/>
        <vnsCMgmts
            host="10.10.11.12"
            port="1234"/>
        <vnsCMgmts
            host="10.10.11.13"
            port="1234"/>
      <vnsCCred name="username" value="admin"/>
      <vnsCCredSecret name="password" value="letmein"/>
      <vnsRsChassisToMChassis tDn="uni/infra/mChassis-Insieme-DevMgr-1.0"/>
    </vnsChassis>
  </fvTenant>
</polUni>
```

CDev オブジェクトをシャーシ オブジェクトと関連付ける XML の例

CDev オブジェクトと Chassis オブジェクトは、次の XML の例に示すように、CDev から Chassis への関係を作成することによって関連付けます。

```
<polUni>
    <fvTenant name="tenant1">
        <vnsLDevVip name="InsiemeCluster" devtype="VIRTUAL">
           ...
           <vnsCDev name="Generic1" devCtxLbl="C1">
              ...
              <vnsRsCDevToChassis tnVnsChassisName="Foo"/>
            </vnsCDev>
            <vnsRsALDevToDevMgr tDn="uni/tn-coke/devMgr-Foo"/>
           ...
        </vnsLDevVip>
    </fvTenant>
</polUni>
```

デバイスとシャーシのコールアウト

ここでは、デバイスおよびシャーシマネージャのパラメータを含むデバイス、クラスタ、およびサービスのコールアウトの例を示します。パラメータはすべてのコールアウトに追加されます。

デバイスの deviceValidate コールアウトの例

次の deviceValidate コールアウトの例に、デバイス固有のコードを太字で示します。

```
2014-10-03 17:38:51,035 DEBUG 140230105585408 [42.42.42.101, 0]:  deviceValidate
{'args': ('1.0',),
 'device': {'creds': {'password': '<hidden>', 'username': 'nsroot'},
            'host': '42.42.42.101',
            'manager': {'creds': {'username': 'admin', 'password': '<hidden>'},
                        'hosts': {'10.10.11.11': {'port': 1234},
                                  '10.10.11.12': {'port': 1234},
                                  '10.10.11.13': {'port': 1234}},
                        'name': 'Foo'},
            'port': 80,
            'version': '1.0',
            'virtual': True}}
```

デバイスの deviceAudit コールアウトの例

次の deviceAudit コールアウトの例に、デバイス固有のコードを太字で示します。

```
2014-10-03 17:38:56,072 DEBUG 140230088800000 [42.42.42.100, 2]:  deviceAudit
{'args': ({(11, '', 'ext'): {'label': 'in', 'state': 0},
    (11, '', 'int'): {'label': 'out', 'state': 0}},
    {(4, 'oneFolder', 'foo'): {'ackedState': 0,
        'state': 0,
        'transaction': 0,
        'value': {(5, 'oneParam', 'foo'): {'ackedState': 0,
            'state': 0,
            'transaction': 0,
```

```
            'value': 'bar'}}}}),
'device': {'creds': {'password': '<hidden>', 'username': 'nsroot'},
    'host': '42.42.42.100',
    'manager': {'creds': {'username': 'admin', 'password': '<hidden>'},
                         'hosts': {'10.10.11.11': {'port': 1234},
                                   '10.10.11.12': {'port': 1234},
                                   '10.10.11.13': {'port': 1234}},
                         'name': 'Foo'},
    'port': 80,
    'version': '1.0',
    'virtual': True}}
```

デバイスの clusterAudit コールアウトの例

次の clusterAudit コールアウトの例に、デバイス固有のコードを太字で示します。

```
2014-10-03 17:39:01,097 DEBUG 140229734295296 [42.42.42.99, 4]:  clusterAudit
{'args': ({(12, '', 'ext'): {'cifs': {'Generic1': 'ext', 'Generic2': 'ext'},
                             'label': 'in',
                             'state': 0},
           (12, '', 'inside'): {'cifs': {'Generic1': 'ext',
                                         'Generic2': 'ext'},
                                'label': 'in',
                                'state': 0},
           (12, '', 'int'): {'cifs': {'Generic1': 'int', 'Generic2': 'int'},
                             'label': 'out',
                             'state': 0},
           (12, '', 'outside'): {'cifs': {'Generic1': 'int',
                                          'Generic2': 'int'},
                                 'label': 'out',
                                 'state': 0}},
          {}),
 'device': {'creds': {'password': '<hidden>', 'username': 'nsroot'},
            'devs': {'Generic1': {'creds': {'password': '<hidden>',
                                            'username': 'nsroot'},
                                  'host': '42.42.42.100',
                                  'port': 80,
                                  'virtual': True},
                     'Generic2': {'creds': {'password': '<hidden>',
                                            'username': 'nsroot'},
                                  'host': '42.42.42.101',
                                  'port': 80,
                                  'virtual': True}},
            'host': '42.42.42.99',
            'manager': {'creds': {'username': 'admin', 'password': '<hidden>'},
                                 'hosts': {'10.10.10.11': {'port': 1234},
                                           '10.10.10.12': {'port': 1234},
                                           '10.10.10.13': {'port': 1234}},
                        'name': 'Foo'},
            'port': 80,
            'version': '1.0',
            'virtual': True}}
```

デバイスの serviceAudit コールアウトの例

次の serviceAudit コールアウトの例に、デバイス固有のコードを太字で示します。

```
2014-10-03 17:39:06,169 DEBUG 140229725902592 [42.42.42.99, 5]:  serviceAudit
{'args': ({(0, '', 4474): {'ackedState': 0,
                           'state': 2,
```

```
                                'transaction': 0,
                                'txid': 10000,
                                'value': {(1, '', 5787): {
                                    ...
                                }},
 'device': {'creds': {'password': '<hidden>', 'username': 'nsroot'},
            'devs': {'Generic1': {'creds': {'password': '<hidden>',
                                            'username': 'nsroot'},
                                  'host': '42.42.42.100',
                                  'port': 80,
                                  'virtual': True},
                     'Generic2': {'creds': {'password': '<hidden>',
                                            'username': 'nsroot'},
                                  'host': '42.42.42.101',
                                  'port': 80,
                                  'virtual': True}},
            'host': '42.42.42.99',
            'manager': {'creds': {'username': 'admin', 'password': '<hidden>'},
                                'hosts': {'10.10.11.11': {'port': 1234},
                                          '10.10.11.12': {'port': 1234},
                                          '10.10.11.13': {'port': 1234}},
                                'name': 'Foo'},
            'port': 80,
            'version': '1.0',
            'virtual': True}}
```

シャーシの deviceValidate コールアウトの例

次の deviceValidate コールアウトの例に、シャーシ固有のコードを太字で示します。

```
2014-11-13 19:33:16,066 DEBUG 140719921972992 [42.42.42.101, 0]: request: deviceValidate

 {'args': ('1.0',),
  'device': {'chassis': {'creds': {'username': 'admin', 'password': '<hidden>'},
                         'hosts': {'10.10.11.11': {'port': 1234},
                                   '10.10.11.12': {'port': 1234},
                                   '10.10.11.13': {'port': 1234}},
                         'name': 'Foo'},
             'creds': {'username': 'nsroot', 'pasword': '<hidden>'},
             'host': '42.42.42.100',
             'port': 80,
             'virtual': True}}
```

シャーシの deviceAudit コールアウトの例

次の deviceAudit コールアウトの例に、シャーシ固有のコードを太字で示します。

```
2014-10-03 17:38:56,072 DEBUG 140230088800000 [42.42.42.100, 2]:  deviceAudit
 {'args': ({(11, '', 'ext'): {'label': 'in', 'state': 0},
            (11, '', 'int'): {'label': 'out', 'state': 0}},
           {(4, 'oneFolder', 'one'): {'ackedState': 0,
                                      'state': 0,
                                      'transaction': 0,
                                      'value': {(5, 'oneParam', 'one'): {'ackedState': 0,
                                                                         'state': 0,
                                                                         'transaction': 0,
                                                                         'value': 'foo'}}}}),
  'device': {'chassis': {'creds': {'username': 'admin', 'password': '<hidden>'},
                         'hosts': {'10.10.11.11': {'port': 1234},
                                   '10.10.11.12': {'port': 1234},
```

```
                                    '10.10.11.13': {'port': 1234}},
                         'name': 'Foo'},
              'creds': {'password': '<hidden>', 'username': 'nsroot'},
              'host': '42.42.42.101',
              'manager': {'creds': {'username': 'admin', 'password': '<hidden>'},
                         'hosts': {'10.10.10.11': {'port': 1234},
                                   '10.10.10.12': {'port': 1234},
                                   '10.10.10.13': {'port': 1234}},
                         'name': 'Foo'},
              'port': 80,
              'version': '1.0',
              'virtual': True}}
```

シャーシの clusterAudit コールアウトの例

次の clusterAudit コールアウトの例に、シャーシ固有のコードを太字で示します。

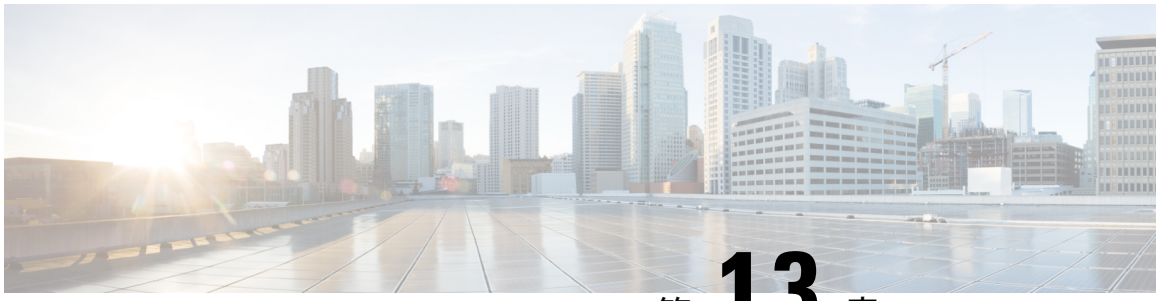
```
2014-10-03 17:39:01,097 DEBUG 140229734295296 [42.42.42.99, 4]:  clusterAudit
 {'args': ({(12, '', 'ext'): {'cifs': {'Generic1': 'ext', 'Generic2': 'ext'},
                              'label': 'in',
                              'state': 0},
            (12, '', 'inside'): {'cifs': {'Generic1': 'ext',
                                          'Generic2': 'ext'},
                                 'label': 'in',
                                 'state': 0},
            (12, '', 'int'): {'cifs': {'Generic1': 'int', 'Generic2': 'int'},
                              'label': 'out',
                              'state': 0},
            (12, '', 'outside'): {'cifs': {'Generic1': 'int',
                                           'Generic2': 'int'},
                                  'label': 'out',
                                  'state': 0}},
           {}),
  'device': {'creds': {'password': '<hidden>', 'username': 'nsroot'},
             'devs': {'Generic1': {'chassis': {'creds': {'password': '<hidden>',
                                                         'username': 'admin'},
                                               'hosts': {'10.10.11.11': {'port': 1234},
                                                         '10.10.11.12': {'port': 1234},
                                                         '10.10.11.13': {'port': 1234}},

                                               'name': 'Foo'},
                                   'creds': {'username': 'nsroot', 'pasword': '<hidden>'},

                                   'host': '42.42.42.100',
                                   'port': 80,
                                   'virtual': True},
                      'Generic2': {'chassis': {'creds': {'password': '<hidden>',
                                                         'username': 'admin'},
                                               'hosts': {'10.10.11.11': {'port': 1234},
                                                         '10.10.11.12': {'port': 1234},
                                                         '10.10.11.13': {'port': 1234}},

                                               'name': 'Foo'},
                                   'creds': {'username': 'nsroot', 'pasword': '<hidden>'},

                                   'host': '42.42.42.101',
                                   'port': 80,
                                   'virtual': True}},
              'host': '42.42.42.99',
              'manager': {'creds': {'password': '<hidden>', 'username': 'admin'},
                         'hosts': {'10.10.11.11': {'port': 1234},
                                   '10.10.11.12': {'port': 1234},
```

```
                                    '10.10.11.13': {'port': 1234}},
                          'name': 'Foo'},
               'port': 80,
               'version': '1.0',
               'virtual': True}}
```

シャーシの serviceAudit コールアウトの例

次の serviceAudit コールアウトの例に、シャーシ固有のコードを太字で示します。

```
2014-10-03 17:39:06,169 DEBUG 140229725902592 [42.42.42.99, 5]:  serviceAudit
 {'args': ...,
  'device': {'creds': {'password': '<hidden>', 'username': 'nsroot'},
             'devs': {'Generic1': {'chassis': {'creds': {'username': 'admin',
                                                         'password': '<hidden>'},
                                               'hosts': {'10.10.11.11': {'port': 1234},
                                                         '10.10.11.12': {'port': 1234},
                                                         '10.10.11.13': {'port': 1234}},
                                     'name': 'Foo'},
                                   'creds': {'username': 'nsroot',
                                             'pasword': '<hidden>'},
                                   'host': '42.42.42.100',
                                   'port': 80,
                                   'virtual': True},
                      'Generic2': {'chassis': {'creds': {'username': 'admin',
                                                         'password': '<hidden>'},
                                               'hosts': {'10.10.11.11': {'port': 1234},
                                                         '10.10.11.12': {'port': 1234},
                                                         '10.10.11.13': {'port': 1234}},
                                     'name': 'Foo'},
                                   'creds': {'username': 'nsroot',
                                             'pasword': '<hidden>'},
                                   'host': '42.42.42.101',
                                   'port': 80,
                                   'virtual': True}},
             'host': '42.42.42.99',
             'manager': {'creds': {'username': 'admin', 'password': '<hidden>'},
                         'hosts': {'10.10.11.11': {'port': 1234},
                                   '10.10.11.12': {'port': 1234},
                                   '10.10.11.13': {'port': 1234}},
                         'name': 'Foo'},
             'port': 80,
             'version': '1.0',
             'virtual': True}}
```



第 13 章

非管理対象モードの設定

非管理対象モードについて

レイヤ 4 ～ レイヤ 7 サービスの挿入機能によって、管理者は 1 つ以上のサービスを 2 つのエンドポイント グループ間に挿入できます。Application Policy Infrastructure Controller（APIC）はサービスにファブリック リソース（VLAN）を割り当て、サービス グラフに指定された設定に従ってファブリック（リーフ スイッチ）とサービス アプライアンスをプログラミングします。サービスをサービス グラフの一部として使用できるようにするには、APIC にそのサービスのデバイス パッケージが必要です。APIC はグラフのインスタンス化時にもサービス アプライアンスをプログラミングします。

APIC で、サービス グラフに対してネットワーク リソースのみを割り当てて、グラフのインスタンス化時にファブリック側のみをプログラミングすることができます。サービス アプライアンスのプログラミングにより適した既存のオーケストレータまたは dev-op ツールが環境にすでにあるなど、さまざまな理由でこれが必要になる場合があります。また、サービス アプライアンスのデバイス パッケージが使用できない場合もあります。

サービスの非管理対象モードでは、ネットワーク リソースの割り当てや、ファブリックのプログラミングに対する APIC の動作を選択できます。非管理対象モードを有効にすると、APIC はネットワーク リソースのみをサービス アプライアンスに割り当て、ファブリック（リーフ）のみをプログラミングするように制限されます。デバイスの設定については、データセンターの管理者が外部から実行します。

管理対象および非管理対象の論理デバイスについて

非管理対象モードは、次の XML コードに示すように、論理デバイス（LDevVip）に managed 設定を導入します。

```
<!-- Specified if the device is a managed device-->
<property name="managed"
         type="scalar:Bool"
         owner="management"
         mod="explicit">
         <default value="true"/>
</property>
```

デバイスは管理対象にも、非管理対象にもできます。デバイスを管理対象として設定すると、Application Policy Infrastructure Controller（APIC）はそのデバイスを管理してグラフのインスタンス化時にプログラミングします。デフォルトで、デバイスは APIC への登録時に管理対象モードに設定されます。

デバイスを非管理対象として設定した場合、つまり managed 設定を false に設定すると、APIC はデバイスをプログラミングしません。APIC は、ネットワーク リソースを割り当ててファブリック側で VLAN/VXLAN のプログラミングのみを実行します。

次の設定は、デバイス クラスタが非管理対象として設定されている場合は必要はありません。

- デバイス パッケージ
- 論理デバイス（vnsLDevViP）とデバイス（CDev）の接続情報（管理 IP アドレス、クレデンシャル、およびインバンド接続情報）
- サポートされる機能タイプ（go-through、go-to）に関する情報
- コンテキスト認識に関する情報（シングル コンテキストかマルチコンテキスト）

この場合も、APIC は論理デバイスおよびデバイスのトポロジ情報（LIF、CIF）を把握する必要があります。この情報は、APIC がリーフ上で適切なポートをプログラミングするために必要です。また、APIC はこの情報をトラブルシューティング ウィザードに使用することもあります。

さらに、APIC はカプセル化の割り当てに使用する DomP との関係も把握する必要があります。

管理対象および非管理対象の機能ノードについて

非管理対象モードは、次の XML コードに示すように、機能ノード（AbsNode）に managed 設定を導入します。

```
<!-- Specified if the function is using a managed device-->
<property name="managed"
              type="scalar:Bool"
              owner="management"
              mod="explicit">
              <default value="true"/>
</property>
```

機能ノードは管理対象にも、非管理対象にもできます。機能ノードを管理対象として設定すると、その機能ノードは管理対象デバイスを使用できます。Application Policy Infrastructure Controller（APIC）は、グラフのインスタンス化時にデバイスをプログラミングします。デフォルトでは、機能ノードをサービス グラフに追加すると、その機能ノードは管理対象モードで設定されます。

機能ノードを非管理対象として設定した場合、つまり `managed` 設定を `false` に設定すると、APIC はパラメータ解決もデバイスのプログラミングも行いません。APIC は、ネットワーク リソースを割り当ててファブリック側でVLAN/VXLANのプログラミングのみを実行します。

次の設定は、機能ノードが非管理対象として設定されている場合は必要はありません。

- `MFunc` の関係
- `AbsFuncProfile`
- 設定パラメータ（`AbsNode` またはエンドポイント グループ上）
- サポートされる機能タイプ（go-through、go-to）に関する情報

この場合も、APICは機能ノードのネットワーク情報（`LIF`、`CIF`）を把握する必要があります。この情報は、APIC がリーフ上でネットワークを適切にプログラミングするために必要です。また、APIC はこの情報をトラブルシューティング ウィザードに使用することもあります。

さらに、次の設定が必要です。

- グラフ インスタンス化時に `LDevVip` の選択を可能にする `LDevCtx`
- グラフ インスタンス化時に `LIf` の選択を可能にする `LIfCtx`
- `LIfCtx` 内のブリッジ ドメイン
- `LIfCtx` でのルート ピアリング
- `LIfCtx` 内のサブネット

レイヤ 4 ～ レイヤ 7 サービスのエンドポイント グループについて

非管理対象モード機能の一部として、Application Policy Infrastructure Controller (APIC) では、グラフのインスタンス化時にグラフ コネクタに使用するエンドポイント グループを指定できます。これにより、グラフ導入のトラブルシューティングが容易になります。APIC は、指定されたレイヤ 4 ～レイヤ 7 サービス エンドポイント グループを使用してリーフにカプセル化情報をダウンロードします。また、APIC はこのエンドポイント グループを使用して仮想デバイスの分散仮想スイッチにポート グループを作成します。さらに、レイヤ 4 ～ レイヤ 7 サービスのエンドポイント グループを使用して、グラフ コネクタのエラー情報や統計情報も集約します。

導入されたグラフ リソースへの可視性の向上に加えて、レイヤ 4 ～ レイヤ 7 サービスのエンドポイント グループも使用して、特定のグラフ インスタンスに使用する静的なカプセル化を指定することもできます。このカプセル化は、複数のグラフ インスタンス間でレイヤ 4 ～ レイヤ 7 サービスのエンドポイント グループを共有することによって、複数のグラフ インスタンス間で共有することもできます。

グラフ コネクタと共に レイヤ 4 ～ レイヤ 7 サービスのエンドポイントをどのように使用できるかを示す XML コードの例については、レイヤ 4 ～ レイヤ 7 サービスのエンドポイント グループとコネクタを関連付ける XML の例 （153 ページ）を参照してください。

グラフ コネクタに対する静的なカプセル化の使用

Application Policy Infrastructure Controller（APIC）は、処理中にさまざまなサービス グラフにカプセル化を割り当てます。一部の使用例では、サービス グラフ内の特定のコネクタに使用するカプセル化を明示的に指定できます。これは静的なカプセル化と呼ばれます。静的なカプセル化は、物理サービスを持つサービス デバイス クラスタがあるサービス グラフ コネクタについてのみサポートされます。仮想サービス デバイスがあるサービス デバイス クラスタは、そのサービス デバイス クラスタに関連付けられた VMware ドメインから動的に割り当てられた VLAN を使用します。

静的なカプセル化は、レイヤ 4 ～ レイヤ 7 サービスのエンドポイント グループの一部としてカプセル化値を指定することによってグラフ コネクタで使用できます。レイヤ 4 ～ レイヤ 7 サービスのエンドポイントで静的なカプセル化の使用方法を示すXML コードの例については、レイヤ 4 ～ レイヤ 7 サービスのエンドポイント グループで静的なカプセル化を使用する XML の例 （153 ページ）を参照してください。

NX-OS スタイルの CLI を使用した物理デバイスの作成

次に、NX-OS スタイルの CLI を使用して物理デバイスを作成する手順の例を示します。

ステップ 1 コンフィギュレーション モードを開始します。

例：

```
apic1# configure
```

ステップ 2 テナントのコンフィギュレーション モードを開始します。

```
tenant tenant_name
```

例：

```
apic1(config)# tenant t1
```

ステップ 3 クラスタを作成します。

例：

```
apic1(config-tenant)# l4l7 cluster name ifav108-asa type physical vlan-domain phyDom5 servicetype
FW
```

ステップ 4 クラスタ デバイスを追加します。

例：

```
apic1(config-cluster)# cluster-device C1
```

ステップ 5 プロバイダー クラスタ インターフェイスを追加します。

例：

```
apic1(config-cluster)# cluster-interface provider
```

ステップ 6 インターフェイスにメンバー デバイスを追加します。

例：

```
apic1(config-cluster-interface)# member device C1 device-interface Po1
apic1(config-member)# interface vpc VPCPolASA leaf 103 104
apic1(config-member)# exit
apic1(config-cluster-interface)# exit
```

ステップ 7 コンシューマ クラスタ インターフェイスを追加します。

例：

```
apic1(config-cluster)# cluster-interface consumer
```

ステップ 8 コンシューマ インターフェイスに同じメンバー デバイスを追加します。

例：

```
apic1(config-cluster-interface)# member device C1 device-interface Po1
apic1(config-member)# interface vpc VPCPolASA leaf 103 104
apic1(config-member)# exit
apic1(config-cluster-interface)# exit
```

ステップ 9 クラスタ作成モードを終了します。

例：

```
apic1(config-cluster)# exit
```

NX-OS スタイルの CLI を使用したハイ アベイラビリティ クラスタの作成

次に、NX-OS スタイルの CLI を使用してハイ アベイラビリティ クラスタを作成する手順の例を示します。

ステップ 1 コンフィギュレーション モードを開始します。

例：

```
apic1# configure
```

ステップ 2 テナントのコンフィギュレーション モードを開始します。

```
tenant tenant_name
```

例：

```
apic1(config)# tenant t1
```

ステップ 3 クラスタを作成します。

例：

```
apic1(config-tenant)# l4l7 cluster name ifav108-asa type physical vlan-domain phyDom5 servicetype
FW
```

ステップ 4 クラスタ デバイスを追加します。

例：

```
apic1(config-cluster)# cluster-device C1
apic1(config-cluster)# cluster-device C2
```

ステップ 5 プロバイダー クラスタ インターフェイスを追加します。

例：

```
apic1(config-cluster)# cluster-interface provider vlan 101
```

ステップ 6 インターフェイスにメンバー デバイスを追加します。

例：

```
apic1(config-cluster-interface)# member device C1 device-interface Po1
apic1(config-member)# interface vpc VPCPolASA leaf 103 104
apic1(config-member)# exit
apic1(config-cluster-interface)# exit
apic1(config-cluster-interface)# member device C2 device-interface Po2
apic1(config-member)# interface vpc VPCPolASA-2 leaf 103 104
apic1(config-member)# exit
apic1(config-cluster-interface)# exit
```

ステップ 7 別のプロバイダー クラスタ インターフェイスを追加します。

例：

```
apic1(config-cluster)# cluster-interface provider vlan 102
```

ステップ 8 最初のインターフェイスからこの新しいインターフェイスに同じメンバー デバイスを追加します。

例：

```
apic1(config-cluster-interface)# member device C1 device-interface Po1
apic1(config-member)# interface vpc VPCPolASA leaf 103 104
apic1(config-member)# exit
apic1(config-cluster-interface)# exit
apic1(config-cluster-interface)# member device C2 device-interface Po2
apic1(config-member)# interface vpc VPCPolASA-2 leaf 103 104
apic1(config-member)# exit
apic1(config-cluster-interface)# exit
```

ステップ 9 クラスタ作成モードを終了します。

例：

```
apic1(config-cluster)# exit
```

NX-OS スタイルの CLI を使用した仮想デバイスの作成

次に、NX-OS スタイルの CLI を使用して仮想デバイスを作成する手順の例を示します。

ステップ 1 コンフィギュレーション モードを開始します。

例：

```
apic1# configure
```

ステップ 2 テナントのコンフィギュレーション モードを開始します。

```
tenant tenant_name
```

例：

```
apic1(config)# tenant t1
```

ステップ 3 クラスタを作成します。

例：

```
apic1(config-tenant)# l4l7 cluster name ifav108-citrix type virtual vlan-domain ACIVswitch servicetype
 ADC
```

ステップ 4 クラスタ デバイスを追加します。

例：

```
apic1(config-cluster)# cluster-device D1 vcenter ifav108-vcenter vm NSVPX-ESX
```

ステップ 5 コンシューマ クラスタ インターフェイスを追加します。

例：

```
apic1(config-cluster)# cluster-interface consumer
```

ステップ 6 コンシューマ インターフェイスにメンバー デバイスを追加します。

例：

```
apic1(config-cluster-interface)# member device D1 device-interface 1_1
apic1(config-member)# interface ethernet 1/45 leaf 102
ifav108-apic1(config-member)# vnic "Network adapter 2"
apic1(config-member)# exit
apic1(config-cluster-interface)# exit
```

ステップ 7 プロバイダー クラスタ インターフェイスを追加します。

例：

```
apic1(config-cluster)# cluster-interface provider
```

ステップ 8 プロバイダー インターフェイスに同じメンバー デバイスを追加します。

例：

```
apic1(config-cluster-interface)# member device D1 device-interface 1_1
apic1(config-member)# interface ethernet 1/45 leaf 102
ifav108-apic1(config-member)# vnic "Network adapter 2"
```

```
apic1(config-member)# exit
apic1(config-cluster-interface)# exit
```

ステップ 9 クラスタ作成モードを終了します。

例：

```
apic1(config-cluster)# exit
```

非管理対象モードの XML の例

以降の項の XML の例で、非管理対象モードの管理方法を示します。

非管理対象の LDevVip オブジェクトを作成する XML の例

次に、非管理対象の `LDevVip` オブジェクトを作成する XML の例を示します。

```
<polUni>
    <fvTenant name="HA_Tenant1">
        <vnsLDevVip name="ADCCluster1" devtype="VIRTUAL" managed="no">
            <vnsRsALDevToDomP tDn="uni/vmmp-VMware/dom-mininet"/>
        </vnsLDevVip>
    </fvTenant>
</polUni>
```

Cisco ACI Virtual Edge については、次の例の XML は非管理対象 `LDevVip` オブジェクト (Cisco ACI Virtual Edge VMM ドメインと、スイッチング モードとしての `ave` に関連付けられたもの) を作成します:

```
<polUni>
 <fvTenant name="HA_Tenant1">
  <vnsLDevVip name="ADCCluster1" devtype="VIRTUAL" managed="no">
   <vnsRsALDevToDomP switchingMode="AVE" tDn="uni/vmmp-VMware/dom-mininet_ave"/>
  </vnsLDevVip>
 </fvTenant>
</polUni>
```

非管理対象の AbsNode オブジェクトを作成する XML の例

次に、非管理対象の `AbsNode` オブジェクトを作成する XML の例を示します。

```
    <fvTenant name="HA_Tenant1">
        <vnsAbsGraph name="g1">
            <vnsAbsTermNodeProv name="Input1">
                <vnsAbsTermConn name="C1">
                </vnsAbsTermConn>
            </vnsAbsTermNodeProv>

            <!-- Node1 provides a service function in un-managed mode -->
            <vnsAbsNode name="Node1" managed="no">
                <vnsAbsFuncConn name="outside" >
                </vnsAbsFuncConn>
                <vnsAbsFuncConn name="inside" >
                </vnsAbsFuncConn>
```

```
            </vnsAbsNode>

            <vnsAbsTermNodeCon name="Output1">
                <vnsAbsTermConn name="C6">
                </vnsAbsTermConn>
            </vnsAbsTermNodeCon>

            <vnsAbsConnection name="CON2" >
                <vnsRsAbsConnectionConns
tDn="uni/tn-HA_Tenant1/AbsGraph-g1/AbsTermNodeCon-Output1/AbsTConn"/>
                <vnsRsAbsConnectionConns
tDn="uni/tn-HA_Tenant1/AbsGraph-g1/AbsNode-Node1/AbsFConn-outside"/>
            </vnsAbsConnection>

            <vnsAbsConnection name="CON1" >
                <vnsRsAbsConnectionConns
tDn="uni/tn-HA_Tenant1/AbsGraph-g1/AbsNode-Node1/AbsFConn-inside"/>
                <vnsRsAbsConnectionConns
tDn="uni/tn-HA_Tenant1/AbsGraph-g1/AbsTermNodeProv-Input1/AbsTConn"/>
            </vnsAbsConnection>

        </vnsAbsGraph>
    </fvTenant>
```

レイヤ 4 ～ レイヤ 7 サービスのエンドポイント グループとコネクタを関連付ける XML の例

次に、レイヤ 4 ～ レイヤ 7 サービスのエンドポイント グループとコネクタを関連付ける XML の例を示します。

```
<fvTenant name="HA_Tenant1">
    <vnsLDevCtx ctrctNameOrLbl="any" descr=""
dn="uni/tn-HA_Tenant1/ldevCtx-c-any-g-any-n-any"
      graphNameOrLbl="any" name="" nodeNameOrLbl="any">
        <vnsRsLDevCtxToLDev tDn="uni/tn-HA_Tenant1/lDevVip-ADCCluster1"/>
        <vnsLIfCtx connNameOrLbl="inside" descr="" name="inside">
            <vnsRsLIfCtxToSvcEPg tDn="uni/tn-HA_Tenant1/ap-sap/SvcEPg-EPG1"/>
            <vnsRsLIfCtxToBD tDn="uni/tn-HA_Tenant1/BD-provBD1"/>
            <vnsRsLIfCtxToLIf tDn="uni/tn-HA_Tenant1/lDevVip-ADCCluster1/lIf-inside"/>
        </vnsLIfCtx>
        <vnsLIfCtx connNameOrLbl="outside" descr="" name="outside">
            <vnsRsLIfCtxToSvcEPg tDn="uni/tn-HA_Tenant1/ap-sap/SvcEPg-EPG2"/>
            <vnsRsLIfCtxToBD tDn="uni/tn-HA_Tenant1/BD-consBD1"/>
            <vnsRsLIfCtxToLIf tDn="uni/tn-HA_Tenant1/lDevVip-ADCCluster1/lIf-outside"/>
        </vnsLIfCtx>
    </vnsLDevCtx>
</fvTenant>
```

レイヤ 4 ～ レイヤ 7 サービスのエンドポイント グループで静的なカプセル化を使用する XML の例

次に、レイヤ 4 ～ レイヤ 7 サービスのエンドポイント グループで静的なカプセル化を使用する XML の例を示します。

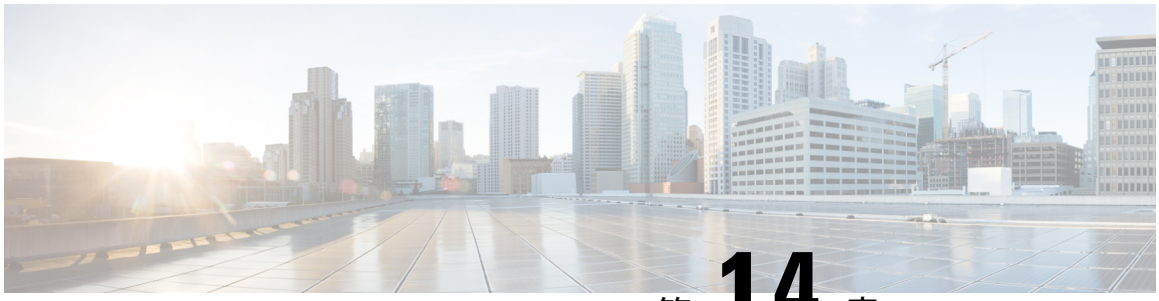
```
<polUni>
    <fvTenant name="HA_Tenant1">
        <fvAp name="sap">
```

