Contrail Networking Architecture Guide

Detailed Technical Description of the Contrail Virtual Networking and Security Platform

– Contrail Virtual Networking and Security Platformの詳細な技術説明 – [機械翻訳版]

2019年11月

オリジナル資料リンク

Juniper Networks,Inc.1133

イノベーションウェイ

カリフォルニア州

Sunnyvale94089USA

408-745-2000

www.juniper.net

本書の情報は、タイトルページの日付時点のものです。

2000年通知
Juniper Networksのハードウェアおよびソフトウェア製品は、Year2000に準拠しています。Junos OSには、2038年までの時間関連の既知の制限はありません。しかし、NTPアプリケーションは、2036年にはある程度の困難性をもつことが知られています。

エンドユーザー使用許諾契約
この技術ドキュメントの対象となるJuniper Networks製品は、Juniper Networksソフトウェアで構成されている(または使用を目的としています)製品です。当該ソフトウェアの使用は、
https://support.juniper.net/support/eula/に掲載されているエンドユーザー使用許諾契約書(「EULA」)の条件に従います。このようなソフトウェアをダウンロード、インストール、または使用することにより、ユーザーはそのEULAの利用規約に同意することになります。

目次

序文

このドキュメントでは、Contrail Networkingが、さまざまな仮想マシンおよびコンテナオーケストレータと連携し、物理ネットワークおよびコンピューティングインフラストラクチャと統合できるスケーラブルな仮想ネットワークプラットフォームを提供する方法について説明します。これにより、利用者は、既存のインフラストラクチャ、手順、およびワークロードを保持しながら、オープンソースのオーケストレーションを利用できます。これにより、中断やコストが軽減されます。

仮想化がパブリッククラウドサービスとプライベートクラウドサービスの両方を提供するための鍵となる技術になるにつれて、ネットワークスケールの問題は、現在まで広く使用されてきた仮想化技術(例えば、L2ネットワーキングを備えたVMwareや、ストックNova、Neutron、またはML2ネットワーキングを備えたOpenStack)によって明らかになりつつあります。Contrail Networkingは、複数のオーケストレータを同時にサポートしながら、最大の環境でマルチテナントネットワークをサポートするように設計された、高度にスケーラブルな仮想ネットワークプラットフォームを提供します。

本当に「グリーンフィールド」であるデータセンターのデプロイは非常に少ないため、新しいインフラストラクチャにデプロイされたワークロードと、以前にデプロイされたワークロードおよびネットワークを統合するための要件はほぼ常に存在します。このドキュメントでは、新しいクラウドインフラストラクチャがデプロイされ、既存のインフラストラクチャとの共存が必要なデプロイメントの一連のシナリオについて説明します。

ユースケース

このドキュメントでは、次の一般的なユースケースについて説明します:

- OpenStackで管理されるデータセンターで高いスケーラビリティと柔軟性を備えたPlatform-as-a-ServiceとSoftware-as-a-Serviceの有効化
- Red Hat OpenShiftを含む、Kubernetesなどのコンテナー管理システムを使用した仮想ネットワーク
- VMware vCenterを実行する既存の仮想化環境で、仮想マシン間でContrail Networking仮想ネットワークを使用できるようにします
- BGPピアリングを備えたゲートウェイルータを使用して、Contrail Networking仮想ネットワークを物理ネットワークに接続します
- ゼロタッチプロビジョニング、基本構成、アンダーレイ構成、オーバーレイ構成など、IPファブリック内の装置のライフサイクル管理を提供します
- 接続されたスイッチでVXLAN VTEPを設定して、OSのプロビジョニングや仮想ネットワークへの接続など、ベアメタルサーバのライフサイクル管理を提供します。

これらのユースケースは、さまざまなデプロイシナリオで特定の要件に対応するために、任意の組合せでデプロイできます。以下の図1は、Contrail Networkingの主な機能領域を示しています。

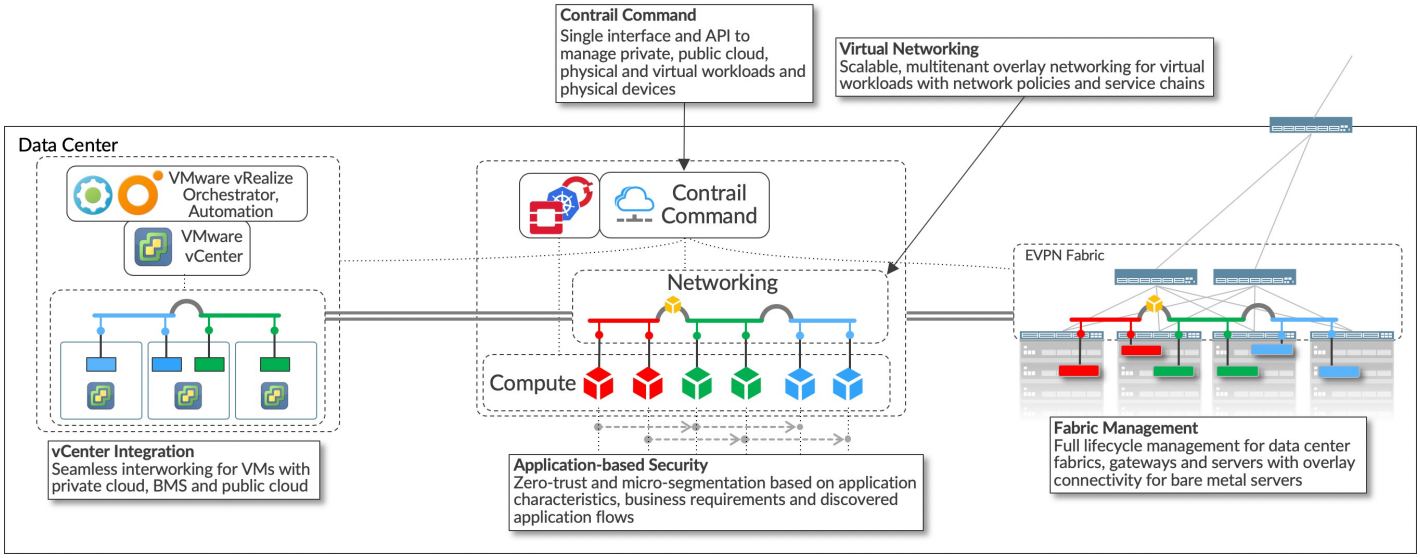

図1:Contrail Networkingの主な機能を示す高レベルの回路図

主なユースケースのサポートを可能にする主な機能領域は、次のとおりです:

- 仮想化ホスト間のカプセル化トンネルを使用した仮想ネットワーク
- 仮想マシンとコンテナのオープンソースオーケストレータ用のプラグイン
- タグに基づくアプリケーションベースのセキュリティポリシー
- VMwareオーケストレーションスタックとの統合
- データセンターファブリック内のスイッチとルータのライフサイクル管理

これらのユースケースでは同じコントローラーと転送コンポーネントが使用されるため、Contrail Networkingはサポートするすべての環境で接続を管理するための一貫したインターフェイスを提供でき、仮想マシン、コンテナ、ベアメタルサーバーに関係なく、さまざまなオーケストレーターによって管理されるワークロード間、および外部ネットワーク内の宛先へのシームレスな接続を提供できます。

Contrail Networking for Virtual Machines and Containers Contrail Networkingの主な機能は、OpenStack、Kubernetes、およびVMwareオーケストレーターを使用して、クラウド環境で仮想ネットワークを管理および実装します。

Contrail Networkingは、各ホストで実行されるvRouter間のオーバーレイネットワークを使用します。現在、世界の主要なサービスプロバイダのワイドエリアネットワークをサポートしていますが、仮想化されたワークロードとクラウド自動化を備えたデータセンターで作業するために再利用されている、実績のある標準ベースのネットワーク技術に基づいています。オーケストレータのネイティブネットワーキング実施よりも、次のような多くの拡張機能を提供します:

- 高度にスケーラブルなマルチテナントネットワーク

- マルチテナントIPアドレス管理
- ネットワークへのフラッディングを回避するためのDHCP、ARPプロキシ
- ローカルDNS解決
- アクセス・コントロール・リストを持つ分散ファイアウォール
- アプリケーションベースのセキュリティポリシー
- ホスト間の分散ロードバランシング
- ネットワークアドレス変換(1:1浮動IPおよびN:1SNAT)
- 仮想ネットワーク機能によるサービスチェーン
- デュアルスタックIPv4およびIPv6
- ゲートウェイルータとのBGPピアリング

次のセクションでは、コントローラがオーケストレータおよびvRouterとどのように相互作用するか、および上記の機能が各vRouterでどのように実装および設定されるかについて詳しく説明します。

Contrail Networkingによるファブリックおよびベアメタルサーバーの管理については、このドキュメントで後述するContrail Networkingのファブリック管理のセクションで説明します。

Contrail Networkingの仕組み

このセクションでは、各ホストでパケットを転送するContrail NetworkingコントローラとvRouterのソフトウェアアーキテクチャについて説明します。また、仮想マシンまたはコンテナが起動され、相互にパケットを交換するときのvRouterとContrail Networkingコントローラ間の相互作用についても説明します。

Orchestratorを使用したContrail Networkingの操作

Contrail Networkingは、2つの主要なソフトウェアで構成されています:

- *Contrail Networking Controller–ネットワークとネットワークポリシーのモデルを維持する一連のソフトウェアサービス。通常、高可用性のために複数のサーバで実行されます。*
- *Contrail Networking vRouter–各仮想化ホストにインストールされ、ネットワークポリシーとセキュリティポリシーを適用し、パケット転送を実行します。*

Contrail Networkingの一般的な展開を以下の図2に示します。

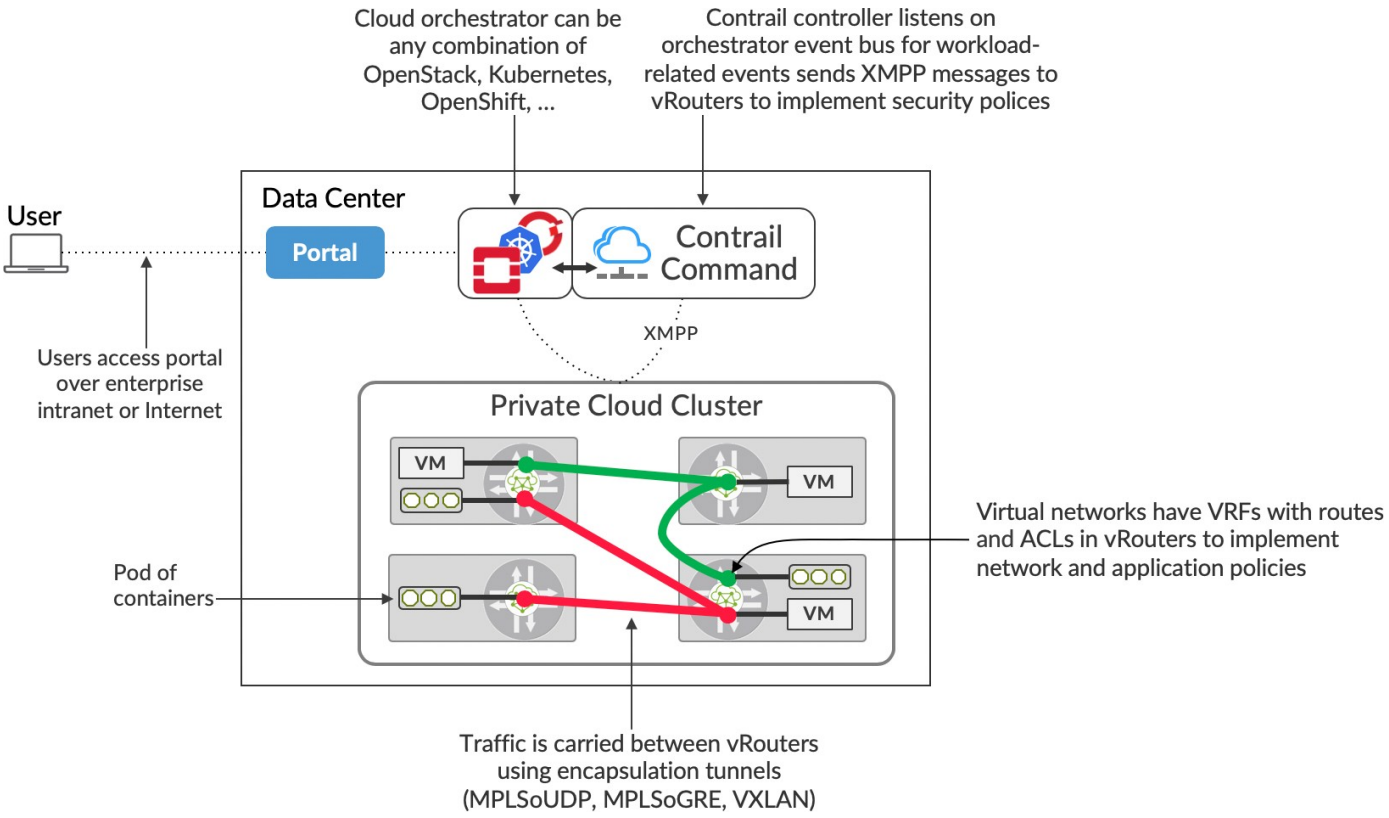

図2:Contrail Networkingは、仮想ワークロード間のカプセル化トンネルを使用します。

Contrail Networking Controllerは、OpenStackやKubernetesなどのクラウド管理システムと統合されており、その機能は、仮想マシン(VM)またはコンテナーが作成されるときに、コントローラまたはOrchestratorで指定されたネットワークポリシーに従ってネットワーク接続が確立されるようにすることです。

Contrail Networkingコントローラは、Orchestratorのネットワークサービスを実装するソフトウェアプラグインを介してOrchestratorと統合されます。たとえば、OpenStackのContrail Networkingプラグインは Neutron APIを実装し、kube-network-managerおよびCNI(コンテナーネットワークインタフェース)部品は Kubernetes(K8s)APIを使用してネットワーク関連の場合をリッスンします。

Contrail Networking vRouterは、コンピュートホストでLinuxブリッジとiptablesユーティリティ、または Open vSwitchネットワーキングを置き換え、コントローラは目的のネットワーキングおよびセキュリティ ポリシーを実装するようにvRouterを設定します。

別のホストで実行されている宛先を持つ1台のホスト上の仮想マシンからのパケットは、UDP経由のMPLS、GRE経由のMPLS、またはVXLANでカプセル化されます。外部ヘッダーの宛先は、宛先仮想マシンが実行されているホストのIPアドレスです。コントローラは、ネットワークポリシーを実装する各vRouterの各VRFに一連のルートをインストールする責任があります。

たとえば、デフォルトでは、同じネットワーク内の仮想マシンは相互に通信できますが、ネットワークポリシーで特に有効になっていない限り、異なるネットワーク内の仮想マシンとは通信できません。コントローラとvRouter間の通信は、広く使用されており、柔軟性のあるメッセージングプロトコルであるXMPPを介して行われます。

クラウド自動化の重要な機能は、利用者がリソースの提供方法や提供場所についての詳細を理解する必要なく、アプリケーションのリソースを要求できることです。これは通常、ユーザが選択できる一連のサービス提供物を提示し、ユーザの要件に応じて必要なメモリ、ディスク、およびCPU容量で仮想マシンまたはコンテナをスピンアップするために、APIコールをクラウドオーケストレータを含む基盤となるシステムに変換するポータルを介して行われます。サービス提供は、特定のメモリ、ディスク、およびCPUが割り当てられたVMと同じくらいシンプルであるか、事前設定された複数のソフトウェアインスタンスで構成されたアプリケーションスタック全体を含むことができます。

Orchestratorとのインタラクション

Contrail NetworkingコントローラとvRouterのアーキテクチャ、およびOrchestratorとのインタラクションを以下の図3に示します。

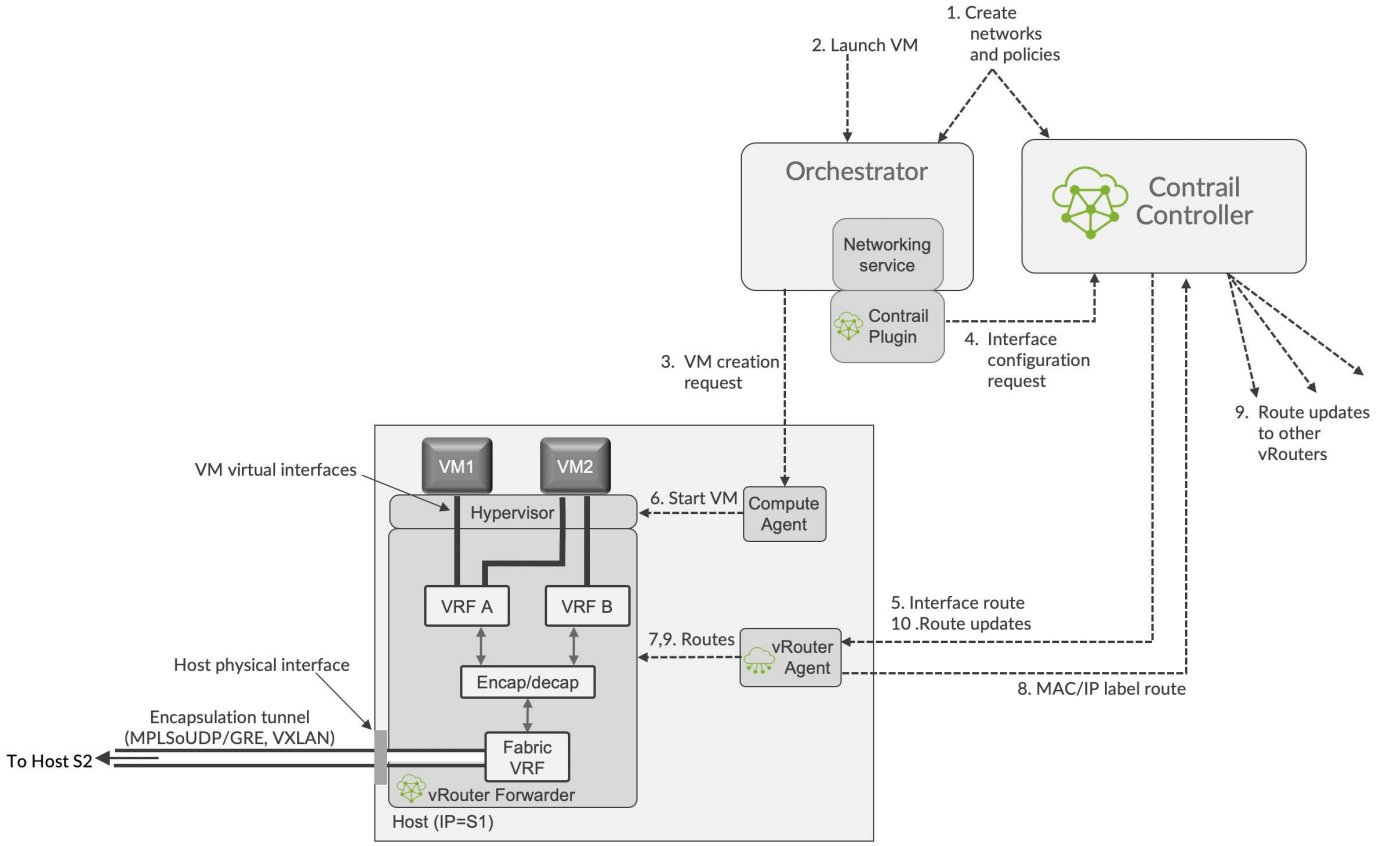

図3:オーケストレータ、Contrail Networking Controller、およびContrail Networking vRouter間の相互作用

ホスト上で実行されている仮想マシンの各インターフェイスは、そのインターフェイスのIPアドレスを含む対応するネットワークの転送テーブルを含むVRFに接続されます。vRouterには、ホストの物理インターフェースに接続するファブリックVRFを含め、そのホストにインターフェイスがあるネットワークのVRFのみがあります。Contrail Networking仮想ネットワークでは、カプセル化トンネルを使用して異なるホスト上のVM間でパケットを転送し、ファブリックVRFとVM VRF間でカプセル化とカプセル化解除が行われます。これについては、次のセクションで詳しく説明します。

新しい仮想ワークロードが作成されると、場合がプラグインに表示され、コントローラに送信されます。コントローラは、仮想ネットワークのVRFにインストールされるルートの要求をエージェントに送信し、エージェントはフォワーダで設定します。

単一のインターフェイスを持つ新しい仮想マシンでネットワークを構成するための論理フローは、次のとおりです:

1. ネットワークおよびネットワークポリシーは、UI、CLI、またはREST APIを使用して、オーケストレータまたはContrail Networkingのいずれかで定義されます。ネットワークは、仮想マシンの作成時にインターフェイスに割り当てられるIPアドレスのプールとして主に定義されます。
2. 仮想マシンは、オーケストレータの利用者が起動するように要求されます。これには、そのインターフェイスがどのネットワークにあるかも含まれます。
3. オーケストレータは、新しい仮想マシンを実行するホストを選択し、そのホスト上のコンピューティングエージェントにそのイメージをフェッチして仮想マシンを起動するように指示します。
4. Contrail Networkingプラグインは、orchestratorのネットワークサービスから場合またはAPIコールを受信し、起動する新しいVMのインターフェイスのネットワークを設定するように指示します。これらの命令はContrail Networking REST呼び出しに変換され、Contrail Networkingコントローラに送信されます。
5. Contrail Networkingコントローラは、指定された仮想ネットワークに接続する新しいVM仮想インターフェイスの要求をvRouterエージェントに送信します。vRouterエージェントは、仮想ネットワークのVRFにVMインターフェースを接続するようにvRouter Forwarderに指示します。VRFが作成され(提示しない場合)、インターフェースが接続されます。
6. コンピューティングエージェントは、仮想マシンを起動します。仮想マシンは通常、DHCPを使用して各インターフェイスのIPアドレスを要求するように設定されます。vRouterは、DHCPリクエストをプロキシ処理し、インターフェースIP、デフォルトゲートウェイ、およびDNSサーバアドレスで応答します。
7. インターフェイスがアクティブになり、DHCPからのIPアドレスがあると、vRouterはVM仮想イ

