



## Red Hat OpenStack Platform 9

# Red Hat のネットワーク機能仮想化ソリューション

Red Hat のネットワーク機能仮想化ソリューションの概要



# Red Hat OpenStack Platform 9 Red Hat のネットワーク機能仮想化ソリューション

---

Red Hat のネットワーク機能仮想化ソリューションの概要

OpenStack Team  
rhos-docs@redhat.com

## 法律上の通知

Copyright © 2018 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution-Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 概要

本書は、Red Hat の NFV (ネットワーク機能仮想化) ソリューションについて紹介し、欧州電気通信標準化機構 (ETSI) の NFV モデルの階層として機能することが可能な各種 Red Hat 製品の特性について概説します。

## 目次

1. 概要 .....	2
1.1. Red Hat と NFV	3
1.2. 設計環境	4
1.2.1. インフラストラクチャー層	5
1.2.2. 仮想化層	5
1.2.3. OpenStack 層	5
1.2.4. 管理層	6
2. テクノロジーの詳細 .....	6
2.1. Compute	6
2.1.1. Non Uniform Memory Access (NUMA) トポロジーの対応	7
2.1.2. 仮想 CPU ピニング	7
2.1.3. ラージページ (および Huge Page) のサポート	7
2.1.4. サポート (PCI パススルー)	8
2.2. ネットワーク	8
2.2.1. Single Root I/O Virtualization (SR-IOV)	8
2.2.2. ポートセキュリティーの拡張機能	8
2.2.3. IPv6 のサポート	9
2.2.4. DPDK Accelerated vSwitch	9
2.2.5. Quality of Service (QoS)	9
2.2.6. OpenDaylight	9
2.3. ストレージ	10
2.3.1. ボリュームの移行	10
2.3.2. スナップショットのインポートおよびエクスポート	10
2.4. 高可用性	10
2.4.1. 高可用性 (keepalived)	10
2.4.2. インスタンスの高可用性 (pacemaker remote)	11
2.5. モニタリング	11
2.6. リアルタイム KVM	12
3. 用語について .....	13
3.1. オープンソース	13
3.2. キャリアグレード	13
3.3. 統合エンジニアリング	13
3.4. 欧州電気通信標準化機構 (ETSI)	14
3.5. Management and Orchestration (MANO)	14
3.6. Virtual Evolved Packet Core (vEPC)	14
3.7. 仮想化 IMS (vIMS: Virtual IP Multimedia Subsystem)	14

## 1. 概要



### 注記

本書では、現在テクノロジープレビューとして提供されている Red Hat のネットワーク機能仮想化ソリューションについて記載します。テクノロジープレビュー機能のサポート範囲に関する情報は <https://access.redhat.com/support/offerings/techpreview/> を参照してください。

**ネットワーク機能仮想化 (NFV: Network Functions Virtualization)** では、仮想化の概念を利用して、あらゆるクラスのネットワークノード機能をビルディングブロックに仮想化します。それらのビルディングブロックを相互に接続して通信サービスを作成することができます。NFV は、専用のハードウェアアプライアンスをソフトウェアと自動化に置き換えることによって、ネットワークの定義、作成、管理を行う新しい方法です。

**仮想ネットワーク機能 (VNF)** は、ネットワーク機能ごとにカスタムのハードウェアアプライアンスを使用する代わりに、標準の大容量サーバー、スイッチ、ストレージでソフトウェアを実行する仮想実行環境で構成されます。ネットワークベンダーの大半は、実行環境として仮想マシンを採用することを検討していますが、効率性と俊敏性を向上させるために、コンテナベースのアプローチを検討しているベンダーもあります。サービスプロバイダーがアプリケーションをさらに重視するようになり、オーバーヘッド (ハードウェアリソースのコストやインストール) をクラウドに移動して仮想化のメリットの拡大を図るに従って、コンテナはますます重要となります。コンテナでは、1つのソフトウェアを中心として、コード、ランタイム、システムツールなど、そのソフトウェアの実行に必要な全要素が含まれたパッケージを作成することができます。これにより、ソフトウェアは別の操作環境でも同じ方法で実行できるようになります。コンテナは、密度の増大、インストール時間の短縮などのメリットをもたらします。