```
            <vnsSvcEPg name="EPG1" encap="vlan-3510">
            </vnsSvcEPg>
        </fvAp>
    </fvTenant>
</polUni>
```

非管理対象モードの動作

非管理対象モードについて次の動作が適用されます。

- パラメータ解決と非管理対象機能：非管理対象機能では、Application Policy Infrastructure Controller（APIC）はパラメータ解決を実行しません。AbsGraph、エンドポイント グループ、またはその他のすべてのレベルでパラメータを設定する必要はありません。
- VDev と非管理対象機能：非管理対象機能では、APIC はパラメータ解決やデバイス側のプログラミングを実行しません。非管理対象サービス グラフ機能では、VDev ツリーは作成されません。
- 非管理対象モードでのルート ピアリング：非管理対象モードはルート ピアリング機能に影響しません。
- 非管理対象モードでの VNIC の自動配置：非管理対象モードは VNIC の配置機能に影響しません。



第 14 章

コピー サービスの設定

コピー サービスについて

すべてのトラフィックを複製する SPAN とは異なり、 Cisco Application Centric Infrastructure (ACI) のコピー サービス機能は、契約での仕様に従って、エンドポイント グループ間のトラフィックのうちコピーの部分だけを選択的に有効にします。ブロードキャスト、不明なユニキャストとマルチキャスト (BUM)、および契約の対象外であるコントロール プレーン トラフィックは、コピーされません。対照的に、SPAN は、エンドポイント グループ、アクセス ポートまたはアップリンク ポートから発するすべてのトラフィックをコピーします。SPAN とは異なり、コピーサービスは、コピーされたトラフィックにヘッダーを追加しません。コピーサービスのトラフィックは、通常のトラフィックの転送への影響を最小限に抑えるため、スイッチ内で内部的に管理されます。

コピー サービスは、コピーされるトラフィックの宛先としてコピー クラスタを指定する、レイヤ 4 ～ レイヤ 7 サービス グラフ テンプレートの一部として構成されます。コピー サービスはサービス グラフ内の異なるホップにタップすることができます。たとえば、コピー サービスは、コンシューマ エンドポイント グループとファイアウォール プロバイダ エンドポイントの間のトラフィック、またはサーバのロードバランサとファイアウォールの間のトラフィックを選択することができます。コピー クラスタは、テナント間で共有することができます。

コピー サービスを使用するには、以下のタスクを実施する必要があります:

- 送信元と宛先エンドポイント グループを特定します。
- 情報カテゴリ、および契約フィルタで許可されている内容に従って、コピー対象を指定する契約を構成します。
- ターゲット デバイスを特定するレイヤ 4 ～ レイヤ 7 のコピー デバイスを構成し、それらが接続するポートを指定します。

- コピー サービスをレイヤ 4 ～ レイヤ 7 サービス グラフ テンプレートの一部として使用します。
- どのデバイスがサービス グラフからのトラフィックを受信するかを指定する、デバイス選択ポリシーを構成します。デバイス選択ポリシーを構成する際には、契約、サービス グラフ、コピー クラスタ、およびコピー デバイス内のクラスタ論理インターフェイスを指定します。

コピー サービスの制限

コピー サービス機能を使用する場合、次の制限が適用されます:

- コピー サービスは、N9K-9300-EX と -FX リーフ スイッチでのみサポートされます。
- ローカルおよびリモートのアナライザ ポートにコピーされるデータ パス トラフィックについては、コピーされたトラフィックではサービス クラス (CoS) および差別化サービス コード ポイント (DSCP) の値が保持されません。これは、コピー アクションの契約が、実際の COS または DSCP 値の変更の前後に、入力または出力 TOR のいずれかで問題となる可能性があるからです。

  特定のエンドポイント入力方向での、データ パスのトラフィックにポリシーを適用する際、トラフィックは、実際の着信トラフィックにポリシーが適用される前にコピーされます。これは、N9K-93108TC-EX および N9K-93180YC-EX スイッチでの ASIC の制限のためです。
- コピー サービスは、コピー クラスタごとに 1 つのデバイスだけをサポートします。
- コピー クラスタは、1 つの論理インターフェイスだけをサポートします。
- コンシューマ エンドポイントまたはプロバイダー エンドポイントでのコピー アナライザは、N9K-93108TC-EX および N9K-93180YC-EX スイッチでのみ設定できます。N9K-93128TX、N9K-9396PX、または N9K-9396TX スイッチでコピー アナライザを設定すると、エラーが発生します。
- `tn-common/ctx-copy` VRF インスタンスは、コピー VRF インスタンスとも呼ばれ、コピー サービスのためのシステム予約コンテキストです。コピー VRF インスタンスは、ブートアップ シーケンス中に、システムにより自動設定されます。コピー VRF インスタンスをユーザが設定または削除することはできません。
- vzAny 契約でのコピー サービスはサポートされていません。

GUI を使用したコピー サービスの設定

この手順では、GUI を使用して、コピー サービスを設定します。

（注） コピー デバイスを設定するときは、context aware パラメータは使用されません。context aware パラメータには `single context` というデフォルト値がありますが、これは無視できます。

ステップ 1 1 つ以上の コピー デバイスを作成します。

コピー デバイスの作成についての詳細は、GUI を使用したコピーデバイスの作成 （157 ページ）を参照してください。

ステップ 2 コピー サービスで使用するサービス グラフ テンプレートを作成します。

サービス グラフ テンプレートの作成についての詳細は、GUI を使用したレイヤ 4 ～ レイヤ 7 サービス グラフ テンプレートの作成 （241 ページ）を参照してください。

a) 1 つ以上のサービス ノードを作成する場合は、**Device Clusters** セクションから、レイヤ 4 ～ レイヤ 7 サービス デバイスを、コンシューマ エンドポイント グループとプロバイダー エンドポイント グループの間にドラッグします。

b) **Device Clusters** セクションから、コピー デバイスを、任意の 2 つのオブジェクトの間にドラッグして 1 つ以上のコピー ノードを作成します。

コピーデバイスをドロップした場所が、コピーデバイスがトラフィックをコピーする、データ フロー内のポイントとなります。

ステップ 3 レイヤ 4 ～ レイヤ 7 サービス グラフ テンプレートを適用します。

サービス グラフ テンプレートを適用する方法の詳細については、GUI を使用したエンドポイント グループへのサービス グラフ テンプレートの適用 （243 ページ）を参照してください。

GUI を使用したコピーデバイスの作成

コピー デバイスは、copy ノードを作成するコピー サービス機能の一部として使用されます。コピー ノードは、どの時点でエンドポイント グループ間のトラフィックをコピーするかを指定します。

この手順では、コピー デバイスの作成のみを行います。コピー サービス機能を使用するために必要なその他の設定は行いません。コピー サービスの設定の詳細については、GUI を使用したコピー サービスの設定 （156 ページ）を参照してください。

始める前に

テナントを作成しておく必要があります。

ステップ 1 メニュー バーで、 **Tenants** > **All Tenants** を選択します。

ステップ 2 作業ウィンドウで、テナントの名前をダブルクリックします。

ステップ 3 [Navigation] ウィンドウで、**Tenant** *tenant_name* > **Services** > **L4-L7** > **Devices** を選択します。

ステップ 4 [Work] ウィンドウで、**Actions** > **Create Copy Devices** を選択します。

ステップ 5 **Create Copy Devices** ダイアログボックスの **General** セクションで、次のフィールドを設定します:

名前	説明
[Name] フィールド	コピーデバイスの名前を入力します。
Device Type ボタン	デバイス タイプです。コピー デバイスは、物理デバイスに限られます。
Physical Domain ドロップダウンリスト	デバイスの物理ドメインを選択します。

ステップ 6 **Device 1** セクションで、+ をクリックしてデバイス インターフェイスを追加し、以下のフィールドを設定して、 **Update** をクリックします:

名前	説明
[Name] フィールド	デバイス インターフェイスの名前を入力します。
Path ドロップダウン リスト	使用するデバイス インターフェイスのポート、ポート チャネル、または仮想ポートチャネルを選択します。コピーデバイスは、そのポート、ポート チャネルまたは仮想ポート チャネルに接続し、そこからトラフィックをコピーします。

ステップ 7 **Cluster** セクションで、+ をクリックしてクラスタ インターフェイスを追加し、以下のフィールドを設定して、**Update** をクリックします:

名前	説明
[Name] フィールド	クラスタ インターフェイスの名前を入力します。
Concrete Interfaces ドロップダウンリスト	使用するクラスタ インターフェイスの、1 つ以上の具体的なインターフェイスを選択します。

Encap フィールド	カプセル化で使用する VLAN を入力します。VLAN 名の書式は次のとおりです:
	`vlan-#`
	# は VLAN の ID です。次に例を示します:
	`vlan-12`

ステップ 8 [Submit] をクリックします。

NX-OS スタイルの CLI を使用したコピー サービスの設定

この手順では、CLI を使用してコピー サービスを設定する例を提供します。

（注） コピー デバイスを設定すると、context aware パラメータは使用されません。context aware パラメータには `single context` というデフォルト値がありますが、これは無視されます。

ステップ 1 コピー クラスタを作成します。

例：

```
l4l7 cluster name Copy_1 type physical vlan-domain phys_scale_copy service COPY function none
  cluster-device Copy_1_Device_1
  cluster-interface Tap_copy vlan 3644
    member device Copy_1_Device_1 device-interface int1
      interface ethernet 1/15 leaf 104
      exit
    member device Copy_1_Device_1 device-interface int2
      interface ethernet 1/15 leaf 105
      exit
    member device Copy_1_Device_1 device-interface int3
      interface ethernet 1/20 leaf 105
      exit
    exit
  exit
```

ステップ 2 抽象グラフとデバイスのコンテキストを作成し、グラフを適用します。

例：

```
l4l7 graph g5 contract c5
  service CP1 device-cluster-tenant t1 device-cluster Copy_1 mode OTHER service COPY
    connector copy cluster-interface Tap_copy
      exit
    exit
  connection C1 terminal consumer terminal provider copyservice CP1 connector copy
  Exit
```

ステップ 3 グラフを契約を接続します。

例：

```
contract c5
  scope tenant
  subject Subject
    access-group default both
    l4l7 graph g5
    exit
  Exit
```

ステップ 4 契約をエンドポイント グループを接続します。

例：

```
  epg epg2210
    bridge-domain member bd5
```

```
  contract consumer c5
  exit
epg epg2211
  bridge-domain member bd5
  contract provider c5
  Exit
```

例

次の例では、両側でコピー デバイスとファイアウォール サービス グラフを作成します。

```
  tenant tenant_cmd_line
    l4l7 graph graph_fire contract fire
     service Fire device-cluster-tenant tenant_cmd_line device-cluster Fire mode FW_ROUTED

        connector consumer cluster-interface Outside_cmdline
          bridge-domain tenant tenant_cmd_line name Consumer_BD_1
          exit
        connector provider cluster-interface Inside_cmdline
          bridge-domain tenant tenant_cmd_line name Provider_BD1
          exit
        exit
      service CP2 device-cluster-tenant tenant_cmd_line device-cluster copy1 mode OTHER
       service COPY
        connector copy cluster-interface int1
          exit
        exit
      service CP3 device-cluster-tenant tenant_cmd_line device-cluster copy1 mode OTHER
       service COPY
        connector copy cluster-interface int1
          exit
        exit
      connection C1 terminal consumer service Fire connector consumer copyservice CP2
       connector copy
      connection C2 terminal provider service Fire connector provider copyservice CP3
       connector copy
      exit
    Exit
```

次の例では、すべてのリンクで接続されているコピーデバイスでワンアームモードでファイアウォールとロード バランスを作成します。

```
    l4l7 graph Graph_LB_Firewall contract c1_firewall
      service Fire device-cluster-tenant Tenant_Firewall_LB device-cluster Firewall_1
mode
       FW_ROUTED
        connector consumer cluster-interface Outside_Firewall
          bridge-domain tenant Tenant_Firewall_LB name BD1_Consumer
          exit
        connector provider cluster-interface Inside_Firewall
          bridge-domain tenant Tenant_Firewall_LB name BD2_Provider
          exit
        exit
      service LB device-cluster-tenant Tenant_Firewall_LB device-cluster LB_1 mode
ADC_ONE_ARM
        connector consumer cluster-interface LB_Inside
          bridge-domain tenant Tenant_Firewall_LB name BD2_Provider
          exit
        connector provider cluster-interface LB_Inside
```

```
      bridge-domain tenant Tenant_Firewall_LB name BD2_Provider
      exit
    Exit
 service CP6 device-cluster-tenant Tenant_Pass2 device-cluster Copy_pass2 mode OTHER

   service-type COPY
    connector copy cluster-interface tap_copy
      exit
    Exit
 service CP7 device-cluster-tenant Tenant_Pass2 device-cluster Copy_pass2 mode OTHER

   service-type COPY
    connector copy cluster-interface tap_copy
      exit
    Exit
 service CP8 device-cluster-tenant Tenant_Pass2 device-cluster Copy_pass2 mode OTHER

   service-type COPY
    connector copy cluster-interface tap_copy
      exit
    exit
  connection C1 terminal consumer service Fire connector consumer copyservice CP6
   connector copy
 connection C2 intra-service service1 Fire connector1 provider service2 LB connector2

   consumer copyservice CP7 connector copy
  connection C3 terminal provider service LB connector provider copyservice CP8
   connector copy
  exit
exit
```

REST API を使用してコピー サービスの設定

コピー デバイスは、copy ノードを作成するコピー サービス機能の一部として使用されます。コピーのノードは、トラフィックをコピーするエンドポイント グループ間のデータ フローのどの時点を指定します。

この手順では、REST API を使用してコピー サービスを設定する例を提供します。

（注） コピー デバイスを設定すると、context aware パラメータは使用されません。context aware パラメータには `single context` というデフォルト値がありますが、これは無視されます。

始める前に

テナントを作成しておく必要があります。

ステップ 1 コピー デバイスを作成します。

例：

```
<vnsLDevVip contextAware="single-Context" devtype="PHYSICAL" funcType="None" isCopy="yes" managed="no"

  mode="legacy-Mode" name="copy0" packageModel="" svcType="COPY" trunking="no">
```

```
      <vnsRsALDevToPhysDomP tDn="uni/phys-phys_scale_copy"/>
      <vnsCDev devCtxLbl="" name="copy_Dyn_Device_0" vcenterName="" vmName="">
        <vnsCIf name="int1" vnicName="">
          <vnsRsCIfPathAtt tDn="topology/pod-1/paths-104/pathep-[eth1/15]"/>
        </vnsCIf>
        <vnsCIf name="int2" vnicName="">
          <vnsRsCIfPathAtt tDn="topology/pod-1/paths-105/pathep-[eth1/15]"/>
        </vnsCIf>
      </vnsCDev>
      <vnsLIf encap="vlan-3540" name="TAP">
        <vnsRsCIfAttN tDn="uni/tn-t22/lDevVip-copy0/cDev-copy_Dyn_Device_0/cIf-[int2]"/>
        <vnsRsCIfAttN tDn="uni/tn-t22/lDevVip-copy0/cDev-copy_Dyn_Device_0/cIf-[int1]"/>
      </vnsLIf>
</vnsLDevVip>
```

ステップ 2 論理デバイス コンテキスト (デバイス選択ポリシーとも呼ばれる) を作成します。

例：

```
<vnsLDevCtx ctrctNameOrLbl="c0" descr="" graphNameOrLbl="g0" name="" nodeNameOrLbl="CP1">
      <vnsRsLDevCtxToLDev tDn="uni/tn-t22/lDevVip-copy0"/>
      <vnsLIfCtx connNameOrLbl="copy" descr="" name="">
        <vnsRsLIfCtxToLIf tDn="uni/tn-t22/lDevVip-copy0/lIf-TAP"/>
      </vnsLIfCtx>
</vnsLDevCtx>
```

ステップ 3 作成し、コピーするグラフ テンプレートを適用します。

例：

```
<vnsAbsGraph descr="" name="g0" ownerKey="" ownerTag="" uiTemplateType="UNSPECIFIED">
      <vnsAbsTermNodeCon descr="" name="T1" ownerKey="" ownerTag="">
        <vnsAbsTermConn attNotify="no" descr="" name="1" ownerKey="" ownerTag=""/>
        <vnsInTerm descr="" name=""/>
        <vnsOutTerm descr="" name=""/>
      </vnsAbsTermNodeCon>
      <vnsAbsTermNodeProv descr="" name="T2" ownerKey="" ownerTag="">
        <vnsAbsTermConn attNotify="no" descr="" name="1" ownerKey="" ownerTag=""/>
        <vnsInTerm descr="" name=""/>
        <vnsOutTerm descr="" name=""/>
      </vnsAbsTermNodeProv>
      <vnsAbsConnection adjType="L2" connDir="provider" connType="external" descr="" name="C1"
        ownerKey="" ownerTag="" unicastRoute="yes">
        <vnsRsAbsConnectionConns tDn="uni/tn-t22/AbsGraph-g0/AbsTermNodeCon-T1/AbsTConn"/>
        <vnsRsAbsConnectionConns tDn="uni/tn-t22/AbsGraph-g0/AbsTermNodeProv-T2/AbsTConn"/>
        <vnsRsAbsCopyConnection tDn="uni/tn-t22/AbsGraph-g0/AbsNode-CP1/AbsFConn-copy"/>
      </vnsAbsConnection>
      <vnsAbsNode descr="" funcTemplateType="OTHER" funcType="None" isCopy="yes" managed="no"
        name="CP1" ownerKey="" ownerTag="" routingMode="unspecified" sequenceNumber="0"
shareEncap="no">
        <vnsAbsFuncConn attNotify="no" descr="" name="copy" ownerKey="" ownerTag=""/>
        <vnsRsNodeToLDev tDn="uni/tn-t22/lDevVip-copy0"/>
      </vnsAbsNode>
</vnsAbsGraph>
```

ステップ 4 エンドポイントのグループに関連付けられている契約でコピー グラフに関係を定義します。

例：

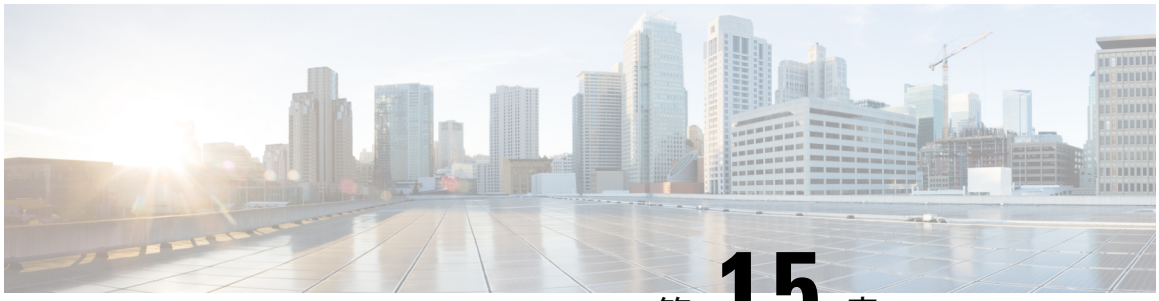
```
<vzBrCP descr="" name="c0" ownerKey="" ownerTag="" prio="unspecified" scope="tenant"
targetDscp="unspecified">
   <vzSubj consMatchT="AtleastOne" descr="" name="Subject" prio="unspecified" provMatchT="AtleastOne"

      revFltPorts="yes" targetDscp="unspecified">
      <vzRsSubjFiltAtt directives="" tnVzFilterName="default"/>
```

```
      <vzRsSubjGraphAtt directives="" tnVnsAbsGraphName="g0"/>
    </vzSubj>
</vzBrCP>
```

ステップ 5 エンドポイント グループを契約を接続します。

例：

```
<fvAEPg name="epg2860">
        <fvRsCons tnVzBrCPName="c0"/>
        <fvRsBd tnFvBDName="bd0"/>
        <fvRsDomAtt tDn="uni/phys-phys_scale_SB"/>
        <fvRsPathAtt tDn="topology/pod-1/paths-104/pathep-[PC_int2_g1]" encap="vlan-2860"
          instrImedcy="immediate"/>
      </fvAEPg>
      <fvAEPg name="epg2861">
        <fvRsProv tnVzBrCPName="c0"/>
        <fvRsBd tnFvBDName="bd0"/>
        <fvRsDomAtt tDn="uni/phys-phys_scale_SB"/>
        <fvRsPathAtt tDn="topology/pod-1/paths-105/pathep-[PC_policy]" encap="vlan-2861"
          instrImedcy="immediate"/>
</fvAEPg>
```



第 15 章

レイヤ 4 ～ レイヤ 7 リソース プールの設定

レイヤ 4 ～ レイヤ 7 リソース プールについて

レイヤ 4 ～ レイヤ 7 リソース プールは、レイヤ 4 ～ レイヤ 7 サービス デバイスの展開に関し、関係する設定をまとめます。関連する設定がパッケージとしてまとめられるので、レイヤ 4 ～ レイヤ 7 サービス デバイスを展開するための Cisco Application Centric Infrastructure (Cisco ACI) Windows Azure パック統合などのような、オーケストレーション レイヤで使用することができます。

外部およびパブリック IP アドレス プールについて

Cisco APIC リリース 3.0(x) 以前で作成されたレイヤ 4 ～ レイヤ 7 リソース プールの場合、パブリック IP アドレス プールと外部 IP アドレス プールは全く同じものであり、単に外部としてマークされているだけです。Cisco APIC リリース 3.1(x) 以降で作成されたレイヤ 4 ～ レイヤ 7 リソース プールの場合、これら 2 つのタイプのアドレス プールは分けられており、区別されます。外部 IP アドレス プールは、レイヤ 4 ～ レイヤ 7 デバイスの外部インタフェースおよび L3Out SVI の IP 割り当てのために使用されます。VPC を通してファブリックに接続するレイヤ 4 ～ レイヤ 7 デバイスの場合、 L3Out の設定のために 3 つの IP アドレス (サイド A のプライマリ IP アドレス、サイド B のプライマリ IP アドレス、およびセカンダリ IP アドレス) が消費されます。一方、ポート チャネルとシングル インターフェイス接続の場合、2 つの IP アドレス (プライマリ IP アドレスおよびセカンダリ IP アドレス) を消費します。

パブリック IP アドレス プールは、ダイナミック NAT の IP アドレスの割り当て (テナント VRF ごとに 1 つ)、ロード バランサ、仮想 IP アドレス (テナント EPG ごとに 1)、およびその他のパブリック NAT IP アドレスを割り当てるために用いられます。

2 つの IP アドレスのタイプを分けることにより、Cisco APIC 管理者は、次のことを行えます。

- IP プールの中でパブリックとマークされている IP アドレスだけをエクスポートします。デバイスレベルのインターフェイス IP アドレスを隠すことができます。
- パブリック IP アドレス プールの IP アドレスのさまざまなブロックに対し、アドレスを取得して、共通のテナント L3Out で利用可能になったときに段階的に追加を行えます。

外部レイヤ 3 ルーテッド ドメインおよび関連付けられた VLAN プールについて

外部 L3Out ルーテッド ドメインは、レイヤ 4 ～ レイヤ 7 デバイスの内部および外部コネクタの両方に L3Out をプロビジョニングするために使用されます。これらの L3Out は、トラフィックが Cisco Application Centric Infrastructure (Cisco ACI) ファブリックの外部から発信すること、および Cisco ACI ファブリック内部のリソースに到達することを可能にします。また、L3Outs は、トラフィックが Cisco ACI ファブリックの内部から発信すること、および Cisco ACI ファブリックの外部に到達することも可能にします。L3Out ルーテッド ドメインに関連付けられる VLAN プール内の VLANは、レイヤ 4 ～ レイヤ 7 サービス デバイスが接続されている特定のリーフまたは VPC リーフスイッチ ペアに対して一意のものである必要があります。レイヤ 4 ～ レイヤ 7 サービス デバイスが複数のリーフまたは VPC リーフスイッチ ペアにわたるものである場合、この制限はそれらのリーフまたは VPC リーフスイッチ ペアにも及びます。

（注） いったんレイヤ 4 ～ レイヤ 7 リソース プールが使用されたら、VLAN ブロックを再設定したり、VLANプールから削除したりするべきではありません。拡張が必要な場合は、現在のVLAN ブロックに VLAN ブロックを追加できます。

VLAN プールのサイズについては、次の考慮点が:

- 外部 IP アドレス プールごとに、1 つの VLAN がダイナミックに割り当てられます。
- レイヤ 4 ～ レイヤ 7 リソース プールにアクセスする、テナント仮想フォワーディングおよびルーティング (VRF) ごとに、1 つの VLAN がダイナミックに割り当てられます。
- 外部ルーテッド ドメインおよび関連付けられている VLAN プールは、レイヤ 4 ～ レイヤ 7 リソース プール全体にわたって使用できます。

OSPF 外部ルーテッド ネットワークの概要

外部ルーテッド ネットワークの設定についての情報は、次の URL の テナントネットワークの外部の *Cisco APIC* レイヤ *3* を参照してください。

http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html

サポートされている管理モードのレイヤ 4 ～ レイヤ 7 のデバイス

レイヤ 4 ～ レイヤ 7 リソース プールは現在のところ、次の管理モードのレイヤ 4 ～ レイヤ 7 デバイスをサポートしています:

リソース プールのタイプ	デバイスのタイプ	デバイスのパッケージのバージョン	デバイスのファームウェアのバージョン	デバイス モデル	サービス タイプ	コンテキストの対応状況
従来型	物理/仮想	CISCO-ASA-1.2 (1.2.7.10) 以降	9.2.1 以降	ASA55xx/ASAv	ファイアウォール	×
従来型	物理/仮想	Citrix-Netscaler-1.0 (11.0 ビルド 65.36) 以降	11.1 ビルド 49.16 以降	Netscaler MPX/VPX/SDX のコンテキスト	ADC	×
クラウドオーケストレータ モード	物理/仮想	CISCO-ASA-1.3 (1.3.10.8) 以降	9.6 以降	ASA55xx/ASAv	ファイアウォール	×
クラウドオーケストレータ モード	物理/仮想	Citrix-Netscaler-2.0 以降	12.0 以降	Netscaler MPX/VPX/SDX のコンテキスト	ADC	×

クラウドオーケストレータモード機能プロファイルの概要

クラウド オーケストレータ モード機能プロファイルとは、標準サービスの簡素化され、整合性のある設定に対して許可されるレイヤ4からレイヤ7デバイス パッケージの機能プロファイルです。ADC レイヤ 4 からレイヤ 7 デバイスにロード バランサ機能を提供する場合、単一のロード バランサ ポリシーはエンドポイント グループにプッシュできます。異なる ADC レイヤ 4 からレイヤ 7 デバイスが選択され、管理者の設定オーバヘッドを簡素化する場合、このロード バランサ ポリシーを変更する必要はありません。レイヤ 7 からレイヤ 4 へのデバイス パッケージは、それぞれのデバイス ベンダーによって提供されます。

GUI を使用してレイヤ 4 ～ レイヤ 7 リソース プールのための IP アドレス プールを作成する

次の手順では、いずれかの GUI モードを使用して、レイヤ 4 ～ レイヤ 7 リソース プールのための IP アドレス プールを作成します。

ステップ 1 メニュー バーで、**Tenants** > **Common** を選択します。

ステップ 2 **Navigation** ウィンドウで、**Tenant Common** > **IP Address Pools** を選択します。

ステップ 3 **Work** ウィンドウで、**Actions** > **Create IP Address Pool** を選択します。

ステップ 4 **Create IP Address Pool** ダイアログボックスで、必要に応じてフィールドに入力します。

Address Ranges には、ゲートウェイ アドレスを含めないでください。ゲートウェイ アドレスは、レイヤ 4 ～ レイヤ7デバイスの外部L3Outのセカンダリ IPアドレスとして使用されます。これはパーベイシブ ゲートウェイになります。

例：

- **Name—`ExtIPPool1`**
- **Gateway Address— `132.121.101.1/24`**
- **Address Block**
  - **From—`132.121.101.2`**
  - **To—`132.121.101.200`**

ステップ 5 [Submit] をクリックします。

GUIを使用したレイヤ4～7リソース プールのダイナミック VLAN プールの作成

次の手順では、GUI モードを使用して、レイヤ 4 ～ 7 リソース プールのダイナミック VLAN プールを作成します。

ステップ 1 メニューバーで、**[Fabric]** > **[Access Policies]** を作成します。

ステップ 2 [Navigation] ウィンドウで、 **[Pools]** > **[VLAN]** の順に選択します。

ステップ 3 [Work] ウィンドウで、**[Actions]** > **[Create VLAN Pool]** の順に選択します。

ステップ 4 [Create VLAN Pool] ダイアログボックスで、下記で指定している項目を除き、必要に応じてフィールドに入力します。

a) [Allocation Mode] ボタンでは、[Dynamic Allocation] をクリックします。

b) [Encap Blocks] テーブルで、[+] をクリックします。

c) [Create Ranges] ダイアログボックスで、下記で指定している項目を除き、必要に応じてフィールドに入力します:

- [Range] フィールドに、目的の VLAN 範囲を入力します。
- [Allocation Mode] ボタンでは、[Inherit alloc mode from parent] をクリックします。

d) [OK] をクリックします。

ステップ 5 [Create VLAN Pool] ダイアログボックスで、[Submit] をクリックします。

GUI を使用して、レイヤ 4 ～ レイヤ 7 のリソース プールのために外部ルーテッド ドメインを作成する

次の手順では、GUI モードを使用して、レイヤ 4 ～ レイヤ 7 のリソース プールのためにダイナミック VLAN プールを作成します。

ステップ 1 メニューバーで、**[Fabric]** > **[Access Policies]** を作成します。

ステップ 2 [Navigation] ウィンドウで、 **[Physical and External Domains]** > **[External Routed Domains]** を選択します。

ステップ 3 [Work] ウィンドウで、**[Actions]** > **[Create Layer 3 Domain]** を選択します。

ステップ 4 [Create Layer 3 Domain] ダイアログ ボックスで、次に指定されている点を除き、必要に応じてフィールドに入力します。

a) [Associated Attachable Entity Profile] ドロップダウンリストでは、すべてのレイヤ 4 ～ レイヤ 7 サービス デバイスの接続先となっている、アタッチ可能なエンティティのプロファイルを選択します。

b) [VLAN Pool] ドロップダウンリストでは、レイヤ 4 ～ レイヤ 7 リソース プールのために作成したダイナミック VLAN プールを選択します。
c) [Security Domains] テーブルで、必要なセキュリティ ドメインを追加します。

ステップ 5 [Submit] をクリックします。

レイヤ 4 ～ レイヤ 7 リソース プールで使用するレイヤ 4 ～ レイヤ 7 デバイスの準備

レイヤ 4 ～レイヤ 7 デバイスの物理接続を設定するには、デバイス内のポート チャネルまたは VPC 設定に関して、各デバイスごとに適切な設定ガイドを参照してください。

（注） コンテキスト認識である ASA55xx ファイアウォール デバイスについて、パス設定は特定の物理 ASA55xx のすべての ASA コンテキストの間で整合性がある必要があります。異なるインターフェイスを使用して ASA コンテキストを設定することは、この設定では許可されていません。

レイヤ 4 ～ レイヤ 7 リソース プールで使用するレイヤ 4 ～ レイヤ 7 デバイスの APIC 設定の検証

次の手順では、GUI モードを使用して、レイヤ 4 ～ レイヤ 7 リソース プールで使用するレイヤ 4 ～ レイヤ 7 デバイスの Cisco Application Policy Infrastructure Controller (Cisco APIC) の設定を検証します。

ステップ 1 メニュー バーで、 **Tenants** > **Common** を選択します。

ステップ 2 [Navigation] ウィンドウで、**Tenant** ***tenant_name*** > **Services** > **L4-L7** > **Devices** > ***ASA_or_NetScaler_logical_device_name*** > ***concrete_device_name*** を選択します。

ステップ 3 **Work** ウィンドウで、**Policy** タブを選択します。

ステップ 4 **Interfaces** テーブルで、少なくとも 2 つのインターフェイスがあり、それぞれがファブリックの検証パス (ポート、ポート チャネル、または VPC) にマップされていることを確認します。

ステップ 5 ASA または NetScaler ごとに、**Cluster > consumer** インターフェイスと **Cluster > provider** インターフェイスの両方が定義されていることを確認します。NetScalers が内部のロード バランシングで使用される場合でも、そのような設定は、テナントがプライベートおよびパブリック両方の IP アドレス ロード バランシングで NetScaler を使用ことを許可するようにします。

ステップ 6 HA 設定では、クラスタ インターフェイスごとに 2 つの具体的なインターフェイスがあることを確認します。これにより、それぞれのポート、ポート チャネル、または VPC が適切に設定されます。

ステップ 7 デバイス タイプごとに、必要なオンボード設定パラメータを設定します。これには、テナント共通のデバイスのレイヤ 4 ～ レイヤ 7 パラメータを通して、NetScaler でのロード バランシングを有効にすることなどが含まれます。

デバイス管理ネットワークとルートの構成

レイヤ 4 ～ レイヤ 7 デバイス上で管理ルートを構成し、直接アウト オブ バンドとなっているデフォルトのルートを削除する必要があります。

次の例では、 Cisco Application Policy Infrastructure Controller (Cisco APIC) の NX-OS スタイル CLI を使用して、ASA ファイアウォールの管理ルートを構成します:

```
apic1(config)# route management 10.24.24.0 255.255.255.0 172.0.0.1
```

次の例では、 Cisco APIC の NX-OS スタイル CLI を使用して、デフォルトのルートを削除します。

```
apic1(config)# no route 0.0.0.0 0.0.0.0 172.0.0.1
```

次の例では、Citrix NetScaler CLI を使用して、NetScaler アプリケーション配信コントローラ (ADC) のロード バランサの管理ルートを構成します:

```
> add route 10.24.24.0 255.255.255.0 172.0.0.1
```

次の例では、Citrix NetScaler CLI を使用して、デフォルト ルートを削除します:

```
> rm route 0.0.0.0 0.0.0.0 172.0.0.1
```

レイヤ 4 ～ レイヤ 7 リソース プールの作成

GUI を使用したレイヤ 4 ～ レイヤ 7 リソース プールの作成

次の手順では、GUI モードを使用してレイヤ 4 ～ レイヤ 7 リソース プールを作成します。いったんリソース プールに、テナントで使用するためのさまざまなコンポーネントを割り当てると、その後でリソース プールを変更することはできません。IP アドレス ブロックの追加、VLAN ブロックを追加して、ASA ファイアウォールまたは Citrix NetScaler などの論理デバイスの追加などの、メンテナンス タスクは実行できます。

ステップ 1 メニュー バーで、**Tenants** > **Common** を選択します。

ステップ 2 [Navigation] ペインで、 **[Tenant Common]** > **[Services]** > **[L4-L7]** > **[L4-L7 Resource Pools]** を選択します。

ステップ 3 **Work** ウィンドウで、**Actions** > **Create L4-L7 Resource Pool** を選択します。

ステップ 4 **Create L4-L7 Resource Pool** ダイアログボックスで、下記で指定している項目を除き、必要に応じてフィールドに入力します:

a) **Private IP Address Subnet** フィールドで、内部デバイス インターフェイスの IP アドレス、内部 VIP アドレス、および内部 L3Out IP アドレスに使用されるサブネットを入力します。
b) **External IP Address Pool** ドロップダウンリストで、サービス グラフとデバイス全体で使用される IP アドレスの動的な割り当てに使用される IP アドレス プールを選択します。必要に応じて新しいIP アドレス プールを作成できます。**Connect Type** では、 **L3 External Network**を選択します。
c) **Public IP Address Pool** テーブルで、NAT IP アドレッシングと VIP アドレッシングで使用される IP アドレスの動的な割り当てに使用される IP アドレス プールを選択します。必要に応じて新しいIP アドレス プールを作成できます。**Connect Type** では、 **L3 External Network**を選択します。
d) **External Routed Domain** ドロップダウンリストで、このレイヤ 4 ～ 7 リソース プールで使用するために作成した外部ルーテッド ドメインを選択します。必要に応じて新しい外部ルーテッド ドメインを作成できます。
e) **外部ルーテッド ネットワーク** テーブルで、テナントが利用できる外部ルーテッド ネットワークを追加します。

   最初の外部ルーテッド ネットワークは自動的に `Default` とマークされます。現時点では、デフォルトのルーテッド ネットワークのみが使用されます。

f) **L4-L7 Devices** テーブルに、このレイヤ 4 ～ レイヤ 7 リソース プールの一部となるレイヤ 4 ～ レイヤ 7 デバイスを追加します。
g) **Functional Profiles** テーブルには、追加したレイヤ 4 ～ レイヤ 7 デバイスに関連付けられているクラウド オーケストレータ モード プロファイルを追加します。たとえば、Citrix Netscaler ADC L4-L7 デバイスを追加した場合には、**Device Package** ドロップダウン メニューから [Netscaler Device Package] オプションを選択してから、**Function Profile** ドロップダウンメニューでクラウド オーケストレータ モードに対応したプロファイルを選択します。

ステップ 5 [Submit] をクリックします。

NX-OS スタイル CLI を使用したレイヤ 4 ～ レイヤ 7 リソース プールの作成

このセクションでは、NX OS スタイルの CLI を使用してレイヤ 4 ～ レイヤ 7 リソース プールを設定するコマンドの例を示します。

ステップ 1 コンフィギュレーション モードを開始します。

```
apic1# configure
```

ステップ 2 テナント共通の設定モードを開始します。

```
apic1(config)# tenant common
```

ステップ 3 レイヤ 4 ～ レイヤ 7 リソース プールを指定します。

```
apic1(config)# l4l7 resource-pool <resource pool name>
```

ステップ 4 リソース プール バージョンを設定します。

```
apic1(config-resource-pool)# version normalized
```

（注） バージョンは次のとおりです。

- **標準**：Cisco APIC リリース 3.1(x) 前に作成されたリソース プール。
- **正規**：Cisco APIC リリース 3.1(x) 後に作成されたリソース プール。クラウド オーケストレータ モードでデバイスとデバイス パッケージをサポートします。

ステップ 5 リソース プールにレイヤ 4 ～ レイヤ 7 デバイスを関連付けます。

```
apic1(config-resource-pool)# l4l7-cluster Dev-ASA-4
apic1(config-resource-pool)# l4l7-cluster Dev-MPX-4
```

ステップ 6 リソース プールに外部 IP アドレス プールとして IP アドレス プールを関連付けます。

```
apic1(config-resource-pool)# address-pool mininetExtPoolL3Ext l3-external
```

ステップ 7 (正規リソース プール）リソース プールにパブリック IP アドレス プールと IP アドレス プールを関連付けます。

```
apic1(config-resource-pool)# public-address-pool mininetPubPoolL3Ext l3-external
```

ステップ 8 外部ルーテッド ドメインに関連付けます。

```
apic1(config-resource-pool)# external-routed-domain L3ServicesDom
```

ステップ 9 リソー スプールのプライベート IP アドレスのサブネットを設定します。

```
apic1(config-resource-pool)# subnet 192.168.254.1/24
```

ステップ 10 共通テナントで L3Out EPG に関連付けます。

```
apic1(config-resource-pool)# l3out vpcDefaultInstP default
```

ステップ 11 （オプション、正規リソース プール）リソース プールにクラウドオーケストレータ モード機能プロファイルを関連付けます。

```
apic1(config-resource-pool)# function-profile ASAContainer
apic1(config-function-profile)# device-package CISCO-ASA-1.2
apic1(config-function-profile)# service-function-profile
CISCO-ASA-1.2/WebServiceProfileGroup/WebPolicyForRoutedModeCloud
apic1(config-function-profile)# exit
apic1(config-resource-pool)#
```

GUI を使用したレイヤ 4 ～ レイヤ 7 リソース プールの設定

リソース プール内のレイヤ 4 ～ レイヤ 7 リソース デバイスの設定

レイヤ 4 ～ レイヤ 7 デバイスをレイヤ 4 ～ レイヤ 7 リソース プールに追加する

（注） 専用 VLAN は、L3Out がテナントのため、そのプライベート VRF 内で作成されるたびに消費されます。レイヤ 3 ドメインに関連付けられているダイナミック VLAN プールは、リソース プールに追加されるデバイスに適合できるように、付加的な VLAN の追加を必要とする場合があります。

新しいレイヤ4～レイヤ7デバイスは、いつでもリソースプールに追加することができます。

ステップ 1 メニュー バーで、**Tenants** > **Common** を選択します。

ステップ 2 [Navigation] ペインで、 **[Tenant Common]** > **[Services]** > **[L4-L7]** > **[L4-L7 Resource Pools]** を選択します。

リソース プ - ルは、[Navigation] ウィンドウに、[L4-L7 Resource Pools] の下のドロップダウンリストとして表示されます。

ステップ 3 デバイスを追加するレイヤ 4 ～ レイヤ 7 リソース プーをクリックします。

ステップ 4 [Work] ウィンドウの **L4-L7 Devices** タブをクリックします。

ステップ 5 **L4-L7 Devices** テーブルで、プラスのアイコン (+) をクリックします。

Create An L4-L7 Device ダイアログが表示されます。

ステップ 6 **Device** ドロップダウン矢印をクリックして、レイヤ 4 ～ レイヤ 7 デバイスを選択します。

ステップ 7 [Submit] をクリックします。

レイヤ 4 ～ レイヤ 7 デバイスをレイヤ 4 ～ レイヤ 7 リソース プールから削除する

リソース プールは、設定されたレイヤ 4 ～ レイヤ 7 デバイスが利用可能出ない限り、どのテナントも使用できません。レイヤ4～レイヤ7デバイスが割り当てられておらず、どのテナントにもエクスポートされていない場合には、次の手順を実行します:

ステップ 1 メニュー バーで、**Tenants** > **Common** を選択します。

ステップ 2 [Navigation] ペインで、 **[Tenant Common]** > **[Services]** > **[L4-L7]** > **[L4-L7 Resource Pools]** を選択します。

リソース プールは、**Navigation** ウィンドウの **L4-L7 Resource Pools** の下で、ドロップダウンリストとして表示されます。

ステップ 3 削除するデバイスが含まれているレイヤ 4 ～ 7 リソース プールをクリックします。

ステップ 4 作業ウィンドウで、**L4-L7 Devices** タブをクリックします。

ステップ 5 削除するレイヤ 4 ～ レイヤ 7 デバイスをハイライトして、**trashcan** のアイコンをクリックします。

確認用のダイアログが表示されます。

ステップ 6 [Yes] をクリックして削除を確定します。

リソー スプールの外部 IP アドレス プールの設定

レイヤ 7 リソース プールにレイヤ 4 への外部 IP アドレス プールの追加

リソースプールを使用している場合またはしないでください削除テナントで使用中では、外部の IP アドレス プールを更新します。

ステップ 1 メニュー バーで、**Tenants** > **Common** を選択します。

ステップ 2 [Navigation] ペインで、 **[Tenant Common]** > **[Services]** > **[L4-L7]** > **[L4-L7 Resource Pools]** を選択します。

リソース プ - ルは、[Navigation] ウィンドウに、[L4-L7 Resource Pools] の下のドロップダウンリストとして表示されます。

ステップ 3 外部の IP アドレス プールを追加するレイヤ 7 リソース プールにレイヤ 4 をクリックします。

ステップ 4 [Work] ウィンドウの **Basic** タブをクリックします。

ステップ 5 **外部 IP アドレス プール** テーブルで、プラス記号アイコンをクリックします (＋)。

外部 IP アドレス プール フィールドが表示されます。

ステップ 6 をクリックして、 **接続タイプ** ドロップダウン矢印] を選択します **L3 外部ネットワーク** 、残りの適切な値を入力 **外部 IP アドレス プール** フィールド。

（注） フィールドの説明を確認するには、右上隅のヘルプ アイコン **(?)** をクリックします。

ステップ 7 **[Update]** をクリックします。

外部 IP アドレス プールをレイヤ 4 ～ レイヤ 7 リソース プールから削除する

（注）
- リソース プールが使用中の場合には、外部 IP アドレス プールもテナントで使用されているので、削除や更新は行わないでください。
- IP アドレス プールの枯渇に対応するために外部 IP アドレス プールの削除、追加、または更新を行う場合には、大規模な IP アドレス プールの追加や削除は行わないでください。これらの状況では、レイヤ 3 ドメインや L3Out と似た構成の、新しい外部 IP アドレス プールを伴うレイヤ 4 ～ レイヤ 7 リソース プールを作成します。
- 外部 IP アドレス プールが設定されていないと、テナントはリソース プールを使用できません。

ステップ 1 メニュー バーで、 **Tenants** > **Common** を選択します。

ステップ 2 [Navigation] ペインで、 **[Tenant Common]** > **[Services]** > **[L4-L7]** > **[L4-L7 Resource Pools]** を選択します。

リソース プ - ルは、[Navigation] ウィンドウに、[L4-L7 Resource Pools] の下のドロップダウンリストとして表示されます。

ステップ 3 削除する外部 IP アドレス プールを持つレイヤ 4 ～ レイヤ 7 リソース プールをクリックします。

ステップ 4 作業ウィンドウで、**Basic** タブをクリックします。

ステップ 5 **External IP Address Pool** テーブルで、削除する外部 IP アドレス プールをクリックしてハイライトし、**trashcan** アイコンをクリックします。

確認用のダイアログが表示されます。

ステップ 6 [Yes] をクリックして削除を確定します。

リソース プールのパブリック IP アドレス プールの設定

パブリック IP アドレス プールをレイヤ 4 ～ レイヤ 7 リソース プールに追加する

（注）
- Cisco APIC リリース 3.0(x) 以前のバージョンで作成されたレイヤ 4 ～ レイヤ 7 リソース プールの場合、外部 IP アドレス プールがパブリック IP アドレス プールとして用いられます。いったんテナントが使用したら、変更してはなりません。
- Cisco APIC リリース 3.1(x) 以降で作成されたレイヤ 4 ～ レイヤ 7 リソース プールの場合、いつでも新しいパブリック IP アドレス プールをリソース プールに追加できます。
- パブリック IP アドレス プールが設定されていないと、テナントはリソース プールを使用できません。

ステップ 1 メニュー バーで、 **Tenants** > **Common** を選択します。

ステップ 2 [Navigation] ペインで、 **[Tenant Common]** > **[Services]** > **[L4-L7]** > **[L4-L7 Resource Pools]** を選択します。

リソース プ - ルは、[Navigation] ウィンドウに、[L4-L7 Resource Pools] の下のドロップダウンリストとして表示されます。

ステップ 3 パブリック IP アドレス プールに追加するレイヤ 4 ～ レイヤ 7 リソース プールをクリックします。

ステップ 4 作業ウィンドウで、**Basic** タブをクリックします。

ステップ 5 **Public IP Address Pool** テーブルで、プラスのアイコン (+) をクリックします。

Public IP Address Pool フィールドが表示されます。

ステップ 6 **Connect Type** ドロップダウン矢印をクリックして **L3 External Network** を選択し、その他の **External IP Address Pool** フィールドに適切な値を入力します。

（注） フィールドの説明を確認するには、右上隅のヘルプ アイコン (**?**) をクリックします。

ステップ 7 **[Update]** をクリックします。

パブリック IP アドレス プールをレイヤ 4 ～ 7 リソース プールから削除する

（注）
- Cisco APIC Release 3.0(x) 以前で作成されたレイヤ 4 ～ 7 リソース プールの場合、外部 IP アドレス プールがパブリック IP アドレス プールとして使用されます。いったんテナントで使用されたら、変更してはなりません。
- Cisco APIC Release 3.1(x) 以降で作成されたレイヤ 4 ～ 7 リソース プールの場合、いずれかのテナントが現在 IP アドレス プールを利用している場合、リソース プールから IP アドレス プールを削除してはなりません。
- パブリック IP アドレスが設定されていない場合、リソース プールはどのテナントからも利用できません。

ステップ 1 メニュー バーで、**[Tenants]** > **[Common]** を選択します。

ステップ 2 [Navigation] ペインで、 **[Tenant Common]** > **[Services]** > **[L4-L7]** > **[L4-L7 Resource Pools]** を選択します。

リソース プ - ルは、[Navigation] ウィンドウに、[L4-L7 Resource Pools] の下のドロップダウンリストとして表示されます。

ステップ 3 削除するパブリック IP アドレス プールが含まれているレイヤ 4 ～ 7 リソース プールをクリックします。

ステップ 4 [Work] ウィンドウで、[Basic] タブをクリックします。

ステップ 5 **[Public IP Address Pool]** テーブルで、削除するパブリック VIP アドレス プールをクリックしてハイライトし、[trashcan] のアイコンをクリックします。

確認用のダイアログが表示されます。

ステップ 6 [Yes] をクリックして削除を確定します。

レイヤ 4 ～ レイヤ 7 リソース プールの外部ルーテッド ドメインの更新

外部ルーテッド ドメインが設定されていないと、テナントはリソース プールを使用できません。

ステップ 1 メニュー バーで、 **Tenants** > **Common** を選択します。

ステップ 2 [Navigation] ペインで、 **[Tenant Common]** > **[Services]** > **[L4-L7]** > **[L4-L7 Resource Pools]** を選択します。

リソース プ - ルは、[Navigation] ウィンドウに、[L4-L7 Resource Pools] の下のドロップダウンリストとして表示されます。

ステップ 3 更新する外部ルーテッド ドメインのあるレイヤ 4 ～ レイヤ 7 リソース プールをクリックします。

ステップ 4 [Work] ウィンドウで、[External] タブをクリックします。

ステップ 5 [External Routed Domain] ドロップダウン矢印をクリックして、レイヤ 3 ドメインを選択します。

ステップ 6 [Submit] をクリックします。

レイヤ 4 からレイヤ 7 リソースプールの外部ルーテッド ネットワークの更新

外部ルーテッド ネットワークが設定されていない場合、リソース プールはどのテナントでも使用できません。

ステップ 1 メニュー バーで、**[Tenants]** > **[Common]** の順に選択します。

ステップ 2 [Navigation] ペインで、 **[Tenant Common]** > **[Services]** > **[L4-L7]** > **[L4-L7 Resource Pools]** を選択します。

リソース プ - ルは、[Navigation] ウィンドウに、[L4-L7 Resource Pools] の下のドロップダウンリストとして表示されます。

ステップ 3 更新する外部ルーテッド ネットワークがあるレイヤ 4 からレイヤ 7 のリソース プールをクリックします。

ステップ 4 [Work] ウィンドウで、[External] タブをクリックします。

ステップ 5 [External Routed Networks] テーブルから、プラスアイコン ([+]) をクリックします。

[External Routed Networks] フィールドが表示されます。

ステップ 6 [External Routed Networks] フィールドに適切な値を入力します。

（注） フィールドの説明については、右上隅のヘルプアイコン ([?]) をクリックしてください。

ステップ 7 [Update] をクリックします。

リソース プールのクラウド オーケストレータ モード機能プロファイルの設定

レイヤ 4 ～ 7 リソース プールにクラウド オーケストレータ モード機能プロファイルを追加する

（注）
- Cisco APIC リリース 3.0(x) 以前のバージョンで作成されたレイヤ 4 ～ レイヤ 7 リソース プールの場合、クラウド オーケストレータ モードの機能プロファイルは利用できません。レイヤ 4 ～ レイヤ 7 デバイスの場合、デフォルトの機能プロファイルが使用されます。
- Cisco APIC Release 3.1(x) 以降で作成されたレイヤ 4 ～ レイヤ 7 リソース プールの場合、どの時点でも、リソース プール内の特定のレイヤ 4 ～ レイヤ 7 デバイス パッケージ用に 1 つだけクラウド オーケストレータ モード機能プロファイルがあるはずです。
- レイヤ 4 ～ 7 リソース プールの機能プロファイルは、サービス グラフの書記作成でのみ使用されます。リソース プールで機能プロファイルが更新された場合、新たに展開されたグラフだけがその設定を受け取ります。リソース プールで機能プロファイルが定義されていない場合は、デフォルトの機能プロファイルが、Cisco ASA と Citrix Netscaler レイヤ 4 ～ レイヤ 7 デバイスで使用されます。

ステップ 1 メニュー バーで、**[Tenants]** > **[Common]** を選択します。

ステップ 2 [Navigation] ペインで、 **[Tenant Common]** > **[Services]** > **[L4-L7]** > **[L4-L7 Resource Pools]** を選択します。

リソース プ - ルは、[Navigation] ウィンドウに、[L4-L7 Resource Pools] の下のドロップダウンリストとして表示されます。

ステップ 3 クラウド オーケストレータ モードの機能プロファイルを追加する、レイヤ 4 ～ レイヤ 7 リソース プールをクリックします。

ステップ 4 [Work] ウィンドウで、[Function Profiles] タブをクリックします。

ステップ 5 [Function Profiles] テーブルで、プラス アイコン ([+]) をクリックします。

[Create a Function Profile] ダイアログ ボックスが表示されます。

ステップ 6 フィールドに適切な値を入力します。

（注） フィールドの説明については、右上隅のヘルプアイコン ([?]) をクリックしてください。

a) [Name] フィールドに名前を入力します。
b) [Device Package] ドロップダウンをクリックして、クラウド オーケストレータ モード機能プロファイルを持つデバイス パッケージをクリックします。
c) [Function Profile] ドロップダウンをクリックして、クラウド オーケストレータ モード機能プロファイルを選択します。

ステップ 7 [Submit] をクリックします。

クラウド オーケストレータ モード機能プロファイルをレイヤ 4 ～ レイヤ 7 リソース プールから削除する

（注）
- Cisco APIC リリース 3.0(x) 以前のバージョンで作成されたレイヤ 4 ～ レイヤ 7 リソース プールの場合、クラウド オーケストレータ モードの機能プロファイルは利用できません。レイヤ 4 ～ レイヤ 7 デバイスの場合、デフォルトの機能プロファイルが使用されます。
- Cisco APIC リリース 3.1(x) 以降で作成されたレイヤ 4 ～ レイヤ 7 リソース プールの場合、特定の時点のリソース プールでの特定のレイヤ 4 ～ レイヤ 7 デバイス パッケージでは、クラウド オーケストレータ モードの機能プロファイルが 1 つだけ存在します。
- クラウド オーケストレータ モードの機能プロファイルがプールのデバイスに合わせて設定されていない限り、どのテナントもリソース プールを使用できません。

ステップ 1 メニュー バーで、**[Tenants]** > **[Common]** を選択します。

ステップ 2 [Navigation] ペインで、 **[Tenant Common]** > **[Services]** > **[L4-L7]** > **[L4-L7 Resource Pools]** を選択します。

リソース プ - ルは、[Navigation] ウィンドウに、[L4-L7 Resource Pools] の下のドロップダウンリストとして表示されます。

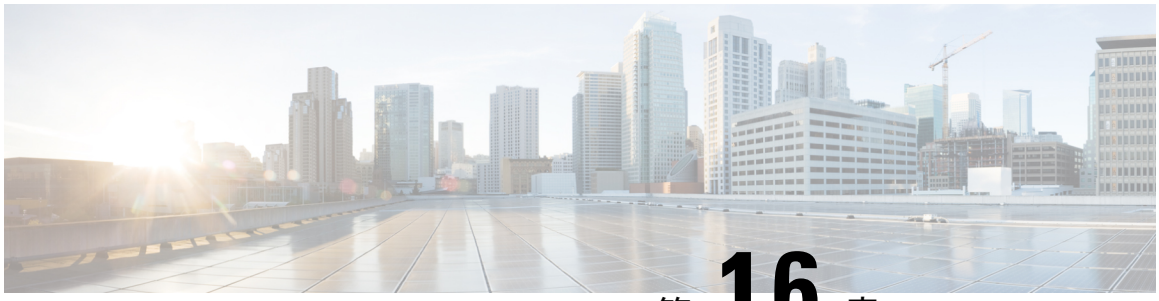
ステップ 3 クラウド オーケストレータ モードの機能プロファイルを追加する、レイヤ 4 ～ レイヤ 7 リソース プールをクリックします。

ステップ 4 [Work] ウィンドウで、[Function Profiles] タブをクリックします。

ステップ 5 [Function Profiles] テーブルで、削除する機能プロファイルをクリックしハイライト表示にし、[trashcan] アイコンをクリックします。

確認用のダイアログが表示されます。

ステップ 6 [Yes] をクリックして削除を確定します。



第 16 章

構成パラメータ

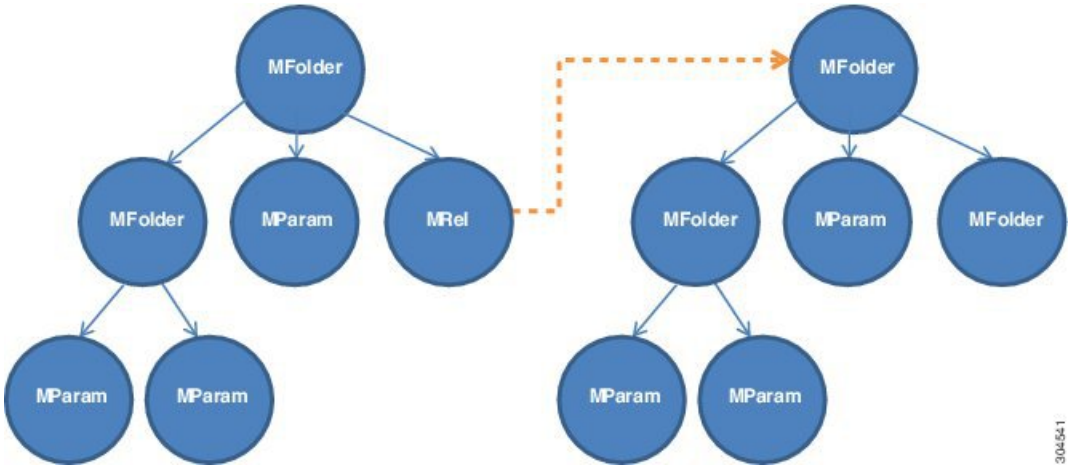
デバイスパッケージ仕様内のコンフィギュレーションパラメータ

デバイス パッケージにはサービス デバイスの仕様を示す XML ファイルが含まれます。この仕様にはデバイス情報およびサービス デバイスによって提供される各種の機能が含まれます。

デバイス仕様の一部として、このファイルにはサービス デバイスによって必要なコンフィギュレーションの宣言が含まれる必要があります。この設定は、グラフのインストール中にサービス デバイスによって提供される各種の機能を設定するために必要です。

次の図は、デバイス パッケージ内のコンフィギュレーション パラメータ階層を示しています。

図 *34* : デバイス パッケージ内のコンフィギュレーション パラメータ階層



MFolder

MFolder は、MParam および他のネストされた MFolder を含むことができるコンフィギュレーション アイテムのグループです。MFolder は次の属性を持ちます。

属性	説明
Key	コンフィギュレーション アイテムのタイプを定義します。キーは、デバイス パッケージで定義されており、上書きすることはできません。キーは、検証だけでなく一致基準として使用されます。
説明	コンフィギュレーション アイテムを説明します。
Cardinality	コンフィギュレーション アイテムの濃度を指定します。濃度のデフォルト値は `1` です。濃度が `N` であれば、Application Policy Infrastructure Controller (APIC) ではコンフィギュレーション パラメータの N インスタンスの設定が可能です。
ScopedBy	パラメータ解決の範囲を指定します。ScopedBy は、APIC がコンフィギュレーション MO からパラメータを解決する場合にパラメータ値を検索する場所を決定します。 デフォルト値は `Epg` です。サポートされる値は `Tenant`、`Ap`、`Bd`、および `Epg` です。
RsConnector	コンフィギュレーション アイテムを MConn に関連付ける関係。
DevCtx	コンフィギュレーションアイテムをデバイス (LDev) 内の特定の物理デバイス (CDev) に関連付けることができます。
Locked	コンフィギュレーション アイテム値がロックされます。一度ロックされると値は変更できません。

MParam

MParamは、単一のコンフィギュレーションパラメータを宣言するコンフィギュレーションパラメータの基本単位です。MParam は次の属性を持ちます。

属性	説明
Key	コンフィギュレーション アイテムのタイプを定義します。キーは、デバイス パッケージで定義されており、上書きすることはできません。キーは、検証だけでなく一致基準として使用されます。
説明	コンフィギュレーション アイテムを説明します。
Cardinality	コンフィギュレーション アイテムの濃度を指定します。濃度のデフォルト値は `1` です。濃度が `N` であれば、APIC ではコンフィギュレーション パラメータの N インスタンスの設定が可能です。
RsConnector	コンフィギュレーション アイテムを MConn に関連付ける関係。
必須	コンフィギュレーション アイテムが必須としてマークされます。
Locked	コンフィギュレーション アイテム値がロックされます。一度ロックされると値は変更できません。
Validation	値の検証方法を指定します。

MRel

MRel は 1 つの MFolder が別の MFolder を参照することを可能にします。MFolder 内の MRel を使用して、管理者は含む側の MFolder を、MRel 内に含まれる RsTarget 関係によって MRel からポイントされる MFolder に関連付けることができます。MRel は次の属性を持ちます。

属性	説明
Key	コンフィギュレーション アイテムのタイプを定義します。キーは、デバイス パッケージで定義されており、上書きすることはできません。キーは、検証だけでなく一致基準として使用されます。
説明	コンフィギュレーション アイテムを説明します。
Cardinality	コンフィギュレーション アイテムの濃度を指定します。濃度のデフォルト値は `1` です。
RsTarget	コンフィギュレーション フォルダを別の MFolder に関連付ける関係。この関係に対する TDn の値はターゲット フォルダの DN です。
RsConnector	コンフィギュレーション アイテムを MConn に関連付ける関係。
必須	コンフィギュレーション アイテムが必須としてマークされます。

デバイス パッケージ仕様の設定スコープ

デバイス仕様ファイルで、コンフィギュレーションアイテムは異なるセクションで配置されます。

MDevCfg

MDevCfg のセクションでは、デバイスを使用するすべてのサービス グラフで共有されるデバイス レベルの設定について説明します。Application Policy Infrastructure Controller（APIC）は、この項で説明されるコンフィギュレーションアイテムを使用して作成されたコンフィギュレーション オブジェクトの参照カウントを実行します。オブジェクトは、デバイスを使用しているすべてのグラフ インスタンスが削除された後にのみサービス デバイスから削除されます。

MFuncCfg

MFuncCfg は、サービス機能に対してローカルで、サービス機能に固有なコンフィギュレーションについて説明します。APIC は、このセクションで説明されるコンフィギュレーション アイテムによって作成されたコンフィギュレーション オブジェクトの参照カウントを実行します。オブジェクトが作成され、サービス機能がインスタンス化または削除されたときに削除されます。

MGrpCfg

MGrpCfg は、デバイスを使用するサービス グラフのすべての機能によって共有される設定を説明します。APIC は、このセクションで説明されるコンフィギュレーション アイテムを使用して作成されたコンフィギュレーション オブジェクトの参照カウントを実行します。オブジェクトは、サービス グラフからすべての機能が削除された後にサービス デバイスから削除されます。

デバイス パッケージ内のコンフィギュレーション パラメータの XML の例

次の XML の例は、デバイス パッケージ内のコンフィギュレーション パラメータを示しています。

```
<vnsMFolder key="VServer" scopedBy="epg">
    <vnsRsConnector tDn="uni/infra/mDev-Acme-ADC-1.0/mFunc-SLB/mConn-external"/>
    <vnsMParam key="vservername" description="Name of VServer" mandatory="true"/>
    <vnsMParam key="vip" description="Virtual IP"/>
    <vnsMParam key="subnet" description="Subnet IP"/>
    <vnsMParam key="port" description="Port for Virtual server"/>
    <vnsMParam key="persistencetype" description="persistencetype"/>
    <vnsMParam key="servicename" description="Service bound to this vServer"/>
    <vnsMParam key="servicetype" description="Service bound to this vServer"/>
    <vnsMParam key="clttimeout" description="Client timeout"/>
    <vnsMFolder key="VServerGlobalConfig"
      description="This references the global configuration">
        <vnsMRel key="ServiceConfig">
            <vnsRsTarget tDn="uni/infra/mDev-Acme-ADC-1.0/mDevCfg/mFolder-Service"/>
        </vnsMRel>
        <vnsMRel key="ServerConfig">
```