ンターフェイスのネクストホップを使用して、仮想マシンのIPアドレスとMACアドレスへのルートをインストールします。

8. vRouterはインターフェイスにラベルを割り当て、MPLSテーブルにラベルルートをインストールします。vRouterは、新しい仮想マシンへのルートを含むXMPPメッセージをコントローラに送信します。

   ルートには、vRouterが実行されているサーバのIPアドレスのネクストホップがあり、割り当てられたばかりのラベルを使用してカプセル化プロトコルを指定します。

9. コントローラは、ネットワークポリシーで許可されているように、同じネットワーク内および他のネットワーク内の仮想マシンを持つ他のvRouterに新しい仮想マシンへのルートを配布します。

10. コントローラは、ポリシーで許可されているように、他の仮想マシンのルートを新しい仮想マシンのvRouterに送信します。

この手順の最後に、新しい仮想マシンを考慮して、データセンター内のすべてのvRouterのVRF内のルートが、設定されたネットワークポリシーを実装するように更新されました。

vRouterのアーキテクチャの詳細

このセクションでは、Contrail Networking vRouterのアーキテクチャについて詳しく説明します。Contrail Networking vRouterの機能コンポーネントの概念図を以下の図4に示します。

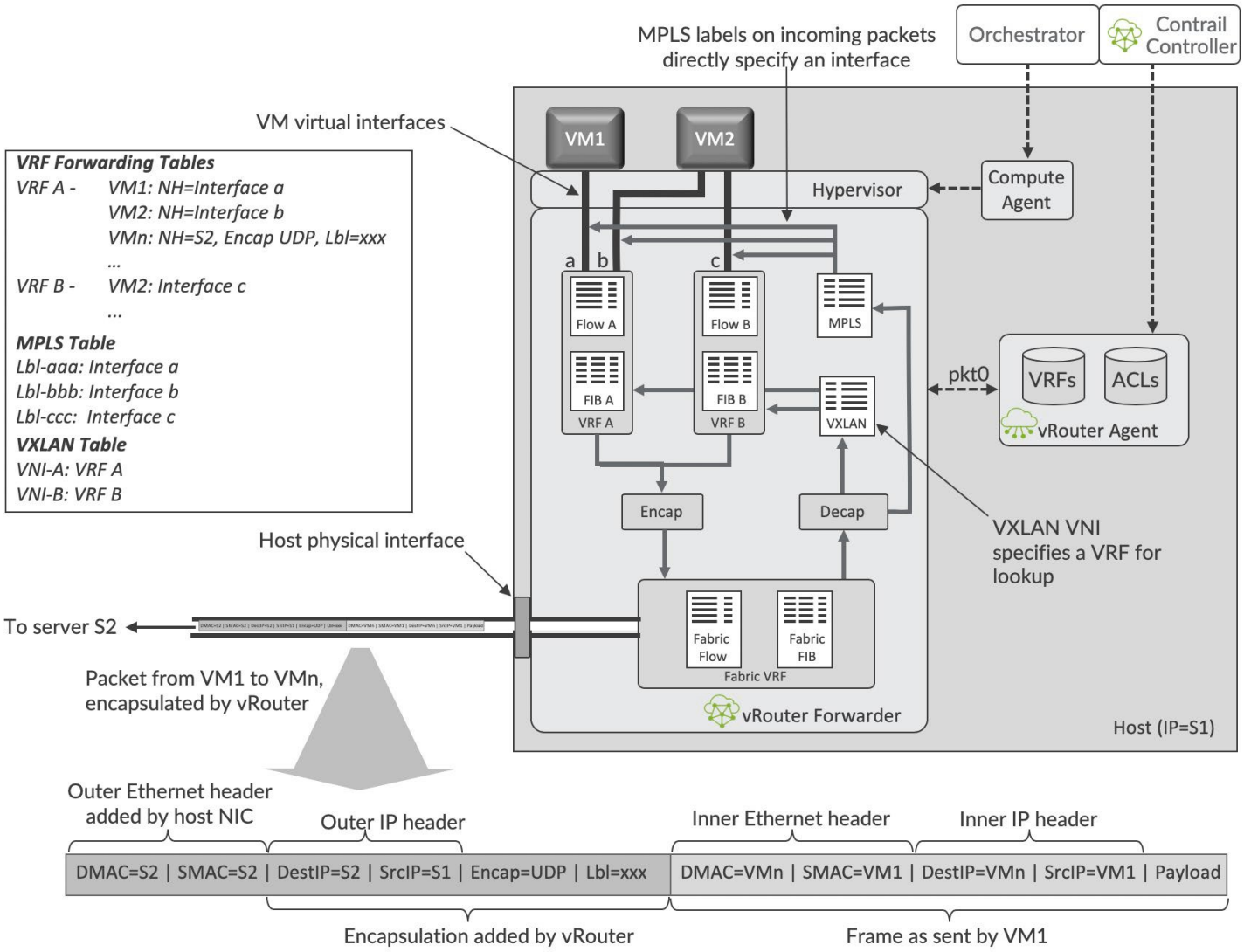

図4:コンピュートノード上のOpenStackおよびContrail Networkingエージェント

vRouterエージェントはホストオペレーティングシステムのユーザー空間で実行されますが、フォワーダはDPDKが使用されている場合はカーネルモジュールとして、ユーザー空間で実行することも、「スマートNIC」とも呼ばれるプログラム可能なネットワークインターフェイスカードで実行することもできます。これらのオプションの詳細については、「vRouterの展開オプション」セクションを参照してください。最も一般的に使用されるカーネルモジュールオプションをここに示します。

エージェントはコントローラとのセッションを維持し、必要なVRF、ルート、およびアクセスコントロールリスト(ACL)に関する情報を送信します。エージェントは、その情報を独自のデータベースに保存し、その情報を使用してフォワーダを設定します。インターフェースがVRFに接続され、各VRFの中継情報ベース(FIB)に中継エントリが設定されます。

各VRFには独自の転送テーブルとフローテーブルがあり、MPLSテーブルとVXLANテーブルはvRouter内でグローバルです。転送テーブルには、目的地のIPおよびMACアドレスの両方のルートが含まれており、IP to MAC協会が代理ARP機能を提供します。

MPLSテーブルのラベルの値は、VMインターフェイスが起動したときにvRouterによって選択され、そのvRouterに対してローカルでのみ有効です。VXLANネットワーク識別子は、Contrail Networkingドメイン内の異なるvRouter内の同じ仮想ネットワークのすべてのVRFにわたってグローバルです。

vRouterでの詳細なパケット処理ロジック

VMからVMに流れるパケットのロジック詳細は、図5と図6で説明されているように、若干異なります。

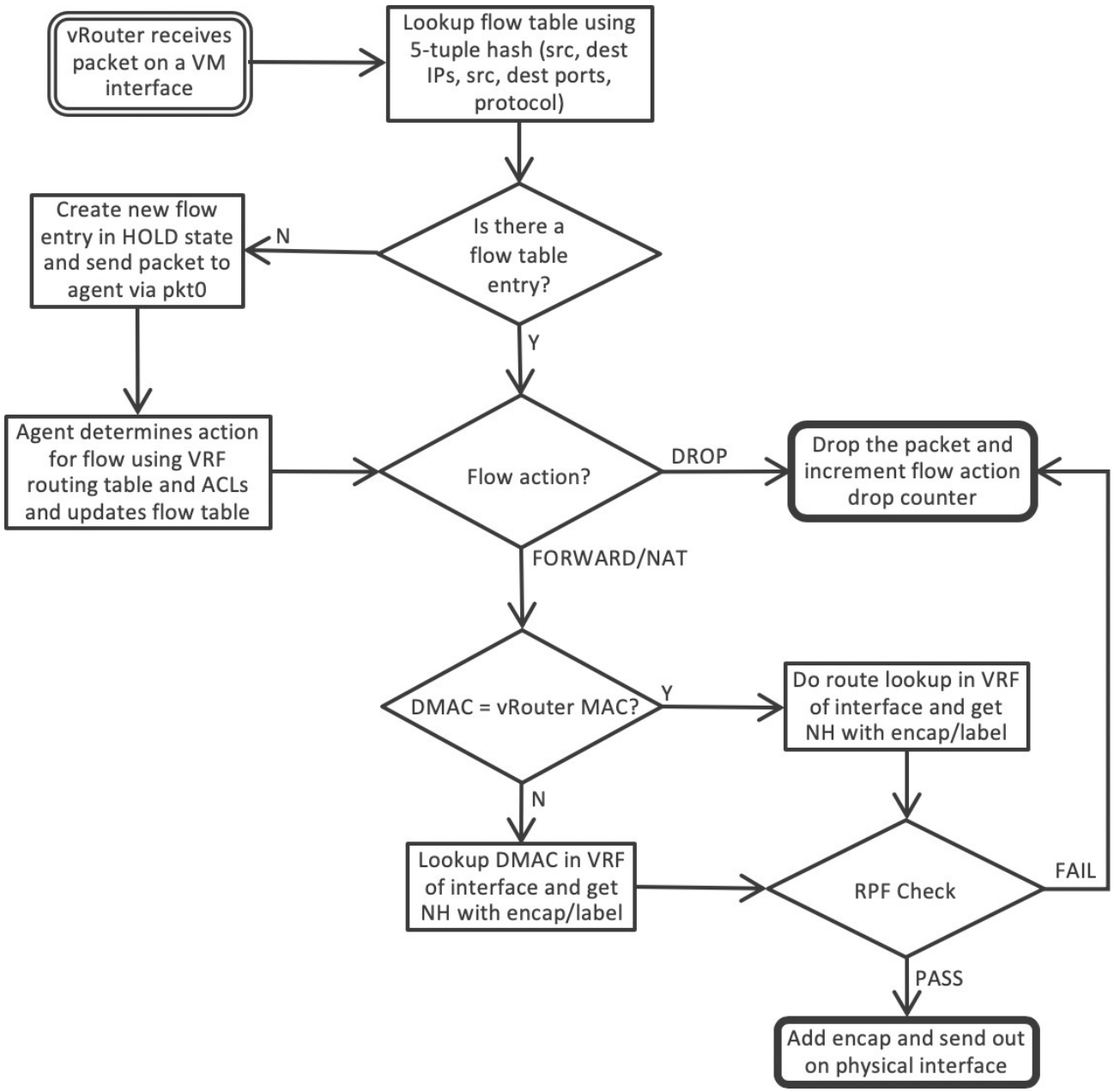

図5:VMインターフェースからvRouterに到着するパケットのロジック

パケットが仮想インタフェースを介してVMから送信されると、フォワーダによって受信されます。フォワーダは、インタフェースが存在するVRFのフローテーブルにパケットの5タプル(プロトコル、送信元と宛先のIPアドレス、送信元と宛先のTCPポートまたはUDPポート)に一致するエントリがあるかどうかを最初にチェックします。これがフローの最初のパケットである場合、エントリは存在せず、フォワーダはpkt0インターフェースを介してパケットをエージェントに送信します。エージェントは、VRFルーティングテーブルとアクセスコントロールリストに基づいてフローのアクションを決定し、その結果でフローテーブルを更新します。アクションには、DROP、FORWARD、NATがあります。パケットが転送される場合、転送者は宛先MACアドレスが自身のMACアドレスであるかどうかを確認します。これは、宛先がVMのサブネット外にあるときにVMがデフォルトゲートウェイにパケットを送信している場合に該当します。その場合、宛先のネクストホップはIPフォワーディングテーブルで検索され、そうでない場合はMACアドレスがルックアップに使用されます。

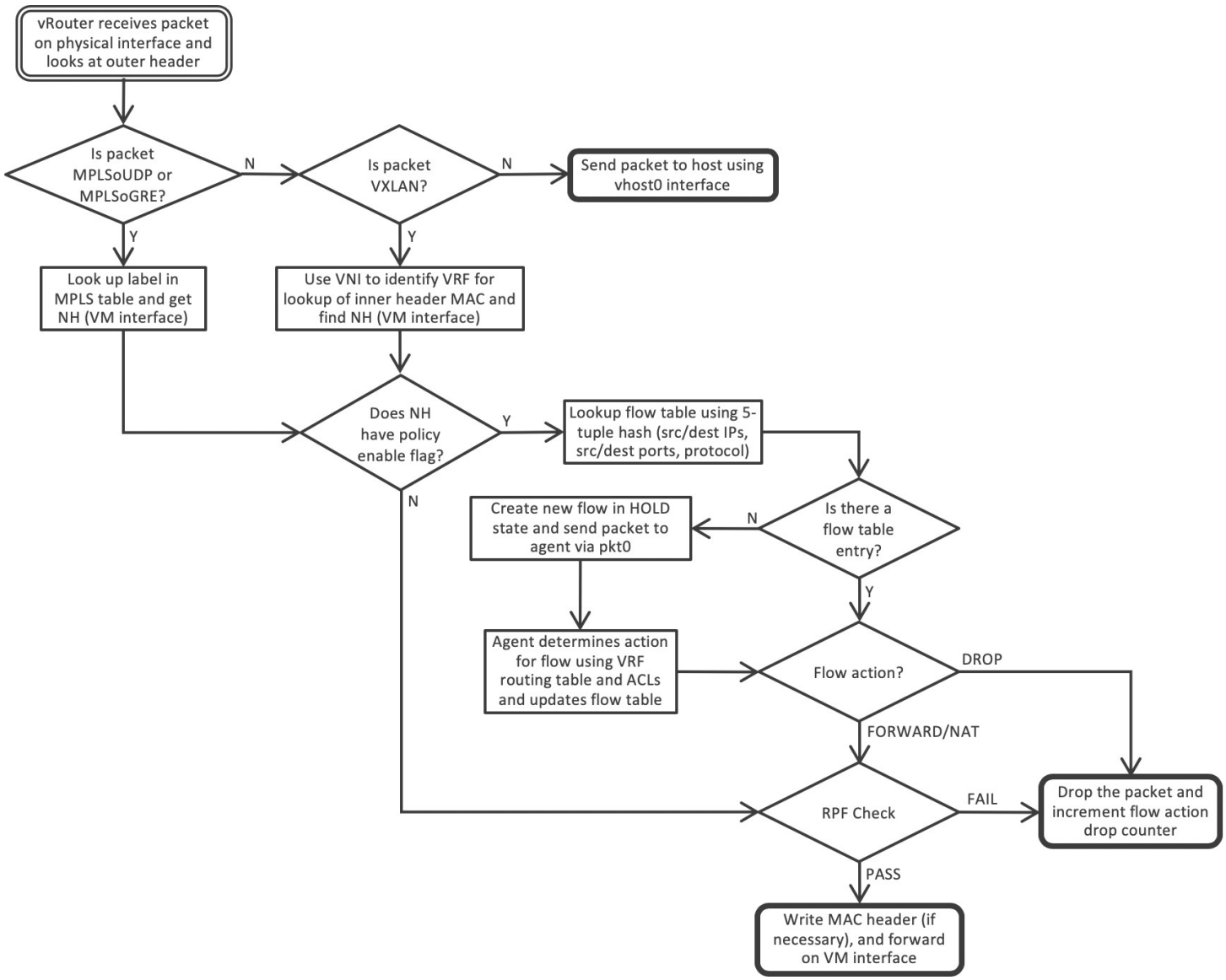

図6:物理ネットワークからvRouterに到着するパケットのロジック

パケットが物理ネットワークから到着すると、vRouterは最初にパケットがサポートされているカプセル化を持っているかどうかを確認します。そうでない場合、パケットはホストオペレーティングシステムに送信されます。MPLS over UDPおよびMPLS over GREの場合、ラベルはVMインターフェースを直接識別しますが、VXLANでは、内部ヘッダーの宛先MACアドレスをVXLANネットワーク識別子(VNI)によって識別されるVRFで検索する必要があります。インターフェイスが識別されると、vRouterは、インターフェイスにポリシーフラグが設定されていない場合(すべてのプロトコルとすべてのTCP/UDPポートが許可されていることを示します)、すぐにパケットを転送できます。それ以外の場合、5タプルはフローテーブル内のフローを検索するために使用され、発信パケットについて説明されているものと同じロジックが使用されます。

同じサブネット内の仮想マシン間のパケットフロー

次の図は、仮想マシンが最初に別の仮想マシンにパケットを送信したときに発生する一連のアクションを示しています。開始点は、両方の仮想マシンが起動し、コントローラが両方のvRouterにL2(MAC)およびL3(IP)ルートを送信して仮想マシン間の通信を可能にしたことです。

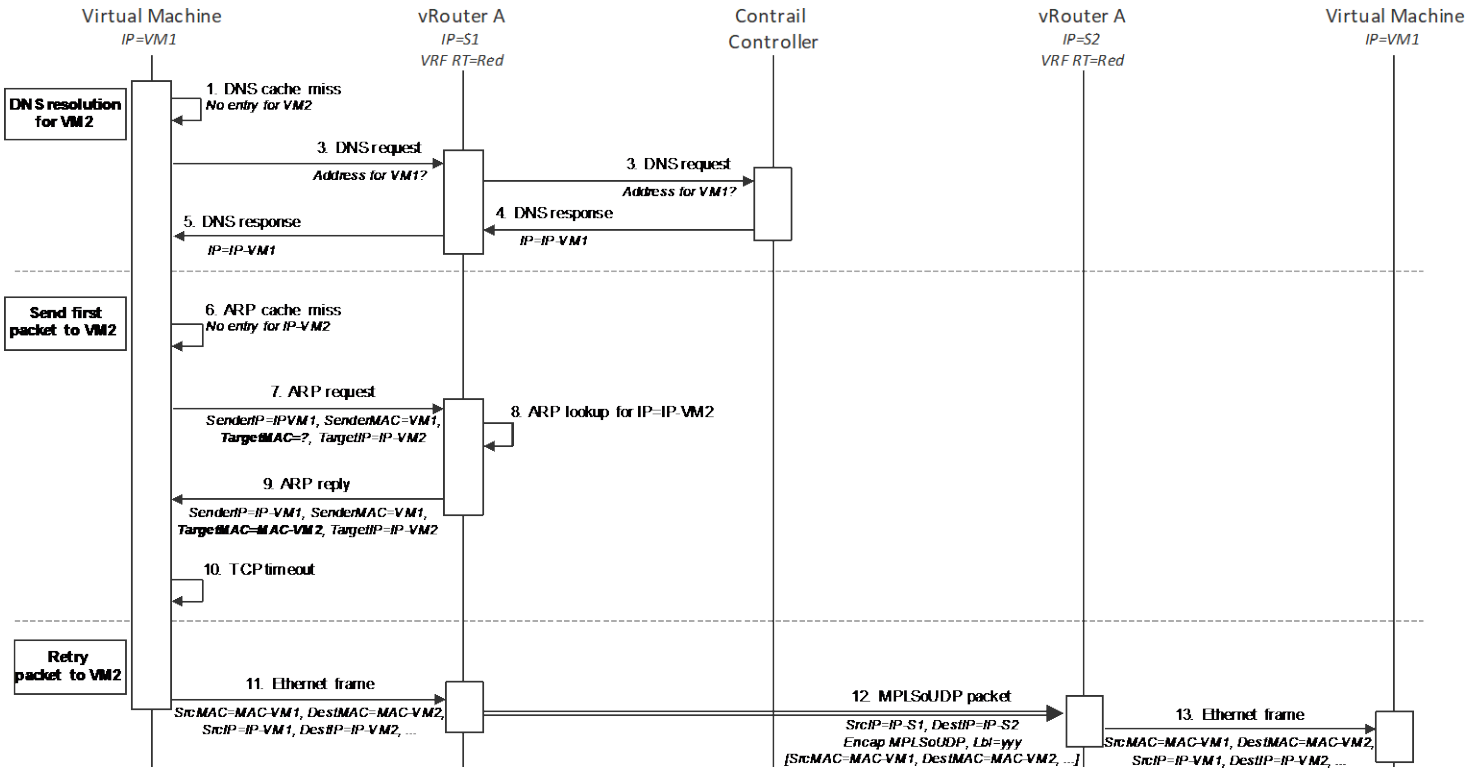

図7:仮想マシンが別の仮想マシンにパケットを送信するシーケンス

1. VM1はVM2にパケットを送信する必要があるため、最初にIPアドレスの独自のDNSキャッシュを検索しますが、これは最初のパケットであるため、エントリはありません。
2. VM1は、インターフェースの起動時にDHCP応答で指定されたDNSサーバアドレスにDNS要求を送信します。

3. vRouterはDNS要求をトラップし、Contrail Networkingコントローラで実行されているDNSサーバに転送します。
4. コントローラ内のDNSサーバは、VM2のIPアドレスで応答します。
5. vRouterは、DNS応答をVM1に送信します。
6. VM1はイーサネットフレームを形成する必要があるため、VM2のMACアドレスが必要です。独自のARPキャッシュをチェックしますが、最初のパケットであるため、エントリはありません。
7. VM1はARP要求を送信します。
8. vRouterはARP要求をトラップし、それ自身の転送表でIP-VM2のMACアドレスを検索し、VM2のために送信したL2/L3ルートで関連性を見つけます。
9. vRouterは、VM2のMACアドレスを持つARP応答をVM1に送信します。
10. VM1のネットワークスタックでTCPタイムアウトが発生します。
11. VM1のネットワークスタックはパケットの送信を再試行し、このときARPキャッシュ内のVM2のMACアドレスを検出し、イーサネットフレームを形成して送信できます。
12. vRouterはVM2のMACアドレスを検索し、カプセル化ルートを見つけます。vRouterは外側のヘッダーを構築し、結果のパケットをS2に送信します。
13. S2のvRouterはパケットをカプセル化解除し、MPLSラベルを検索して、元のイーサネットフレームを送信する仮想インターフェイスを識別します。イーサネットフレームはインターフェイスに送信され、VM2によって受信されます。

異なるサブネット内の仮想マシン間のパケットフロー

異なるサブネットの宛先にパケットを送信する場合のシーケンスは同じですが、VM1はデフォルトゲートウェイのMACアドレスを持つイーサネットフレームでパケットを送信し、そのIPアドレスは、vRouterがVM1のブート時に提供したDHCP応答で提供されたものです。VM1がゲートウェイIPアドレスのARP要求を実行すると、vRouterは独自のMACアドレスで応答します。VM1がそのゲートウェイMACアドレスを使用してイーサネットフレームを送信すると、vRouterはフレーム内のパケットの宛先IPアドレスを使用してVRF内の転送テーブルを検索し、宛先VMが実行されているホストへのカプセル化トンネルを介しているルートを見つけます。

サービスチェーン

サービスチェーンは、ネットワークポリシーによって、2つのネットワーク間のトラフィックが1つ以上のネットワークサービス(仮想ネットワーク機能(VNF)とも呼ばれます)を通過する必要があることが指定されている場合に形成されます。ネットワークサービスは、Contrail Networkingでサービスとして識別される仮想マシンに実装され、ポリシーに含まれます。Contrail Networkingは、OpenStack環境とvCenter環境の両方でサービスチェーンをサポートします。2つの仮想マシン間のサービスチェーンの概念を図8に示します。

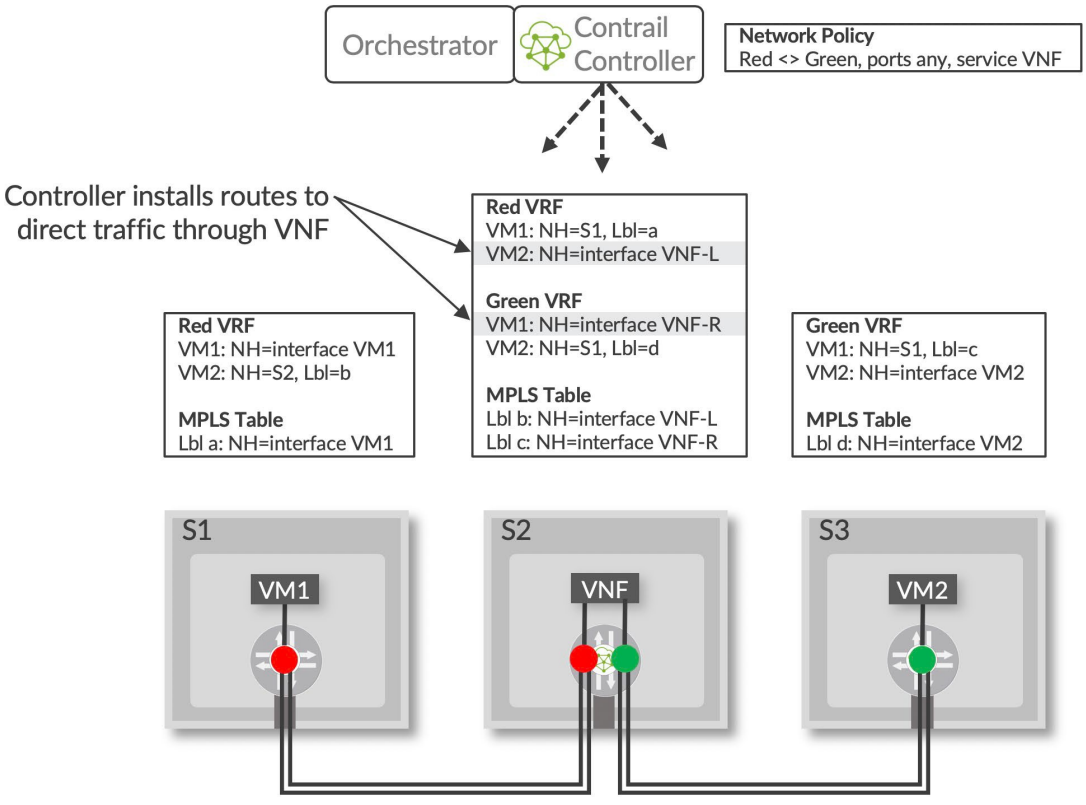

図8:特別なルートは、サービスチェーン内のVNFを介してトラフィックをダイレクトします。

VMがサービスインスタンス(VNF)になるようにコントローラで構成されていて、サービスがネットワークポリシーに含まれている場合、コントローラはVNFを介してトラフィックを転送するVNFの「左」および「右」インターフェイスのVRFにルートをインストールします。カプセル化ルートがVNF vRouterによってコントローラに戻されると、ルートはRedおよびGreen VRFを持つ他のvRouterに配布され、最終結果として、RedおよびGreenネットワーク間を流れるトラフィックをサービスインスタンスを通過させるルートのセットになります。ラベル「Left」および「Right」は、VNFのブート時にアクティブになる順序に基づいてインターフェイスを識別するために使用されます。VNFには、到着するインターフェイスに基づいてパケットを適切に処理する設定が必要です。

Contrail Networkingで実装されているように、サービスチェーンルートは、わかりやすくするためにここでは示していないですが、原則は同じである特殊なVRFにインストールされます。

各種サービスチェーンのシナリオを以下の図9に示し、それぞれの簡単な説明を次に示します。

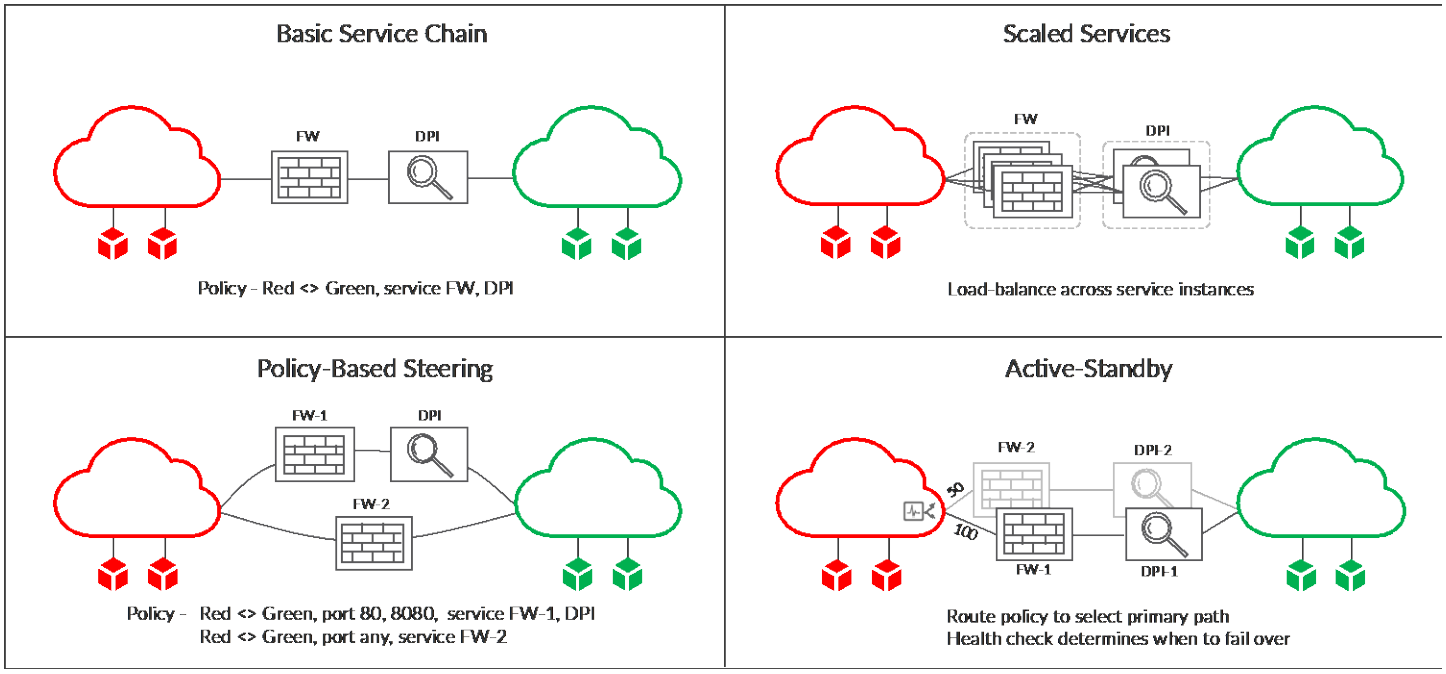

図9:Contrail Networkingで接続されたコンテナ

基本的なサービスチェーン

最初のパネルでは、RedネットワークとGreenネットワーク間のネットワークポリシーを編集してサービスFWとDPIを含めることで、単純なサービスチェーンが作成されています。これらは、OpenStackまたはvCenterで以前に起動され、Contrail NetworkingでRedおよびGreenネットワークのインターフェイスを持つサービスインスタンスとして設定された仮想マシンです。ポリシーが保存され、2つのネットワークに適用されると、Red仮想マシンまたはGreen仮想マシンが接続されたすべてのvRouterのルートが変更され、サービスチェーン経由で2つのネットワーク間のトラフィックが送信されます。たとえば、ポリシーを変更する前に、Redネットワーク内の各VRFには、仮想マシンが実行されているホストのネクストホップと、ホストvRouterによって指定され、コントローラによって送信されたラベルを持つ、Greenネットワーク内の各仮想マシンへのルートがあります。ルートは、FWサービスインスタンスの入力VRFのネクストホップと、FW leftインターフェースに指定されたラベルを持つように変更されます。権FWインターフェースを持つVRFには、DPIの左インターフェースを指すすべての緑の宛先のルートが含まれ、権DPIインターフェースを持つVRFには、実行されているホストのネクストホップと元のラベルを持つすべての緑の宛先のルートが含まれます。逆方向のトラフィックのルーティングも同様に処理されます。

スケールアウトサービス

単一の仮想マシンにサービスチェーンのトラフィック要件を処理するキャパシティがない場合は、2番目のパネルに示すように、同じタイプの複数の仮想マシンをサービスに含めることができます。

これが行われると、トラフィックは両端のサービスチェーンの入力インターフェースフェイス間でECMPを使用してロードバランシングされ、チェーンのレイヤ間でもロードバランシングされます。

新しいサービスインスタンスは、Contrail Networkingで必要に応じて追加できます。また、ECMPハッシュアルゴリズムは通常、ターゲットの数が変更されるとほとんどのセッションを他の経路に移動しますが、Contrail Networkingでは、既存のフローの経路は「vRouterでのパケット処理ロジックの詳細」セクションで説明されているフローテーブルから決定されるため、これは新しいフローに対してのみ発生します。この挙動は、フロー内のすべてのパケットを表示する必要があるステートフルサービスに不可欠です。そうしないと、フローがブロックされ、ユーザセッションがドロップされます。

フローの逆方向のトラフィックが、フローの元と同じサービスインスタンスを通過するように、フローテーブルにもデータが入力されます。

https://datatracker.ietf.org/doc/draft-ietf-bess-service-chaining/のインターネットドラフトには、ステートフルサービスを備えたスケールアウトされたサービスチェーンの詳細が記載されています。

ポリシーベースのステアリング

異なるタイプのトラフィックを異なるサービスチェーンに渡す必要がある場合があります。これは、ネットワークまたはセキュリティポリシーに複数の条件を含めることによって、Contrail Networkingで実現できます。図の例では、ポート80と8080のトラフィックはファイアウォール(FW-1)とDPIの両方を通過する必要がありますが、他のすべてのトラフィックはファイアウォール(FW-2)を通過するだけで、FW-1とは異なる構成になっている可能性があります。

アクティブスタンバイサービスチェーン

一部のシナリオでは、トラフィックは通常、特定のサービスチェーンを通過することが望まれますが、そのチェーンで問題が検出された場合は、トラフィックをバックアップに切り替える必要があります。これは、スタンバイサービスチェーンがあまり好まれない地理的な場所に配置されている場合があります。

アクティブ/スタンバイ構成は、Contrail Networkingの2つのステップで実現されます。最初に、ルートポリシーは、優先されるアクティブチェーンのイングレスに対してより高いローカルプリファレンス値を指定して、各サービスチェーンのイングレスに適用されます。次に、サービスインスタンスが到達可能であるかどうか、またはチェーンの反対側の宛先に到達できるかどうかをテストできるヘルスチェックが各チェーンにアタッチされます。ヘルスチェックが失敗すると、通常アクティブなサービスチェーンへのルートが取り消され、トラフィックはスタンバイを通過します。

アプリケーションベースのセキュリティポリシー

従来のファイアウォールポリシーには、個々のIPアドレスまたはサブネット範囲に基づくルールが含まれています。あらゆる規模のデータセンターにおいて、これは、ファイアウォールルールの普及につながります。これは、作成時には管理が困難で、トラブルシューティング時には理解が困難です。

これは、サーバーまたはVMのIPアドレスが、アプリケーション、アプリケーション所有者、場所、またはその他の財産権に関連していないためです。たとえば、以下の図10に示すように、2つのデータ・センターがあり、開発および本番環境に3層アプリケーションをデプロイするエンタープライズについて考えてみます。

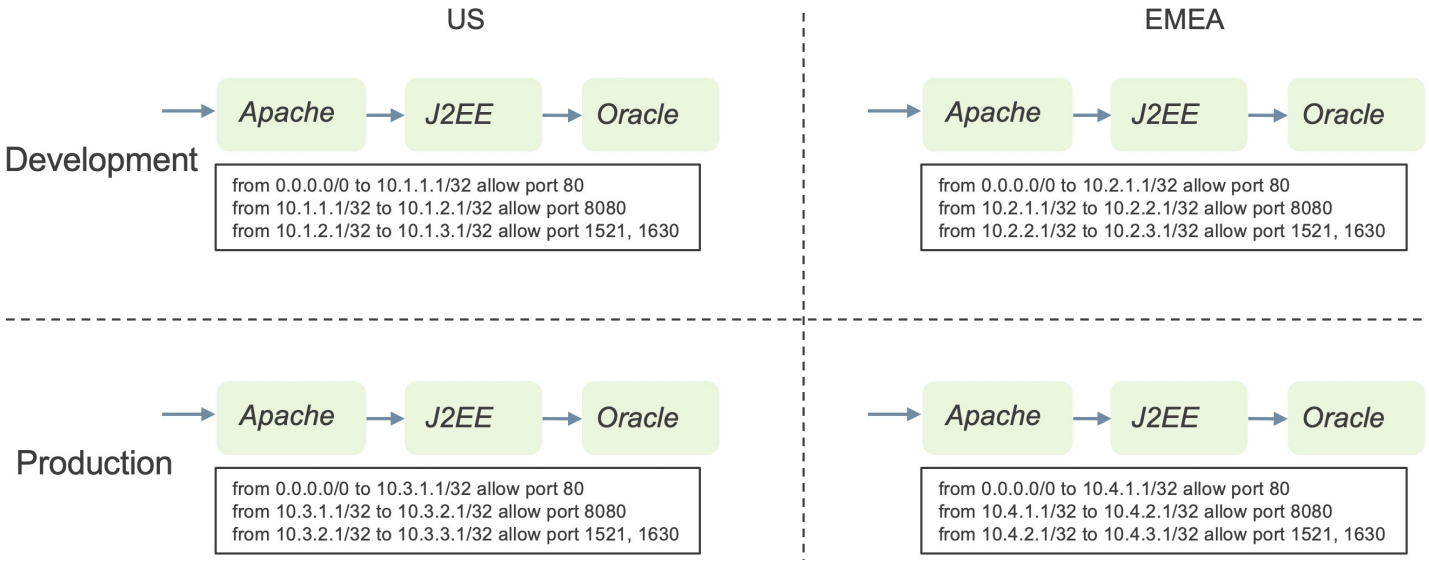

図10:アプリケーションスタックの複数のインスタンスには、複数のファイアウォールルールが必要です。

このエンタープライズでは、アプリケーションの各インスタンスのレイヤーは、同じインスタンス内の次のレイヤーとのみ通信同一のことが要件です。これには、次に示すように、アプリケーションインスタンスごとに個別のポリシーが必要です。問題をトラブルシューティングする場合、管理者はIPアドレスとアプリケーションインスタンスの関係を把握している必要があり、新しいインスタンスがデプロイされるたびに、新しいファイアウォールルールを記述する必要があります。

Contrail Networkingコントローラは、プロジェクト、ネットワーク、vRouters、VM、およびインターフェースに適用可能なタグに基づいてセキュリティポリシーをサポートします。タグは、オブジェクトモデル内で、タグが適用されたオブジェクトに含まれるすべてのオブジェクトにプロパゲートされます。タグには名前と値があります。Contrail Networkingディストリビューションの一部として、いくつかのタグ名が提供されています。タグタイプの一般的な用途を以下の表に示します:

タグ名	一般的な使用法	例
アプリケーション	エンドユーザーまたはその他のサービスによってアクセスされるサービスをサポートするために、さまざまなタイプのソフトウェアインスタンスのセットを実行する仮想マシンのグループを特定します。ヒートスタックに対応できます。	LAMPスタック、Hadoopクラスタ、NTPサーバの設定、Openstack/Contrail Networkingクラスタ
階層	同じ機能を実行する、アプリケーションスタック内の同じタイプのソフトウェアインスタンスのセット。このようなインスタンスの数は、異なるスタックのパフォーマンス要件に従ってスケーリングできます。	Apache Webサーバー、Oracleデータベースサーバー、Hadoopスレーブノード、OpenStackサービスコンテナー
展開	一連の仮想マシンの目的を示します。通常、スタック内のすべての仮想マシンに適用されます。	開発、テスト、製造
サイト	スタックの場所を示します。通常はデータセンターの粒度です。	米国東部、ロンドン、ネバダ-2
カスタム	必要に応じて新しいタグを作成できます。	インスタンス名
ラベル	複数のラベルを適用して、スタック内およびスタック間のデータフローを細かく制御できます。	顧客アクセス、財務ポータル、db-client-access

表に示すように、Contrail Networkingで提供されるタグタイプに加えて、利用者は必要に応じて独自のカスタムタグ名を作成でき、データフローをより細かく調整するために使用できるラベルタイプタグがあります。

アプリケーションポリシーには、TCPまたはUDPポート番号のセットであるタグ値とサービスグループに基づくルールが含まれます。最初に、セキュリティ管理者はアプリケーションスタックにアプリケーションタイプのタグを割り当て、次にアプリケーションのソフトウェアコンポーネントごとにTierタイプのタグを割り当てます。これを下の図11に示します。

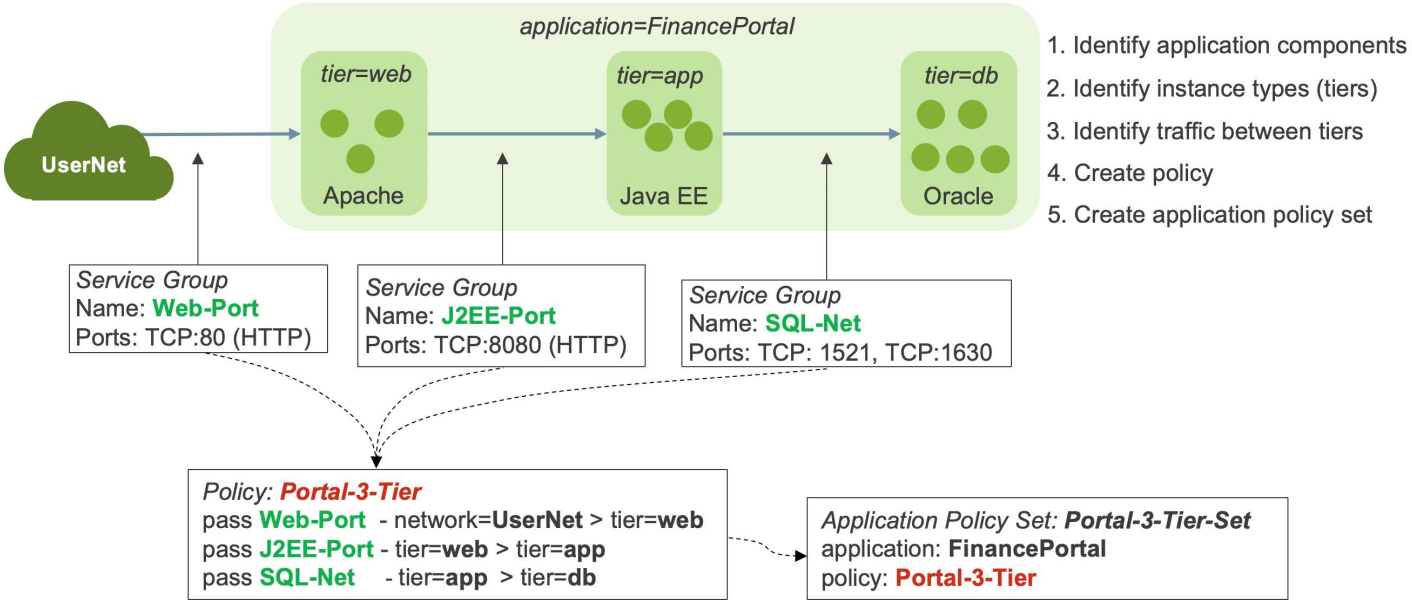

図11:アプリケーションポリシーは、タグとサービスグループに基づいています。

この例では、アプリケーションにFinancePortalというタグが付けられ、階層にはweb、app、dbというタグが付けられています。サービスグループは、アプリケーションスタックおよび各レイヤー間のトラフィックフローに対して作成されています。次に、セキュリティ管理者は、必要なトラフィックフローのみを許可するルールを含むPortal-3-Tierと呼ばれるアプリケーションポリシーを作成します。その後、アプリケーションポリシー設定はアプリケーションタグFinancePortalに関連付けられ、アプリケーションポリシーPortal-3-Tierが含まれます。この時点で、アプリケーションスタックを起動し、Contrail Networkingコントローラの各種VMにタグを適用できます。これにより、コントローラは、アプリケーションポリシー設定を適用するために各vRouterに送信する必要があるルートを計算し、それらがそれぞれのvRouterに送信されます。各ソフトウェアコンポーネントのインスタンスが1つの場合、各vRouterのルーティングテーブルは次のようになります:

ホスト	VRF	ソース	宛先	ポート	ルート
S1	ネット-ウェブ	0.0.0.0/0 10.1.1.3/32 10.1.1.3/32	10.1.1.3/32 10.1.2.3/32 0.0.0.0/0	80 8080 任意	VM-web NH=S2、Lbl=10のインターフェイス インターネットにルーティングします
S2	ネットアプリ	10.1.1.3/32 10.1.2.3/32 10.1.2.3/32	10.1.2.3/32 10.1.3.3/32 10.1.1.3/32	8080 1521, 1630 1521, 1630	VM-app NH=S3、Lbl=12NH=S1、Lbl=5のインターフェイス
S3	Net-db	10.1.2.3/32 10.1.3.3/32	10.1.3.3/32 10.1.2.3/32	1521, 1630 1521, 1630	VM-db NH=S2、Lbl=12のインターフェイス

ネットワークと仮想マシンには、その階層の名前が付けられます。実際には、エンティティ名と層の関係は、通常、それほど単純ではありません。表に示すように、ルートは、アプリケーションポリシーで指定されたとおりにのみトラフィックを有効にしますが、ここでは、タグベースのルールは、vRouterが適用できるネットワークアドレスベースのファイアウォールルールに変換されています。

アプリケーションスタックが正常に作成されたら、以下の図12に示すように、スタックの別のデプロイが作成されたときの動作を見てみましましょう。

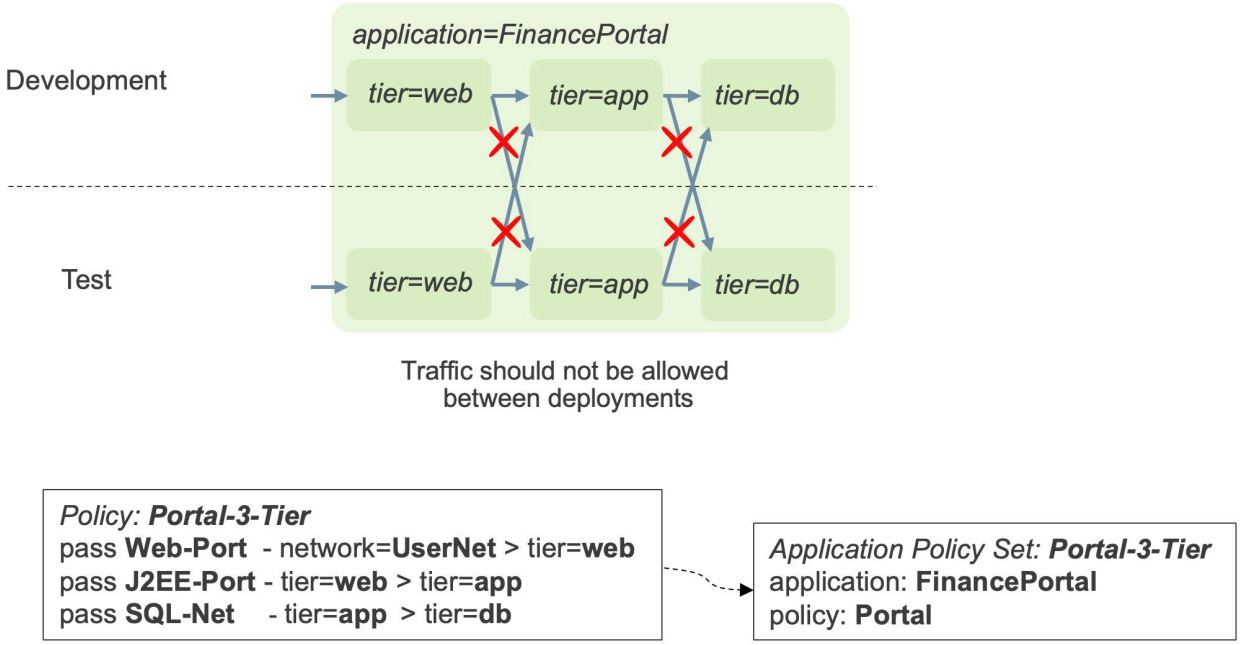

図12:元のポリシーでは、デプロイメント間のトラフィックフローが許可されます。

元のポリシーには、1つの展開のレイヤから別の展開のレイヤにトラフィックが流れないようにするものは何もありません。この挙動を変更するには、各スタックの各コンポーネントにデプロイタグをタグ付けし、アプリケーションポリシーに一致条件を追加して、デプロイタグが一致した場合にのみトラフィックが階層間を流れるようにします。更新されたポリシーを下の図13に示します。

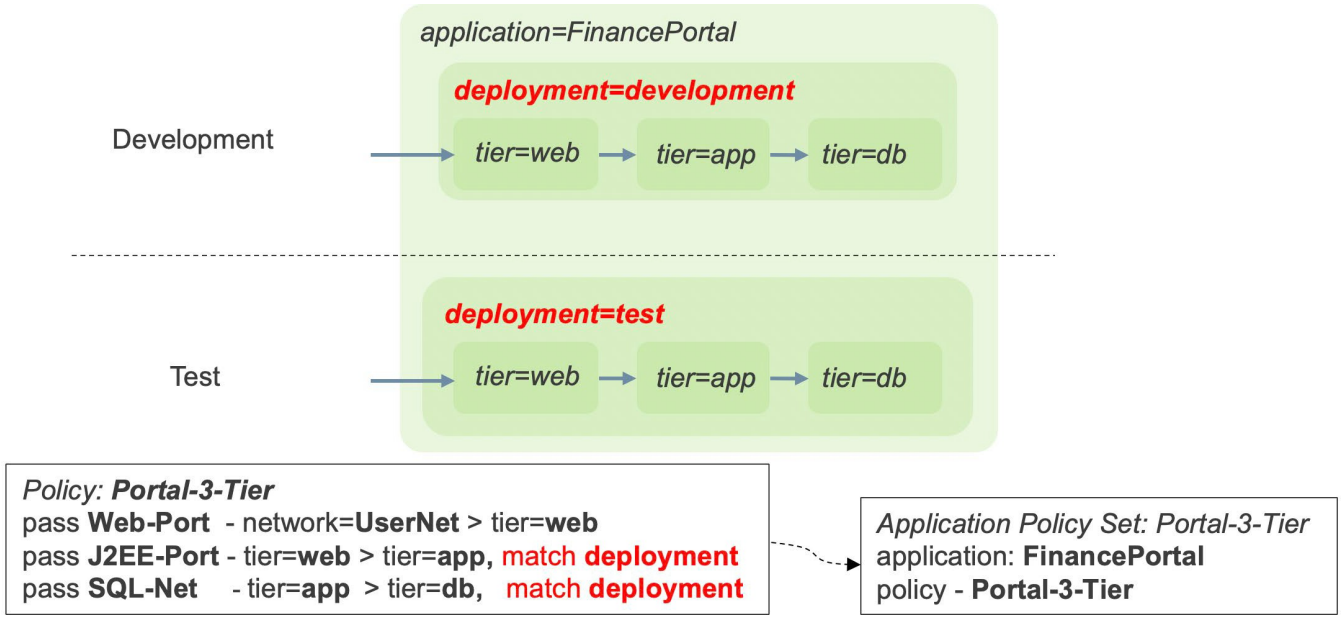

図13:スタック間を流れるトラフィックを防ぐデプロイメントタグを追加します

これで、トラフィックフローは、同じスタック内のコンポーネント間でのみトラフィックが流れるという厳密な要件に準拠します。

さまざまなタイプのタグを適用すると、セキュリティポリシーを単一のポリシー内の複数のディメンションに適用できます。たとえば、以下の図14では、単一のポリシーでサイトに基づいて個々のスタック内のトラフィックをセグメント化できますが、サイト内でのデータベース層の共有は可能です。

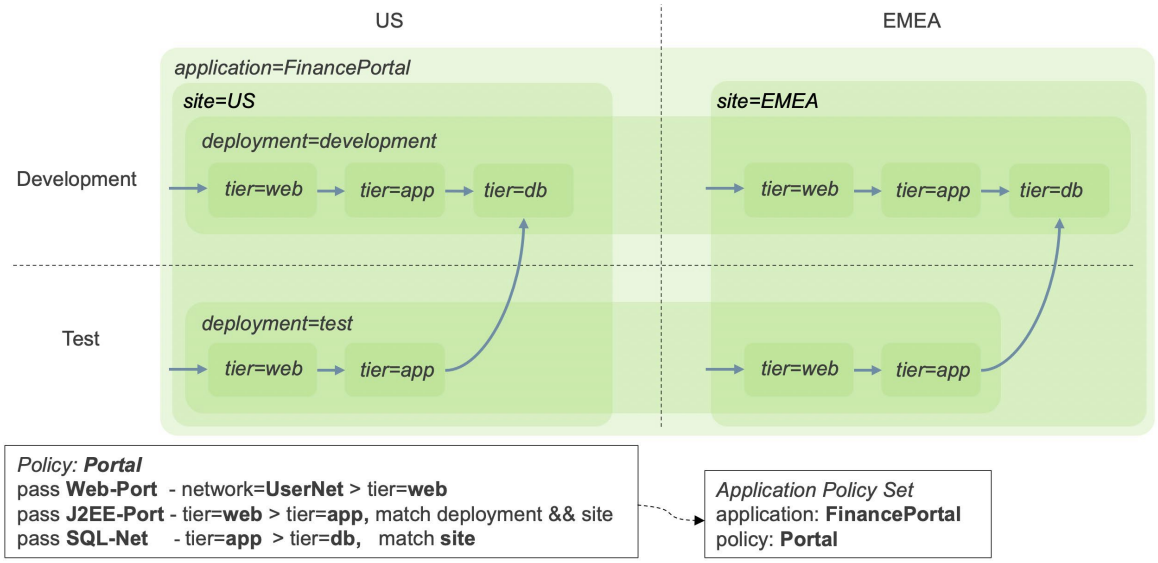

図14:タグを使用してサイト内のトラフィックを制限しますが、リソース共有を許可します

サイトとデプロイの同じ組合せ内に複数のスタックがデプロイされている場合、以下の図15に示すように、インスタンス名のカスタムタグを作成し、インスタンスタグの一致条件を使用して必要な制限を作成できます。

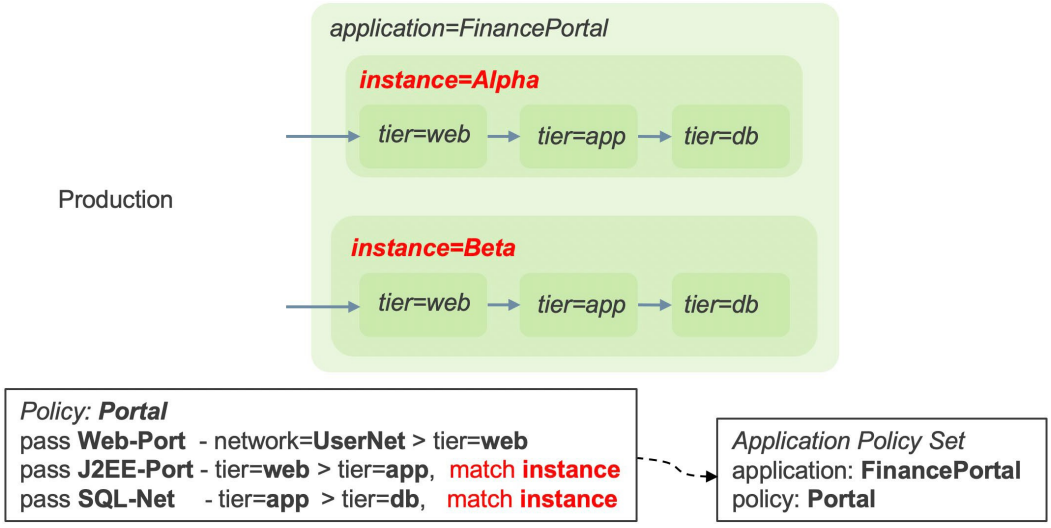

図15:カスタムタグを使用したスタックのセグメント化

Contrail Networkingのアプリケーションポリシー機能は、非常に強力な適用フレームワークを提供すると同時に、ポリシーの劇的な簡素化と数の削減を可能にします。

vRouterの展開オプション

vRouterには、さまざまな利点と使いやすさを提供する複数の展開オプションがあります:

- **カーネルモジュール–これはデフォルトのデプロイメントモードです。**
- **DPDK–転送アクセラレーションはIntelライブラリを使用して提供されます。**
- **SR–IOV–仮想マシンからNICへの直接アクセスを提供します。**
- **スマートNIC–vRouterフォワーダはプログラム可能なNICで実装されていま**

す。これらのオプションを図16に示します。

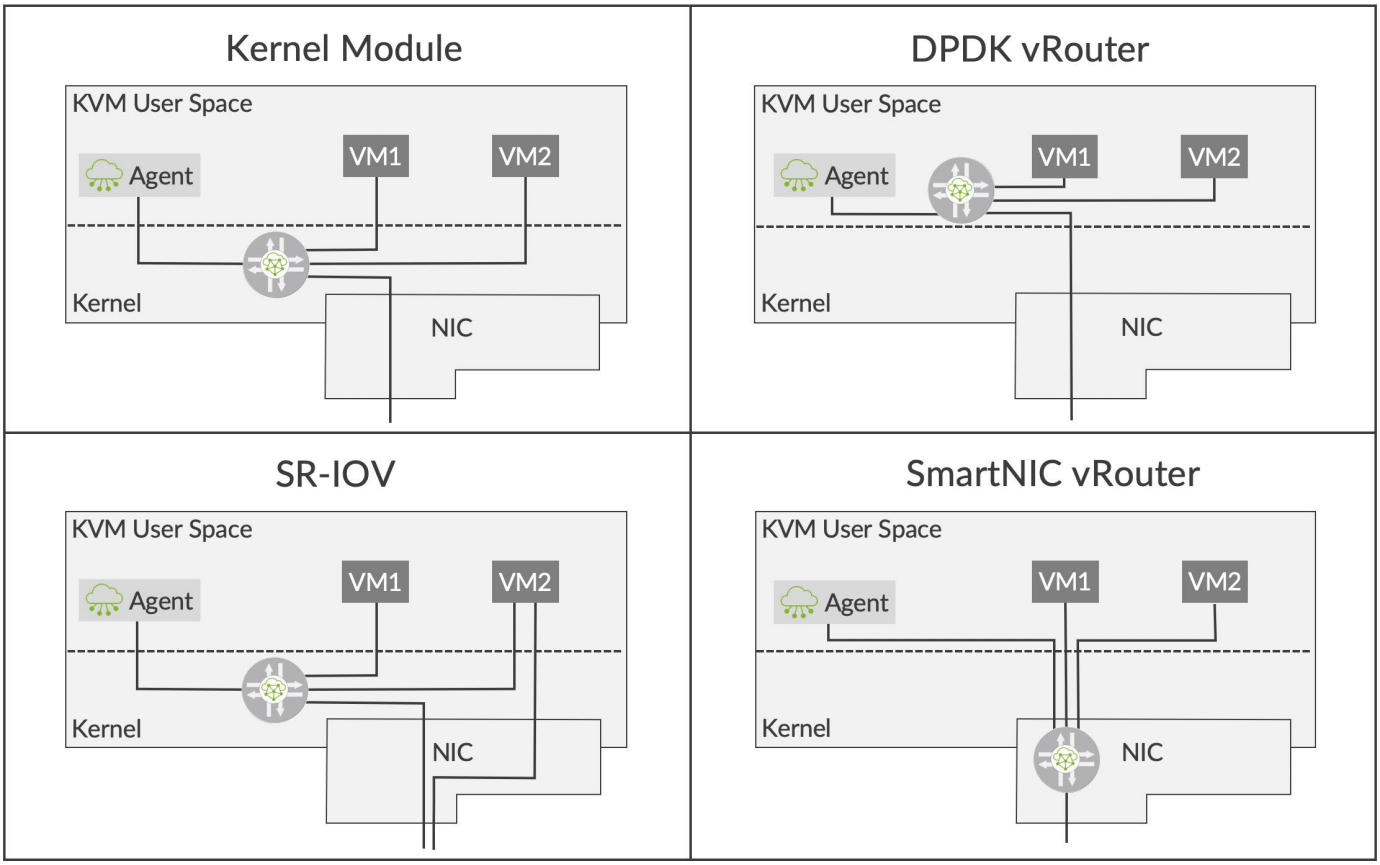

図16:vRouter展開オプション

各オプションの特徴と利点を以下に説明します。

カーネルモジュールvRouter

今日のデフォルトのデプロイオプションは、vRouterフォワーダがKVMカーネルで実行されるモジュールに実装されるようにするためのもので、Linux iptablesユーティリティまたはOpen vSwitch転送機能の使用に代わるものです。カーネルで実行すると、フォワーダはKVMのネットワークスタックを通過するときにネットワークトラフィックに直接アクセスでき、フォワーダがユーザー空間でプロセスとして実行した場合に達成できる履行を大幅に向上させることができます。実装されている最適化の中には、次のものがあります:

- TCPセグメンテーションオフロード
- ラージ受信オフロード
- マルチキューvirtioパケット処理の利用

カーネルモジュールアプローチにより、利用者はContrail Networkingを使用して、基盤となるサーバとNICハードウェアへの依存性を最小限に抑えながらネットワーク仮想化を実装できます。ただし、サポートされているのは特定のカーネルバージョンのKVMのみで、詳細はContrail Networkingの各バージョンのリリースノートに記載されています。

DPDK vRouter

インテル社のデータ・プレーン開発キット(DPDK)は、KVMネットワーク・スタックを介さずに、ユーザー・スペースで実行されているアプリケーションがNICに直接アクセスできるようにするライブラリーとドライバーのセットです。ユーザー領域で実行され、DPDKをサポートするバージョンのvRouterフォワーダを使用できます。DPDK vRouterは、未変更のVMを使用するカーネルモジュールと比較してパケットスループットを高速化し、ゲストVMでもDPDKが有効になっている場合は、パフォーマンスをさらに向上させることができます。

DPDK vRouterは、パケットを継続的に待機するパケット転送専用にCPUコアを割り当てることによって機能します。これらのコアは、ゲストVMの実行には使用できないだけでなく、常に100%のCPU使用率で実行されるため、一部の環境では問題になる可能性があります。

SR-IOV(Single Root-Input/Output Virtualization)

SR-IOVは、厳密にはvRouter自体の展開オプションではありませんが、帯域幅の最大化が重要で、(DPDKのように)パケット転送専用のコアが望ましくない場合に、一部のアプリケーションでvRouterとともに使用できます。SR-IOVを使用すると、ハイパーバイザーがCPUに対して行うのと同様に、NICのハードウェアリソースを複数のクライアント間で共有できます。これにより、仮想マシンインターフェースからNICに直接アクセスできるため、データパスはハイパーバイザーネットワークスタックをバイパスし、パフォーマンスが向上します。SR-IOVは、VMが物理ネットワークと仮想ネットワーク間のゲートウェイ機能を実行しているときに役立つことがありますが、SR-IOVではvRouterのバイパスが伴うため、インターフェイスはContrail Networking仮想ネットワークに参加せず、ネットワークポリシーおよびネットワークサービスに参加しません。

スマートNIC vRouter

プログラマブルな一部の新しいNICが利用可能になりつつあります。Contrail Networking vRouterフォワーダ機能は、これらの新しいNICに実装できます。これにより、特に一部の環境で支配的な小さなパケットサイズの場合に、パフォーマンスに大きな利点がもたらされます。
さらに、転送はサーバーのx86CPUからほぼ完全にオフロードされるため、より多くのVM用にコアを解放できます。

コントローラマイクロサービス

Contrail Networking(5.0以降)の新しいバージョンでは、Dockerコンテナに基づくマイクロサービスアーキテクチャが使用されています。マイクロサービスは、以下の図17に示すように、「ポッド」にグループ化され、それ自体が役割にグループ化されます。

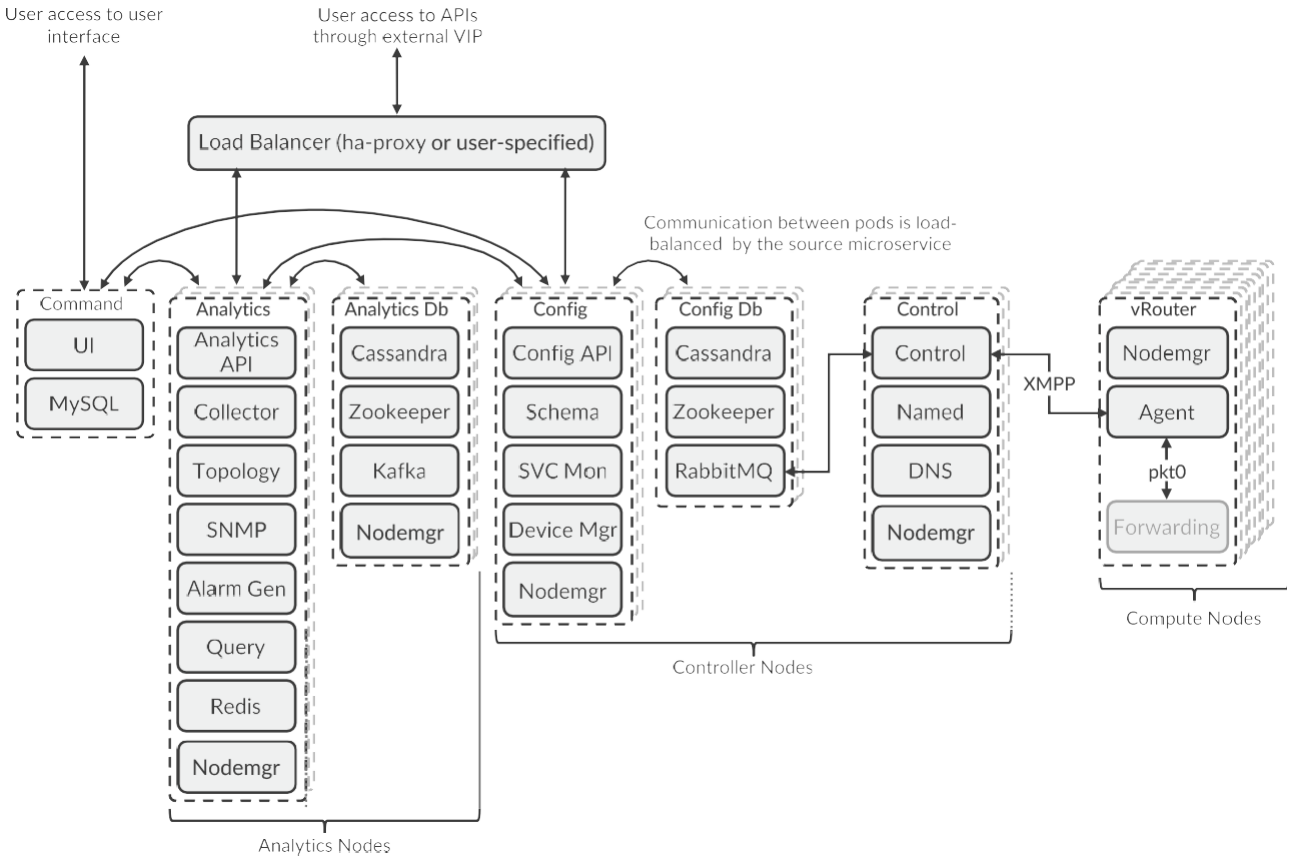

図17:Contrail Networkingマイクロサービスアーキテクチャ

アーキテクチャは構成可能です。つまり、各Contrail Networkingロールとポッドは、特定のデプロイメントのレジリエンスと履行の要件をサポートするために、異なるサーバーで実行される複数のインスタンスを使用して個別にスケーリングできます。