```
            <vnsRsTarget tDn="uni/infra/mDev-Acme-ADC-1.0/mDevCfg/mFolder-Server"/>
        </vnsMRel>
        <vnsMRel key="VipConfig">
            <vnsRsTarget
              tDn="uni/infra/mDev-Acme-ADC-1.0/mDevCfg/mFolder-Network/mFolder-vip"/>
           <vnsRsConnector tDn="uni/infra/mDev-Acme-ADC-1.0/mFunc-SLB/mConn-external"/>

        </vnsMRel>
    </vnsMFolder>
</vnsMFolder>
```

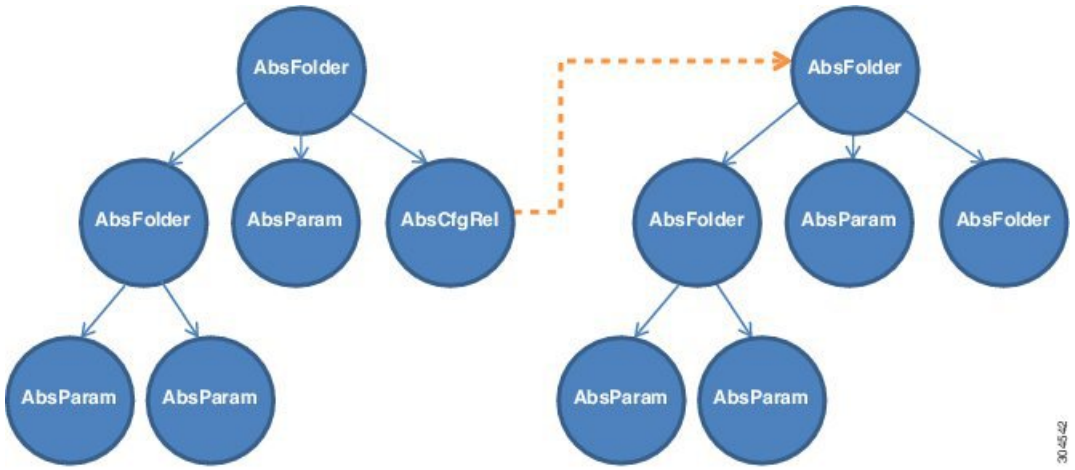
抽象機能プロファイル内のコンフィギュレーションパラメータ

抽象プロファイルを使用すると、管理者はコンフィギュレーションパラメータのデフォルト値を設定できます。抽象機能プロファイルには値を持つコンフィギュレーションパラメータが含まれます。これらの値がグラフ インスタンス作成時にデフォルト値として使用されます。

抽象機能プロファイルはサービス グラフの機能ノードに接続されます。抽象機能プロファイルで指定されたデフォルト値は、グラフのインスタンス化の際にサービスデバイスに機能をレンダリングする場合に使用されます。

次の図は、抽象機能プロファイル内のコンフィギュレーション パラメータ階層を示しています。

図 *35* : 抽象機能プロファイル内部のコンフィギュレーション パラメータ階層



AbsFolder

AbsFolder は、AbsParam および他のネストされた AbsFolder を含むことができるコンフィギュレーション アイテムのグループです。デバイス パッケージ内に各 AbsFolder の MFolder が必要

です。Application Policy Infrastructure Controller（APIC） は、各 AbsFolder を検証して、パッケージ内に AbsFolder に対応する MFolder が存在することを確認します。AbsFolder には、次の属性があります。

属性	説明
Key	コンフィギュレーション アイテムのタイプを定義します。キーは、デバイス パッケージで定義されており、上書きすることはできません。キーは、検証だけでなく一致基準として使用されます。
説明	コンフィギュレーション アイテムを説明します。
Cardinality	コンフィギュレーション アイテムの濃度を指定します。デフォルト値は `1` です。
ScopedBy	パラメータ解決の範囲を指定します。ScopedBy は、APIC がコンフィギュレーション MO からパラメータを解決する場合にパラメータ値を検索する場所を決定します。 デフォルト値は `Epg` です。サポートされる値は `Tenant`、`Ap`、`Bd`、および `Epg` です。
DevCtx	コンフィギュレーション アイテムをデバイス クラスタ（LDev）内の特定の物理デバイス（CDev）に関連付けることができます。
Locked	コンフィギュレーション アイテム値がロックされます。一度ロックされると値は変更できません。

AbsParam

AbsParam はコンフィギュレーション パラメータの基本単位です。 AbsParam は単一のコンフィギュレーション パラメータを定義します。AbsFolder と同様、各 AbsParam に対してデバイス仕様内に対応する MFolder が存在する必要があります。APIC は仕様を検証して、パッケージ内に AbsParam に対応する MFolder が存在することを確認します。AbsParam の値は、MParam 内で指定される検証メソッドを使用して検証されます。AbsParam には、次の属性があります。

属性	説明
Key	コンフィギュレーション アイテムのタイプを定義します。キーは、デバイス パッケージで定義されており、上書きすることはできません。キーは、検証だけでなく一致基準として使用されます。
値	特定のコンフィギュレーション アイテムの値を保持します。値は MParam ではサポートされません。
説明	コンフィギュレーション アイテムを説明します。
Cardinality	コンフィギュレーション アイテムの濃度を指定します。デフォルト値は `1` です。
必須	コンフィギュレーション アイテムが必須としてマークされます。

属性	説明
Locked	コンフィギュレーション アイテム値がロックされます。一度ロックされると値は変更できません。
Validation	コンフィギュレーション パラメータの検証に使用する検証メカニズムを指定します。

AbsRel

AbsRel は 1 つの AbsFolder が別の AbsFolder を参照することを可能にします。AbsRel には、次の属性があります。

属性	説明
Key	コンフィギュレーション アイテムのタイプを定義します。キーは、デバイス パッケージで定義されており、上書きすることはできません。キーは、検証だけでなく一致基準として使用されます。
値	特定のコンフィギュレーション アイテムの値を保持します。値は MParam ではサポートされません。
説明	コンフィギュレーション アイテムを説明します。
Cardinality	コンフィギュレーション アイテムの濃度を指定します。デフォルト値は 1 です。
必須	コンフィギュレーション アイテムが必須としてマークされます。

抽象機能プロファイルの設定スコープ

抽象機能プロファイルでは、コンフィギュレーション パラメータはデバイス パッケージ内の場合と似た方法で構成されます。3 種類のスコープがあります。

AbsDevCfg

このセクションは、デバイスパッケージ内のデバイス レベル設定と宣言される、コンフィギュレーション アイテムのデフォルト値を提供します。コンフィギュレーション アイテムは MDevCfg で指定されます。

各コンフィギュレーションアイテムに対して、デバイスパッケージに同等のコンフィギュレーション アイテムが存在する必要があります。

このセクションで説明される設定は、デバイスを使用するサービス グラフで共有されます。Application Policy Infrastructure Controller（APIC）は、このセクションで説明されるコンフィギュレーション アイテムを使用して作成されたコンフィギュレーション オブジェクトの参照カウントを実行します。オブジェクトは、デバイスを使用しているすべてのグラフインスタンスが削除された後にのみサービス デバイスから削除されます。

AbsGrpCfg

このセクションは、デバイスパッケージ内のデバイス レベル設定と宣言される、コンフィギュレーションアイテムのデフォルト値を提供します。コンフィギュレーションアイテムはMGrpCfgで指定されます。

各コンフィギュレーションアイテムに対して、デバイスパッケージに同等のコンフィギュレーション アイテムが存在する必要があります。

このセクションで説明される設定は、デバイスを使用するサービス グラフのすべての機能で共有されます。APIC は、このセクションで説明されるコンフィギュレーション アイテムを使用して作成されたコンフィギュレーションオブジェクトの参照カウントを実行します。オブジェクトは、デバイスを使用しているすべてのグラフインスタンスが削除された後にのみサービスデバイスから削除されます。

AbsFuncCfg

このセクションは、デバイスパッケージ内の機能レベル設定と宣言される、コンフィギュレーションアイテムのデフォルト値を提供します。コンフィギュレーションアイテムはMFuncCfgで指定されます。

各コンフィギュレーションアイテムに対して、デバイスパッケージに同等のコンフィギュレーション アイテムが存在する必要があります。

このセクションは、サービス機能にローカルな設定を説明するために使用されます。このセクションで説明されている設定は、サービス機能に固有のものです。APIC は、このセクションで説明されるコンフィギュレーション アイテムによって作成されたコンフィギュレーションオブジェクトの参照カウントを実行します。オブジェクトが作成され、サービス機能がインスタンス化または削除されたときに削除されます。

コンフィギュレーションパラメータを持つ抽象機能プロファイルに対する XML POST の例

次の XML POST の例は、コンフィギュレーション パラメータを持つ抽象機能プロファイルを示しています。

```
<vnsAbsFuncProfContr name = "NP">
    <vnsAbsFuncProfGrp name = "Grp1">
        <vnsAbsFuncProf name = "P1">
            <vnsRsProfToMFunc tDn="uni/infra/mDev-Acme-ADC-1.0/mFunc-SLB"/>
            <vnsAbsDevCfg name="D1">
                <vnsAbsFolder key="Service" name="Service-Default" cardinality="n">
                    <vnsAbsParam name="servicetype" key="servicetype" value="TCP"/>
                    <vnsAbsParam name="serviceport" key="serviceport" value="80"/>
                    <vnsAbsParam name="maxclient" key="maxclient" value="1000"/>
                    <vnsAbsParam name="maxreq" key="maxreq" value="100"/>
                    <vnsAbsParam name="cip" key="cip" value="enable"/>
                    <vnsAbsParam name="usip" key="usip" value="enable"/>
                    <vnsAbsParam name="sp" key="sp" value=""/>
                    <vnsAbsParam name="svrtimeout" key="svrtimeout" value="60"/>
                    <vnsAbsParam name="clttimeout" key="clttimeout" value="60"/>
                    <vnsAbsParam name="cka" key="cka" value="NO"/>
                    <vnsAbsParam name="tcpb" key="tcpb" value="NO"/>
```

```
                    <vnsAbsParam name="cmp" key="cmp" value="NO"/>
                </vnsAbsFolder>
            </vnsAbsDevCfg>
            <vnsAbsFuncCfg name="SLB">
                <vnsAbsFolder key="VServer" name="VServer-Default">
                    <vnsAbsParam name="port" key="port" value="80"/>
                    <vnsAbsParam name="persistencetype" key="persistencetype"
                      value="cookie"/>
                    <vnsAbsParam name="clttimeout" key="clttimeout" value="100"/>
                    <vnsAbsParam name="servicetype" key="servicetype" value="TCP"/>
                    <vnsAbsParam name="servicename" key="servicename"/>
                </vnsAbsFolder>
            </vnsAbsFuncCfg>
        </vnsAbsFuncProf>
    </vnsAbsFuncProfGrp>
</vnsAbsFuncProfContr>
```

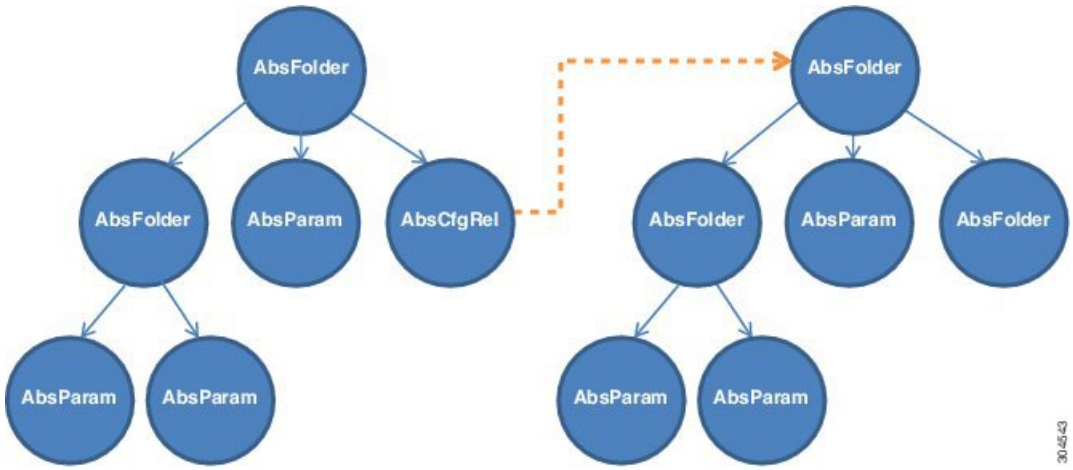
サービス グラフでの抽象機能ノード内のコンフィギュレーション パラメータ

サービス グラフ内の機能ノードを使用して、管理者はコンフィギュレーション パラメータの値を設定できます。これらの値は、グラフのインストール時に使用されます。

抽象機能ノードでは、コンフィギュレーションパラメータは抽象機能プロファイル内の場合と似た方法で構成されます。

次の図は、抽象機能ノード内のコンフィギュレーション パラメータ階層を示しています。

図 *36* : 抽象機能ノード内のコンフィギュレーション パラメータ



AbsDevCfg

このセクションは、デバイスパッケージ内のデバイス レベル設定と宣言される、コンフィギュレーションアイテムのデフォルト値を提供するために使用されます。コンフィギュレーションアイテムは MDevCfg で指定されます。

これらの各コンフィギュレーションアイテムに対して、デバイスパッケージに同等のコンフィギュレーション アイテムが存在する必要があります。

AbsGrpCfg

このセクションは、デバイスパッケージ内のデバイス レベル設定と宣言される、コンフィギュレーションアイテムのデフォルト値を提供するために使用されます。コンフィギュレーションアイテムは MGrpCfg で指定されます。

これらの各コンフィギュレーションアイテムに対して、デバイスパッケージに同等のコンフィギュレーション アイテムが存在する必要があります。

このセクションで説明される設定は、デバイスを使用するサービス グラフのすべての機能で共有されます。Application Policy Infrastructure Controller（APIC）は、この項で説明されるコンフィギュレーション アイテムを使用して作成されたコンフィギュレーション オブジェクトの参照カウントを実行します。オブジェクトは、サービス グラフからすべての機能が削除された後にサービス デバイスから削除されます。

AbsFuncCfg

このセクションは、デバイスパッケージ内の機能レベル設定と宣言される、コンフィギュレーション アイテムのデフォルト値を提供するために使用されます。コンフィギュレーション アイテムは MFuncCfg で指定されます。

これらの各コンフィギュレーションアイテムに対して、デバイスパッケージに同等のコンフィギュレーション アイテムが存在する必要があります。

このセクションは、サービス機能にローカルな設定を説明するために使用されます。このセクションで説明されている設定は、サービス機能に固有のものです。APIC は、このセクションで説明されるコンフィギュレーション アイテムによって作成されたコンフィギュレーション オブジェクトの参照カウントを実行します。オブジェクトが作成され、サービス機能がインスタンス化または削除されたときに削除されます。

AbsFolder

AbsFolder は、AbsParam および他のネストされた AbsFolder を含むことができるコンフィギュレーション アイテムのグループです。デバイス パッケージ内に各 AbsFolder の MFolder が必要です。APIC は、各 AbsFolder を検証して、パッケージ内に AbsFolder に対応する MFolder が存在することを確認します。AbsFolder には、次の属性があります。

属性	説明
Key	コンフィギュレーション アイテムのタイプを定義します。キーは、デバイス パッケージで定義されており、上書きすることはできません。キーは、検証だけでなく一致基準として使用されます。

属性	説明
説明	コンフィギュレーション アイテムを説明します。
Cardinality	コンフィギュレーション アイテムの濃度を指定します。デフォルト値は `1` です。
ScopedBy	パラメータ解決の範囲を指定します。ScopedByは、APICがコンフィギュレーション MO からパラメータを解決する場合にパラメータ値を検索する場所を決定します。 デフォルト値は `Epg` です。サポートされる値は `Tenant`、`Ap`、`Bd`、および `Epg` です。
RsCfgToConn	コンフィギュレーション アイテムを AbsConn に関連付ける関係。
DevCtx	コンフィギュレーションアイテムをデバイス(LDev)内の特定の物理デバイス(CDev) に関連付けることができます。
Locked	コンフィギュレーション アイテム値がロックされます。一度ロックされると値は変更できません。

AbsParam

AbsParamはコンフィギュレーションパラメータの基本単位です。 AbsParamは単一のコンフィギュレーション パラメータを定義します。AbsFolder と同様、各 AbsParam に対してデバイス仕様内に対応する MFolder が存在する必要があります。APIC は仕様を検証して、パッケージ内に AbsParam に対応する MFolder が存在することを確認します。AbsParam の値は、MParam 内で指定される検証メソッドを使用して検証されます。AbsParam には次の属性があります。

属性	説明
Key	コンフィギュレーション アイテムのタイプを定義します。キーは、デバイス パッケージで定義されており、上書きすることはできません。キーは、検証だけでなく一致基準として使用されます。
値	特定のコンフィギュレーション アイテムの値を保持します。値は MParam ではサポートされません。
説明	コンフィギュレーション アイテムを説明します。
Cardinality	コンフィギュレーション アイテムの濃度を指定します。デフォルト値は `1` です。
RsCfgToConn	コンフィギュレーション アイテムを MConn に関連付ける関係。
必須	コンフィギュレーション アイテムが必須としてマークされます。
Locked	コンフィギュレーション アイテム値がロックされます。一度ロックされると値は変更できません。
Validation	コンフィギュレーション パラメータの検証に使用する検証メカニズムを指定します。

AbsRel

AbsRel は 1 つの AbsFolder が別の AbsFolder を参照することを可能にします。AbsRel には次の属性があります。

属性	説明
Key	コンフィギュレーション アイテムのタイプを定義します。キーは、デバイス パッケージで定義されており、上書きすることはできません。キーは、検証だけでなく一致基準として使用されます。
値	特定のコンフィギュレーション アイテムの値を保持します。値は MParam ではサポートされません。
説明	コンフィギュレーション アイテムを説明します。
Cardinality	コンフィギュレーション アイテムの濃度を指定します。デフォルト値は 1 です。
RsCfgToConn	コンフィギュレーション アイテムを MConn に関連付ける関係。
必須	コンフィギュレーション アイテムが必須としてマークされます。
Locked	コンフィギュレーション アイテム値がロックされます。一度ロックされると値は変更できません。

コンフィギュレーション パラメータを持つ抽象機能ノードに対する XML POST の例

次の XML POST の例は、コンフィギュレーション パラメータを持つ抽象機能ノードを示しています。

```
<vnsAbsNode name = "SLB" funcType="GoTo" >
   <vnsRsDefaultScopeToTerm tDn="uni/tn-tenant1/AbsGraph-G3/AbsTermNode-Output1/outtmnl"/>

    <vnsAbsFuncConn name = "C4" direction = "input">
        <vnsRsMConnAtt tDn="uni/infra/mDev-Acme-ADC-1.0/mFunc-SLB/mConn-external" />
    </vnsAbsFuncConn>
    <vnsAbsFuncConn name = "C5" direction = "output">
        <vnsRsMConnAtt tDn="uni/infra/mDev-Acme-ADC-1.0/mFunc-SLB/mConn-internal" />
    </vnsAbsFuncConn>

    <vnsAbsDevCfg>
        <vnsAbsFolder key="Network" name="Network" scopedBy="epg">
            <!-- Following scopes this folder to input terminal or Src Epg -->
            <vnsRsScopeToTerm
tDn="uni/tn-tenant1/AbsGraph-G3/AbsTermNode-Output1/outtmnl"/>

            <!-- VIP address -->
            <vnsAbsFolder key="vip" name="vip" scopedBy="epg">
                <vnsAbsParam name="vipaddress" key="vipaddress" value=""/>
            </vnsAbsFolder>

            <!-- SNIP address -->
            <vnsAbsFolder key="snip" name="snip" scopedBy="epg">
                <vnsAbsParam name="snipaddress" key="snipaddress" value=""/>
```

```
            </vnsAbsFolder>

        </vnsAbsFolder>

        <vnsAbsFolder key="Service" name="Service" scopedBy="epg" cardinality="n">
            <vnsRsScopeToTerm
tDn="uni/tn-tenant1/AbsGraph-G3/AbsTermNode-Output1/outtmnl"/>
            <vnsAbsParam name="servicename" key="servicename" value=""/>
            <vnsAbsParam name="servername" key="servername" value=""/>
            <vnsAbsParam name="serveripaddress" key="serveripaddress" value=""/>
        </vnsAbsFolder>
    </vnsAbsDevCfg>

    <vnsAbsFuncCfg>
        <vnsAbsFolder key="VServer" name="VServer" scopedBy="epg">
            <vnsRsScopeToTerm
tDn="uni/tn-tenant1/AbsGraph-G3/AbsTermNode-Output1/outtmnl"/>
            <!-- Virtual Server Configuration -->
            <vnsAbsParam name="vip" key="vip" value=""/>
            <vnsAbsParam name="vservername" key="vservername" value=""/>
            <vnsAbsParam name="servicename" key="servicename"/>
            <vnsRsCfgToConn tDn="uni/tn-tenant1/AbsGraph-G3/AbsNode-Node2/AbsFConn-C4"
/>
        </vnsAbsFolder>
    </vnsAbsFuncCfg>
    <vnsRsNodeToMFunc tDn="uni/infra/mDev-Acme-ADC-1.0/mFunc-SLB"/>
</vnsAbsNode>
```

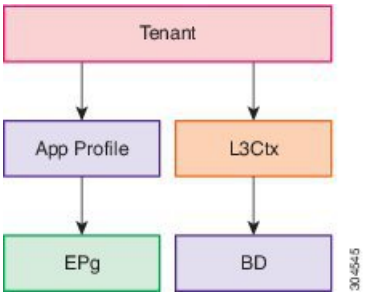
各種の設定 MO 内のコンフィギュレーション パラメータ

管理者は EPG、テナント、BD、または AP などの各種の Application Policy Infrastructure Controller（APIC） MO の一部としてサービス機能に対するコンフィギュレーション パラメータを指定できます。グラフがインスタンス化されると、APIC は各種の場所からパラメータを検索することでグラフに必要な設定を解決します。インスタンス化では、パラメータ値はデバイス スクリプトに解決され、渡されます。

各種の MO 内でコンフィギュレーション パラメータを保持できることの柔軟性により、管理者は単一のサービス グラフを設定し、グラフを異なるテナントまたはエンド ポイント グループ（EPG）に対して異なる設定で使用できます。

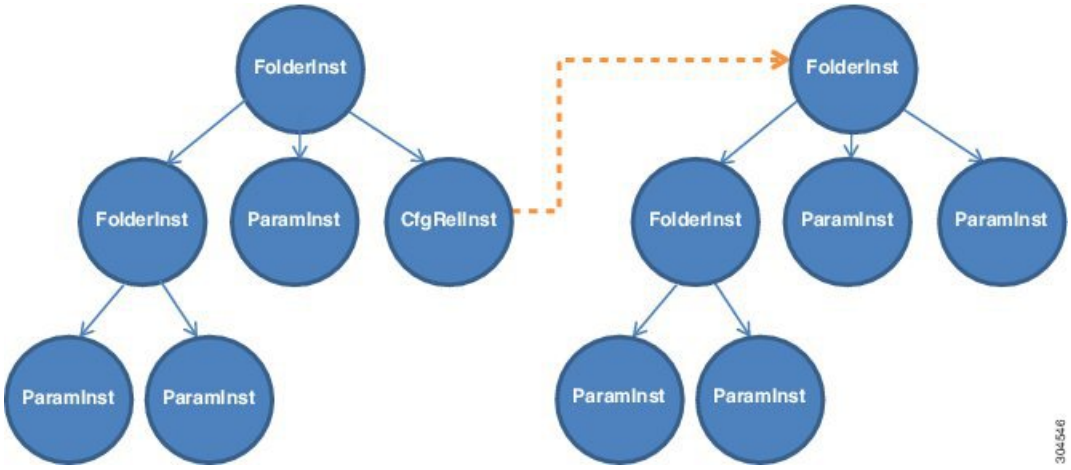
次の図は、APIC MO の階層を示しています。

図 *37 : APIC MO* の階層



次の図は、各種のコンフィギュレーション MO 内のコンフィギュレーション パラメータを示しています。

図 *38* : 各種の設定 *MO* 内のコンフィギュレーション パラメータ



FolderInst

FolderInst は、ParamInst および他のネストされた FolderInst を含むことができるコンフィギュレーション アイテムのグループです。FolderInst は次の属性を持ちます。

属性	説明
Key	コンフィギュレーション アイテムのタイプを定義します。キーは、デバイス パッケージで定義されており、上書きすることはできません。キーは、検証だけでなく一致基準として使用されます。
ctrctNameOrLbl	パラメータの解決時に一致する FolderInst を検索します。FolderInst をパラメータ解決で使用するには、このプロパティはサービス グラフと関連付けられたコントラクト名と一致する必要があります。一致していない場合、FolderInst はスキップされ、値はこの FolderInst から使用されません。 このフィールドの値を [any] にして、この FolderInst がすべてのコントラクトで使用されるようにできます。
graphNameOrLbl	パラメータの解決時に一致する FolderInst を検索します。FolderInst をパラメータ解決で使用するには、このプロパティはサービス グラフ名と一致する必要があります。一致していない場合、FolderInst はスキップされ、値はこの FolderInst から使用されません。 この FolderInst がすべてのサービス グラフで使用されるようにするには、このフィールドの値を [any] にできます。

属性	説明
nodeNameOrLbl	パラメータの解決時に一致する FolderInst を検索します。FolderInst をパラメータ解決で使用するには、このプロパティはノード名と一致する必要があります。一致していない場合、FolderInst はスキップされ、値はこの FolderInst から使用されません。 このフィールドの値を [any] にして、この FolderInst がサービス グラフ内のすべてのノードで使用されるようにできます。

ParamInst

ParamInst はコンフィギュレーション パラメータの基本単位です。ParamInst は単一のコンフィギュレーション パラメータを定義します。FolderInst と同様、各 ParamInst に対してデバイス仕様内に対応する MParam が存在する必要があります。APIC は仕様を検証して、パッケージ内に ParamInst に対応する MParam が存在することを確認します。ParamInst の値は、対応する MParam 内で指定される検証メソッドを使用して検証されます。ParamInst には、次の属性があります。

属性	説明
Key	コンフィギュレーション アイテムのタイプを定義します。キーは、デバイス パッケージで定義されており、上書きすることはできません。キーは、検証だけでなく一致基準として使用されます。
値	特定のコンフィギュレーション アイテムの値を保持します。値は MParam ではサポートされません。

CfgRelInst

CfgRelInst には、次の属性があります。

属性	説明
Key	コンフィギュレーション アイテムのタイプを定義します。キーは、デバイス パッケージで定義されており、上書きすることはできません。キーは、検証だけでなく一致基準として使用されます。
値	ターゲット FolderInst のパスを保持します。

コンフィギュレーション パラメータを持つアプリケーション EPG の XML POST の例

次の XML の例は、デバイス パッケージ内のコンフィギュレーション パラメータを示しています。

```
<fvAEPg dn="uni/tn-acme/ap-myApp/epg-app"  name="app">

    <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="any" nodeNameOrLbl="any"
key="Monitor"
```

```
      name="monitor1">
        <vnsRsFolderInstToMFolder
tDn="uni/infra/mDev-Acme-ADC-1.0/mDevCfg/mFolder-Monitor"/>
        <vnsParamInst name="weight" key="weight" value="10"/>
    </vnsFolderInst>

    <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="any" nodeNameOrLbl="any"
key="Service"
      name="Service1">
        <vnsParamInst name="servicename" key="servicename" value="crpvgrtst02-8010"/>
        <vnsParamInst name="servicetype" key="servicetype" value="TCP"/>
        <vnsParamInst name="servername" key="servername" value="s192.168.100.100"/>
        <vnsParamInst name="serveripaddress" key="serveripaddress"
value="192.168.100.100"/>
        <vnsParamInst name="serviceport" key="serviceport" value="8080"/>
        <vnsParamInst name="svrtimeout" key="svrtimeout" value="9000" />
        <vnsParamInst name="clttimeout" key="clttimeout" value="9000" />
        <vnsParamInst name="usip" key="usip" value="NO" />
        <vnsParamInst name="useproxyport" key="useproxyport" value="" />
        <vnsParamInst name="cip" key="cip" value="ENABLED" />
        <vnsParamInst name="cka" key="cka" value="NO" />
        <vnsParamInst name="sp" key="sp" value="OFF" />
        <vnsParamInst name="cmp" key="cmp" value="NO" />
        <vnsParamInst name="maxclient" key="maxclient" value="0" />
        <vnsParamInst name="maxreq" key="maxreq" value="0" />
        <vnsParamInst name="tcpb" key="tcpb" value="NO" />
        <vnsCfgRelInst name="MonitorConfig" key="MonitorConfig" targetName="monitor1"/>
    </vnsFolderInst>

    <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="G2" nodeNameOrLbl="any"
key="Network"
      name="Network">
        <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="G2" nodeNameOrLbl="any"
key="vip"
          name="vip">
            <vnsParamInst name="vipaddress1" key="vipaddress" value="10.10.10.200"/>
        </vnsFolderInst>
        <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="G2" nodeNameOrLbl="any"
          devCtxLbl="C1" key="snip" name="snip1">
            <vnsParamInst name="snipaddress" key="snipaddress" value="192.168.1.200"/>
        </vnsFolderInst>
        <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="G2" nodeNameOrLbl="any"
          devCtxLbl="C2" key="snip" name="snip2">
    <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="G1" nodeNameOrLbl="any"
key="Network"
      name="Network">
        <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="G1" nodeNameOrLbl="any"
key="vip"
          name="vip">
            <vnsParamInst name="vipaddress1" key="vipaddress" value="10.10.10.100"/>
        </vnsFolderInst>
        <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="G1" nodeNameOrLbl="any"
          devCtxLbl="C1" key="snip" name="snip1">
            <vnsParamInst name="snipaddress" key="snipaddress" value="192.168.1.100"/>
        </vnsFolderInst>
        <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="G1" nodeNameOrLbl="any"
          devCtxLbl="C2" key="snip" name="snip2">
            <vnsParamInst name="snipaddress" key="snipaddress" value="192.168.1.101"/>
        </vnsFolderInst>
        <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="G1" nodeNameOrLbl="any"
          devCtxLbl="C3" key="snip" name="snip3">
            <vnsParamInst name="snipaddress" key="snipaddress" value="192.168.1.102"/>
        </vnsFolderInst>
    </vnsFolderInst>
```

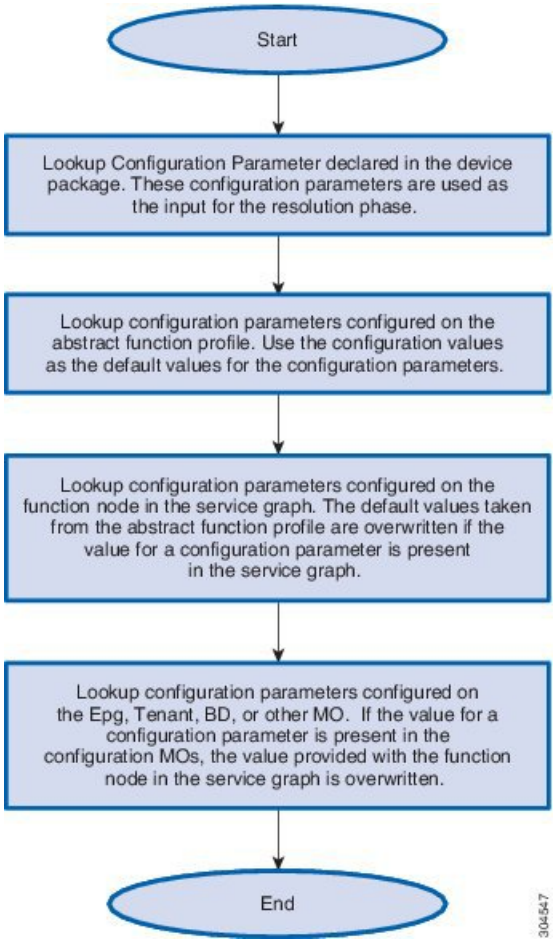
```
    <!-- SLB Configuration -->
    <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="any" nodeNameOrLbl="any"
key="VServer"
      name="VServer">
        <!-- Virtual Server Configuration -->
        <vnsParamInst name="port" key="port" value="8010"/>
        <vnsParamInst name="vip" key="vip" value="10.10.10.100"/>
       <vnsParamInst name="vservername" key="vservername" value="crpvgrtst02-vip-8010"/>