サーバー間のContrail Networkingサービスのレイアウトは、デプロイツールによって読み取られる設定ファイルによって制御されます。このファイルは、Ansible(Playbookを使用)またはHelm(チャートを使用)のいずれかです。すべてのサービスが同じ仮想マシンで実行されるシンプルなオールインワン展開、複数の仮想マシンまたはベアメタルサーバーを含む高可用性の例を含む、Playbookおよびチャートの例を使用できます。

デプロイツールとその使用方法の詳細については、Contrail Networkingのドキュメントページを参照してください。

Contrail Networkingを使用したOpenStack Orchestration

OpenStackは、仮想マシンとコンテナーの主要なオープンソースオーケストレーションシステムです。Contrail Networkingは、OpenStackのNeutronネットワーキングサービスの実装を提供し、多くの追加機能も提供します。

OpenStackでは、利用者のグループは「プロジェクト」に割り当てられます。このプロジェクトでは、VMやネットワークなどのリソースはプライベートであり、他のプロジェクトの利用者には表示されません(特に有効になっていない限り)。VPNを使用すると、許可された宛先へのルートのみがコンピュートノード上のvRouterのVRFに配布され、vRouterが実行するプロキシサービスによってフラッディングが発生しないため、ネットワーク層でのプロジェクト隔離の実施がストレートになります。

図3より前のバージョンでは、ネットワークサービスはNeutronで、コンピュートエージェントはNova(OpenStackコンピュートサービス)です。

Contrail Networkingは、両方がOpenStack環境にデプロイされている場合に、VMとDockerコンテナー間のシームレスなネットワークを提供できます。

以下の図18に示すように、OpenStackのContrail Networkingプラグインは、NeutronネットワーキングAPIからContrail Networkingコントローラで実行されるContrail Networking API呼び出しへのマッピングを提供します。

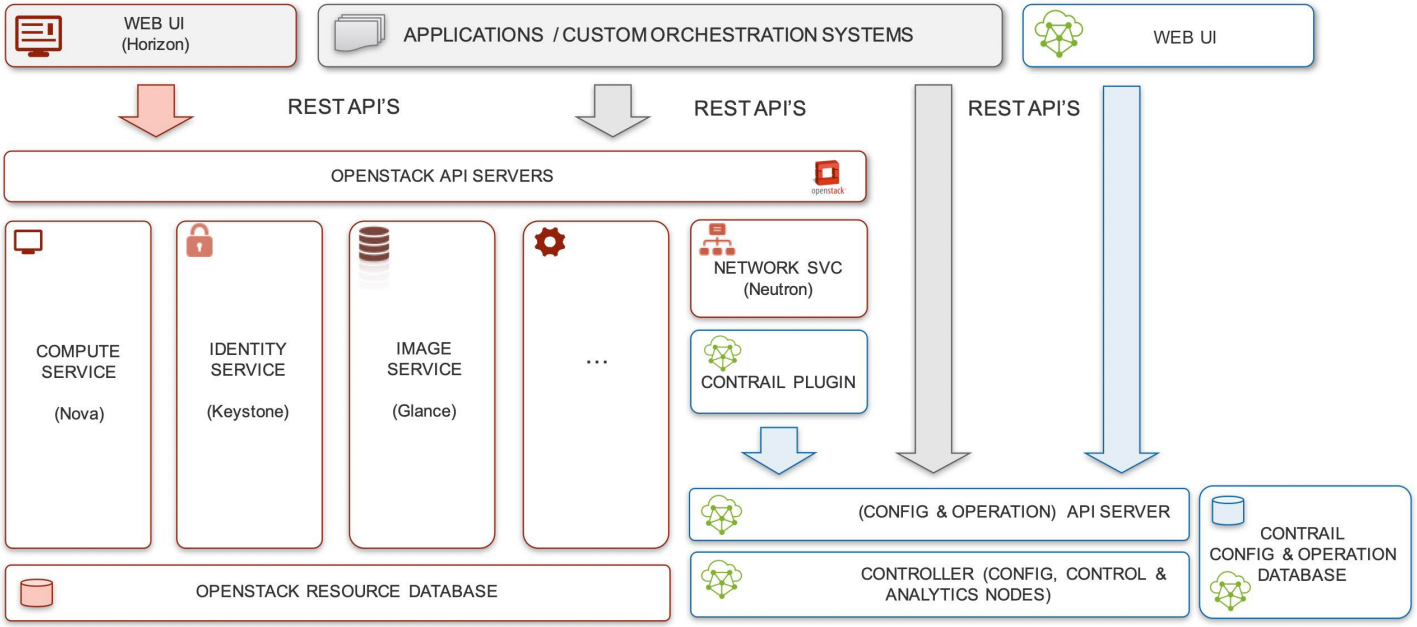

図18:Contrail NetworkingはOpenStack Neutron APIのスーパーセットを実装します。

Contrail Networkingは、ネットワークとサブネットワーク、およびOpenStackネットワークポリシーとセキュリティグループの定義をサポートします。これらのエンティティはOpenStackまたはContrail Networkingのいずれかで作成でき、変更は2つのシステム間で同期されます。さらに、Contrail NetworkingはOpenStack LBaaS v2APIをサポートします。ただし、Contrail NetworkingはOpenStackよりも豊富なネットワーク機能のスーパーセットを提供するため、多くのネットワーク機能はContrail Networking APIおよびGUIを介してのみ使用できます。これには、外部ルータへの接続を有効にするためのルート対象の割り当て、サービスチェーン、BGPルートポリシーの設定、およびアプリケーションポリシーが含まれます。

「アプリケーションベースのセキュリティポリシー」セクションで説明されているように、アプリケーションセキュリティは、OpenStackがContrail Networkingを使用する場合に完全にサポートされます。Contrail Networkingタグは、プロジェクト、ネットワーク、ホスト、VM、またはインターフェイスレベルで適用でき、タグが適用される目的物に含まれるすべてのエンティティに適用されます。

さらに、Contrail Networkingは、OpenStack Heatテンプレートを使用して制御できるネットワークワーキングとセキュリティのための一連のリソースをサポートしています。

Contrail Networkingを使用したKubernetes Container Orchestration

コンテナを使用すると、各VMが独自の完全なゲストOSを実行する仮想マシンとは異なり、同じホストOSで実行しながら、プロセスを互いに分離して動作させることができます。通常、コンテナで実行されているアプリケーションは、仮想マシンで実行されている同じアプリケーションよりもはるかに高速に起動し、パフォーマンスが向上します。これは、データセンターやNFVでコンテナを使用することに広く関心がある理由の1つです。Dockerは、コンテナをオペレーティングシステムのバージョン間で移植できるようにするソフトウェアレイヤーであり、Kubernetesによってshimレイヤーとして使用され、サーバー上のコンテナの作成と破棄を管理します。

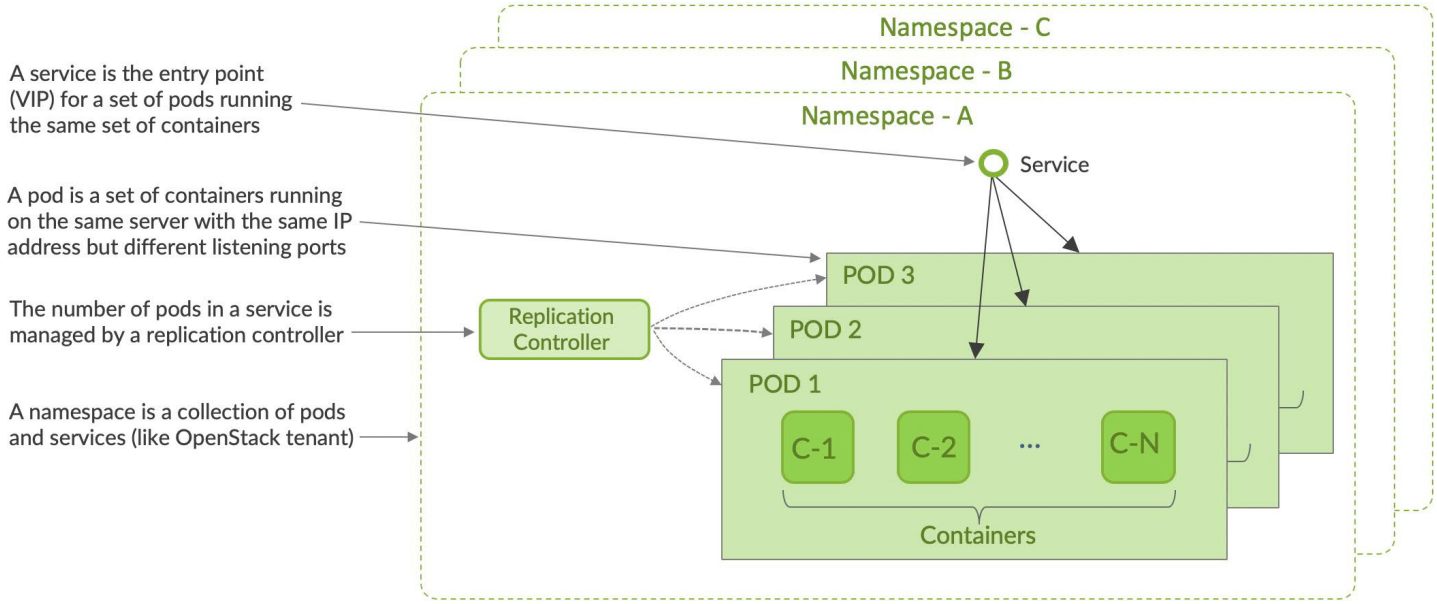

図19:Kubernetesはコンテナをポッドとサービスに編成します

上の図19に示すように、Kubernetesはコンテナのグループを管理し、一緒にいくつかの機能を実行し、Podと呼ばれます。ポッド内のコンテナーは同じサーバー上で実行され、IPアドレスを共有します。同じポッドの設定(通常は異なるサーバーで実行)は、サービスを形成し、サービス宛てのネットワークトラフィックをサービス内の特定のポッドに向ける必要があります。

デフォルトのKubernetesネットワーキング実装では、特定のポッドの選択は、送信側ポッドでネイティブのKubernetes APIを使用してアプリケーション自体によって実施されるか、非ネイティブアプリケーションの場合は、送信側サーバーでLinux iptablesに実装された仮想IPアドレスを使用してロードバランシングプロキシによって実施されます。アプリケーションの大部分は非ネイティブです。これは、Kubernetesを念頭に置いて開発されなかった既存のコードのポートであるため、ロードバランシングプロキシが使用されるためです。

Kubernetes環境の標準ネットワークは実質的にフラットで、どのポッドも他のポッドと通信できます。あるネームスペースのポッド(OpenStackのプロジェクトと同様)から別のネームスペースのポッドへの通信は、対象ポッドの名前またはそのIPアドレスが既知の場合は妨げられません。このモデルは、単一の企業に属するハイパースケールのデータセンターでは適切ですが、多くのエンドカスタマ間でデータセンターが共有されているサービスプロバイダや、異なるグループのトラフィックを互いに隔離する必要がある企業には適していません。

Contrail Networking仮想ネットワークは、Kubernetes環境に統合して、OpenStackと同様の方法でさまざまなマルチテナントネットワーク機能を提供できます。

Kubernetesを使用したContrail Networkingのこの設定を以下の図20に示します。

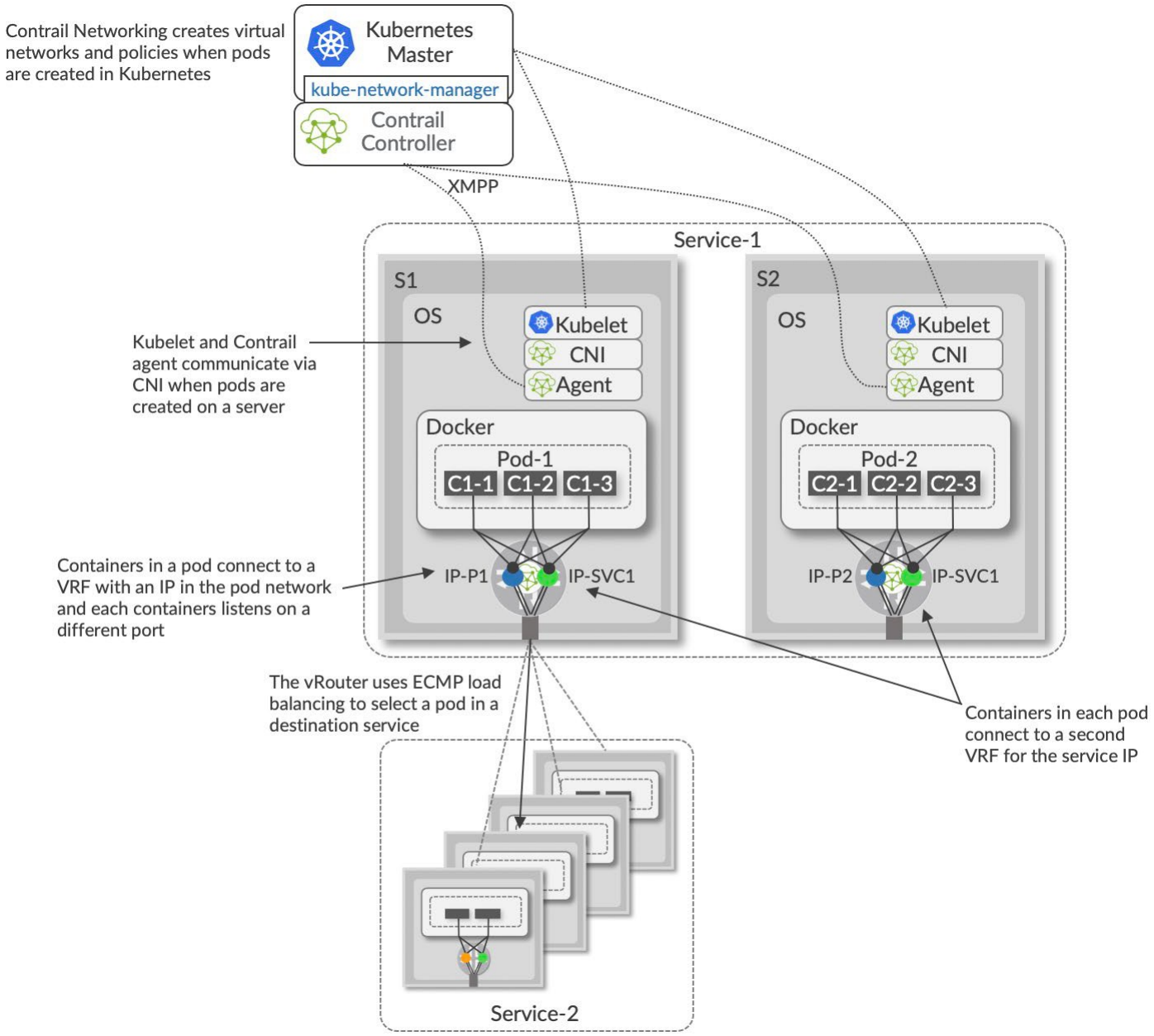

図20:Contrail Networkingで接続されたKubernetesポッド

KubernetesオーケストレーションおよびDockerコンテナを使用したContrail Networkingのアーキテクチャは、OpenStackおよびKVM/QEMUに似ており、ホストLinux OSでvRouterが実行され、仮想ネットワーク転送テーブルを持つVRFが含まれています。ポッド内のすべてのコンテナーは、単一のIPアドレス(図ではIP-1、IP-2)を持つネットワークスタックを共有しますが、異なるTCPまたはUDPポートでリッスンし、各ネットワークスタックのインターフェイスはvRouterのVRFに接続されます。kubenetwork-managerと呼ばれる処理は、Kubernetes APIを使用してネットワーク関連のメッセージをリッスンし、Contrail Networking APIに送信します。

サーバー上にポッドが作成されると、新しいインターフェイスを正しいVRFに接続するために、コンテナネットワークインターフェイス(CNI)を介してローカルkubeletとvRouterエージェント間の通信が行われます。サービス内の各ポッドには、仮想ネットワーク内の一意のIPアドレスと、サービス内のすべてのポッドで同じ浮動IPアドレスが割り当てられます。サービスアドレスは、他のサービスのポッド、または外部クライアントやサーバからサービスにトラフィックを送信するために使用されます。ポッドからサービスIPにトラフィックが送信されると、そのポッドに接続されたvRouterは、宛先サービスを形成する個々のポッドのインターフェイスに解決されるサービスIPアドレスへのルートを使用してECMPロードバランシングを実行します。Kubernetesクラスタの外部からサービスIPにトラフィックが送信されると、負荷分散はContrail Networkingコントローラとピアリングされているゲートウェイルータによって実行されます。KubernetesクラスタでContrail Networking仮想ネットワークが使用されている場合、Kubernetesプロキシのロードバランシングは必要ありません。

Kubernetesでサービスとポッドが作成または削除されると、kube-network-managerプロセスはKubernetes APIで対応するイベントを検出し、Contrail Networking APIを使用してKubernetesクラスタに設定されているネットワークモードに従ってネットワークポリシーを適用します。
さまざまなオプションを次の表にまとめます。

ネットワークモード	**ネットワークポリシー**	**影響**
Kubernetes default	Any-to-any、テナント分離なし	どのコンテナも、他のコンテナやサービスと通信できます。
名前空間の分離	Kubernetes名前空間はプロジェクトにマップされます Contrail Networkingで	ネームスペース内のコンテナーは互いに通信できます
サービス分離	各ポッドは独自の仮想ネットワークにあり、サービスIPアドレスのみがポッドの外部からアクセスできるようにセキュリティポリシーが適用されます。	ポッド内の通信は有効ですが、サービスIPアドレスのみがポッドの外部からアクセス可能です。
コンテナの分離	同じポッド内のコンテナー間のゼロ信頼。	ポッド内であっても、コンテナー間で特に許可された通信のみが有効になります。特定のサービスに対する特定のポッドのみを有効にできます。

Contrail Networkingは、OpenStackと同じ方法で、Kubernetesワールドに多くの強力なネットワーキング機能を提供します:

- IPアドレス管理
- DHCP
- DNS
- 負荷分散
- ネットワークアドレス変換(1:1浮動IPおよびN:1SNAT)
- アクセス制御リスト
- アプリケーションベースのセキュリティ

Contrail NetworkingおよびVMware vCenter

このセクションでは、Contrail vRouterがESXiホストにインストールされ、仮想マシンの仮想ネットワークとサービスを提供するユースケースについて説明します。Contrail Networkingは、仮想マシン間で仮想ネットワークを提供する物理スイッチの構成にも使用できます。これについては、『Contrail Networking』の「ファブリック管理」で説明されています。

VMware vCenterは、仮想化プラットフォームとして広く使用されていますが、異なるサブネットにある仮想マシン間のネットワークを実現し、vCenterクラスタの外部の宛先を使用するには、ネットワークゲートウェイを手動で構成する必要があります。Contrail Networkingは、既存のvCenter環境にデプロイして、以前にリストされたすべてのネットワーク機能を提供できますが、利用者がvCenter GUIおよびAPIを使用して仮想マシンを作成および管理するために必要になったワークフローは保持されます。さらに、vRealize OrchestratorおよびvRealize AutomationでのContrail Networkingのサポートが実装されているため、仮想ネットワークやネットワークポリシーの作成など、Contrail Networkingの一般的なタスクを、これらのツールに実装されているワークフローに含めることができます。

VMware vCenterを使用したContrail Networkingのアーキテクチャを以下の図21に示します。

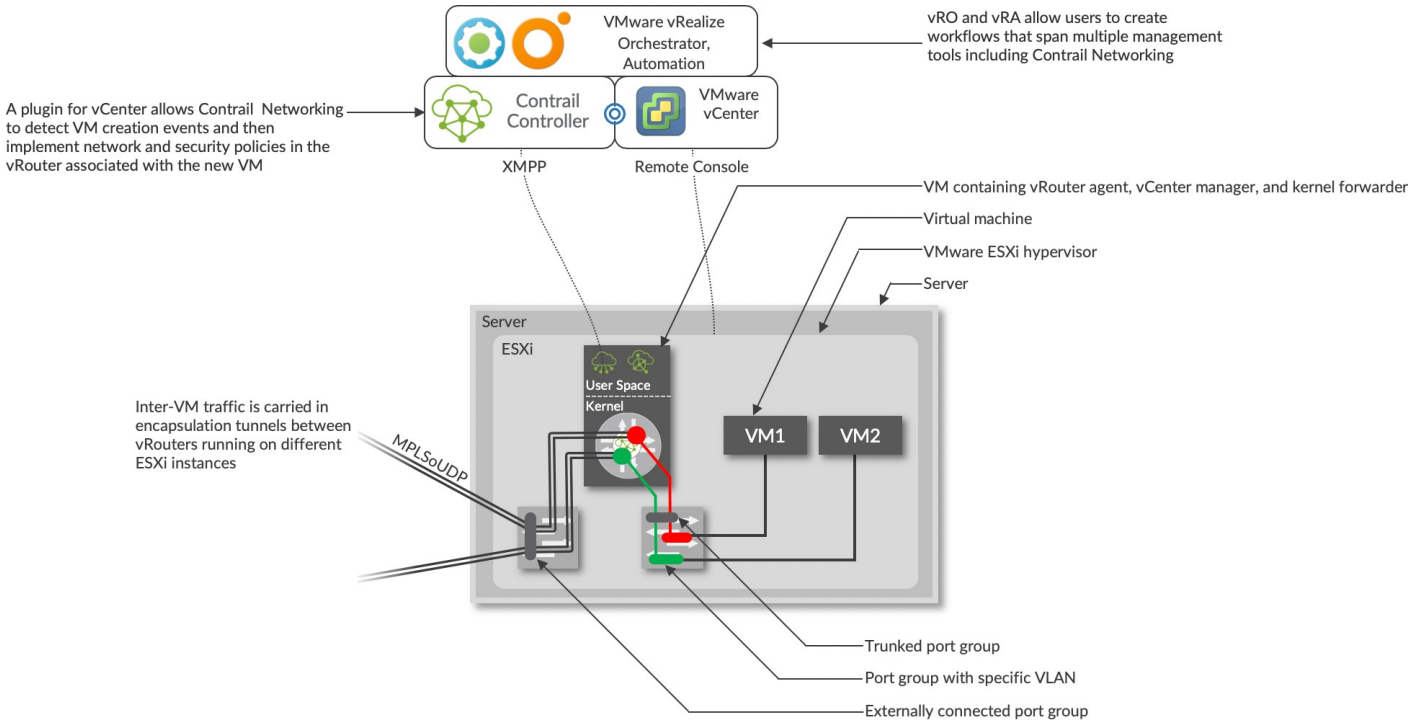

図21:vRouterはvCenter環境の仮想マシンで実行されます。

仮想ネットワークとポリシーは、Contrail Networkingで直接作成するか、vRO/vRAワークフローのContrail Networkingタスクを使用して作成します。

vCenter、GUI、またはvRO/vRAを使用して仮想マシンが作成されると、Contrail NetworkingのvCenterプラグインに対応するメッセージがvCenterメッセージバスに表示されます。これは、仮想マシンが作成されるサーバでvRouterを構成するためのContrail Networkingのトリガーです。各VMの各インターフェイスは、インターフェイスが存在する仮想ネットワークに対応するポートグループに接続されます。ポートグループには、vCenterの「VLAN override」オプションを使用してContrail Networkingコントローラによって設定されたVLANが関連付けられており、ポートグループのすべてのVLANはトランクポートグループを介してvRouterに送信されます。Contrail Networkingコントローラは、インターフェイスのVLANと、そのサブネットを含む仮想ネットワークのVRFとの間でマッピングします。VLANタグは削除され、VRFにおけるルートルックアップは、vRouterにおける詳細なパッケージ処理ロジックの項に記載されているように実行されます。

Contrail Networking with vCenterを使用すると、Contrail Networkingがこのドキュメントで前述したように提供するすべてのネットワークサービスおよびセキュリティサービスに利用者がアクセスできます。これには、ゼロトラストマイクロセグメンテーション、DHCPプロキシ、DNS、DHCPなどがあり、ネットワークフラッディング、簡単なサービスチェーン、ほぼ無制限のスケール、物理ネットワークとのシームレスな相互接続が回避されます。

ネストされたKubernetesとOpenStackまたはvCenter

前のセクションでは、コンテナが実行されるKVMホストが、何らかの方法で事前にプロビジョニングされていることを前提としていました。代わりに、OpenStackまたはvCenterを使用してコンテナーが実行されるVMをプロビジョニングし、Contrail NetworkingでOpenStackまたはvCenterによって作成されたVMとKubernetesによって作成されたコンテナー間の仮想ネットワークを管理することもできます。これを下の図22に示します。

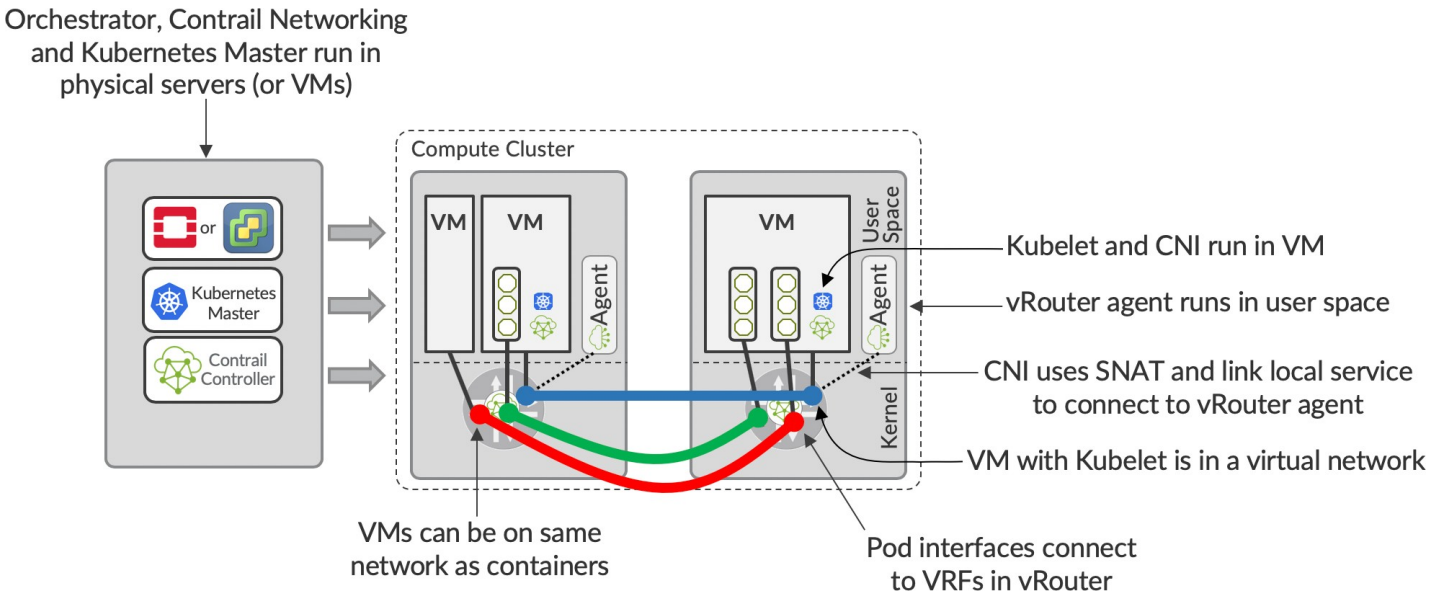

図22:ネストモードを使用してVMで実行されているKubernetesポッド

Orchestrator(OpenStackまたはvCenter)、Kubernetes Master、およびContrail Networkingは、一連のサーバーまたはVMで実行されています。Orchestratorは、Contrail Networkingでコンピューティングクラスタを管理するように設定されているため、各サーバにvRouterがあります。仮想マシンをスピンアップして、KubeletとContrail Networking用のCNIプラグインを実行するように設定できます。これらの仮想マシンは、Kubernetesマスターが Contrail Networkingによって管理されるネットワークを使用してコンテナを実行できるようになります。同じContrail Networkingがオーケストレータとkubernetesの両方のネットワークを管理しているため、VM間、コンテナ間、VMとコンテナ間でシームレスなネットワーキングが可能です。

ネストされた設定では、Contrail Networkingは前述のように同じレベルの分離を実現します。また、複数のKubernetesマスターが共存し、Kubeletを実行している複数のVMが同じホストで実行される可能性があります。これにより、マルチテナントKubernetesをサービスとして提供できます。

物理ネットワークへの接続

どのデータセンターでも、一部の仮想マシンが外部のパブリックIPアドレスにアクセスし、データセンター外のユーザーがパブリックIPアドレスを介して一部の仮想マシンにアクセスする必要があります。Contrail Networkingは、これを実現するいくつかの方法を提供します:

- BGP対応ゲートウェイへのVPN接続
- vRouterの送信元NAT
- 下地布へのvRouterのローカルゲートウェイ

これらはそれぞれ異なるユースケースで適用可能であり、それぞれ外部デバイスやネットワークの設定にさまざまな依存関係があります。

外部ネットワークへの接続方法については、次の項で説明します。

BGP対応ゲートウェイ

外部接続を実現する1つの方法は、パブリックIPアドレスの範囲を使用して仮想ネットワークを作成し、ネットワークをゲートウェイルータに拡張することです。ゲートウェイルーターがJuniper MXシリーズルーターの場合、デバイスでの設定はContrail Networkingによって自動的に行うことができます。これを下の図23に示します。

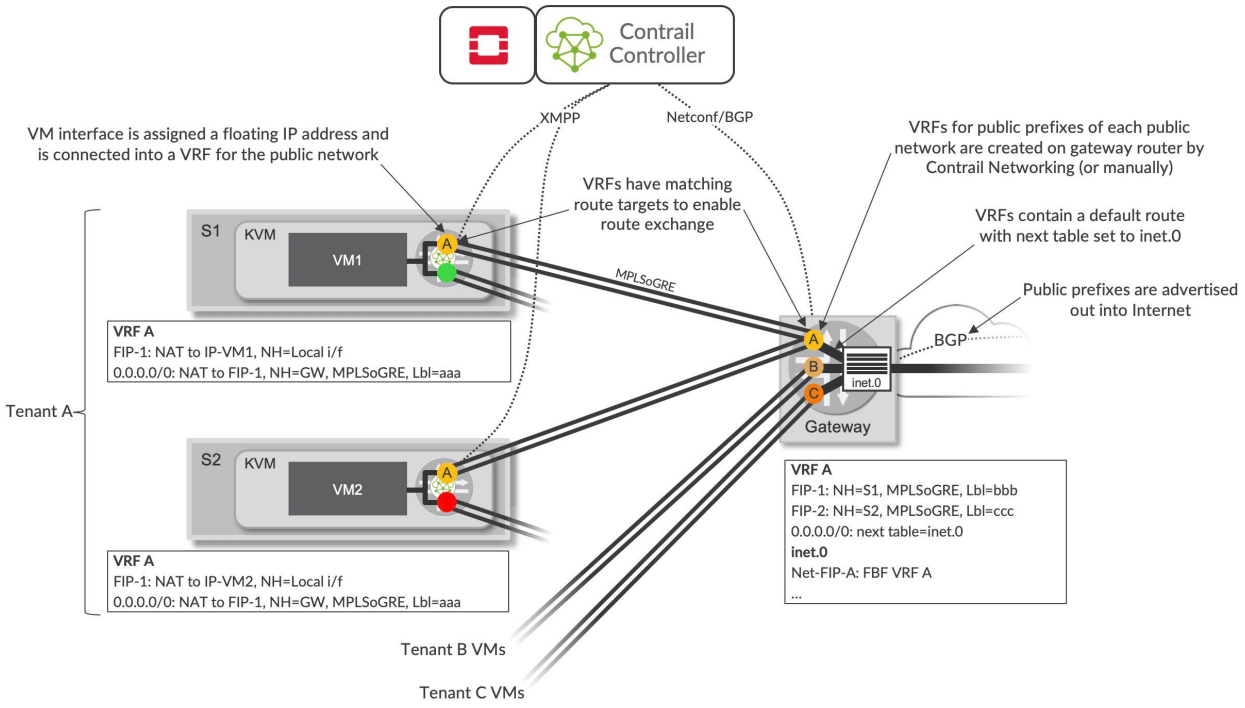

図23:フローティングIPアドレスを使用したBGPピアリングを使用した外部ネットワークへの接続

ネットワークAはContrail Networkingで定義され、パブリックにアドレス指定可能なIPアドレスのサブネットを含みます。このパブリック仮想ネットワークは、ゲートウェイルータに拡張するようにContrail Networkingで設定されます。これにより、Contrail Networkingは、仮想ネットワークのルートターゲットと一致するVRF(Aというラベルの VRFなど)をゲートウェイに自動的に作成します。Contrail Networkingは、コンピュートノード上のvRouterからVRFに到着するトラフィックのルートルックアップをメインのinet.0ルーティングテーブル(インターネットのパブリック宛先へのルートを含みます)で発生させるデフォルトルートを使用して、このVRFを設定します。フォワーディングフィルターがインストールされています。これにより、ネットワークAの宛先とゲートウェイに到着するトラフィックは、Contrail Networkingが作成したVRFで検索されます。ルータは、VRFを介してContrail Networkingコントローラにデフォルトルートをアドバタイズします。

ネットワークAはContrail Networkingでフローティング IPアドレスプールに設定され、このようなアドレスが既存のVMインターフェイスに割り当てられると、仮想マシンのvRouterに追加のVRF(ネットワークAなど)が作成され、インターフェイスは元のVRFに接続されるだけでなく、新しいパブリックVRFに接続されます(図23では緑色または赤色)。フローティングIPアドレスのVRFは、フローティングIPアドレスと仮想マシンに設定されているIPアドレスの間で1:1NATを実行します。VMはこの追加の接続を認識せず、DHCP経由で受信した元の仮想ネットワークのアドレスを使用してトラフィックの送受信を継続します。vRouterはフローティングIPアドレスへのルートをコントローラにアドバタイズし、このルートはBGP経由でゲートウェイに送信され、パブリックVRFにインストールされます(例:VRF A)。Contrail Networkingコントローラは、物理ルータ上のVRFを介してvRouterにデフォルトルートを送信し、これはvRouterのパブリックVRFにインストールされます。

これらのアクションの結果、vRouter上のパブリックVRFには、仮想マシンのローカルインターフェイスを介したフローティングIPアドレスへのルートと、ルータ上のVRFを介したデフォルトルートが含まれます。ゲートウェイ上のVRFには、inet.0ルーティングテーブル経由のデフォルトルート(フィルターベースの転送を使用して実装)があり、割り当てられた各フローティングIPアドレスへのホストルートがあります。inet.0ルーティングテーブルには、対応するVRF経由で各フローティングIPネットワークへのルートがあります。

テナントが独自のパブリックIPアドレス範囲を所有する場合(図を参照)、複数の個別のパブリックサブネットを独自のVRFを持つ個別のフローティングIPアドレスプールとして使用できます。逆に、1つのフローティングIPアドレスプールを複数のテナント間で共有できます(図を参照)。

ジュニパー以外のデバイスが使用されている場合、またはContrail Networkingがゲートウェイでの設定変更を許可されていない場合、BGPセッション、パブリックネットワークプレフィックス、およびスタティックルートをゲートウェイに手動で設定するか、設定ツールで設定できます。この方法は、ルータがエンタープライズVPNのプロバイダエッジ(PE)ルータの役割とデータセンターゲートウェイの役割を組み合わせている場合に使用されます。
一般的に、この場合、VRFはVPN管理システムによって作られることになります。Contrail Networkingクラスタ内の仮想ネットワークは、一致するルートターゲットが仮想ネットワークで設定され、コントローラとゲートウェイ/PE間でルートが交換されると、エンタープライズVPNに接続されます。

送信元NAT

Contrail Networkingを使用すると、複数のVMまたはコンテナが同じ外部IPアドレスを共有できるソースベースのNATサービスを介してネットワークを接続できます。ソースNATは、各vRouterで分散サービスとして実装されます。VMからインターネットに送信されるトラフィックのネクストホップはSNATサービスになり、送信側VMに固有のvRouterホストと送信元ポートに変更された送信元アドレスを持つアンダーレイネットワークのゲートウェイに転送されます。vRouterは、パケットを返す際に宛先ポートを使用して、発信元のVMにマップバックします。

アンダーレイでのルーティング

Contrail Networkingでは、アンダーレイを使用して接続するネットワークを作成できます。基礎となるものがルーティングIPファブリックの場合、コントレールネットワーキングコントローラーは、基礎切り替えとルートを交換するように設定されます。これにより、仮想ワークロードはアンダーレイネットワークから到達可能な任意の宛先に接続でき、仮想ワークロードを外部ネットワークに接続するための物理ゲートウェイよりもはるかにシンプルな方法を提供します。重複するIPアドレスがファブリックに接続されていないことに注意する必要があります。そのため、この機能は、マルチテナントサービスプロバイダではなく、レガシーリソースにクラウドを接続する企業にとって便利です。

Contrail Networkingにおけるファブリック管理

Contrail Networkingの重要な機能は、生地構成を通して、ゼロタッチのプロビジョニングからスイッチやルータの全ライフサイクルを管理できることです。装置ライフサイクル管理機能は、Ethernet VPN(EVPN)、VXLAN、NETCONFなどのオープンスタンダードに基づいています。装置のライフサイクル全体がサポートされています。装置検出、ベース構成のプロビジョニング、アンダーレイ接続の構成、ベアメタルサーバーのアタッチされたワークロードのオーバーレイネットワーキングです。このセクションでは、Contrail NetworkingとIPファブリック内のスイッチが工場出荷時の設定から完全に機能する環境になるための相互作用について説明します。

Contrail Networkingは、ファブリックを管理下に置く2つのモードをサポートします:

- *Greenfield–装置は工場出荷状態で始まります。*
- *ブラウンフィールド–デバイスには管理IPアドレスとループバックの相互接続がすでに設定されています。*

ファブリック管理の範囲

ファブリック管理Contrailの範囲は、以下の図24にまとめられています。

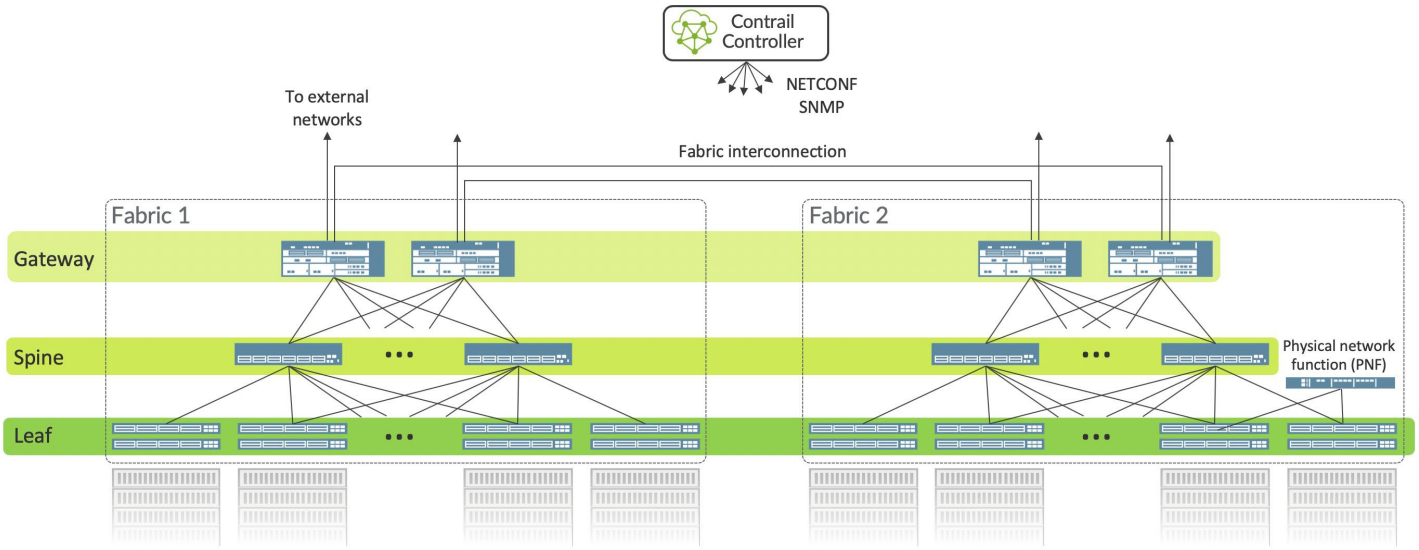

図24:Contrail Networkingでのファブリック管理の範囲

ファブリックは、装置の設定が異なる役割(ゲートウェイ、スパイン、リーフ)を実行する、接続された装置のグループです。スパインデバイスがゲートウェイ機能をサポートしている場合(QFX10000シリーズスイッチなど)、個別のゲートウェイレイヤーは省略できます。各デバイスには、ルートリフレクタ、SDNゲートウェイなど、1つ以上のルーティング/ブリッジング役割が割り当てられます。ルーティング/ブリッジングの役割については、次のセクションで詳細に説明します。Contrail Networkingは、複数のファブリックとそれらの間の接続を管理できます。さらに、VLANまたはアクセスポートを使用してサーバへの接続を管理したり、VXLAN仮想ネットワークを使用してサーバのグループを接続したり、外部ネットワークへの接続を提供したりできます。サーバー管理の詳細については、以下の「Lifecycle Management」および「Virtual Networking for Bare Metal Servers」を参照してください。さらに、Contrail NetworkingはVMware vCenterと統合し、vCenterで作成されたポートグループのファブリックで接続を提供できます。これについては、「Contrail Networking and VMware vCenter」で説明されています。