        <vnsParamInst name="servicename" key="servicename" value="crpvgrtst02-8010"/>
        <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="any" nodeNameOrLbl="any"
          key="VServerGlobalConfig" name="VServerGlobalConfig">
          <vnsCfgRelInst name="ServiceConfig" key="ServiceConfig" targetName="Service1"/>

            <vnsCfgRelInst name="VipConfig" key="VipConfig" targetName="Network/vip"/>
        </vnsFolderInst>
    </vnsFolderInst>
</fvAEPg>
```

パラメータ解決

グラフ インスタンス作成時に、Application Policy Infrastructure Controller（APIC）はサービス グラフの各機能に対してコンフィギュレーション パラメータを解決します。解決が完了すると、パラメータ値がデバイス スクリプトに渡されます。デバイス スクリプトはこれらのパラメータ値を使用してサービス アプライアンス上でサービスを設定します。

次のフロー チャートは、パラメータの解決手順について説明しています。

図 *39* : パラメータ解決



パラメータ解決時の MO の検索

Application Policy Infrastructure Controller（APIC）は、コンフィギュレーション パラメータを取得する適切なコンフィギュレーション MO の検出に 2 つの主なコンストラクトを使用します。

RsScopeToTerm

機能ノードまたは AbsFolder に対する RsScopeToTerm 関係は、グラフに対するパラメータを持つコンフィギュレーション MO と接続されるサービス グラフの端末ノードを示します。APIC は、グラフ コンフィギュレーション パラメータを検出するために、RsScopeToTerm 内の指定の端末ノードと接続されたコンフィギュレーション MO を使用します。

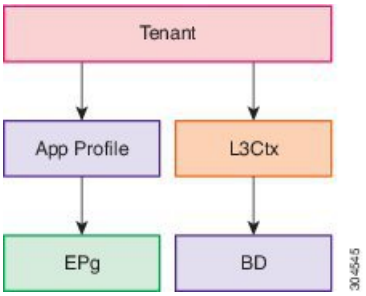
指定された RsScopeToTerm コンフィギュレーションがない場合、APIC はデフォルトでプロバイダー EPG に接続された端末を使用します。

ScopedBy 属性

ScopedBy 属性はパラメータの解決に使用する開始 MO の検出に使用されます。たとえば、scopedBy に「EPG」の値がある場合、APIC はエンドポイント グループからパラメータ解決を開始します。APIC は、階層を上ってパラメータを解決し、アプリケーション プロファイルの次にテナントに上ってコンフィギュレーション パラメータを解決します。

次の図は、APIC MO の階層を示しています。

図 *40 : APIC MO* の階層



ロールベースのアクセスコントロールルールの拡張について

マルチテナント環境でのレイヤ 4 ～ レイヤ 7 設定では、従来のロールベースのアクセス コントロール（RBAC）ドメインとロールモデルの定義を使用してテナント管理者が作成できない特定のオブジェクトを作成するには、管理者が介入する必要がありました。Application Policy Infrastructure Controller（APIC）では、オブジェクトの作成に必要な権限をテナント管理者に付与できるように、管理情報ツリー（MIT）でRBAC権限をより詳細に指定できます。また、テナント管理者は、管理者の介入なしに、セルフサービスを介してRBACルールを作成し、テナントサブツリーの下にあるリソースの権限をシステム内の他のテナントやユーザに付与することもできます。

ロールベースのアクセス コントロール ルールのアーキテクチャ

ロールベースのアクセス コントロール（RBAC）ルールには、ロールベースのアクセス コントロール（RBAC）モデルを強化して加筆性ルールを許可するブール型の`allowWrites`フィールドがあります。`allowWrites`フィールドがない場合に定義できるのは、読み取り RBAC ルールのみになります。

`RbacRule` クラスは次のように定義します。

```
Class aaa:RbacRule (CONCRETE)
Encrypted: false
Exportable: true
Persistent: true
Configurable: true
Write Access: [aaa, admin]
Read Access: [aaa, admin]
```

RBACルールにより、ユーザはセキュリティ ドメインから特定のオブジェクトで始まるサブツリーを読み取ることができます。

```
DN FORMAT:   [1] uni/rbacdb/rule-{[objectDn]}-dom-{domain}
```

表 *2 : aaa:RbacRule* プロパティの概要

プロパティ	タイプ	クラス	説明
aaa:Boolean	scalar:Enum8	allowWrites（aaa:RbacRule:allowWrites）	読み取り/書き込みまたは読み取りルール。
naming:Name	string:Basic	domain（aaa:RbacRule:domain）	カウントオブジェクトのドメイン。`aaa:ARbacRule:domain` を無効にします。
reference:BinRef		objectDn（aaa:RbacRule:objectDn）	`aaa:ARbacRule:objectDn` を無効にします。

`PartialRbacRule` クラスは `fvTenant` クラスの下に定義され、テナントがRBACルール（セルフサービス）を作成できるようにします。`PartialRbacRule` クラスは次のように定義されます。

```
Class aaa:PartialRbacRule (CONCRETE)
Encrypted: false
Exportable: true
Persistent: true
Configurable: true
Write Access: [aaa, admin]
Read Access: [aaa, admin]
```

表 *3 : aaa:PartialRbacRule* プロパティの概要

プロパティ	タイプ	クラス	説明
aaa:Boolean	scalar:Enum8	allowWrites（aaa:PartialRbacRule:allowWrites）	読み取り/書き込みまたは読み取りルール。
naming:Name	string:Basic	domain（aaa:PartialRbacRule:domain）	カウントオブジェクトのドメイン。
reference:BinRef		monPolDn（aaa:PartialRbacRule:monPolDn）	この監視可能なオブジェクトにアタッチするモニタリングポリシー。
reference:BinRef		partialObjectDn（aaa:PartialRbacRule:partialObjectDn）	

テナントによる `PartialRbacRule` クラスの作成では、`partialObjectDn` の正当性を確認する必要があります。`partialObjectDn` がテナント サブツリーの下にあれば有効です。親テナント サブツリー外の識別名は許可されていません。

管理者は、システム内の識別名を指す RbacRule を作成できます。テナント管理者が作成できるのは、テナント管理者のテナント サブツリー内にある識別名を指す PartialRbacRule のみです。

ロールベース アクセス コントロール ルールのシステム フロー

レイヤ4～レイヤ7ポリシーの設定前、設定中、または設定後に、テナント管理者は、特定のファイアウォールとロード バランサ デバイスへのアクセス権を自分のテナント ユーザに付与する PartialRbacRule の作成を選択することができます。各リソース グループを表す aaaDomain を作成し、個々に割り当てることでアクセスが実現します。次にセットアップの例を示します。

テナント	Acme
ユーザ	acme-admin acme-firewall-1-admin acme-firewall-2-admin acme-loadbalancer-1-admin acme-loadbalancer-2-admin
ファイアウォール デバイス	Firewall1 Firewall2
ロード バランサ デバイス	LB1 LB2

テナント管理者ユーザの acme-admin は、デバイスの Firewall1、Firewall2、LB1、および LB2 を作成したいと考えています。各デバイスに対する完全な書き込みアクセス許可をユーザごとに割り当てる必要があります。たとえば、ユーザ acme-firewall-1-admin にはデバイス Firewall1 ポリシーへの書き込み権限のみが必要ですが、ユーザ acme-loadbalancer-1-admin にはデバイス LB1 ポリシーへの書き込み権限のみが必要です。これを実現するには、acme-admin ユーザが、次のアクセス権を付与する 4 つの PartialRbacRule を作成する必要があります。

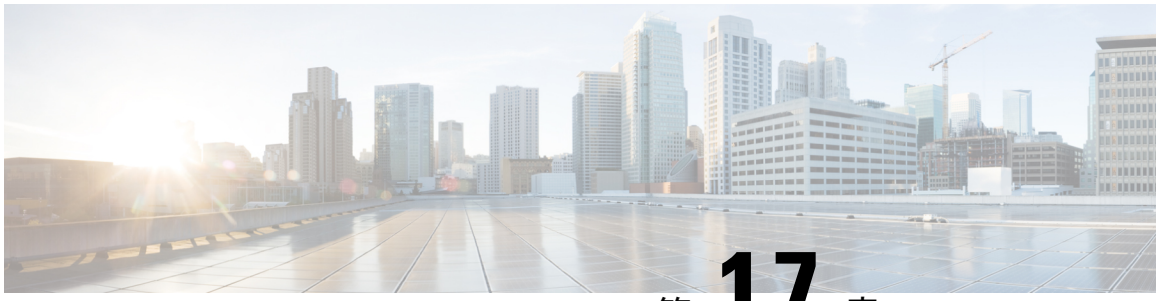
- Firewall1 識別名：ドメイン acme-firewall1 による書き込みが可能
- Firewall2 識別名：ドメイン acme-firewall2 による書き込みが可能
- LB1 識別名：ドメイン acme-lb1 による書き込みが可能
- LB2 識別名：ドメイン acme-lb2 による書き込みが可能

ユーザには次の権限が割り当てられます。

- ユーザ：acme-firewall-1-admin
  - ドメイン acme：read-all 権限
  - ドメイン acme-firewall1：テナント管理/書き込み

- ユーザ：acme-firewall-2-admin
  - ドメイン acme：read-all 権限
  - ドメイン acme-firewall2：テナント管理/書き込み
- ユーザ：acme-lb-1-admin
  - ドメイン acme：read-all 権限
  - ドメイン acme-lb1：テナント管理/書き込み
- ユーザ：acme-lb-2-admin
  - ドメイン acme：read-all 権限
  - ドメイン acme-lb2：テナント管理/書き込み

上記 4 人のユーザのいずれも、ドメイン acme の権限によって、acme テナント サブツリーを読み取れますが、どのノードにも書き込めません。テナント acme-lb2 のテナント管理/書き込みの権限によって、ユーザは LB2 ポリシー サブツリーのみに書き込むことができます。



第 17 章

サービス グラフ テンプレートの使用

GUI を使用したサービス グラフ テンプレートとコントラクトおよび EPG の関連付け

GUI を使用して、サービス グラフ テンプレートをコントラクトとエンドポイント グループ（EPG）に関連付ける必要があります。

（注） サービス グラフ テンプレートをコントラクトと EPG に関連付けるには、GUI のみを使用できます。

サービス グラフ テンプレートをコントラクトと EPG に関連付ける手順については、GUI の使用方法 （237 ページ） を参照してください。

NX-OS スタイルの CLI を使用したサービス グラフ テンプレートの作成

次に、サービス グラフ テンプレートの作成手順を示します。

ステップ 1 コンフィギュレーション モードを開始します。

例：

```
apic1# configure
```

ステップ 2 テナントのコンフィギュレーション モードを開始します。

```
tenant tenant_name
```

例：

```
apic1(config)# tenant t1
```

ステップ 3 サービス グラフをテンプレートに関連付けます。

```
l4l7 graph graph_name contract contract_name
```

パラメータ	説明
グラフ	サービス グラフ テンプレートの名前。
contract	サービス グラフ テンプレートで使用するコントラクトの名前。

例：

```
apic1(config-tenant)# l4l7 graph GraphL3asa contract ContractL3ASA
```

ステップ 4 機能ノードを追加します。

```
service node_name [device-cluster-tenant tenant_name] [device-cluster device_name] [mode
deployment_mode]
```

パラメータ	説明
service	追加するサービス ノードの名前。
device-cluster-tenant	デバイス クラスタのインポート元のテナント。グラフを設定するテナントと同じテナントにデバイス クラスタがない場合にのみ、このパラメータを指定します。
device-cluster	このサービス ノードに使用するデバイス クラスタの名前。
mode	導入モード。値は次のとおりです。 • ADC_ONE_ARM：ワンアーム モードを指定します。 • ADC_TWO_ARM：ツーアーム モードを指定します。 • FW_ROUTED：ルーテッド（GoTo）モードを指定します。 • FW_TRANS：トランスペアレント（GoThrough）モードを指定します。 • OTHERS モードを指定しないと、導入モードは使用されません。

例：

```
apic1(config-graph)# service Node1 device-cluster-tenant common device-cluster ifav108-asa-2 mode
 FW_ROUTED
```

ステップ 5 コンシューマ コネクタを追加します。

```
connector connector_type [cluster-interface interface_type]
```

パラメータ	説明
コネクタ	サービス グラフ内のコネクタのタイプ。値は次のとおりです。 • provider • consumer
cluster-interface	デバイス クラスタ インターフェイスのタイプ。値は次のとおりです。 • provider • consumer テナント Common 内のサービス グラフ テンプレートの場合は、このパラメータを指定しないでください。

例：

```
apic1(config-service)# connector consumer cluster-interface consumer
```

ステップ 6 テナントをコネクタに関連付け、コネクタ コンフィギュレーション モードを終了します。

```
l4l7-peer tenant tenant_name out L3OutExternal epg epg_name
  redistribute redistribute_property
exit
```

パラメータ	説明
テナント	コネクタに関連付けるテナントの名前。
out	レイヤ 3 Outside の名前。
epg	エンドポイント グループの名前。
redistribute	再配布プロトコルのプロパティ。

例：

```
apic1(config-connector)# l4l7-peer tenant t1 out L3OutExternal epg L3ExtNet
  redistribute connected,ospf
apic1(config-connector)# exit
```

ステップ 7 プロバイダーに対してステップ 5 と 6 を繰り返します。

例：

```
apic1(config-service)# connector provider cluster-interface provider
apic1(config-connector)# l4l7-peer tenant t1 out L3OutInternal epg L3IntNet
  redistribute connected,ospf
apic1(config-connector)# exit
```

ステップ 8 （任意） ルータを追加し、ノード コンフィギュレーション モードを終了します。

```
rtr-cfg router_ID
exit
```

パラメータ	説明
rtr-cfg	ルータの ID。

テナント Common でサービス グラフ テンプレートを作成する場合は、この手順をスキップします。

例：

```
apic1(config-service)# rtr-cfg router-id1
apic1(config-service)# exit
```

ステップ 9 1 つの接続をコンシューマ コネクタに、もう 1 つをプロバイダー コネクタに関連付けてから、サービス グラフ コンフィギュレーション モードを終了します。

```
connection connection_name terminal terminal_type service node_name
  connector connector_type
exit
```

パラメータ	説明
connection	コネクタに関連付ける接続の名前。
terminal	端末のタイプ。値は次のとおりです。 • provider • consumer
service	サービス グラフのノードの名前。
コネクタ	コネクタのタイプ。値は次のとおりです。 • provider • consumer

例：

```
apic1(config-graph)# connection C1 terminal consumer service Node1 connector consumer
apic1(config-graph)# connection C2 terminal provider service Node1 connector provider
apic1(config-graph)# exit
```

ステップ 10 コンフィギュレーション モードを終了します。

例：

```
apic1(config-tenant)# exit
apic1(config)# exit
```

REST API を使用したサービス グラフ テンプレートの設定

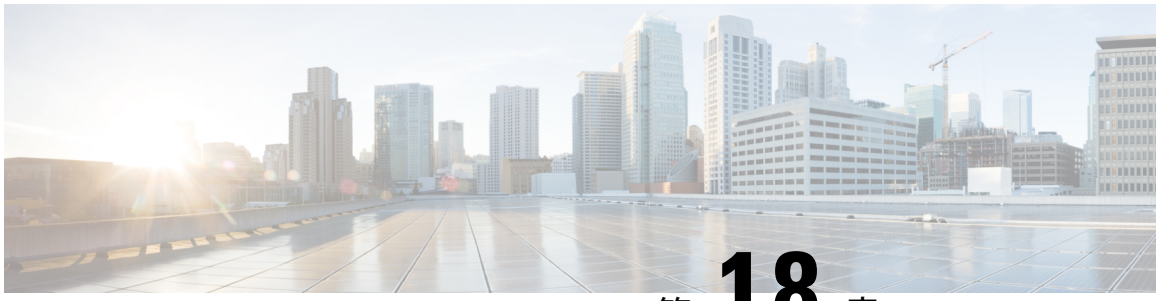
次の REST API を使用してサービス グラフ テンプレートを設定できます。

```
<polUni>
    <fvTenant dn="uni/tn-acme" name="acme">
      <!—L3 Network-->
      <fvCtx name="MyNetwork"/>
        <!-- Bridge Domain for MySrvr EPG -->
        <fvBD name="MySrvrBD">
           <fvRsCtx tnFvCtxName="MyNetwork" />
           <fvSubnet ip="10.10.10.10/24">
           </fvSubnet>
        </fvBD>
        <!-- Bridge Domain for MyClnt EPG -->
        <fvBD name="MyClntBD">
          <fvRsCtx tnFvCtxName="MyNetwork" />
          <fvSubnet ip="20.20.20.20/24">
          </fvSubnet>
        </fvBD>
        <fvAp dn="uni/tn-acme/ap-MyAP" name="MyAP">
            <fvAEPg dn="uni/tn-acme/ap-MyAP/epg-MyClnt" name="MyClnt">
                <fvRsBd tnFvBDName="MySrvrBD" />
                <fvRsDomAtt tDn="uni/vmmp-Vendor1/dom-MyVMs" />
                <fvRsProv tnVzBrCPName="webCtrct">
                </fvRsProv>
                <fvRsPathAtt tDn="topology/pod-1/paths-17/pathep-[eth1/21]"
encap="vlan-202"/>
                <fvRsPathAtt tDn="topology/pod-1/paths-18/pathep-[eth1/21]"
encap="vlan-202"/>
            </fvAEPg>
            <fvAEPg dn="uni/tn-acme/ap-MyAP/epg-MySRVR" name="MySRVR">
                <fvRsBd tnFvBDName="MyClntBD" />
                <fvRsDomAtt tDn="uni/vmmp-Vendor1/dom-MyVMs" />
                <fvRsCons tnVzBrCPName="webCtrct">
                </fvRsCons>
                <fvRsPathAtt tDn="topology/pod-1/paths-17/pathep-[eth1/21]"
encap="vlan-203"/>
                <fvRsPathAtt tDn="topology/pod-1/paths-18/pathep-[eth1/21]"
encap="vlan-203"/>
            </fvAEPg>
        </fvAp>
    </fvTenant>
</polUni>
```

REST API を使用したセキュリティ ポリシーの作成

次の REST API を使用してセキュリティ ポリシーを作成することができます。

```
<polUni>
    <fvTenant dn="uni/tn-acme" name="acme">
        <vzFilter name="HttpIn">
            <vzEntry name="e1" prot="6" dToPort="80"/>
        </vzFilter>
        <vzBrCP name="webCtrct">
            <vzSubj name="http">
                <vzRsSubjFiltAtt tnVzFilterName="HttpIn"/>
            </vzSubj>
        </vzBrCP>
    </fvTenant>
</polUni>
```



第 18 章

サービス グラフのモニタリング

GUI を使用したサービス グラフ インスタンスのモニタリング

サービス グラフ テンプレートを設定し、エンドポイント グループ（EPG）およびコントラクトにグラフをアタッチした後は、サービス グラフ インスタンスをモニタできます。モニタリングには、グラフ インスタンスの状態、グラフ インスタンスの機能、機能に割り当てられたリソース、および機能に指定されたパラメータの表示が含まれます。

ステップ 1 メニュー バーで、**[Tenants]** > **[All Tenants]** の順に選択します。

ステップ 2 [Work] ペインで、サービス グラフをモニタするテナントの名前をダブルクリックします。

ステップ 3 **[Navigation]** ペインで、**[Tenant]** ***tenant_name*** > **[Services]** > **[L4-L7]** > **[Deployed Graph Instances]** の順で選択します。[Work] ペインは、アクティブなサービス グラフ インスタンスに関する次の情報を表示します。

名前	説明
[Service Graph] カラム	サービス グラフ テンプレートの名前。
[Contract] カラム	サービス グラフ テンプレートに表示されるコントラクトの名前。
[Contained By] カラム	サービス グラフ テンプレートを含むネットワークの名前。

名前	説明
[State] カラム	サービス グラフ テンプレートの状態。[applied] の状態は、グラフが適用され、グラフ ポリシーがファブリックおよびサービス デバイス内でアクティブであることを意味します。
[Description] カラム	サービス グラフの説明

ステップ 4 [Deployed Service Graphs] ブランチを展開します。アクティブなサービス グラフ インスタンスがブランチの下にリストされます。

ステップ 5 サービス グラフ インスタンスをクリックして、[Work] ペインにそのインスタンスに関する追加情報を表示します。デフォルト ビューはグラフのトポロジです。[Work] ペインのタブのいずれかをクリックして、そのグラフのビューを変更できます。

ステップ 6 グラフ インスタンスのいずれかのブランチを展開します。グラフ インスタンスの機能は、インスタンスの下に表示されます。

ステップ 7 機能をクリックして、[Work] ペインにその機能に関する追加情報を表示します。デフォルト ビューはその機能のポリシーです。[Work] ペインのタブのいずれかをクリックして、その機能のビューを変更できます。[Work] ペインには、ポリシーに関する次の情報が表示されます。

名前	説明
[POLICY] タブ	機能のプロパティ、機能に割り当てられたリソース、および機能のパラメータ。
[FAULTS] タブ	機能ノードで生じている問題。
[HISTORY] タブ	機能ノードで発生したイベントの履歴。

ステップ 8 [Navigation] ペインで、[Deployed Device] をクリックします。[Work] ペインにデバイスのインスタンスに関する情報が表示されます。

GUI を使用したサービス グラフ エラーのモニタリング

サービス グラフ テンプレートを設定し、エンドポイント グループ（EPG）およびコントラクトにグラフをアタッチした後は、サービス グラフ テンプレートのエラーをモニタできます。

ステップ 1 メニュー バーで、**[Tenants]** > **[All Tenants]** の順に選択します。

ステップ 2 [Work] ペインで、サービス グラフをモニタするテナントの名前をダブルクリックします。

ステップ 3 [Navigation] ウィンドウで、**Tenant *tenant_name*** > **Services** > **L4-L7** > **Deployed Graph Instances** を選択します。

ステップ 4 エラーを表示するグラフ インスタンスのブランチを展開します。グラフ インスタンスの機能は、インスタンスの下に表示されます。

ステップ 5 機能のいずれかをクリックします。デフォルトで、[Work] ペインはその機能のポリシーを示します。

ステップ 6 [Work] ペインの [FAULTS] タブをクリックします。[Work] ペインが機能ノードのエラーを表示します。

サービス グラフ エラーの解決

1 つ以上のサービス グラフ テンプレート エラーを発見した場合、問題の解決はエラーによって異なります。次の表は、エラーの説明とエラーを解決する方法を説明しています。

表 *4 :* コネクタのエラー

Fault	CLI ラベル	説明と解決法
missing-connection	connection associated with a connector not found	グラフ コネクタの設定が無効です。コネクタに関連付けられた接続が見つかりませんでした。
missing-nodeinst	NodeInst associated with a connector not found	グラフ コネクタの設定が無効です。コネクタに関連付けられた NodeInst が見つかりませんでした。
conn-nonrenderable	Graph connector could not be rendered.	グラフ コネクタの設定が無効です。グラフをレンダリングできませんでした。
invalid-bd	BD associated with a connector is not valid	グラフ コネクタの設定が無効です。コネクタの関連ブリッジ ドメインが無効です。
invalid-ctx	Ctx associated with a connector is not valid.	グラフ コネクタの設定が無効です。コネクタの関連する Ctx が無効です。
missing-peer-conn	Peer connector associated with a connector not found.	グラフ コネクタの設定が無効です。接続のピア コネクタが見つかりませんでした。

表 *5 : AbsGraph* および *GraphInst* エラー

Fault	CLI ラベル	説明と解決法
invalid-abstract-graph-config	invalid abstract graph config	抽象グラフ設定が無効です。

Fault	CLI ラベル	説明と解決法
missing-mandatory-param	mandatory param not found	必要な設定パラメータが解決できませんでした。パッケージの必須パラメータをチェックし、AbsGraph にパラメータがあることを確認します。
param-cardinality-error	invalid param cardinality	コンフィギュレーション パラメータは、濃度の要件を満たしていません。cardinality=n を指定しないでパラメータの複数のインスタンスが指定されているかどうかを確認します。
epp-download-failure	epp download failure	グラフ ポリシーがスイッチのダウンロードに失敗しました。
param-duplicate-name-failure	duplicate param name	同じ名前のパラメータの複数の同一コピーが検出されました。
id-allocation-failure	id allocation failure	一意のネットワーク リソース（VLAN VXLAN）を割り当てることができませんでした。
missing-ldev	No cluster found	クラスタが見つかりませんでした。
context-cardinality-violation-failure	invalid cluster context cardinality	クラスタは必要なテナント機能（マルチテナントまたはシングル テナント）をサポートしていません。
function-type-mismatch-failure	invalid function type	機能タイプが選択したデバイスでサポートされていません。AbsNode 機能タイプと解決された LDevVip 機能タイプが一致するか確認します。
invalid-abstract-graph-config-param	invalid abstract graph config param	抽象グラフ コンフィギュレーション パラメータが無効です。
missing-mparam	No parameter definition found	必要なパラメータ定義が見つかりませんでした。
missing-abs-graph	no abs graph found	抽象グラフ設定がグラフ インスタンスにありません。

Fault	CLI ラベル	説明と解決法
invalid-param-config	invalid param config	パラメータ設定が無効です。
invalid-param-scope	invalid parameter scope	パラメータ スコープが無効です。AbsGraph の vnsRsScopeToTerm パラメータが正しいかどうか確認します。
invalid-ldev	Invalid cluster	クラスタ設定が無効です。解決した LDevVip のステータスを確認して、エラーを解決します。
missing-tenant	no tenant found	グラフに対してテナントが見つかりませんでした。
internal-error	internal error	内部エラーがグラフ処理中に発生しました。
resource-allocation-failure	resource allocation failure	グラフ処理中に必要なリソースを割り当てることができませんでした。
missing-abs-function	no abstract function found	抽象機能の定義が見つかりません。
param-validation-failed	param validation failure	コンフィギュレーション パラメータ値が無効です。
missing-mconn	No connector found	必要なコネクタが見つかりませんでした。
cdev-missing-mgmt-ip	no mgmt ip found for cdev	具象デバイスに対して管理 IP アドレスが見つかりませんでした。vnsCMgmt が解決する vnsCDev に存在するかどうかを確認します。
invalid-graphinst	invalid graphinst config	グラフ インスタンスが無効です。
missing-interface	no interface found	インターフェイスが見つかりませんでした。
missing-bd	no bd found	ブリッジ ドメインが見つかりませんでした。
missing-terminal	Terminal node is missing a terminal	端末ノードに端末がありません。端末ノードの設定を確認してください。

Fault	CLI ラベル	説明と解決法
missing-namespace	no vlan/vxlan namespace found	VLAN または VXLAN の割り当てに必要なネームスペースが見つかりません。解決された fvnsVlanInstp と関係がある phyDomp パラメータまたは vmmDomp パラメータが解決された vnsLDevVip に設定されていることを確認します。
missing-mfunc	No function found in device package	デバイス パッケージで必要な機能が見つかりません。パッケージ内にすべての AbsNode 機能タイプがあることを確認します。
missing-lif	no cluster interface found	必要なクラスタ インターフェイスが見つかりませんでした。vnsLDevVip の vnsLIf パラメータが正しく設定されていることを確認します。
invalid-absfunc-profile	Abstract Function Profile config is invalid	抽象機能のプロファイル設定が無効です。このエラーは、プロファイルで指定されている無効なコンフィギュレーション パラメータが要因として考えられます。
missing-cdev	No device found	具象デバイスがクラスタ内に見つかりませんでした。有効な vnsCDev が解決された vnsLDevVip の下に存在することを確認してください。
inappropriate-devfolder	Illegal folder in configuration	対応するフォルダがデバイス パッケージで見つかりませんでした。
invalid-devctx	Device context is not legal for this folder	デバイス パッケージではこのフォルダにデバイス コンテキストを指定することはできません。

Fault	CLI ラベル	説明と解決法
insufficient-devctx	Folder must have one value for each associated CDev	フォルダは具象デバイスに固有です。フォルダは、各具象デバイスに対して少なくとも 1 つの値を持つ必要があります。
cdev-missing-cif	No interface defined	具象デバイスには少なくとも 1 つのインターフェイスを定義する必要があります。
cdev-missing-pathinfo	Missing path for interface	物理サービス アプライアンスでは、インターフェイスがどのリーフ ポートに接続されているかを把握する必要があります。vnsCifPathAtt パラメータが、解決された vnsCDev の下のすべての vnsCIf に存在することを確認します。
missing-cif	Device interfaces does not match cluster	デバイス インターフェイスは、クラスタに設定されているインターフェイスに一致させる必要があります。vnsCIf パラメータおよび vnsLIf パラメータが、解決された vnsLDevVip の下に存在することを確認します。
ldevvip-missing-mgmt-ip	No Mgmt ip found for LDevVip	LDevVip に対して管理 IP アドレスが見つかりませんでした。
lif-invalid-MIf	LIf has an invalid MifLbl	LIf に含まれる MIfLbl がデバイス パッケージに存在しません。
lif-invalid-CIf	LIf has an invalid CIf	LIf に含まれる CIf がありません。具象デバイスおよび CIf の設定を確認します。
missing-function-node	Abstract graph missing function node	抽象グラフには、少なくとも 1 つの機能ノードが存在する必要があります。
graph-loop-detected	Abstract graph config has a loop	抽象グラフ設定が無効です。設定にループがあります。

Fault	CLI ラベル	説明と解決法
gothrough-routing-enabled-both	Both the legs of go through node has routing enabled	通過ノードの両方のレッグでルーティングが有効になっています。
invalid-terminal-nodes	Abstract graph has invalid number of terminal nodes	抽象グラフは少なくとも 2 つの端末ノードを持つ必要があります。
missing-ldev-ctx	No device context found for LDev	デバイスのデバイス コンテキストが見つかりませんでした。vnsLDevCtx にコントラクト、グラフおよびノードに一致する値があることを確認します。
arp-flood-enabled	ARP flood is enabled on the management end point group	ARP フラッディングは管理エンドポイントのグループに対して無効です。
folderinst-validation-failed	FolderInst has key, that is not found in MFolder	FolderInst のキーおよび値は MFolder 仕様を尊重する必要があります。
paraminst-validation-failed	ParamInst has key and/or value, that are not found in MParam	ParamInst のキーおよび値は MParam 仕様を尊重する必要があります。
invalid-mfolder	FolderInst points to an invalid MFolder	FolderInst は有効な MFolder をポイントする必要があります。
invalid-mparam	ParamInst points to an invalid MParam	ParamInst は有効な MParam をポイントする必要があります。
devfolder-validation-failed	DevFolder has key, that is not found in MFolder	DevFolders のキーおよび値は MFolder 仕様を尊重する必要があります。
devparam-validation-failed	DevParam has key and/or value, that are not found in MParam	DevParam のキーおよび値は MParam 仕様を尊重する必要があります。
cdev-missing-virtual-info	Virtual Object Info is missing in CDev	LDevVip のタイプが Virtual の場合は仮想オブジェクト情報を指定する必要があります。

Fault	CLI ラベル	説明と解決法
invalid-rsmconnatt	Relationship to metaconnector is invalid	メタコネクタのDNを修正し、正しい MDev 階層にバインドすることを確認します。

GUI を使用した仮想デバイスのモニタリング

サービス グラフ テンプレートを設定し、エンドポイント グループ（EPG）およびコントラクトにグラフをアタッチした後は、テナントの仮想デバイスをモニタできます。仮想デバイスをモニタリングすると、どのデバイスが使用中か、どの VLAN がデバイス用に設定されているかや、デバイスに渡されるパラメータ、デバイスの統計、およびデバイスの健全性を確認できます。

ステップ 1 メニュー バーで、**[Tenants]** > **[All Tenants]** の順に選択します。

ステップ 2 [Work] ペインで、サービス グラフをモニタするテナントの名前をダブルクリックします。

ステップ 3 ナビゲーション ペインで、次のように選択します。 **テナント** ***tenant_name*** > **サービス** > **L4 L7** > **デバイスの導入** 。

ステップ 4 導入されたデバイスのいずれかをクリックします。デフォルトでは、[Work] ペインに導入済みのデバイスのポリシーが表示されます。ビューを変更するには、[Work] ペインのタブをクリックします。タブは、仮想デバイスに関する以下の情報を表示します。

タブ	説明
[POLICY] タブ	使用中のデバイス、デバイス内で設定されたVLAN、およびデバイスに渡されたパラメータ。
[OPERATIONAL] タブ	さまざまなデバイスから受信する統計情報。
[HEALTH] タブ	デバイスの状態。

NX-OSスタイルのCLIを使用したデバイスクラスタとサービス グラフ ステータスのモニタリング

この項のコマンドで、NX-OS スタイルの CLI を使用してデバイス クラスタとサービス グラフ ステータスをモニタする例を示します。

デバイス クラスタの動作情報の表示

次に、デバイス クラスタの動作情報を表示するコマンドを示します。

```
show l4l7-cluster tenant tenant_name cluster device_cluster_name
```

例：

```
apic1# show l4l7-cluster tenant HA_Tenant1 cluster Firewall
tenant-graph : HA_Tenant1-g2,HA_Tenant1-g1


Device Cluster    : Firewall
Cluster Interface : consumer1
Encap             : vlan-501
Pctag             : 32773
Devices           : FW2(int),FW1(int)
Graphs            : HA_Tenant1-g1
Contracts         : HA_Tenant1-c1


Device Cluster    : Firewall
Cluster Interface : provider1
Encap             : vlan-502
Pctag             : 32774
Devices           : FW2(ext),FW1(ext)
Graphs            : HA_Tenant1-g1
Contracts         : HA_Tenant1-c1
```

デバイス クラスタの動作ステータスの表示

次に、デバイス クラスタの動作ステータスを表示するコマンドを示します。

```
apic1# show l4l7-graph tenant tenant_name [graph graph_name]
```

例：

次に、HA_Tenant1 テナントのステータスの高レベル出力を提供する例を示します。

```
apic1# show l4l7-graph tenant HA_Tenant1
Graph           : g1
Total Instances : 1
Encaps Used     : vlan-501,vlan-502,vlan-503,vlan-504
Device Used     : uni/tn-HA_Tenant1/lDevVip-Firewall

Graph           : g2
Total Instances : 1
Encaps Used     : vlan-501,vlan-502,vlan-503,vlan-504
Device Used     : uni/tn-HA_Tenant1/lDevVip-Firewall
```

次に、HA_Tenant1 に関連付けられた g1 サービス グラフの詳細出力を提供する例を示します。

```
apic1# show l4l7-graph tenant HA_Tenant1 graph g1
Graph           : HA_Tenant1-g1
Graph Instances : 1

Consumer EPg  : HA_Tenant1-consEPG1
Provider EPg  : HA_Tenant1-provEPG1
Contract Name : HA_Tenant1-c1
Config status : applied

Function Node Name : Node1
 Connector   Encap       Bridge-Domain  Device Interface
 ----------  ----------  ----------     ---------------------
 consumer    vlan-3001   provBD1        consumer
 provider    vlan-3335   consBD1        provider
```

デバイス クラスタのエラーの表示

次に、デバイス クラスタのエラーを表示するコマンドを示します。

```
show faults l4l7-cluster
```

例：

```
apic1# show faults l4l7-cluster
Code            : F0772
Severity        : minor
Last Transition : 2015-09-01T01:41:13.767+00:00
Lifecycle       : soaking-clearing
Affected object : uni/tn-ts1/lDevVip-d1/lIf-ext/fault-F0772
Description     : LIf configuration ext for L4-L7 Devices d1 for tenant ts1
                  is invalid.