このセクションでは、サーバー間のオーバーレイネットワークをサポートするためにファブリックを構成する方法を中心に説明します。

図25は、Contrail Networkingがオーバーレイネットワーキングをサポートするファブリックをセットアップする方法を示しています。

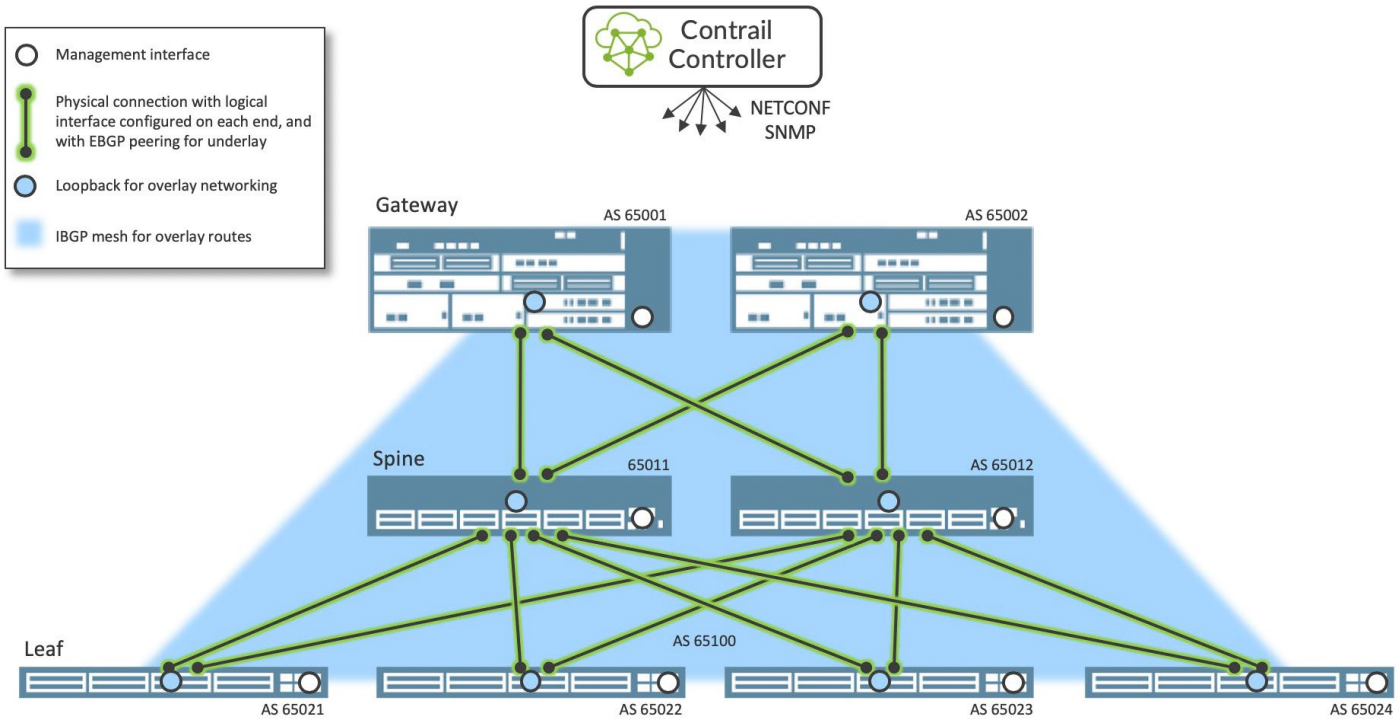

図25:アンダーレイ接続用のEBGPとオーバーレイ制御プレーン用のIBGPメッシュを備えたクローズベースのIPファブリック

各スパインは、別々のゲートウェイが使用されている場合、各リーフと各ゲートウェイに接続されます。デバイス間に複数の物理接続が存在する可能性があります。存在する場合、Contrail Networkingはこれらをリンクアグリゲーショングループ(LAG)として設定できます。論理インターフェイスは、各接続で設定されます。接続されたインターフェースには/31サブネットからアドレスが割り当てられ、接続されたインターフェースのペアまたはセットごとに異なるサブネットが使用されます。各デバイスには異なる自律システム(AS)数が割り当てられ、これらのランを使用するEBGPセッションが各接続を介して実行され、各ループバックアドレスがファブリック内のすべてのスイッチにアドバタイズされるようにします。ループバックインタフェース間の接続はアンダーレイネットワークを形成します。IBGPメッシュは、物理サーバがファブリックに接続されている場合に、物理サーバのオーバーレイルートを分散するために使用されます(このドキュメントで後述します)。Contrailは、スパインまたはゲートウェイレイヤーのルートリフレクタを使用して、これらのオーバーレイルートを分散します。

Contrail Networkingを使用してファブリックを管理する方法の詳細については、『Data Center:Contrail Enterprise Multicloud for Fabric Managementソリューションガイド』を参照してください。ファブリックの設計、構成、および操作に関する一般情報は、『Cloud Data Center Architecture Guide』およびジュニパー Webサイトで入手可能な『Data Center Deployment with EVPN/VXLAN』に記載されています。Contrail Networkingでのファブリック管理は、これらのドキュメントに記載されている設計原則と構成の詳細に従います。

ファブリックライフサイクル管理で使用される主要概念

ファブリックの作成プロセスを詳細に説明する前に、Contrail Networkingの重要な概念(ロール、ネームスペース、および仮想ポートグループ)をいくつか紹介する必要があります。

役割

Contrail Networkingのファブリック管理機能は、「ノードプロファイル」と呼ばれる概念を使用して、各デバイスタイプが持つことができる役割と機能を指定します。Contrail Networkingでサポートされる各特定のデバイスモデルには、デバイスが実行できるロールのリストを含む対応するノードプロファイルがあります。役割には、ファブリック内の場所(ゲートウェイ、スパイン、リーフ)を記述する部分と、その場所でデバイスが実行するネットワーク機能を記述する部分の2つがあります。たとえば、Juniper Networks QFX5100e-48s-6qスイッチは、リーフまたはスピンのいずれかの役割を果たし、リーフとしてサーバ用のアクセスポートを提供することができます。対照的に、Juniper Networks QFX511048s-4cスイッチは、QFX5100e-48s-6qと同じ役割を果たすことができ、また、VXLANネットワーク間の集中ルーティングを実行し、データセンターゲートウェイとしての役割を果たすことができます。

次のセクションでは、異なるデバイスタイプで使用可能なロールを指定するAnsible設定ファイルのエントリの例を示します。このファイルでは、「CRB」は集中ルーティングブリッジングを表します。

```
juniper-qfx5100e-48s-6q:
  - CRB-Access@leaf
  - null@spine
   juniper-qfx5110-48s-4c:
  - CRB-Access@leaf
  - null@spine
  - CRB-Gateway@spine
  - DC-Gateway@spine
```

同じ設定ファイルの下位に、各ロールに設定された機能が指定されます。例えば:

```
null@spine:
  - 基本
  - ip_clos
  - overlay_bgp
  - overlay_networkingCRB-
Gateway@spine:
  - 基本
  - ip_clos
  - overlay_bgp
  - overlay_evpn
  - overlay_evpn_gateway
```

- overlay_security_group
- overlay_lag
- overlay_multi_homing
- overlay_networking
- overlay_evpn_type5

これらの各機能には、対応するAnsible Playbook(Jinja2テンプレート付き)があり、機能がデバイスに適用されるロールに存在する場合に実行されます。次の表では、Contrail Networkingで定義されている各種ネットワークロールと、それらが適用できる物理ロールについて説明します。各ネットワーク役割のサポートは、特定の物理役割で採用されている特定のデバイスモデルによって異なります。

役割	説明
ヌル	エッジルーティングとブリッジングが使用されている場合にのみスパインに適用されます。
ルートリフレクタ	ルートリフレクタとして機能するように、各ファブリックに少なくとも1つのデバイスを指定します。通常、すべてのスパインまたはゲートウェイにこの役割が与えられます。
CRB-Gateway	集中ルーティングブリッジング。Contrailで仮想ネットワークを接続するための分散論理ルーターを作成すると、作成される各ネットワークのIRBを含むVRFと、タイプ5ルートを使用したサブネットルート配布が発生します。
CRBアクセス	ベアメタルサーバーが接続されるリーフ装置に適用します。
CRB-MCAST-Gateway	マルチキャストプロトコルと入力方向レプリケーションのサポートを提供します。
ERB-UCAST-Gateway	エッジルーテッドブリッジング。アクセスインタフェースおよび論理ルーティングでのIGMPスヌーピングは、タイプ5ルートでサポートされます。
DCゲートウェイ	外部ネットワークへの接続を提供する装置。折りたたまれたゲートウェイアーキテクチャが使用されている場合はスパインレイヤのデバイスに適用し、存在する場合はゲートウェイデバイスを分離します。
DCIゲートウェイ	ファブリック間の接続に使用されます。
AR-Replicator	装置は、BUMトラフィックのアシストレプリケーション(AR)を実行します。
ARクライアント	装置は、ARを実行する別の装置にBUMトラフィックを送信します。

注:通常、ファブリック内の仮想ネットワーク間のすべてのルーティングには、リーフスイッチとスパインスイッチでCRB役割またはERB役割が適用されますが、これは必須ではなく、両方の役割が同じファブリック内で共存できます。

名前空間

ネームスペースは、Contrail Networkingによって値を描画および割り当てることができるプールです。

これらの値は、たとえば、ループバックアドレスを割り当てるサブネットや、スパインデバイスとリーフデバイス間のポイントツーポイント接続にBGP自律システム(AS)番号を割り当てる番号範囲を指定するために使用されます。名前空間は、通常、ファブリックが最初に作成されるときに指定されます。

グリーンフィールドファブリックを作成するプロセスを、ブラウンフィールドシナリオに関連する相違点の注釈とともに以下に説明します。

仮想ポートグループ

仮想ポートグループ(VPG)は、EVPNセグメント識別子(ESI)リンク集約グループ(LAG)を形成するように設定されるスイッチポートのグループです。同じESIを持つインターフェイスは、一般に同じサーバ上にあり、リンクがダウンした場合にサーバがリンク全体のロードバランシングとフェイルオーバーを備えたボンディングインターフェイスをサポートできるように、これらのインターフェイスでLink Aggregation Control Protocol(LACP)が実行されます。

各VPGには1つ以上のVLAN(タグなしを含みます)が関連付けられており、それぞれがContrail Networking仮想ネットワーク、一連のセキュリティグループ、ポートプロファイル(現在はストーム制御設定のみが含まれています)に関連付けられています。VPG内のVLANは、サーバポートに設定されているVLANと一致する必要があります。Contrail Networkingが管理する仮想ネットワーク間の論理ルーティング、またはファブリックに接続されたワークロードが外部ネットワークにアクセスできるようにするためのゲートウェイルータの設定などのレイヤ3サービスを提供するためにContrail Networkingが使用される場合、VPGの仮想ネットワークサブネットはサーバで設定されたものと一致する必要があります。

ファブリック作成の手順

ファブリックの作成と管理のプロセスには、4つの主要な段階があります:

- *ファブリック作成:名前空間(IPアドレスの割り当てプールなど)が指定されている場所。*
- *デバイス検出:インターフェースと接続が検出されました。アンダーレイ接続とオーバーレイ制御プレーンが指定されています。*
- *役割の割り当て:利用者は、各デバイスのファブリック役割とルーティング/ブリッジング役割を指定します。*
- *自動設定:Ansible Playbookを実行して、アンダーレイ接続、オーバーレイ制御プレーン、および各デバイスの役割を設定します。*

各ステージの詳細については、次のセクションで説明します。

ファブリックの作成

ファブリック構成の最初の段階では、Contrail Commandインターフェースを使用して次の情報を構成します:

- ファブリックの名前
- 装置のタイプ(ノードプロファイル)
- ゲートウェイ付き管理サブネット
- アンダーレイAS範囲
- スパインリーフのポイントツーポイント接続用のファブリックサブネット
- ループバック・サブネット

この情報が入力されると、利用者はデバイス検出段階に進むことができます。

デバイス検出

グリーンフィールド展開では、装置は工場出荷状態で追跡され、管理インターフェイスから定期的にDHCP要求が発行されます。ラック中、管理インターフェイスは、Contrail Networkingコントローラへのアクセスを提供するVLANに接続する必要があります。クラスターに埋め込まれているのは、DHCPサーバを含むOpenStackのベアメタル・アラーム管理機能の一部です。管理サブネットがネームスペースとして指定されている場合、Contrail Networkingはサブネットとゲートウェイを DHCPサーバの設定ファイルに設定します。次にラックされたデバイスがDHCPリクエストを発行すると、DHCPサーバはIPアドレスとデフォルトゲートウェイで対応します。

検出段階では、Contrail Networkingはデバイスが管理IPアドレスを送信されたことを検出し、管理IPアドレスを含む一部の基本構成をデバイスにプッシュし、NETCONF、SNMP、LLDPを有効にするAnsible Playbookを実行できます。後続のPlaybookは、名前、モデル、インターフェイスのリストなど、装置に関するファクトを取得します。ネイバー接続は、SNMPを使用してLLDPテーブルから取得されます。利用者はファブリック内の装置の数を入力できるため、その数の装置が見つかると検出処理は終了します。

次の段階(アンダーレイ接続の設定)は、スパインとリーフデバイス間のポイントツーポイント接続の設定、各デバイスでのループバックの設定、および接続されたデバイス間のEBGPセッションの設定によって行われます。これにより、直接接続がない場合、各デバイスはネクストホップとしてネイバーを使用して、ファブリック内の他のすべてのデバイスへのルートを受信します。これでアンダーレイの設定は完了です。

ブラウンフィールドシナリオでは、装置はすでに管理接続で設定されています。pingスイープはデバイスを検出し、その後に前述のインターフェース検出処理が続きます。

役割の割り当て

検出フェーズの後にContrail Command GUIにデバイスモデルが表示されると、各デバイスに役割を割り当てることが可能になります。前述したように、ロールには、ファブリック内の物理的な役割(スパイン/リーフ)とルーティング/ブリッジングの役割の2つの部分があります。デバイスの物理的役割を指定すると、割り当てられた物理的役割を持つそのデバイスモデルで使用可能なルーティング/ブリッジング役割が利用者に提示されます。

自動設定

すべてのデバイスにロールが指定されると、利用者はAutoconfigureボタンを押します。次に、プレイブックのセットが実行され、各デバイスで指定されたロールが設定されます。

まず、ネイバー間接続を可能にするために、ネイバーデバイス間のポイントツーポイント接続が、それらの接続上のEBGPセッションと共に設定されます。次に、ファブリックの作成時に指定されたAS番号を使用して、ファブリック内の装置ループバック間のIBGPセッションを設定することによって、オーバーレイ制御プレーンが作成されます。これを実現するために、Ansible Playbookのシーケンスが実行されます。

この時点で、ベアメタルサーバー(BMS)またはVMware ESXiサーバーをスイッチポートに接続し、それらのポートでインターフェイスを設定してVXLAN仮想ネットワークに配置する準備が整います。

デバイス動作

Contrail Networkingは、次のデバイス動作をサポートします:

- ファブリック内の装置の追加/削除
- 装置の新しいOSバージョンへのソフトウェアアップグレード
- ファブリック内のデバイスを交換するためのRMA手順

ベアメタルサーバのライフサイクル管理と仮想ネットワーク

Contrail Networkingは、OSのプロビジョニングや、接続されているスイッチのポートの設定など、ベアメタルサーバーのライフサイクル管理を提供します。仮想マシンとコンテナのContrail Networkingに存在するのと同じネットワーク機能とセキュリティ機能が、スイッチのインターフェイスに実装され、それらに接続されているベアメタルサーバーに適用されます。Contrail Networkingでは、ファブリックスイッチでEVPNセッションを使用して、物理ワークロードと仮想ワークロード間のシームレスなネットワークを可能にします。

ベアメタルサーバーの管理、プロビジョニング、およびネットワークの機能アーキテクチャを図26に示します。

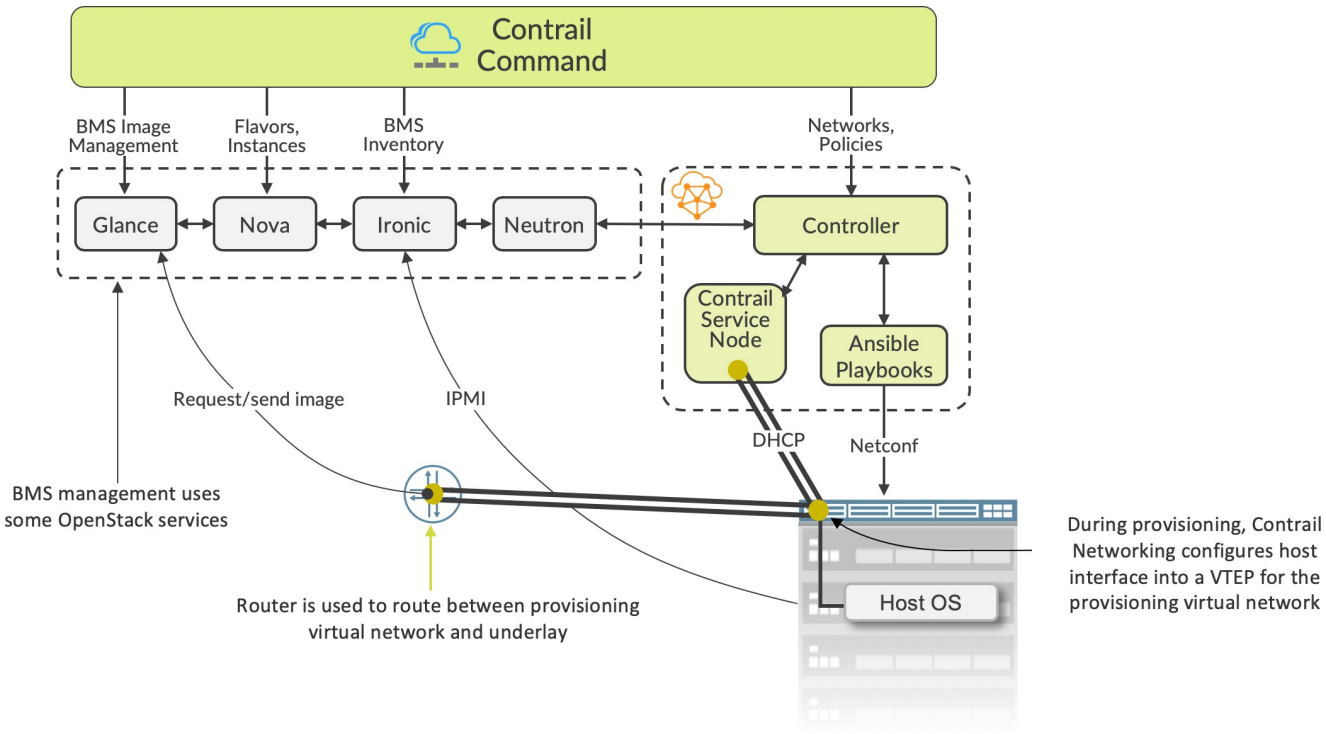

図26:Contrail Networkingを使用してベアメタルサーバーをプロビジョニングするための機能コンポーネント

ベアメタルサーバープロビジョニングは、Contrail Networkingに組み込まれたOpenStack部品によって処理されます。Contrail Commandは、Contrail Networkingで使用されるGlance、Nova、Ironicの機能のGUIを提供します。Ironicは、Novaサービスを使用してベアメタルサーバーを管理できるようにするOpenStackのサービスです。Novaがベアメタルサーバーに関する命令を受信すると、Ironicがフルフィルメントを引き継ぎます。Contrail Service Node(CSN)は、ContrailのIPアドレス管理からIPアドレスを受信するベアメタルサーバーのDHCPプロキシです。サーバーをファブリックに接続し、そのネットワークを構成するための一連のイベントには、主に2つのステップがあります:

- 接続されているスイッチポート、各インターフェイスのMACアドレス、ボンディング接続とマルチホーム接続のどちらが使用されているかを識別して、サーバーをインフラストラクチャインベントリに追加します。
- 規定するイメージと、それを規定するサーバーを識別します。VXLANネットワークでスイッチインターフェイスを設定し、Glanceイメージからサーバをプロビジョニングします。

物理サーバとそれで実行されているオペレーティングシステムは、Contrail Networkingでは個別のオブジェクトとして扱われることに注意してください。

サーバは、Contrail Networkingによって完全に管理できます。Contrail Networkingは、サーバが接続されているスイッチに設定されているVXLANオーバーレイネットワークを使用して接続を提供するだけでなく、オペレーティングシステムをプロビジョニングすることもできます。すでにIPアドレスが設定されている既存のサーバは、Contrail仮想ネットワークに接続することもできます。

Contrail Networkingが物理サーバーのライフサイクル管理と仮想ネットワークを管理する場合に関係する一連の操作については、ホワイトペーパー『Fabric and Server Lifecycle Management with Contrail Networking』で詳しく説明されています。このホワイトペーパーについては、ジュニパー Webサイトで入手できます。以下のセクションでは、物理サーバの仮想ネットワークにおけるパケットフローに関するホワイトペーパーの内容の一部を要約します。

物理サーバの仮想ネットワーク

このセクションでは、同一および異なる仮想ネットワーク上のサーバー間、および物理サーバーと仮想ワークロード(仮想マシンまたはコンテナ)間のパケットフローについて説明します。

同じ仮想ネットワーク内のサーバ間のパケット

ファブリック内のスイッチに接続され、同一の仮想ネットワーク内に配置されるように構成された2台のサーバを図27に示します。サーバは、各スイッチにVTEPも構成されているContrail Networkingによってプロビジョニングされています。このドキュメントでは、仮想ネットワークのルートターゲットを「Red」と呼びます。ただし、実際の値はtarget:xxx:yyy(xxxとyyyは数値)の形式になり、VNIは「Red」と呼ばれます。このドキュメントの規則として、(R)などの括弧内の文字は、ルート更新にアタッチされたルートターゲットまたはVNIを示します。