Code            : F1085
Severity        : cleared
Last Transition : 2015-09-01T01:39:04.696+00:00
Lifecycle       : retaining
Affected object : uni/tn-ts1/lDevVip-d1/rsmDevAtt/fault-F1085
Description     : Failed to form relation to MO uni/infra/mDev-CiscoInternal-
                  NetworkOnly-1.0 of class vnsMDev

Code            : F1690
Severity        : minor
Last Transition : 2015-09-01T01:39:04.676+00:00
Lifecycle       : soaking
Affected object : uni/tn-ts1/lDevVip-d1/vnsConfIssue-missing-
                  namespace/fault-F1690
Description     : Configuration is invalid due to no vlan/vxlan namespace
                  found
```

サービス グラフのエラーの表示

次に、サービス グラフのエラーを表示するコマンドを示します。

```
show faults l4l7-graph
```

例：

```
apic1# show faults l4l7-graph
Code            : F1690
Severity        : minor
Last Transition : 2015-11-25T20:07:33.635+00:00
Lifecycle       : raised
DN              : uni/tn-HA_Tenant1/AbsGraph-WebGraph/vnsConfIssue-invalid-
                  abstract-graph-config-param/fault-F1690
Description     : Configuration is invalid due to invalid abstract graph
                  config param
```

デバイス クラスタの実行コンフィギュレーションの表示

次に、デバイス クラスタの実行コンフィギュレーションを表示するコマンドを示します。

```
show running-config tenant tenant_name l4l7 cluster
```

例：

```
apic1# show running-config tenant common l4l7 cluster
# Command: show running-config tenant common l4l7 cluster
# Time: Thu Nov 26 00:35:59 2015
  tenant common
```

```
   l4l7 cluster name ifav108-asa type physical vlan-domain phyDom5 service FW function
 go-through
      cluster-device C1
      cluster-interface consumer_1
        member device C1 device-interface port-channel1
          interface vpc VPCPolASA leaf 103 104
          exit
        exit
      cluster-interface provider_1
        member device C1 device-interface port-channel1
          interface vpc VPCPolASA leaf 103 104
          exit
        exit
      exit
```

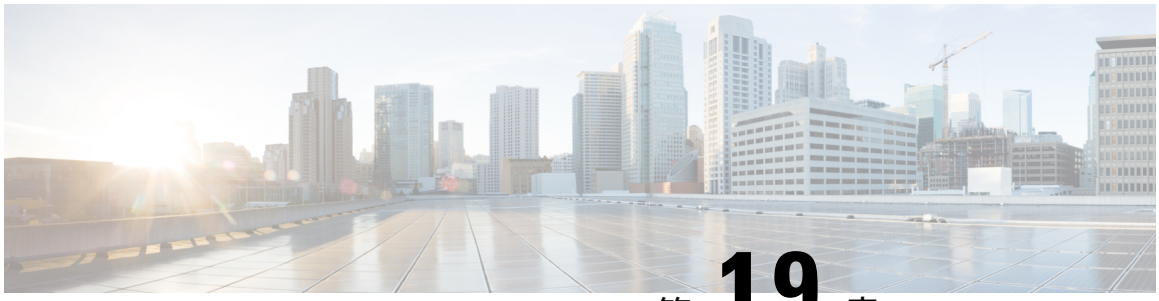
サービス グラフの実行コンフィギュレーションの表示

次に、サービス グラフの実行コンフィギュレーションを表示するコマンドを示します。

```
show running-config tenant tenant_name l4l7 graph
```

例：

```
apic1# show running-config tenant common l4l7 graph
# Command: show running-config tenant common l4l7 graph
# Time: Thu Nov 26 00:35:59 2015
  tenant T1
    l4l7 graph Graph-Citrix contract Contract-Citrix
      service N1 device-cluster-tenant common device-cluster ifav108-citrix mode
ADC_ONE_ARM
        connector provider cluster-interface pro
          bridge-domain tenant common name BD4-Common
          exit
        connector consumer cluster-interface pro
          bridge-domain tenant common name BD4-Common
          exit
        exit
      connection C1 terminal consumer service N1 connector consumer
      connection C2 terminal provider service N1 connector provider
      exit
```



第 19 章

多層アプリケーションとサービス グラフの設定

多層アプリケーションとサービス グラフについて

[Multi-Tier Application with Service Graph Quick Start] ダイアログは、ブリッジ ドメイン、EPG、VRF、サービス、契約など、サービス グラフのコンポーネントを構成するための、統一された方法を提供します。Cisco APIC の別々の場所で各オブジェクトを設定しなくても、[Quick Start] ダイアログは、必要な設定を収集し、それらをシンプルで組織的なステップバイ ステップのプロセスにまとめます。

GUI を使用した多階層アプリケーション プロファイルの作成

始める前に

手順を実行中に、使用可能な場合または前に、次のオブジェクトを設定します。

- **テナント:** 手順を実行する前に少なくとも 1 つのテナントを設定します。
- **VMM ドメイン プロファイル:** デバイス仮想サービスを使用すると、レイヤ 7 デバイスのクラスタ (デバイスがホストされる) をレイヤ 4 で、Virtual Machine Manager (VMM) ドメイン プロファイルと、VM を設定します。
- **外部ルーテッド ネットワーク:** 外部ルーテッド ネットワークにサービス デバイスを接続する場合は、(L3Out) ネットワークの外部レイヤ 3 を設定します。

ステップ 1 [Quick Start] の [Multi-Tier Application] ダイアログにアクセスします。

a) メニュー バーで、[Tenant] > [All Tenants] の順にクリックします。
b) [All Tenants] 作業ペインで、テナントの名前をダブルクリックします。
c) [Navigation] ペインで、[Tenant *tenant_name*] > [Quick Start] > [Multi-tier Application] を選択します。
d) [Work] ペインで、[Configure Multi-tier Application] をクリックします。
[Create Application Profile] ダイアログが表示されます。
e) [Start] をクリックします。

ステップ 2 [STEP 2 > EPGs] ダイアログ ボックスで、プロファイルの基本を設定し、ブリッジ ドメインと EPG を設計します。

a) [Application Profile] フィールドで、プロファイルの一意の名前を入力します。
b) （オプション）このプロファイルで 1 個以上のデバイスが仮想である場合は、[VMM Domain Profile] ドロップダウンリストから仮想マシン マネージャ（VMM）ドメイン プロファイルを選択します。

（注） [VMM Domain Profile] ドロップダウン リストで表示および選択されるように、この手順を実行する前に VMM ドメイン プロファイルを作成する必要があります（[Virtual Networking] > [VMM Domains]）。

c) （オプション）コンシューマまたはプロバイダー EPG が外部ルーテッド ネットワークに属している場合は、[Consumer L3 Outside] および [Provider L3 Outside] フィールド（またはいずれか）のドロップダウンリストからネットワークを選択します。

（注） 外部ルーテッド ネットワークが [L3 Outside] ドロップダウンリストに表示されて選択できるように、この手順を実行する前に外部ルーテッド ネットワークを作成する必要があります（[Tenants] > テナント > [Networking] > [External Routed Networks]）。

d) ブリッジ ドメイン ボタンについて、EPG ゲートウェイ IP アドレスが単一の共有サブネットか、 EPG ごとに設定されるかを決定します。

[Shared] を選択した場合、[Shared Gateway IP] フィールドが表示されます。[Per EPG] を選択した場合、手順 f に進みます。

e) [Bridge Domain] ボタンから [Shared] ボタンを選択した場合、[Shared Gateway IP] フィールドの EPG で共有されるゲートウェイの IPv4 アドレスを入力します。
f) アプリケーション階層（EPG）の [Name] フィールドに EPG の名前を入力します。
g) [Bridge Domain] ボタンから [Per EPG] を選択した場合、EPG で使用されるゲートウェイの IPv4アドレスを入力します。[Bridge Domain] ボタンから [Shared] を選択した場合、[Shared Gateway IP] フィールドに入力した IP アドレスが表示されます。
h) （オプション）[+] をクリックし、手順 g に従い別の EPG を追加して EPG を設定します。3 つの EPG が必要な場合はこの手順を繰り返します。
i) [Next] をクリックします。

ステップ 3 [STEP 3 > Services] ダイアログで、必要に応じて、EPG の近隣にあるサービスに含まれるものを設定します。

a) (オプション）[Share same device] ボックスのチェックをオンにして、すべての EPG でファイアウォール ロード バランサを共有します。

b) （オプション）各 EPG の間で、このプロファイルに含むファイアウォール（**FW**）またはロード バランサ（**ADC**）を選択します。
c) （オプション）EPG 間で複数のデバイスを追加する場合は、**< Toggle >** をクリックしてデバイスを再配置します。
d) [Next] をクリックします。

ステップ 4 （ファイアウォールとロード バランサ）**[STEP 4 >]** ダイアログとファイアウォールまたはロード バランサの設定セクションで、サービス デバイスを設定します。

a) **[デバイス タイプ]** ボタンでは、**[物理]** または **[仮想]** を選択します。
b) **[デバイス タイプ]** に **[物理]** を選択した場合、**[物理ドメイン]** ドロップダウン リストからドメインを選択します。**[デバイス タイプ]** に **[仮想]** を選択した場合、**[VMM ドメイン]** ドロップダウン リストおよび **[デバイス 1 VM]** ドロップダウン リストからホストされたデバイスの仮想マシン（VM）からドメインを選択します。
c) **[ノード タイプ]** ボタンでは、**[One-Arm]** または **[Two-Arm]** を選択します。デバイスがコンシューマ コネクタ（one-arm）のみを有するか、コンシューマとプロバイダ（two-arm）を有するか決定します。
d) **[ビュー]** ボタンでは、**[単一ノード]** または **[HA ノード]** を選択します。**[HA ノード]** を選択した場合、2 番目のインターフェイス（物理デバイス）または 2 番目の VNIC（仮想デバイス）がコネクタの設定に含まれており、仮想デバイスでは 2 番目の仮想マシンを選択する必要があります。

ステップ 5 （ファイアウォールのみ）**[STEP 4 >]** ダイアログおよびコンシューマとプロバイダセクションで、ファイアウォール コンシューマとプロバイダ コネクタを設定します。

a) **[IP]** フィールドの物理デバイスでは、ファイアウォール デバイスのレイヤ 4 ～ レイヤ 7 ポリシー ベースのリダイレクト ポリシーにコンシューマ/プロバイダ インターフェイス IP アドレスを入力します。仮想デバイスでは、コンシューマ/プロバイダ インターフェイスの IP アドレスを入力します。
b) **[MAC]** フィールドで、ファイアウォール デバイスのレイヤ 4 ～ レイヤ 7 ポリシー ベースのリダイレクト ポリシーの MAC アドレスを入力します。
c) **[ゲートウェイ IP]** フィールドで、ルート ゲートウェイ IP アドレスを入力します。
d) 物理デバイスでは、**[デバイス 1 インターフェイス]** ドロップダウン リストで、インターフェイスを選択します。仮想デバイスでは、**[デバイス 1 VNIC]** ドロップダウン リストで vNIC を選択します。**[ビュー]** ボタンから **[HA]** ノードを選択した場合、**[デバイス 2 VNIC]** ドロップダウン リストで 2 番目の vNIC を選択する必要があります。
e) （物理デバイスのみ）**[Encap]** フィールドで、インターフェイスのポート カプセル化を入力します。

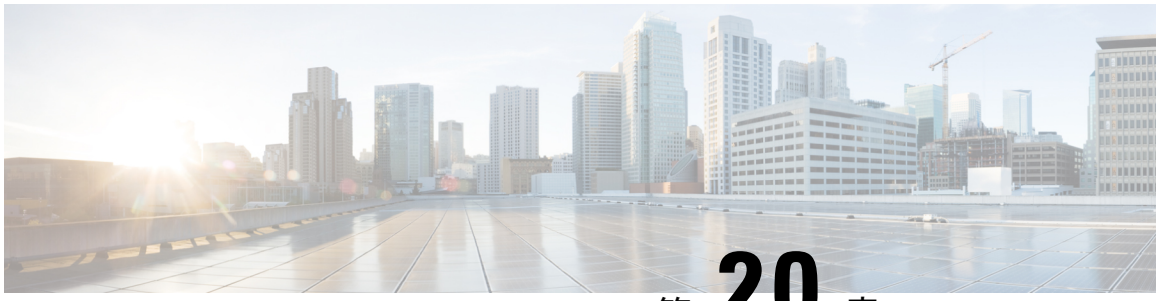
ステップ 6 (ロード バランサのみ)**[手順 4 >]** ダイアログおよびコンシューマとプロバイダ セクションで、ロード バランサ コンシューマおよびプロバイダ コネクタを設定します。

a) **[ゲートウェイ IP]** フィールドで、ルート ゲートウェイ IP アドレスを入力します。
b) 物理デバイスでは、**[デバイス 1 インターフェイス]** ドロップダウン リストで、インターフェイスを選択します。仮想デバイスでは、**[デバイス 1 VNIC]** ドロップダウン リストで vNIC を選択します。**[ビュー]** ボタンから **[HA]** ノードを選択した場合、**[デバイス 2 VNIC]** ドロップダウン リストで 2 番目の vNIC を選択する必要があります。
c) （物理デバイスのみ）**[Encap]** フィールドで、インターフェイスのポート カプセル化を入力します。
d) コネクタで L3 トラフィックを終端させるには、[L3 Destination (VIP)] ボックスをオンのままにします。コネクタが L3 宛先ではない場合はオフにします。

（注） このパラメータのデフォルトは有効（オン）です。ただし、ポリシーベース リダイレクトがインターフェイスで設定されている場合、この設定は考慮されません。

ステップ 7 追加でデバイスを設定する場合は、[Next] をクリックし、各デバイスごとに手順 4 ～ 6 を繰り返します。

ステップ 8 [Finish] をクリックします。



第 20 章

サービス コンフィギュレーションの管理に対する管理ロールの設定

権限について

Application Policy Infrastructure Controller（APIC）で設定したロールに権限を付与できます。権限は、ロールが実行できるタスクを決定します。管理者ロールには次の特権を付与できます。

特権	説明
nw-svc-policy	ネットワーク サービス ポリシー権限では次を実行できます。 • サービス グラフ テンプレートの作成 • アプリケーション エンドポイント グループ（EPG）およびコントラクトへのサービス グラフ テンプレートのアタッチ • サービス グラフのモニタ
nw-svc-device	ネットワーク サービス デバイス権限では次を実行できます。 • デバイスの作成 • 具象デバイスの作成 • デバイス コンテキストの作成

（注） インフラストラクチャ管理者のみがデバイス パッケージを APIC にアップロードできます。

デバイス管理のロールの設定

デバイスを管理するためのロールを有効化するには、そのロールに次の特権を付与する必要があります。

- nw-svc-device

サービス グラフ テンプレート管理のロールの設定

サービス グラフ テンプレートを管理するためのロールを有効化するには、そのロールに次の特権を付与する必要があります。

- nw-svc-policy

デバイス パッケージのアップロードのロールの設定

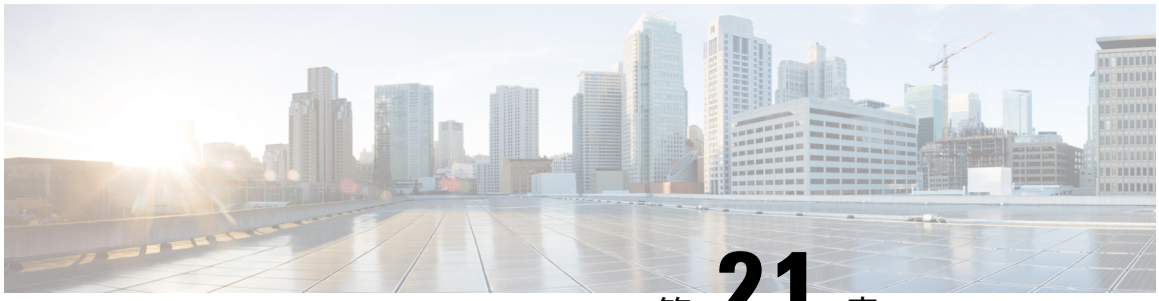
デバイス パッケージは、APIC インフラ管理者特権でのみアップロードできます。インフラ管理者はデバイス パッケージをアップロードします。他のすべてのテナント管理者はデバイス パッケージに対して読み取り専用アクセスを持ちます。テナント管理者は、デバイスパッケージで使用可能なさまざまな機能にアクセスできます。

デバイスをエクスポートするためのロールの設定

デバイスをエクスポートして、テナント間でデバイスを共有することができます。**nw-device** ロールを持つテナントはデバイスを作成できます。デバイスを所有するテナントがこれらを別のテナントと共有する場合、共有には **nw-svc-devshare** 特権が必要です。

nw-svc-devshare 特権を使用すると、テナントはデバイスをエクスポートできます。

（注） インポートされたデバイスを使用できるようにするには、インポートされたデバイスを持つ他のテナントが **nw-svc-policy** 特権を持つ必要があります。



第 21 章

自動化の開発

REST API について

自動化は、Application Policy Infrastructure Controller（APIC）のノースバウンド Representational State Transfer （REST）API を使用します。APIC UI を通じて実行できる処理はすべて、ノースバウンド API を使用した XML ベースの REST POST を使用して実行できます。たとえば、これらの API 経由でのイベントのモニタ、EPG のダイナミックな有効化、およびポリシーの追加などを実行できます。

また、ノースバウンド REST API を使用して、デバイスがオンボードになったことの通知や、エラーをモニタできます。両方のケースで特定のアクションをトリガするイベントをモニタできます。たとえば、特定のアプリケーション層で発生したエラーを検出し、接続の切断がありリーフ ノードがダウンした場合、これらのアプリケーションを他の場所に再展開するアクションをトリガーできます。パケット ドロップが検出された特定のコントラクトがある場合、これらのコントラクトの複数のコピーを特定のアプリケーション上で有効化できます。また、レポートされた問題に基づいて特定のカウンタをモニタできる統計モニタリング ポリシーを使用できます。

APIC ノースバウンド API にサブミットされた XML ファイルを構成する方法については、『*Cisco APIC Layer 4 to Layer 7 Device Package Development Guide*』 を参照してください。

『*Cisco APIC Management Information Model Reference*』 で定義されている次の Python API はノースバウンド API を使用した REST POST コールのサブミットに使用できます。

- `vns:LDevVip`：デバイス クラスタをアップロードします
- `vns:CDev`：デバイスをアップロードします
- `vns:LIf`：論理インターフェイスを作成します
- `vns:AbsGraph`：グラフを作成します
- `vz:BrCP`：契約にグラフを追加します

REST API を使用した自動化の例

ここでは、REST API を使用してタスクを自動化する例を示します。

次の REST 要求は、ブロードキャスト ドメインを持つテナント、レイヤ 3 ネットワーク、アプリケーション エンドポイント グループ、およびアプリケーション プロファイルを作成します。

```
<polUni>
    <fvTenant dn="uni/tn-acme" name="acme">

        <!—L3 Network-->
        <fvCtx name="MyNetwork"/>

        <!-- Bridge Domain for MySrvr EPG -->
        <fvBD name="MySrvrBD">
            <fvRsCtx tnFvCtxName="MyNetwork"/>
            <fvSubnet ip="10.10.10.10/24">
            </fvSubnet>
        </fvBD>

        <!-- Bridge Domain for MyClnt EPG -->
        <fvBD name="MyClntBD">
            <fvRsCtx tnFvCtxName="MyNetwork"/>
            <fvSubnet ip="20.20.20.20/24">
            </fvSubnet>
        </fvBD>

        <fvAp dn="uni/tn-acme/ap-MyAP" name="MyAP">

            <fvAEPg dn="uni/tn-acme/ap-MyAP/epg-MyClnt" name="MyClnt">
                <fvRsBd tnFvBDName="MySrvrBD"/>
                <fvRsDomAtt tDn="uni/vmmp-Vendor1/dom-MyVMs"/>
                <fvRsProv tnVzBrCPName="webCtrct"> </fvRsProv>
                <fvRsPathAtt tDn="topology/pod-1/paths-17/pathep-[eth1/21]"
                  encap="vlan-202"/>
                <fvRsPathAtt tDn="topology/pod-1/paths-18/pathep-[eth1/21]"
                  encap="vlan-202"/>
           </fvAEPg>

            <fvAEPg dn="uni/tn-acme/ap-MyAP/epg-MySRVR" name="MySRVR">
                <fvRsBd tnFvBDName="MyClntBD"/>
                <fvRsDomAtt tDn="uni/vmmp-Vendor1/dom-MyVMs"/>
                <fvRsCons tnVzBrCPName="webCtrct"> </fvRsCons>
                <fvRsPathAtt tDn="topology/pod-1/paths-17/pathep-[eth1/21]"
                  encap="vlan-203"/>
                <fvRsPathAtt tDn="topology/pod-1/paths-18/pathep-[eth1/21]"
                 encap="vlan-203"/>
            </fvAEPg>
        </fvAp>
    </fvTenant>
</polUni>
```

次の REST 要求は VLAN ネームスペースを作成します。

```
<polUni>
    <infraInfra>
        <fvnsVlanInstP name="MyNS" allocMode="dynamic">
            <fvnsEncapBlk name="encap" from="vlan-201" to="vlan-300"/>
        </fvnsVlanInstP>
    </infraInfra>
</polUni>
```

次の REST 要求は VMM ドメインを作成します。

```
<polUni>
    <vmmProvP vendor="Vendor1">
        <vmmDomP name="MyVMs">
            <infraRsVlanNs tDn="uni/infra/vlanns-MyNS-dynamic"/>
            <vmmUsrAccP name="admin" usr="administrator" pwd="in$1eme"/>
            <vmmCtrlrP name="vcenter1" hostOrIp="192.168.64.186">
                <vmmRsAcc tDn="uni/vmmp-Vendor1/dom-MyVMs/usracc-admin"/>
            </vmmCtrlrP>
        </vmmDomP>
    </vmmProvP>
</polUni>
```

次の REST 要求は物理ドメインを作成します。

```
<polUni>
    <physDomP name="phys">
        <infraRsVlanNs tDn="uni/infra/vlanns-MyNS-dynamic"/>
    </physDomP>
</polUni>
```

次の REST 要求は管理対象デバイス クラスタを作成します。

```
<polUni>
    <fvTenant dn="uni/tn-acme" name="acme">
        <vnsLDevVip name="ADCCluster1" contextAware=1>
            <vnsRsMDevAtt tDn="uni/infra/mDev-Acme-ADC-1.0"/>
            <vnsRsDevEpg tDn="uni/tn-acme/ap-services/epg-ifc"/>
            <vnsRsALDevToPhysDomP tDn="uni/phys-phys"/>

            <vnsCMgmt name="devMgmt" host="42.42.42.100" port="80"/>

            <vnsCCred name="username"value="admin"/>

            <vnsCCredSecret name="password" value="admin"/>
        </vnsLDevVip>
    </fvTenant>
</polUni>
```

次の REST 要求は非管理対象デバイス クラスタを作成します。

```
<polUni>
    <fvTenant name="HA_Tenant1">

        <vnsLDevVip name="ADCCluster1" devtype="VIRTUAL" managed="no">
            <vnsRsALDevToDomP tDn="uni/vmmp-VMware/dom-mininet"/>
        </vnsLDevVip>

    </fvTenant>
</polUni>
```

次の REST 要求はデバイス クラスタ コンテキストを作成します。

```
<polUni>
    <fvTenant dn="uni/tn-acme" name="acme">
       <vnsLDevCtx ctrctNameOrLbl="webCtrct" graphNameOrLbl="G1" nodeNameOrLbl="Node1">

            <vnsRsLDevCtxToLDev tDn="uni/tn-acme/lDevVip-ADCCluster1"/>
            <vnsLIfCtx connNameOrLbl="ssl-inside">
                <vnsRsLIfCtxToLIf tDn="uni/tn-acme/lDevVip-ADCCluster1/lIf-int"/>
            </vnsLIfCtx>
            <vnsLIfCtx connNameOrLbl="any">
                <vnsRsLIfCtxToLIf tDn="uni/tn-acme/lDevVip-ADCCluster1/lIf-ext"/>
            </vnsLIfCtx>
        </vnsLDevCtx>
```

```
    </fvTenant>
</polUni>
```

次の要求は、ルーティング ピアリングに使用されるデバイス クラスタ コンテキストを作成します。

```
<polUni>
    <fvTenant dn="uni/tn-coke{{tenantId}}" name="coke{{tenantId}}">
        <vnsRtrCfg name="Dev1Ctx1" rtrId="180.0.0.12"/>
            <vnsLDevCtx ctrctNameOrLbl="webCtrct1" graphNameOrLbl="WebGraph"
              nodeNameOrLbl="FW">
                <vnsRsLDevCtxToLDev tDn="uni/tn-tenant1/lDevVip-Firewall"/>
                <vnsRsLDevCtxToRtrCfg tnVnsRtrCfgName="FwRtrCfg"/>
                <vnsLIfCtx connNameOrLbl="internal">
                    <vnsRsLIfCtxToInstP
tDn="uni/tn-tenant1/out-OspfInternal/instP-IntInstP"
                      status="created,modified"/>
                 <vnsRsLIfCtxToLIf tDn="uni/tn-tenant1/lDevVip-Firewall/lIf-internal"/>

                </vnsLIfCtx>
                <vnsLIfCtx connNameOrLbl="external">
                 <vnsRsLIfCtxToInstP tDn="uni/tn-common/out-OspfExternal/instP-ExtInstP"

                      status="created,modified"/>
                 <vnsRsLIfCtxToLIf tDn="uni/tn-tenant1/lDevVip-Firewall/lIf-external"/>

            </vnsLIfCtx>
        </vnsLDevCtx>
    </fvTenant>
</polUni>
```

（注） テナント（レイヤ3 Outside）の外部接続の設定については、『*Cisco APIC* ベーシック コンフィギュレーション ガイド』を参照してください。

次の REST 要求はデバイス クラスタの論理インターフェイスを追加します。

```
<polUni>
    <fvTenant dn="uni/tn-acme" name="acme">
        <vnsLDevVip name="ADCCluster1">
            <vnsLIf name="C5">
                <vnsRsMetaIf tDn="uni/infra/mDev-Acme-ADC-1.0/mIfLbl-outside"/>
                <vnsRsCIfAtt tDn="uni/tn-acme/lDevVip-ADCCluster1/cDev-ADC1/cIf-int"/>
            </vnsLIf>
            <vnsLIf name="C4">
                <vnsRsMetaIf tDn="uni/infra/mDev-Acme-ADC-1.0/mIfLbl-inside"/>
                <vnsRsCIfAtt tDn="uni/tn-acme/lDevVip-ADCCluster1/cDev-ADC1/cIf-ext"/>
            </vnsLIf>
        </vnsLDevVip>
    </fvTenant>
</polUni>
```

次の REST 要求は物理デバイス クラスタの具象デバイスを追加します。

```
<polUni>
    <fvTenant dn="uni/tn-acme" name="acme">
        <vnsLDevVip name="ADCCluster1">
            <vnsCDev name="ADC1" devCtxLbl="C1">
                <vnsCIf name="int">
                    <vnsRsCIfPathAtt tDn="topology/pod-1/paths-17/pathep-[eth1/22]"/>
                </vnsCIf>
                <vnsCIf name="ext">
```

```
                    <vnsRsCIfPathAtt tDn="topology/pod-1/paths-17/pathep-[eth1/21]"/>
                </vnsCIf>
                <vnsCIf name="mgmt">
                    <vnsRsCIfPathAtt tDn="topology/pod-1/paths-17/pathep-[eth1/20]"/>
                </vnsCIf>
                <vnsCMgmt name="devMgmt" host="172.30.30.100" port="80"/>
                <vnsCCred name="username" value="admin"/>
                <vnsCCred name="password" value="admin"/>
            </vnsCDev>
            <vnsCDev name="ADC2" devCtxLbl="C2">
                <vnsCIf name="int">
                    <vnsRsCIfPathAtt tDn="topology/pod-1/paths-17/pathep-[eth1/23]"/>
                </vnsCIf>
                <vnsCIf name="ext">
                    <vnsRsCIfPathAtt tDn="topology/pod-1/paths-17/pathep-[eth1/24]"/>
                </vnsCIf>
                <vnsCIf name="mgmt">
                    <vnsRsCIfPathAtt tDn="topology/pod-1/paths-17/pathep-[eth1/30]"/>
                </vnsCIf>
                <vnsCMgmt name="devMgmt" host="172.30.30.200" port="80"/>
                <vnsCCred name="username" value="admin"/>
                <vnsCCred name="password" value="admin"/>
            </vnsCDev>
        </vnsLDevVip>
    </fvTenant>
</polUni>
```

次の REST 要求は仮想デバイス クラスタの具象デバイスを追加します。

```
<polUni>
    <fvTenant dn="uni/tn-coke5" name="coke5">
        <vnsLDevVip name="Firewall5" devtype="VIRTUAL">
            <vnsCDev name="ASA5" vcenterName="vcenter1" vmName="ifav16-ASAv-scale-05">
                <vnsCIf name="Gig0/0" vnicName="Network adapter 2"/>
                <vnsCIf name="Gig0/1" vnicName="Network adapter 3"/>
                <vnsCIf name="Gig0/2" vnicName="Network adapter 4"/>
                <vnsCIf name="Gig0/3" vnicName="Network adapter 5"/>
                <vnsCIf name="Gig0/4" vnicName="Network adapter 6"/>
                <vnsCIf name="Gig0/5" vnicName="Network adapter 7"/>
                <vnsCIf name="Gig0/6" vnicName="Network adapter 8"/>
                <vnsCIf name="Gig0/7" vnicName="Network adapter 9"/>
                <vnsCMgmt name="devMgmt" host="3.5.3.170" port="443"/>
                <vnsCCred name="username" value="admin"/>
                <vnsCCredSecret name="password" value="insieme"/>
            </vnsCDev>
        </vnsLDevVip>
    </fvTenant>
</polUni>
```

次の REST 要求は管理対象サービス グラフを作成します。

```
<polUni>
    <fvTenant name="acme">
        <vnsAbsGraph name = "G1">

        <vnsAbsTermNode name = "Input1">
            <vnsAbsTermConn name = "C1" direction = "output">
            </vnsAbsTermConn>
        </vnsAbsTermNode>

        <!-- Node1 Provides SLB functionality -->
        <vnsAbsNode name = "Node1" funcType="GoTo" >
            <vnsRsDefaultScopeToTerm
                tDn="uni/tn-acme/AbsGraph-G1/AbsTermNode-Output1/outtmnl"/>
```

```
            <vnsAbsFuncConn name = "C4" direction = "input">
              <vnsRsMConnAtt tDn="uni/infra/mDev-Acme-ADC-1.0/mFunc-SLB/mConn-external"/>

                <vnsRsConnToLIf tDn="uni/tn-acme/lDevVip-ADCCluster1/lIf-C4"/>
            </vnsAbsFuncConn>

            <vnsAbsFuncConn name = "C5" direction = "output">
              <vnsRsMConnAtt tDn="uni/infra/mDev-Acme-ADC-1.0/mFunc-SLB/mConn-internal"/>

                <vnsRsConnToLIf tDn="uni/tn-acme/lDevVip-ADCCluster1/lIf-C5"/>
            </vnsAbsFuncConn>

            <vnsRsNodeToMFunc tDn="uni/infra/mDev-Acme-ADC-1.0/mFunc-SLB"/>
        </vnsAbsNode>

        <vnsAbsTermNode name = "Output1">
            <vnsAbsTermConn name = "C6" direction = "input">
            </vnsAbsTermConn>
        </vnsAbsTermNode>

        <vnsAbsConnection name = "CON1">
            <vnsRsAbsConnectionConns
                tDn="uni/tn-acme/AbsGraph-G1/AbsTermNode-Input1/AbsTConn"/>
            <vnsRsAbsConnectionConns
                tDn="uni/tn-acme/AbsGraph-G1/AbsNode-Node1/AbsFConn-C4"/>
        </vnsAbsConnection>

        <vnsAbsConnection name = "CON3">
            <vnsRsAbsConnectionConns
                tDn="uni/tn-acme/AbsGraph-G1/AbsNode-Node1/AbsFConn-C5"/>
            <vnsRsAbsConnectionConns
                tDn="uni/tn-acme/AbsGraph-G1/AbsTermNode-Output1/AbsTConn"/>
        </vnsAbsConnection>
    </vnsAbsGraph>
  </fvTenant>
</polUni>
```

次の REST 要求は非管理対象モードでサービス グラフを作成します。

```
<polUni>
    <fvTenant name="HA_Tenant1">
        <vnsAbsGraph name="g1">

            <vnsAbsTermNodeProv name="Input1">
                <vnsAbsTermConn name="C1">
                </vnsAbsTermConn>
            </vnsAbsTermNodeProv>

            <!-- Node1 Provides LoadBalancing functionality -->
            <vnsAbsNode name="Node1" managed="no">
                <vnsRsDefaultScopeToTerm
                  tDn="uni/tn-HA_Tenant1/AbsGraph-g1/AbsTermNodeProv-Input1/outtmnl"/>
                <vnsAbsFuncConn name="outside" attNotify="true">
                </vnsAbsFuncConn>
                <vnsAbsFuncConn name="inside" attNotify="true">
                </vnsAbsFuncConn>
            </vnsAbsNode>

            <vnsAbsTermNodeCon name="Output1">
                <vnsAbsTermConn name="C6">
                </vnsAbsTermConn>
            </vnsAbsTermNodeCon>

            <vnsAbsConnection name="CON2" adjType="L3" unicastRoute="yes">
                <vnsRsAbsConnectionConns
```

```
                tDn="uni/tn-HA_Tenant1/AbsGraph-g1/AbsTermNodeCon-Output1/AbsTConn"/>
            <vnsRsAbsConnectionConns
                tDn="uni/tn-HA_Tenant1/AbsGraph-g1/AbsNode-Node1/AbsFConn-outside"/>
        </vnsAbsConnection>

        <vnsAbsConnection name="CON1" adjType="L2" unicastRoute="no">
            <vnsRsAbsConnectionConns
                tDn="uni/tn-HA_Tenant1/AbsGraph-g1/AbsNode-Node1/AbsFConn-inside"/>
            <vnsRsAbsConnectionConns
                tDn="uni/tn-HA_Tenant1/AbsGraph-g1/AbsTermNodeProv-Input1/AbsTConn"/>
        </vnsAbsConnection>

    </vnsAbsGraph>
  </fvTenant>
</polUni>
```

次の REST 要求はセキュリティ ポリシー（コントラクト）を作成します。

```
<polUni>
    <fvTenant dn="uni/tn-acme" name="acme">
        <vzFilter name="HttpIn">
            <vzEntry name="e1" prot="6" dToPort="80"/>
        </vzFilter>

        <vzBrCP name="webCtrct">
            <vzSubj name="http">
                <vzRsSubjFiltAtt tnVzFilterName="HttpIn"/>
            </vzSubj>
        </vzBrCP>
    </fvTenant>
</polUni>
```

次の REST 要求はアプリケーション EPG からのグラフ コンフィギュレーション パラメータを提供します。

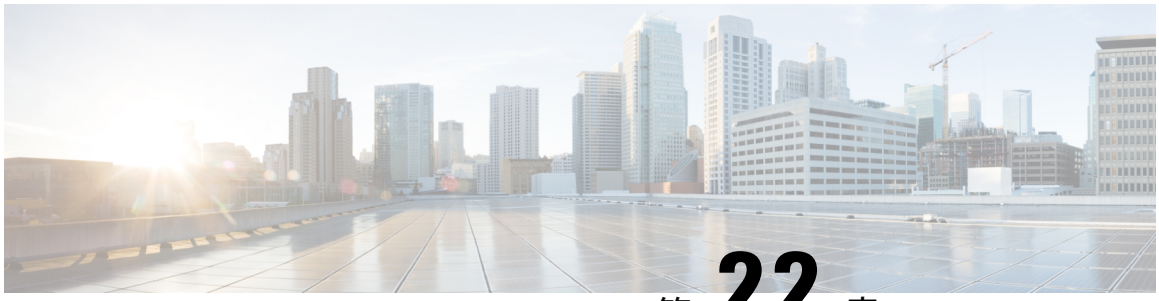
```
<polUni>
   <fvTenant dn="uni/tn-acme" name="acme">

      <!-- Application Profile -->
      <fvAp dn="uni/tn-acme/ap-MyAP" name="MyAP">

          <!-- EPG 1  -->
          <fvAEPg dn="uni/tn-acme/ap-MyAP/epg-MyClnt" name="MyClnt">
              <fvRsBd tnFvBDName="MyClntBD"/>
              <fvRsDomAtt tDn="uni/vmmp-Vendor1/dom-MyVMs"/>
              <fvRsProv tnVzBrCPName="webCtrct">
              </fvRsProv>
              <fvRsPathAtt tDn="topology/pod-1/paths-17/pathep-[eth1/20]"
encap="vlan-201"/>
              <fvSubnet name="SrcSubnet" ip="192.168.10.1/24"/>
          </fvAEPg>

          <!-- EPG 2 -->
          <fvAEPg dn="uni/tn-acme/ap-MyAP/epg-MySRVR"  name="MySRVR">
             <fvRsBd tnFvBDName="MyClntBD"/>
             <fvRsDomAtt tDn="uni/vmmp-Vendor1/dom-MyVMs"/>
             <fvRsCons tnVzBrCPName="webCtrct">
             </fvRsCons>

           <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="any" nodeNameOrLbl="any"

               key="Monitor" name="monitor1">
                 <vnsParamInst name="weight" key="weight" value="10"/>
             </vnsFolderInst>
```

```
<vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="any" nodeNameOrLbl="any"

    key="Service" name="Service1">
      <vnsParamInst name="servicename" key="servicename"
        value="crpvgrtst02-8010"/>
      <vnsParamInst name="servicetype" key="servicetype" value="TCP"/>
      <vnsParamInst name="servername" key="servername"
        value="s192.168.100.100"/>
      <vnsParamInst name="serveripaddress" key="serveripaddress"
        value="192.168.100.100"/>
      <vnsParamInst name="serviceport" key="serviceport" value="8080"/>
      <vnsParamInst name="svrtimeout" key="svrtimeout" value="9000"/>
      <vnsParamInst name="clttimeout" key="clttimeout" value="9000"/>
      <vnsParamInst name="usip" key="usip" value="NO"/>
      <vnsParamInst name="useproxyport" key="useproxyport" value=""/>
      <vnsParamInst name="cip" key="cip" value="ENABLED"/>
      <vnsParamInst name="cka" key="cka" value="NO"/>
      <vnsParamInst name="sp" key="sp" value="OFF"/>
      <vnsParamInst name="cmp" key="cmp" value="NO"/>
      <vnsParamInst name="maxclient" key="maxclient" value="0"/>
      <vnsParamInst name="maxreq" key="maxreq" value="0"/>
      <vnsParamInst name="tcpb" key="tcpb" value="NO"/>
      <vnsCfgRelInst name="MonitorConfig" key="MonitorConfig"
        targetName="monitor1"/>
    </vnsFolderInst>

    <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="any"
      nodeNameOrLbl="any" key="Network" name="Network">
      <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="any"
        nodeNameOrLbl="any" key="vip" name="vip">
          <vnsParamInst name="vipaddress1" key="vipaddress"
            value="10.10.10.100"/>
      </vnsFolderInst>
      <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="any"
        nodeNameOrLbl="any" devCtxLbl="C1" key="snip" name="snip1">
          <vnsParamInst name="snipaddress" key="snipaddress"
            value="192.168.1.100"/>
      </vnsFolderInst>
      <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="any"
        nodeNameOrLbl="any" devCtxLbl="C2" key="snip" name="snip2">
          <vnsParamInst name="snipaddress" key="snipaddress"
            value="192.168.1.101"/>
      </vnsFolderInst>
      <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="any"
        nodeNameOrLbl="any" devCtxLbl="C3" key="snip" name="snip3">
          <vnsParamInst name="snipaddress" key="snipaddress"
            value="192.168.1.102"/>
      </vnsFolderInst>
    </vnsFolderInst>

    <!-- SLB Configuration -->
    <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="any"
     nodeNameOrLbl="any" key="VServer" name="VServer">
      <!-- Virtual Server Configuration -->
      <vnsParamInst name="port" key="port" value="8010"/>
      <vnsParamInst name="vip" key="vip" value="10.10.10.100"/>
      <vnsParamInst name="vservername" key="vservername"
        value="crpvgrtst02-vip-8010"/>
      <vnsParamInst name="servicename" key="servicename"
        value="crpvgrtst02-8010"/>
      <vnsParamInst name="servicetype" key="servicetype" value="TCP"/>
      <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="any"
        nodeNameOrLbl="any" key="VServerGlobalConfig"
```

```
name="VServerGlobalConfig">
                      <vnsCfgRelInst name="ServiceConfig" key="ServiceConfig"
                        targetName="Service1"/>
                      <vnsCfgRelInst name="VipConfig" key="VipConfig"
                        targetName="Network/vip"/>
                  </vnsFolderInst>
              </vnsFolderInst>
          </fvAEPg>
      </fvAp>
   </fvTenant>
</polUni>
```

次の REST 要求はコントラクトにサービス グラフをアタッチします。