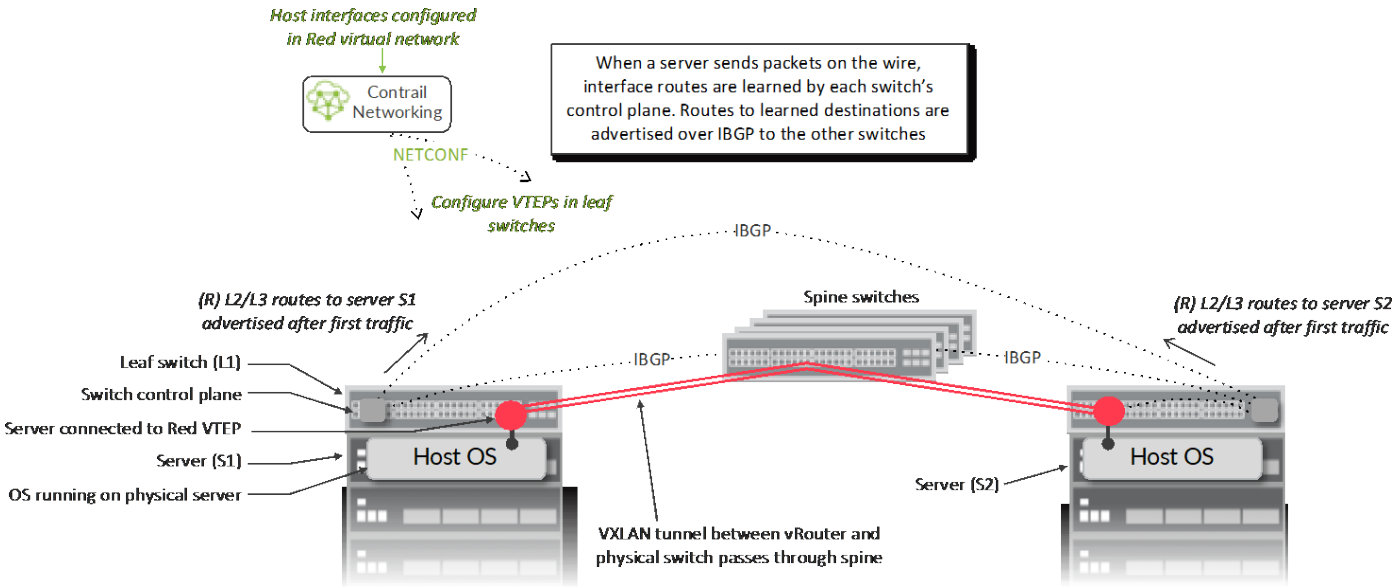

図27:VXLANオーバーレイトンネルを使用した2台のサーバ間の接続

切り替え間にはIBGPセッションがあり(通常、スパイン切り替えのルートリフレクタを使用して実装されます)、切り替えに接続されているサーバへのさまざまなルートは、切り替え間で交換されます(リーフツーリーフ接続は常にスパイン切り替えを物理的に通過することに注意してください)。上記のホワイトペーパーで詳細に説明されているように、トラフィックがネットワークに送信され、スイッチのブリッジングテーブルにデータが入力されると、サーバへのルートがインストールされてアドバタイズされます。

この例では、各スイッチは、自身をトンネルの宛先とするRed VXLANトンネルを介して、接続されているサーバにルートをアドバタイズします。サーバS2宛てのパケットがS1から送信されると、リーフ切り替えL1は、切り替えL2へのVXLANトンネルを介してルーティングテーブルでS2へのルートを検出し、スパイン切り替えのそれぞれを介してS2へのECMPルートが存在することになります。リーフはスパインへのルートを選択し、VNIがRedに設定されたVXLANカプセル化内のL2にパケットを転送してから、そのスパインにルーティングします。S2はパケットをカプセル化解除し、サーバインタフェースに送信します。

ECMPは両方向で使用されるため、正方向と逆方向のトラフィックは異なるスパインスイッチを通過できます。

異なるネットワーク内のサーバ間のパケット

異なるネットワーク内のサーバ間の通信が必要な場合は、それらの間でルーティングを設定する必要があります。これは、Contrail Networkingで、目的のネットワークを含む論理ルーターを設定することによって実行されます。分散論理ルーターが作成されると、ContrailはNETCONFを使用して、接続される仮想ネットワークごとにIRBを含むVRFでファブリック内のスイッチを設定します。各IRBは、そのネットワークのデフォルトゲートウェイアドレスで設定され、そのネットワークのVNIを持つVTEPでも設定されます。これらのVRFは、スパインスイッチ(中央集中型ブリッジング-CRB)またはリーフスイッチ(エッジルーティングブリッジング-ERB)で作成するように指定できます。

CRBの使用例を以下の図28に示します。

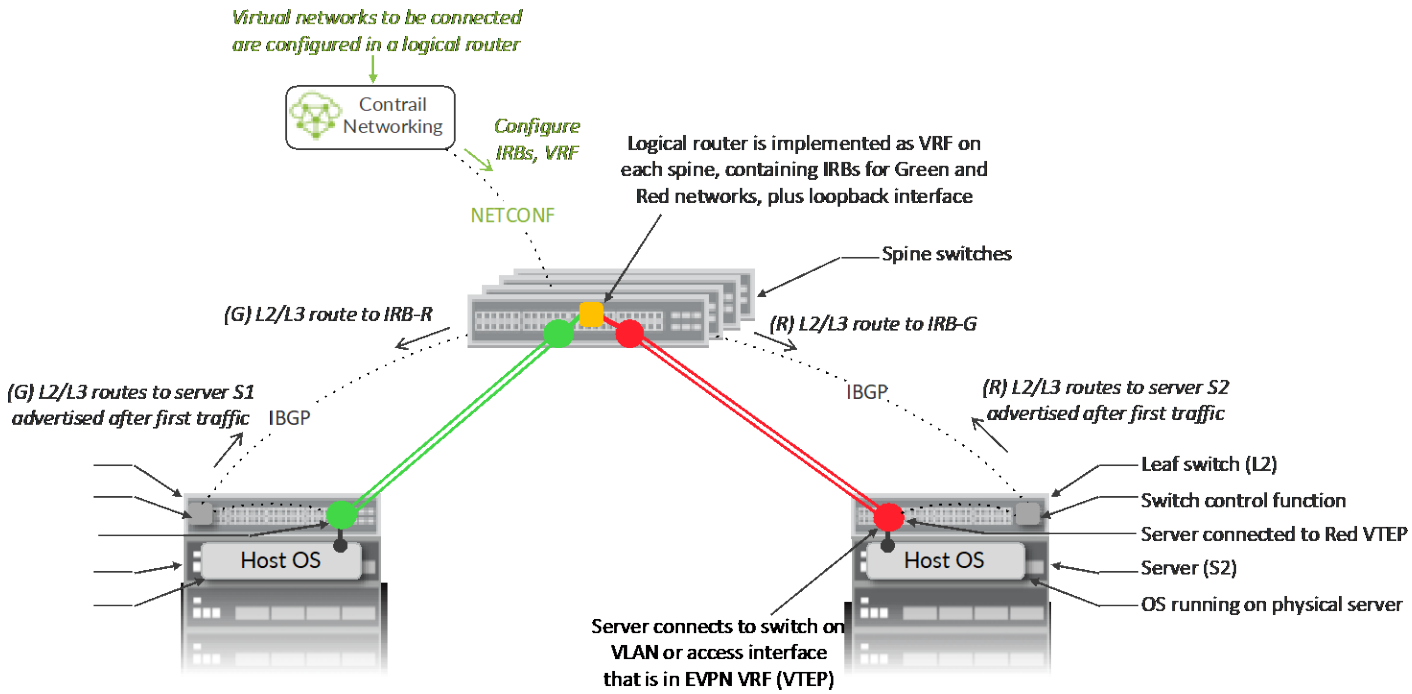

図28:論理ルーターは、CRBのスパインスイッチにIRBを持つVRFとして実装されています。

IRBが各VRFに設定されると、IRBゲートウェイアドレスのBGPルートがスパインによって各リーフスイッチに送信されます。リーフスイッチは、ECMP経由で使用するスパインを選択します。これは、フォワードトラフィックとリバーストラフィックが異なるスパインを通過できることを意味します。

ERBを選択し、Contrail Networkingで分散論理ルーターを作成すると、分散論理ルーターで設定されたネットワークにサーバーインターフェースを持つ各リーフスイッチに、対応するVRFが設定されます。これを下の図29に示します。

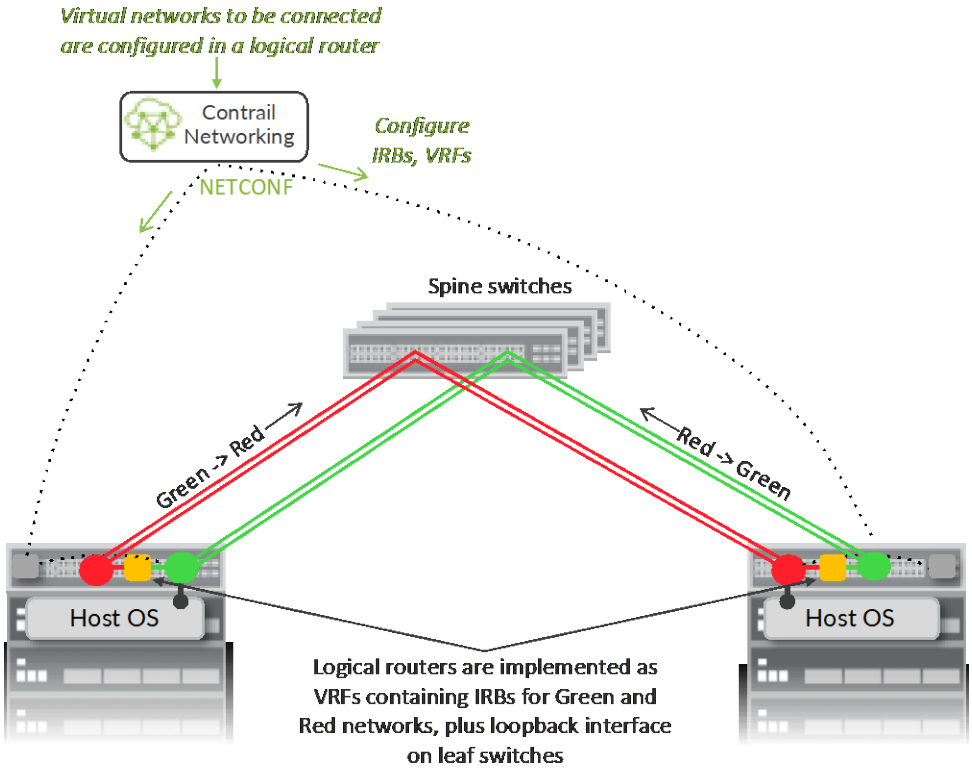

図29:論理ルーターは、ERBのスパインスイッチにIRBを持つVRFとして実装されています。

別のネットワーク内の宛先を持つ1つのネットワーク内のサーバによって送信されたトラフィックは、ローカルリーフスイッチでローカルにルーティングされてから、宛先ネットワークのVNIとともにVXLANトンネルで送信されます。VXLANトンネルのリーフトーリーフトラフィックはスパインスイッチでルーティングされ、リーフスイッチはスパイン間でECMPロードバランシングを使用するため、フォワードトラフィックとリバーストラフィックは異なるスパインを通過できます。

物理サーバと同じ仮想ネットワーク内の仮想マシン間のトラフィック

Contrail Networkingでは、オーバーレイネットワークを使用して物理ワークロードと仮想ワークロードをシームレスに相互接続できます。

図30は、同じネットワーク内にインターフェイスを持つ仮想マシンと物理サーバを示しています。この図

は、vRouterによって使用されるXMPPメッセージとスイッチで使用されるEVPNルートの間を仲介するContrail Controllerを介して交換されるルートを示しています。

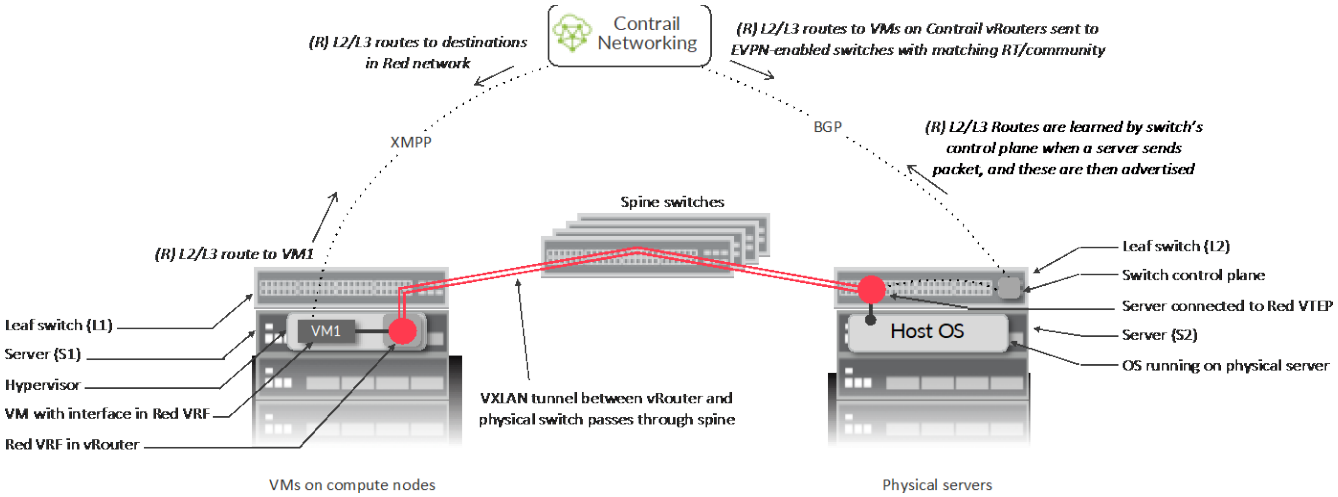

図30:同じネットワーク内のサーバとVM間のトラフィック

VM1とS2の間のトラフィックは、一方の側のvRouterで終端し、もう一方の側のリーフスイッチで終端するVXLANトンネルで伝送されます。

異なる仮想ネットワーク内の物理サーバと仮想マシン間のトラフィック

このセクションでは、VXLANを管理するためにEVPNを実行しているスイッチに物理サーバが接続され、サーバが通信する必要があるVMとは異なるVNI(この場合はGreen)を持つVTEPに接続されている場合の動作について説明します。

図31は、分散論理ルーターがIRBを持つVRFとして実装される方法、および集中ルーティングブリッジングが使用される場合のルートの交換方法を示しています。

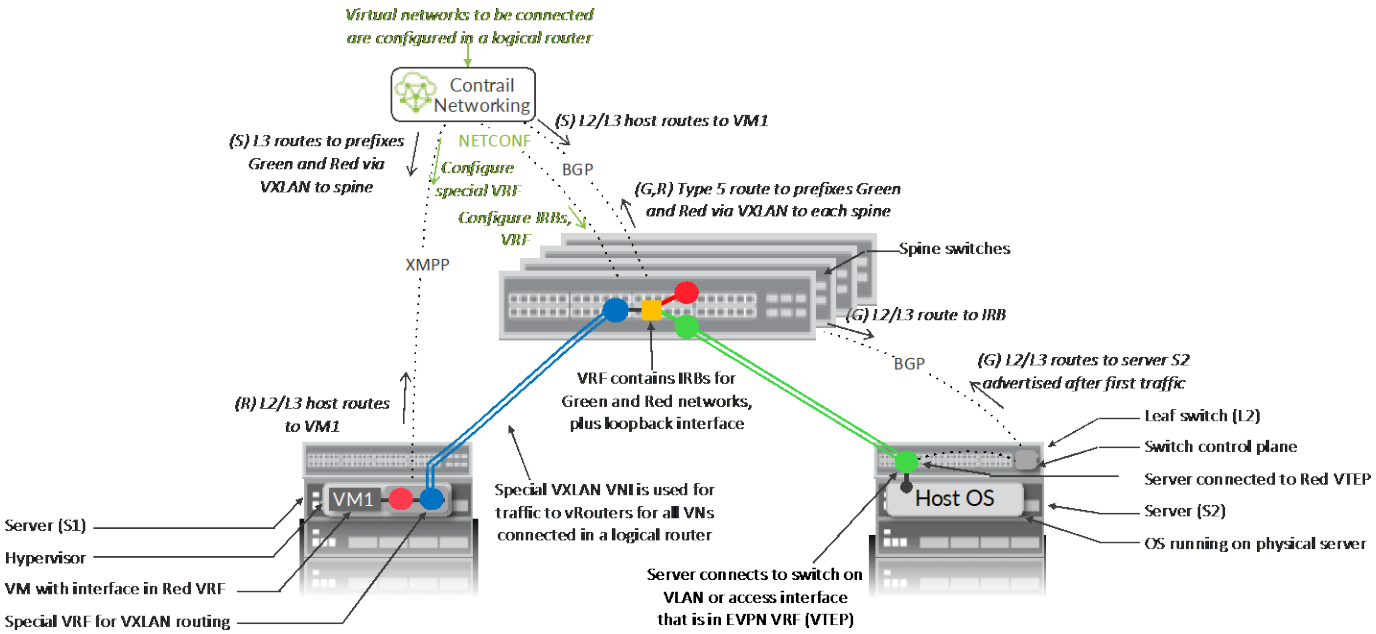

図31:異なるネットワーク内のサーバとVM間のトラフィック

スパインスイッチは、制御プレーンとしてEVPNを使用し、Contrail Controller内の制御ノードのIPアドレスをピアとして使用するように設定されています。Contrail NetworkingでGreenネットワークが作成されると、管理者はネットワークをスパインスイッチに拡張するように指定します。これにより、Contrail Networkingは各スパインスイッチにVRFルーティングインスタンスを作成します。各VRFには2つのIRBが含まれており、これらはRedおよびGreenネットワークのゲートウェイアドレスとRedおよびGreenサブネットで設定されています。仮想環境へのルーティングはVRFのメインルーティングテーブルにあり、発信トラフィックは論理ルーターによって接続されているすべての仮想ネットワークに使用される特別なVNIを使用してVXLANを使用するように設定されます。論理ルーターごとに異なる特別なVNI(この場合はBlue)が使用されます。Red VMを実行しているvRouterでは、その特別なVNIを使用して追加のVRFが作成され、Red VRFにデフォルトルートがインストールされます。これにより、Greenネットワーク宛てのトラフィックが特別なVRFに送信され、次にスパインスイッチのVRFに送信され、最後に宛先にオンになります。

エッジルーティングブリッジングが使用されると、論理ルーターは関連するIRBおよびVTEPとともにリーフデバイスに配置されます。

VMware vCenterの物理ネットワーク構成

Contrail Networkingリリース1910以降では、VMware vCenterとContrail Networkingファブリック管理の統合がサポートされています。この統合により、Contrail Networkingは、VMインターフェイスが設定されているDistributed Port Groups(DPG)に期待される接続を実装するようにスイッチを設定できます。これは、この統合のためにデプロイされているvCenter、Contrail vCenter Fabric Manager(CVFM)へのプラグインを使用して実現されます。このプラグインは、仮想マシンの変更イベント(作成/削除/変更)を監視し、新しいDPG内のインターフェイスを持つ仮想マシンがESXiホストで起動されたときに、Contrail Networkingが対応する仮想ポートグループ(VPG)を作成するようにします。VPGは、DPGで指定されたVLANで設定され、VXLAN仮想ネットワークを使用して、同じDPG内にあるすべてのVMを接続します。この手順は、VMware vCenterシステムとESXiホストにはまったく影響しません。

CVFMデザインの概要

図32は、Contrail Networkingコントロールノードにインストールされているプラグインを示しています。CVFMプラグインは、vCenter環境の変更を検出し、新しい構成をContrail Device Managerにプッシュします。次に、Contrail Device Managerは、これらの設定をQFXシリーズスイッチなどのファブリック装置にプッシュします。

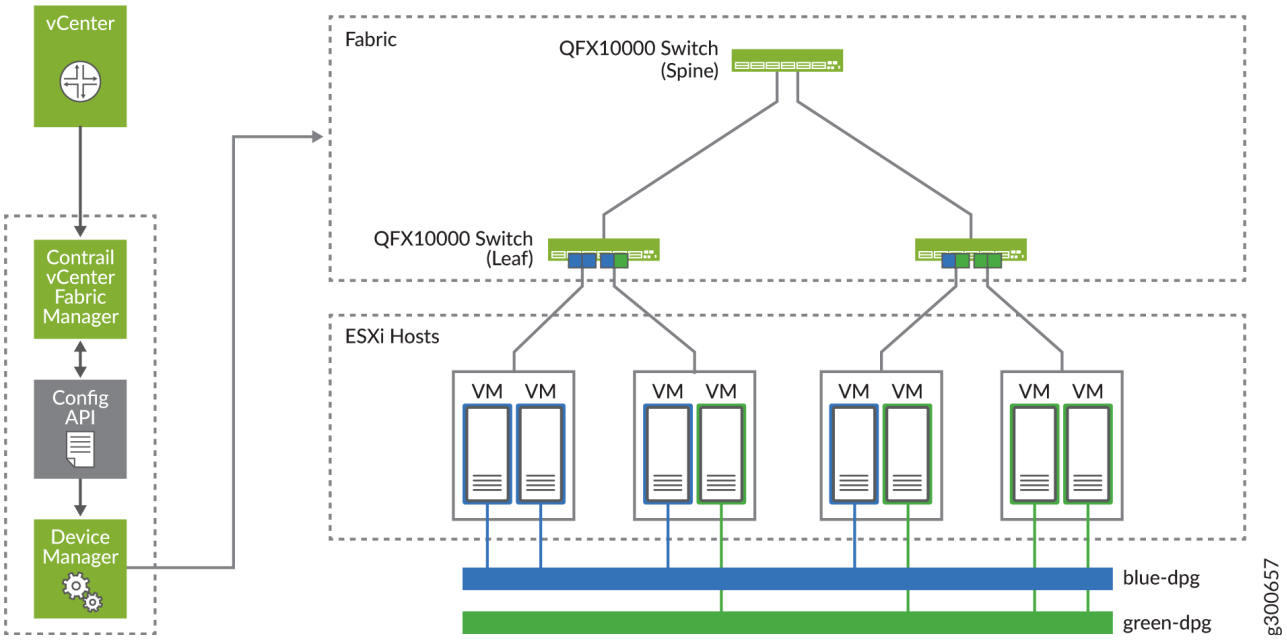

図32:Contrail vCenter Fabric Managerプラグイン

リーフスイッチとスパインスイッチ(QFXシリーズ)は、ESXiホスト環境の仮想マシンに接続されます。VLANは、これらのQFXシリーズスイッチのDPGで設定されます。CVFMプラグインは、vCenterでネットワーク変更イベントが検出されると、VLANの設定を自動的に追加および削除します。