```
<polUni>
    <fvTenant name="acme">
        <vzBrCP name="webCtrct">
            <vzSubj name="http">
                <vzRsSubjGraphAtt graphName="G1" termNodeName="Input1"/>
            </vzSubj>
        </vzBrCP>
    </fvTenant>
</polUni>
```



第 22 章

GUI の使用方法

GUI を使用したレイヤ 4 ～ レイヤ 7 サービスの導入

GUI を使用して、レイヤ 4 ～ レイヤ 7 サービスを導入することができます。次の順序で手順を実行します。

1. デバイス パッケージをインポートします。

   GUI を使用したデバイス パッケージのインポート （238 ページ）を参照してください。

2. 機能プロファイルを作成します。

   GUI を使用した機能プロファイルの作成 （238 ページ）を参照してください。

3. サービス グラフ テンプレートを作成します。

   GUI を使用したレイヤ 4 ～ レイヤ 7 サービス グラフ テンプレートの作成 （241 ページ）を参照してください。

4. デバイスを作成します。

   GUI を使用したレイヤ 4 ～ レイヤ 7 デバイスの作成 （13 ページ）を参照してください。

   （オプション）デバイスを変更します。

   デバイスの変更 （242 ページ）を参照してください。

5. エンドポイント グループ（EGP）にサービス グラフ テンプレートを適用します。

GUI を使用したエンドポイント グループへのサービス グラフ テンプレートの適用 （243 ページ）を参照してください。

GUI を使用したデバイス パッケージのインポート

サービス グラフに基づいていかなる設定も実行する前に、適切なデバイス パッケージを Application Policy Infrastructure Controller（APIC）にダウンロードしてインストールする必要があります。デバイスパッケージは、所有しているデバイスと、そのデバイスで何が実行できるかを APIC に対して指定します。

（注） クラウド オーケストレータ モードを使用するデバイス パッケージを選択すると、簡潔なインターフェイスが得られます。クラウドオーケストレータ モードデバイスパッケージは、Cisco APIC で自動的に作成されます。デバイス パッケージが誤って削除された場合、もう一度アップロードすることができます。クラウドオーケストレータ モードの設定 （245 ページ）も参照してください。

ステップ 1 適切なデバイス パッケージをダウンロードします。パートナーのリストは、次の URL にあります。

http://www.cisco.com/c/en/us/solutions/data-center-virtualization/ecosystem.html

この URL は、適切なデバイス パッケージをダウンロードできる [Partner Ecosystem] ページです。

ステップ 2 プロバイダー管理者として APIC にログインします。

ステップ 3 メニュー バーで、**[L4-L7 Services]** > **[Packages]** を選択します。

ステップ 4 [Navigation] ペインで、[L4-L7 Service Device Types] をクリックします。

ステップ 5 [Work] ペインで、**[Actions]** > **[Import Device Package]** を選択します。[Import Device Package] ダイアログ ボックスが表示されます。

ステップ 6 [Browse...] をクリックし、使用するデバイス パッケージを参照します。

デバイス パッケージの作成については、『*Cisco APIC Layer 4 to Layer 7 Device Package Development Guide*』を参照してください。

ステップ 7 [Open] をクリックします。

ステップ 8 [Submit] をクリックします。

GUI を使用した機能プロファイルの作成

機能プロファイルはサービス グラフ テンプレートにデフォルト値を提供します。次の手順で、新しい機能プロファイルの作成方法を説明します。

ステップ 1 メニュー バーで、**[Tenants]** > **[All Tenants]** の順に選択します。

ステップ 2 [Work] ペインで、テナントの名前をダブルクリックします。

ステップ 3 ナビゲーション ウィンドウで、**Tenant** *tenant_name* > **Services** > **L4-L7** > **Function Profiles** を選択します。

ステップ 4 **Function Profiles** を右クリックし、**Create L4-L7 Services Function Profile** を選択します。

ステップ 5 **Create L4-L7 Services Function Profile** ダイアログボックスで、必要に応じてフィールドに適切な値を入力します。ただし、下で指定しているものを除きます:

a) [Profile Group] ドロップダウン リストで、[Create Function Profile Group] を選択します。

プロファイル グループは、プロファイルをグループ化して整理するための機能です。たとえば、Web アプリケーション、レガシー アプリケーション、電子メール アプリケーション用に 1 つのプロファイルを作成することもできます。グループを作成し、そのグループにプロファイルを配置できます。使用可能な既存のグループがあるか確認します。存在しない場合は [Create L4-L7 Services Function Profile Group] ウィンドウでグループに名前を付け、説明を入力して新しいグループを作成できます。

ステップ 6 **Create L4-L7 Services Function Profile Group** ダイアログボックスで、必要に応じて適切な値をフィールドに入力します。

ステップ 7 [Submit] をクリックします。

Create L4-L7 Services Function Profile ダイアログボックスに戻ります。プロファイル グループは適切に作成され、保存されています。これは **Create L4-L7 Services Function Profile** ダイアログボックスに表示されます。

サービス プロファイルを特定の機能用に作成します。[Create L4-L7 Services Function Profile] の [Device Function] ドロップダウン リストから選択した機能に対して、プロファイルを作成します。デバイス パッケージをインポートした後、ドロップダウンリストにはデバイス パッケージと、Application Policy Infrastructure Controller（APIC）で使用可能なサービス機能のリストが表示されます。

ステップ 8 **Create L4-L7 Services Function Profile** ダイアログボックスで、**Copy Existing Profile Parameters** チェックボックスをオフにします。

ステップ 9 **Device Function** ドロップダウン リストから、デバイスの機能を選択します。
その機能に含まれる各パラメータと共に、オプションが表示されます。プロファイルはパラメータにデフォルト値を提供することを目的としています。

（注） この時点では、パラメータに値はないので、追加します。追加する値がデフォルト値として使用されます。機能プロファイルは、これらの値を指定した後に、グラフ テンプレートで使用できるようになります。これらの値はデフォルト値としてグラフ テンプレートに適用されます。つまり、グラフ テンプレートを使用していて特定のパラメータに値を指定しなければ、APIC がプロファイルをルックアップし、値があるかどうかを確認します。値が存在する場合、APIC はその値を使用します。

ステップ 10 **Features and Parameters** セクション (**Create L4-L7 Services Function Profile** ダイアログボックスの下) で、値を追加します。このセクションには 2 つのタブ、[Basic Parameters] と [All Parameters] があります。[Basic Parameters] タブには、パッケージに必須（必要）とマークされたパラメータのリストが含まれています。**All Parameters** タブには必要なパラメータのリストと、高度な設定を行うためのいくつかの追加/オプション ラメータのリストが含まれています。**Basic Parameters** パラメータが公開されている理由は、それら

が基本設定の一部であり、管理者にはこれらを入力することが期待されているからです。**All Parameters** はオプションであるため、機能をカスタマイズしない限り、省略することもできます。

ステップ 11 (オプション) 次の手順に従って、クラウド オーケストレータ モード機能プロファイルを作成します:

a) **All Parameters** または **Basic Parameters** タブでフォルダまたはパラメータをダブルクリックします。選択したフォルダまたはパラメータに対応する行が開きます。

b) **Path from Schema** を指定します:

- フォルダのパスを指定する場合、**Path from Schema** カラムのドロップダウンリストには、可能なすべてのフォルダ パスが一覧表示されます。スキーマ内でフォルダのマップ先になっているパスを選択します。
- パラメータのパスを指定する場合には、次の手順に従います:

1. **Path from Schema** フィールドの編集アイコンをクリックします。**Manage Path-From-Schema** ダイアログが表示されます。
2. **Specify Path-From-Schema** をクリックして有効にします。
3. **Path** ドロップダウン矢印をクリックして、パスを選択します。
4. パラメータ エディタで + をクリックして、ドロップダウンリストからパラメータを選択します。
5. 完了したら、**Ok** をクリックします。**Create L4-L7 Services Function Profile** に戻ります。
6. (オプション) 次のフィールドに値を入力します:
   - **Value** – 選択したパラメータのグラフを展開する際に、UI でデフォルト値が表示されるようにするには、ここに値を入力します。
   - **Hint** – グラフを展開する際、UI で値を入力するときに表示されるテキストを指定します。

c) **Update** をクリックします。

ステップ 12 [Submit] をクリックします。
これで、機能プロファイルは作成・保存されました。

GUI を使用した既存の機能ファイルを使用しての新しい機能プロファイルの作成

この手順では、既存の機能プロファイルを使用して新しい機能プロファイルを作成します。

ステップ 1 メニュー バーで、**[Tenants]** > **[All Tenants]** の順に選択します。

ステップ 2 [Work] ペインで、テナントの名前をダブルクリックします。

ステップ 3 [Navigation] ペインで、 **[Tenant]*[tenant_name]*** > **[Services]** > **[L4-L7]** > **[Function Profiles]** の順に選択します。

ステップ 4 **[Function Profiles]** を右クリックし、**[Create L4-L7 Services Function Profile]** を選択します。

ステップ 5 **[Create L4-L7 Services Function Profile]** ダイアログボックスで、下記で指定している項目を除き、必要に応じてフィールドに入力します。

a) **[Profile]** ドロップダウン リストで、ベンダーが指定する既存のプロファイルを選択します。
選択したプロファイルに基づいて、新しいプロファイルにパラメータが挿入されます。

b) 必要に応じて、この既存のプロファイルに変更を加えたり、パラメータを追加したりします。

ステップ 6 [Submit] をクリックします。

GUI を使用したレイヤ 4 ～ レイヤ 7 サービス グラフ テンプレートの作成

サービス グラフ テンプレートは、機能プロファイルを使用して提供可能な一連のレイヤ 4 ～ レイヤ 7 機能、レイヤ 4 ～ レイヤ 7 デバイス、またはコピーデバイスと、それらに関連付けられた設定です。サービス グラフ テンプレートは、レイヤ 4 ～ レイヤ 7 デバイスまたはコピーデバイス上およびファブリック上に「レンダリングされる」、または設定される契約と関連付ける必要があります。

始める前に

テナントを作成しておく必要があります。

ステップ 1 メニュー バーで、 **Tenants** > **All Tenants** を選択します。

ステップ 2 作業ウィンドウで、テナントの名前をダブルクリックします。

ステップ 3 ナビゲーション ウィンドウで、**Tenant *tenant_name*** > **Services** > **L4-L7** > **Service Graph Templates** を選択します。

ステップ 4 ナビゲーション ウィンドウで、**Service Graph Templates** を右クリックして、**Create a L4-L7 Service Graph Template** を選択します。

Create L4-L7 Service Graph Template ダイアログボックスが表示されます。

ステップ 5 必要に応じては、1 つ以上のレイヤ 4 ～ レイヤ 7 デバイスをまたはコピー デバイスを作成します。

a) **Device Clusters** ペイン (**Create L4-L7 Service Graph Template** ダイアログボックス) でドロップダウン矢印をクリックして、**Create L4-L7 Devices** または **Create Copy Devices** を選択します。
対応するダイアログボックスが表示されます。

b) ダイアログボックスに従い、ダイアログボックスに表示される適切な値を入力して **Next** をクリックし、完了するまで続けます。

（注） ダイアログボックス内のフィールドの説明については、右上隅のヘルプ アイコンをクリックして、ヘルプ ファイルを表示してください。

c) 完了したら、**Finish** をクリックします。

Create L4-L7 Service Graph Template ダイアログボックスに戻ります。

ステップ 6 **Create L4-L7 Service Graph Template** ダイアログボックスに適切な値を入力します。

（注） ダイアログボックス内のフィールドの説明については、右上隅のヘルプ アイコンをクリックして、ヘルプ ファイルを表示してください。

ステップ 7 （任意）(既存のサービス グラフ テンプレートを複製場合のみ) 複製したサービス グラフ テンプレートからノードを削除する場合は、ノードを右クリックして、**Remove Node** を選択します。

ステップ 8 サービス ノードを作成するには、**Device Clusters** セクションからレイヤ 4 ～ レイヤ 7 デバイスをドラッグして、コンシューマエンドポイントとプロバイダエンドポイントの間にドロップします。コピー ノードを作成するには、コピー デバイスをドラッグ アンド ドロップします。既存のサービス グラフ テンプレートを複製し、それにサービス グラフテンプレートに使用するすべてのノードが含まれている場合には、この手順はオプションです。

複数のデバイスをドラッグ アンド ドロップして、複数のノードを作成することができます。サービス ノードの最大数は 3 ですが、他のデバイスはそれ以上ドラッグ アンド ドロップできます。

コピーデバイスをドロップした場所が、データ フローの中で、コピーデバイスがトラフィックをコピーする場所になります。

ステップ 9 1 つまたは複数のサービス ノードを作成した場合、レイヤ 4 ～ レイヤ 7 デバイスごとの ***device_name*** **Information** セクションで、入力を完了してください。フィールドは、デバイスのタイプによって異なります。

（注） フィールドの説明については、右上隅のヘルプアイコンをクリックして、ヘルプ ファイルを表示してください。

ステップ 10 完了したら、**Submit** をクリックします。

ステップ 11 （任意） **Navigation** ウィンドウで、サービス グラフ テンプレートをクリックします。
作業ウィンドウには、そのサービス グラフ テンプレートのグラフィック トポロジが表示されます。

デバイスの変更

デバイスを作成した後で、そのデバイスを変更することができます。

（注） デバイスを作成するか、または既存のクラスタにデバイスを追加するには、「デバイスの作成」の手順を使用する必要があります。

ステップ 1 メニュー バーで、**[Tenants]** > **[All Tenants]** の順に選択します。

ステップ 2 [Work] ペインで、テナントの名前をダブルクリックします。

ステップ 3 [Navigation] ウィンドウで、**Tenant** ***tenant_name*** > **Services** > **L4-L7** > **Devices** > ***device_name*** を選択します。[Work] ウィンドウにデバイスに関する情報が表示されます。

ステップ 4 **General** セクションではいくつかのパラメータを変更するコ音ができます。

Device 1 セクションでは、インターフェイスの追加、または既存のインターフェイスのパスの変更を行えます。インターフェイスを追加するには、+ ボタンをクリックします。パスを変更するには、変更するパスをダブルクリックします。

ステップ 5 パラメータを変更した後、**Submit** をクリックします。

GUI を使用したエンドポイント グループへのサービス グラフ テンプレートの適用

次の手順で、エンドポイント グループへのサービス グラフ テンプレートの適用法を説明します。

始める前に

次を作成しておく必要がります。

- アプリケーション エンドポイント グループ
- サービス グラフ テンプレート

ステップ 1 メニュー バーで、**[Tenants]** > **[All Tenants]** の順に選択します。

ステップ 2 [Work] ペインで、テナントの名前をダブルクリックします。

ステップ 3 [Navigation] ウィンドウで、**Tenant** ***tenant_name*** > **Services** > **L4-L7** > **Service Graph Templates** > ***template_name*** を選択します。

ステップ 4 [Navigation] ウィンドウで、EPG を適用する ***template_name*** を右クリックし、**Apply L4-L7 Service Graph Template** を選択します。

Apply L4-L7 Service Graph Template To EPGs ダイアログボックスが表示されます。レイヤ 4 ～ レイヤ 7 サービス グラフ テンプレートをコンシューマ エンドポイント グループとプロバイダー エンドポイント グループに関連付けます。

ステップ 5 [Apply L4-L7 Service Graph Template To EPGS STEP 1] > [Contract] ダイアログボックスで、適切な値を入力して契約を設定します。

a) EPG 内契約を設定する場合は、[Configure an Intra-EPG Contract] チェックボックスをオンにして、[EPG / Network] ドロップダウンリストから EPG とネットワークの組み合わせを選択します。

b) 標準契約を設定する場合は、該当するドロップダウンリストでコンシューマ/プロバイダー EPG とネットワークの組み合わせを選択します。

c) [Contract] フィールドで適切なオプション ボタンをクリックして、新しい契約を作成するか既存の契約を選択します。[Create A New Contract] を選択した場合、フィルタを設定するには、[No Filter (Allow All Traffic)] チェックボックスをオフにします。[+] をクリックしてフィルタ エントリを追加し、完了したら [Update] をクリックします。

ステップ 6 [次へ] をクリックします。

[STEP 2] > [Graph] ダイアログが表示されます。

ステップ 7 [*device_name* Information] セクションで、赤色のボックスで示された必須フィールドを設定します。

（注） 優先グループ（契約なしのエンドポイント間通信）にコネクタを含めるには、[Service EPG Policy] ドロップダウンリストから設定済みポリシーを選択します。

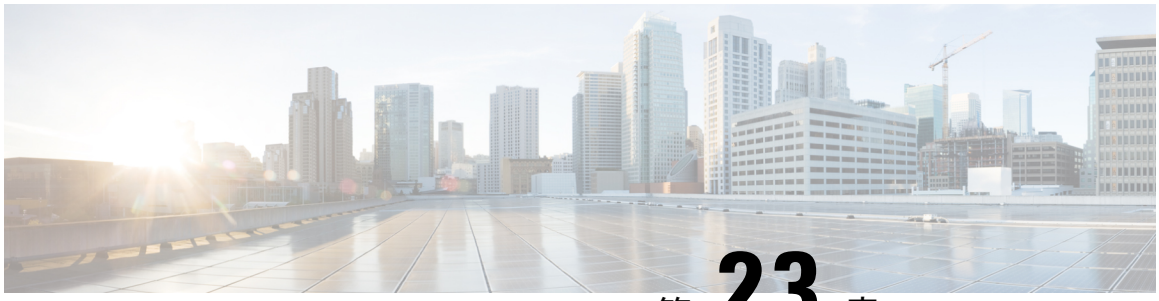
ステップ 8 [次へ] をクリックします。

[STEP 3] > [device_name Information] ダイアログが表示されます。

ステップ 9 [Required Parameters] と [All Parameters] タブで、必要に応じてパラメータを設定します。

ステップ 10 [Finish] をクリックします。

サービス グラフ テンプレートがアクティブになりました。



第 23 章

クラウド オーケストレータ モードの設定

クラウド オーケストレータ モードの概要

Cisco APIC が Azure、 v Realize、OpenStack、または CliQR などのクラウド オーケストレータで機能する環境で、クラウドオーケストレータは一般的に、ベンダーの設定パラメーターのセマンティクスを認識している必要があります。しかし、クラウドオーケストレータモードで、Cisco APIC は、LB-aas および FW-aas インターフェイスを提供して、サービス グラフのロードバランサーとファイアウォールを設定するための統合インターフェイスを作成する標準セットのパラメータを有効にします。クラウド オーケストレータ モードは、Cisco ACI ファブリックのロードバランサおよびファイアウォールをプロビジョニングする統合インターフェイスでも機能できます。その結果、オーケストレータでは、ベンダーの設定パラメータのセマンティクスを認識する必要はありません。

クラウド オーケストレータ モードのスキーマ

ファイアウォールのスキーマ

クラウド オーケストレータ モードのスキーマは、Cisco APIC で自動的に作成されたデバイス パッケージ (CISCO CloudMode デバイス パッケージ) として公開されます。

図 *41 :* ファイアウォールのインターフェイス

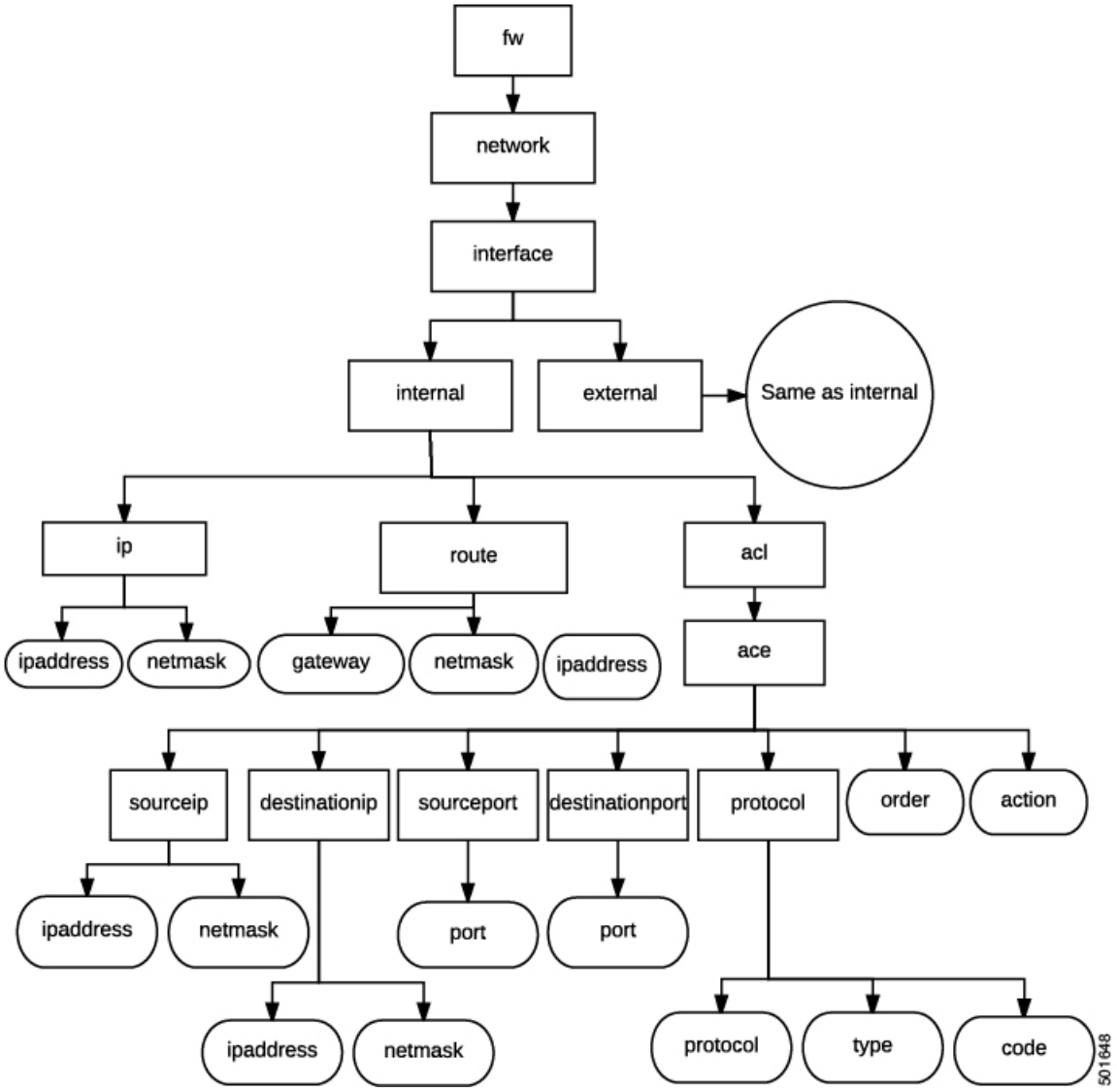


図 *42 :* ファイアウォールの *NAT*

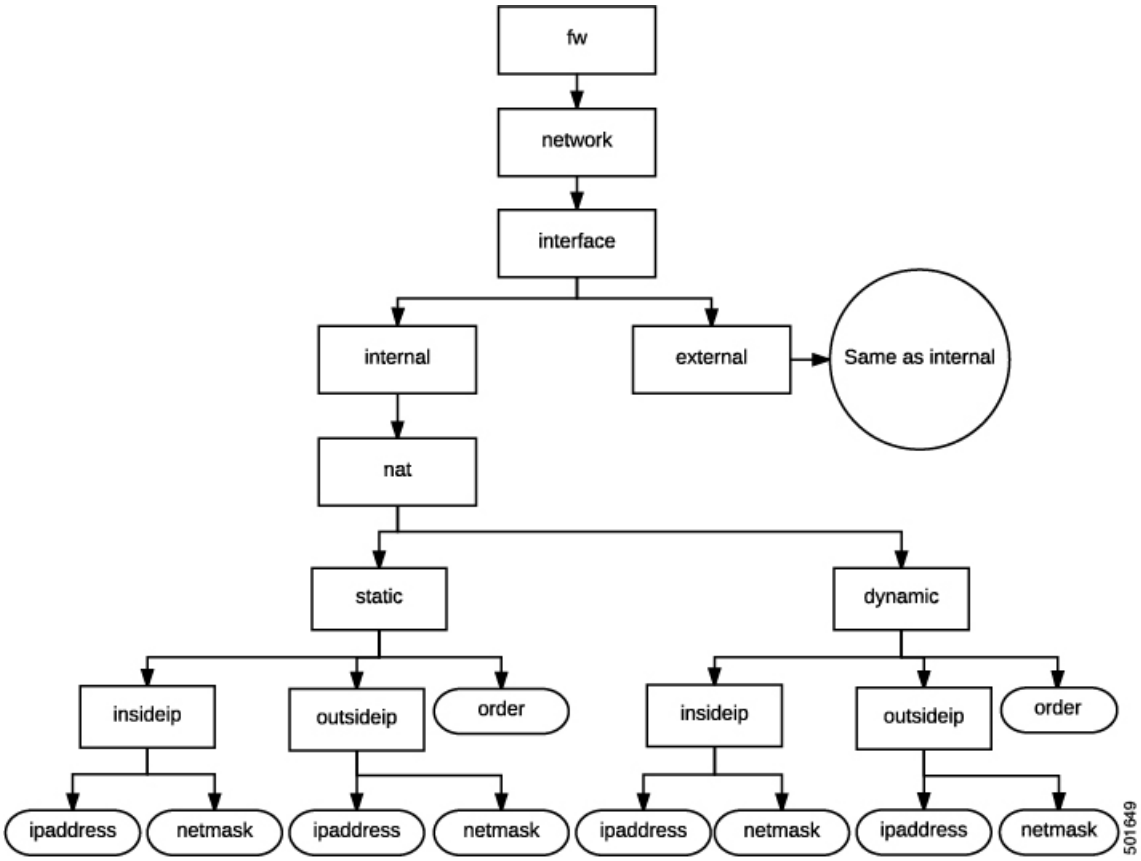


図 *43 :* ファイアウォールの *PAT*

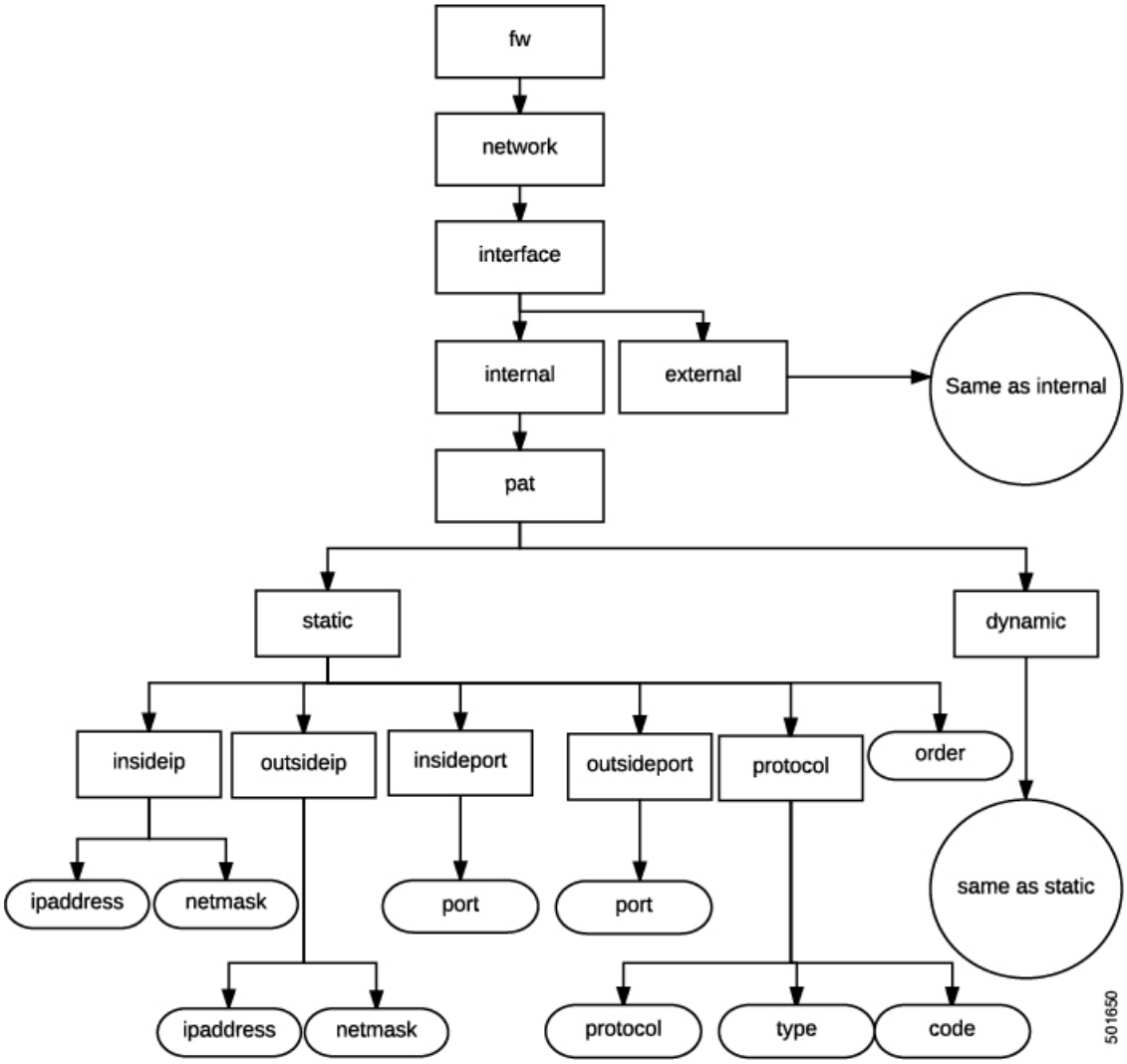
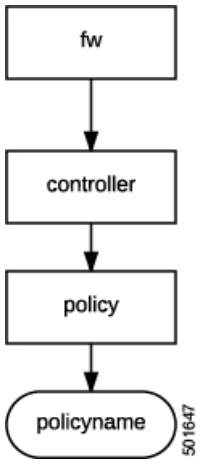


図 *44 :* ファイアウォールのコントローラ



ロード バランサのスキーマ

クラウド オーケストレータ モードのスキーマは、Cisco APIC で自動的に作成されたデバイス パッケージ (CISCO CloudMode デバイス パッケージ) として公開されます。

図 *45 :* ロード バランサのインターフェイス

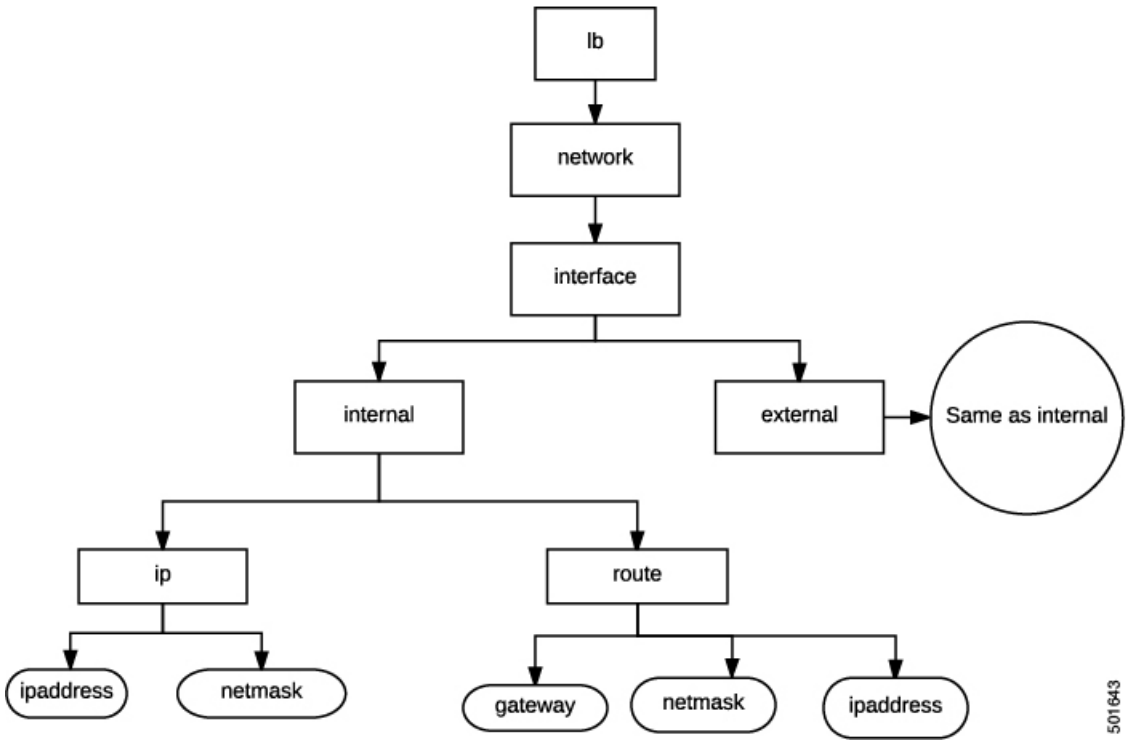


図 *46 :* ロード バランサの *NAT*

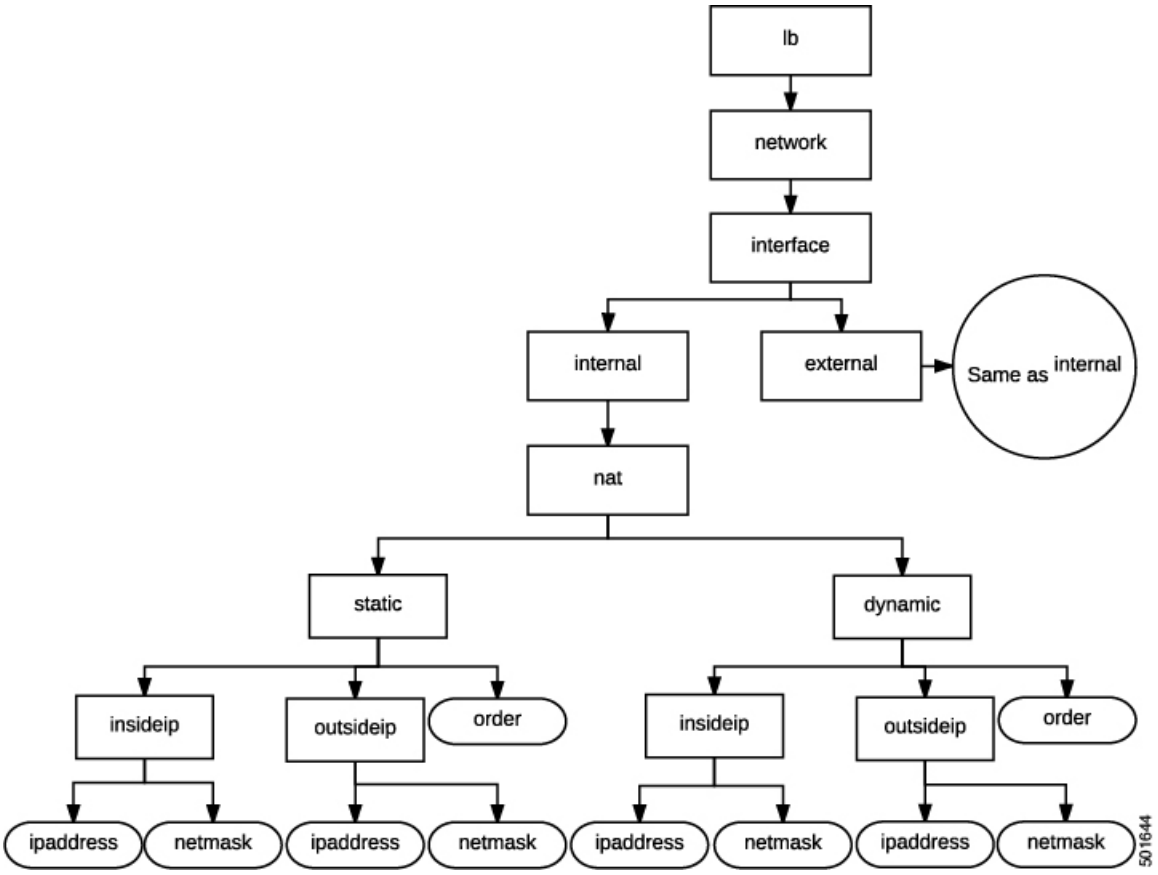


図 *47* : ロード バランサの *PAT*

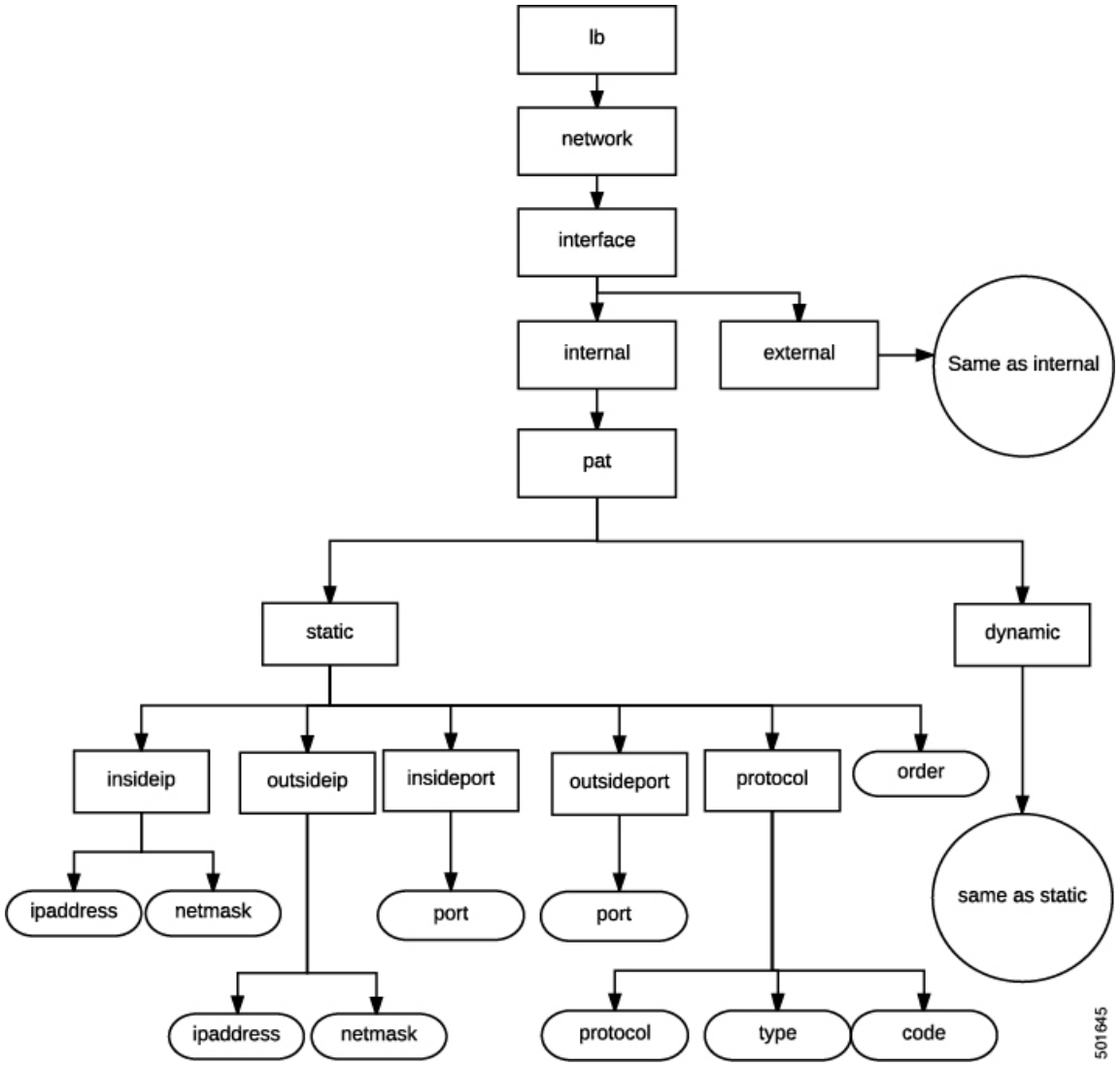


図 *48 :* ロード バランサのコントローラ

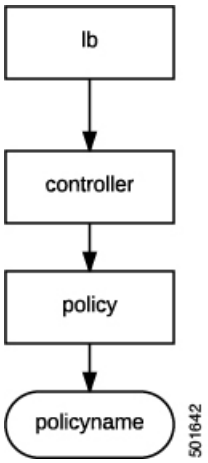
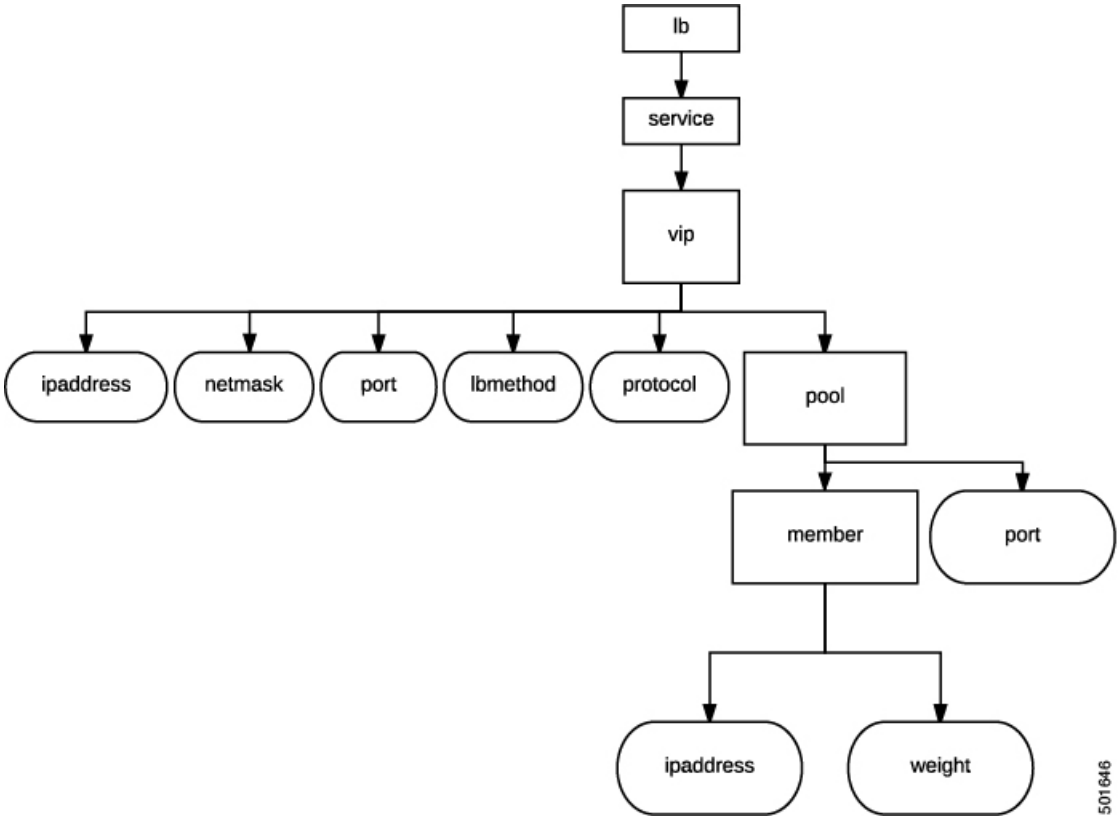


図 *49 :* ロード バランサ サービス



GUI を使用したクラウド オーケストレータ モードの設定

次のアクションを実行する際には、GUI でクラウド オーケストレータ モードを設定します:

- **関数プロファイルの作成** – レイヤ 4 ～ レイヤ 7 機能プロファイルを作成する際、既存のプロファイルを選択するときに、クラウド オーケストレータ モードを使用するオプションが表示されます。**Profile** プロパティがドロップダウン メニューとして表示され、サポートされている機能プロファイルが一覧表示されます。クラウド オーケストレータ モード プロファイルは、プロファイル名で判別できます。

  （注） クラウド オーケストレータ モードを使用する、ベンダーからのデバイス パッケージは、既存の機能プロファイルすべてと対応した、クラウドモードでの機能プロファイルの作成済みコピーが含まれています。

  機能プロファイルの詳細については、GUI を使用した機能プロファイルの作成 （238 ページ）を参照してください。

- **レイヤ 4 ～ レイヤ 7 サービス グラフ テンプレートの作成** – レイヤ 4 ～ レイヤ 7 サービス グラフ テンプレートの作成の際、クラウド オーケストレータ モードを使用するオプションがサービス ノードを作成するときに表示されます。**Profile** プロパティがドロップダウン メニューとして表示され、サポートされているプロファイイルが一覧表示されます。クラウド オーケストレータ モード プロファイルは、プロファイル名で判別できます。

  レイヤ 4 ～ レイヤ 7 サービスを作成するときにサービス ノードを作成する方法の詳細については、GUI を使用したレイヤ 4 ～ レイヤ 7 サービス グラフ テンプレートの作成 （241 ページ）を参照してください。

- **エンドポイント グループにサービス グラフ テンプレートを適用する** – クラウド オーケストレータ モード プロファイルで EPG にサービス グラフ テンプレートを適用するときには、選択したプロファイルに対応するクラウド オーケストレータ モード インターフェイスが表示されます。

  EPG にサービス グラフ テンプレートを適用する方法の詳細については、GUI を使用したエンドポイント グループへのサービス グラフ テンプレートの適用 （243 ページ）を参照してください。

REST API を使用したファイアウォールの設定

次の REST API はファイアウォールを設定します。

```
<fvTenant  name="Tenant1">
  <fvAp name="ap1">
   <fvAEPg name="epg3">
    <vnsSvcPol ctrct="ctrct_fw" graph="Graph_FW" node="FW">
     <vnsFolderInst ctrctNameOrLbl="ctrct_fw" graphNameOrLbl="Graph_FW" nodeNameOrLbl="FW"
 key="fw" name="fw">
      <vnsRsFolderInstToMFolder
tDn="uni/infra/mDev-CISCO-CloudMode-1.0/mFunc-FW/mFolder-fw"/>
     <vnsFolderInst ctrctNameOrLbl="ctrct_fw" graphNameOrLbl="Graph_FW" nodeNameOrLbl="FW"
 key="network" name="network">
       <vnsFolderInst ctrctNameOrLbl="ctrct_fw" graphNameOrLbl="Graph_FW"
```

```
nodeNameOrLbl="FW" key="interface" name="interface">
        <vnsFolderInst ctrctNameOrLbl="ctrct_fw" graphNameOrLbl="Graph_FW"
nodeNameOrLbl="FW" key="internal" name="internal">
         <vnsFolderInst ctrctNameOrLbl="ctrct_fw" graphNameOrLbl="Graph_FW"
nodeNameOrLbl="FW" key="ip" name="ip">
          <vnsParamInst name="ipaddress" key="ipaddress" value="2.2.2.2"/>
          <vnsParamInst name="netmask" key="netmask" value="255.255.255.0"/>
         </vnsFolderInst>
        </vnsFolderInst>
        <vnsFolderInst ctrctNameOrLbl="ctrct_fw" graphNameOrLbl="Graph_FW"
nodeNameOrLbl="FW" key="external" name="external">
         <vnsFolderInst ctrctNameOrLbl="ctrct_fw" graphNameOrLbl="Graph_FW"
nodeNameOrLbl="FW" key="ip" name="ip">
          <vnsParamInst name="ipaddress" key="ipaddress" value="1.1.1.1"/>
          <vnsParamInst name="netmask" key="netmask" value="255.255.255.0"/>
         </vnsFolderInst>
         <vnsFolderInst ctrctNameOrLbl="ctrct_fw" graphNameOrLbl="Graph_FW"
nodeNameOrLbl="FW" key="acl" name="acl">
          <vnsFolderInst ctrctNameOrLbl="ctrct_fw" graphNameOrLbl="Graph_FW"
nodeNameOrLbl="FW" key="ace" name="ace">
           <vnsParamInst name="action" key="action" value="PERMIT"/>
           <vnsParamInst name="order" key="order" value="10"/>
           <vnsFolderInst ctrctNameOrLbl="ctrct_fw" graphNameOrLbl="Graph_FW"
nodeNameOrLbl="FW" key="protocol" name="protocol">
            <vnsParamInst name="protocol" key="protocol" value="TCP"/>
           </vnsFolderInst>
           <vnsFolderInst ctrctNameOrLbl="ctrct_fw" graphNameOrLbl="Graph_FW"
nodeNameOrLbl="FW" key="sourceip" name="sourceip">
            <vnsParamInst name="ipaddress" key="ipaddress" value="0.0.0.0"/>
            <vnsParamInst name="netmask" key="netmask" value="0"/>
           </vnsFolderInst>
           <vnsFolderInst ctrctNameOrLbl="ctrct_fw" graphNameOrLbl="Graph_FW"
nodeNameOrLbl="FW" key="destinationip" name="destinationip">
            <vnsParamInst name="ipaddress" key="ipaddress" value="10.10.10.0"/>
            <vnsParamInst name="netmask" key="netmask" value="24"/>
           </vnsFolderInst>
           <vnsFolderInst ctrctNameOrLbl="ctrct_fw" graphNameOrLbl="Graph_FW"
nodeNameOrLbl="FW" key="destinationport" name="destinationport">
            <vnsParamInst name="port" key="port" value="80"/>
           </vnsFolderInst>
          </vnsFolderInst>
         </vnsFolderInst>
        </vnsFolderInst>
       </vnsFolderInst>
      </vnsFolderInst>
     </vnsFolderInst>
    </vnsSvcPol>
   </fvAEPg>
  </fvAp>
</fvTenant>
```

REST API を使用したロード バランサの設定

次の REST API がロード バランサを設定します:

```
<fvTenant  name="Tenant1">
  <fvAp name="ap1">
   <fvAEPg name="epg2">
    <vnsSvcPol ctrct="ctrct_lb" graph="Graph_ADC" node="ADC">
     <vnsFolderInst ctrctNameOrLbl="ctrct_lb" graphNameOrLbl="Graph_ADC"
nodeNameOrLbl="ADC" key="lb" name="lb">
```

```
      <vnsRsFolderInstToMFolder
tDn="uni/infra/mDev-CISCO-CloudMode-1.0/mFunc-LB/mFolder-lb"/>
      <vnsFolderInst ctrctNameOrLbl="ctrct_lb" graphNameOrLbl="Graph_ADC"
nodeNameOrLbl="ADC" key="network" name="network">
       <vnsFolderInst ctrctNameOrLbl="ctrct_lb" graphNameOrLbl="Graph_ADC"
nodeNameOrLbl="ADC" key="interface" name="interface">
        <vnsFolderInst ctrctNameOrLbl="ctrct_lb" graphNameOrLbl="Graph_ADC"
nodeNameOrLbl="ADC" key="internal" name="internal">
         <vnsFolderInst ctrctNameOrLbl="ctrct_lb" graphNameOrLbl="Graph_ADC"
nodeNameOrLbl="ADC" key="ip" name="ip">
          <vnsParamInst name="ipaddress" key="ipaddress" value="2.2.2.2"/>
          <vnsParamInst name="netmask" key="netmask" value="255.255.255.0"/>
         </vnsFolderInst>
        </vnsFolderInst>
        <vnsFolderInst ctrctNameOrLbl="ctrct_lb" graphNameOrLbl="Graph_ADC"
nodeNameOrLbl="ADC" key="external" name="external">
         <vnsFolderInst ctrctNameOrLbl="ctrct_lb" graphNameOrLbl="Graph_ADC"
nodeNameOrLbl="ADC" key="ip" name="ip">
          <vnsParamInst name="ipaddress" key="ipaddress" value="1.1.1.1"/>
          <vnsParamInst name="netmask" key="netmask" value="255.255.255.0"/>
         </vnsFolderInst>
        </vnsFolderInst>
       </vnsFolderInst>
      </vnsFolderInst>
      <vnsFolderInst ctrctNameOrLbl="ctrct_lb" graphNameOrLbl="Graph_ADC"
nodeNameOrLbl="ADC" key="service" name="service">
       <vnsFolderInst ctrctNameOrLbl="ctrct_lb" graphNameOrLbl="Graph_ADC"
nodeNameOrLbl="ADC" key="vip" name="vip1">
        <vnsParamInst name="lbmethod" key="lbmethod" value="LEAST_CONNECTIONS"/>
        <vnsParamInst name="protocol" key="protocol" value="TCP"/>
        <vnsParamInst name="ipaddress" key="ipaddress" value="3.3.3.3"/>
        <vnsParamInst name="port" key="port" value="80"/>
        <vnsFolderInst ctrctNameOrLbl="ctrct_lb" graphNameOrLbl="Graph_ADC"
nodeNameOrLbl="ADC" key="pool" name="pool1">
         <vnsParamInst name="port" key="port" value="80"/>
        </vnsFolderInst>
       </vnsFolderInst>
      </vnsFolderInst>
     </vnsFolderInst>
    </vnsSvcPol>
   </fvAEPg>
  </fvAp>
</fvTenant>
```