NFV は汎用のクラウドベースのインフラストラクチャーでネットワーク機能を仮想化して、コストの削減とイノベーションの拡大を図りつつ、レガシーのインフラストラクチャーよりも高い俊敏性、柔軟性、容易性、効率性、スケーラビリティを提供します。

NFV 環境は、仮想サーバー、ストレージやその他のネットワークなど、完全に仮想化されたインフラストラクチャーをサポートするのに必要なネットワークコンポーネントを集約して提供するように設計されています。NFV は大容量サービス、スイッチ、ストレージハードウェア上で実行される標準の IT 仮想化技術を活用して、ネットワーク機能を仮想化します。管理およびオーケストレーションのロジックにより、これらのサービスはデプロイ、維持されます。NFV には、Virtual Function (VF) のホストや接続に使用するリソースのプールである **ネットワーク機能仮想化インフラストラクチャー (NFVI)** という別のコンポーネントもあります。NFVI リソースは、**仮想化インフラストラクチャーマネージャー (VIM)** で管理、制御されます。これらのリソースは、必要に応じて VNF と **ネットワーク機能仮想化の管理とオーケストレーション (NFV MANO)** で使用されます。NFV の主な目的の1つに、ハードウェアとソフトウェアを切り離すことが挙げられます。ハードウェアとソフトウェアを切り離すには、追加の管理層が必要になります。MANO は、仮想マシンおよび VNF と対話して仮想マシンおよび VNF のライフサイクルを管理する **仮想ネットワーク機能マネージャー (VNFM)** を定義します。MANO が定義するもう1つの重要なコンポーネントに **オーケストレーター (Orchestrator)** があります。Orchestrator は、**オペレーションサポートシステムまたはビジネスサポートシステム (OSS/BSS)** などのさまざまなデータベースやシステムや VNF マネージャーと連携します。Orchestrator が VNF のインスタンスまたは仮想マシンを作成する場合は、VNF マネージャーにこのようなインスタンスを1台作成するように要求します。

NFV の実装は、ビジネス面で主に次のようなメリットをもたらします。

- ビジネスニーズの変化に対応するために新しいハードウェアをインストールする必要はないので、新たなネットワークサービスを迅速にデプロイすることが可能となり、市場投入までの時間が短縮

されます。NFVにより、プロバイダーはサービスを試験的にデプロイして発展させ、顧客のニーズを満たす最適な状態を判断することができるので、新規サービスの展開に伴うリスクが軽減されます。

- NFVにより、需要の変化に対応してサービスを迅速にスケールアップできるため、俊敏性と柔軟性が提供されます。また、業界標準のどのようなサーバーハードウェア上でもソフトウェアを使用してサービスを提供できるのでイノベーションをサポートします。
- セキュリティーやパフォーマンスを犠牲にせず、週/日単位ではなく、時間/分単位で顧客のニーズに対応します。
- 1回限りの投資が可能なので、資本支出が削減されます。
- 設備のスペース、電力、冷却要件を減らして運用コストを削減します。

## 1.1. Red Hat と NFV

Red Hat では、**Red Hat Solution for NFV** というオープンソースをベースとしたキャリアグレードのソリューションを提供しています。このソリューションは、**OpenStack** などの既存のオープンソース製品で利用できるように NFV 機能を追加して、通信サービスプロバイダー (CSP) が IT およびネットワークの収束を実現するのをサポートします。

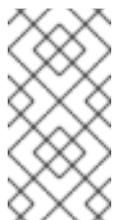
幅広いカテゴリーに分類される Red Hat NFV ソリューションの最も一般的なユースケースは以下のとおりです。

**モビリティ:** vEPC および vIMS などのモバイル CSP のデプロイサービスをサポートします。

**マネージドサービス:** マネージドサービスとは、サービスプロバイダーが大中小規模のエンタープライズに提供するサービスのことで、最も一般的なサービスは、マネージド CPE (Customer Premise Equipment)、マネージドファイアウォール、VPN 終端のゲートウェイ、侵入検出および分析です。

マネージドサービスでは、サービスプロバイダーは CPEaaS や FWaaS などの「aaS (as a Service)」を提供します。マネージドサービスには、VNF が仮想化を提供しますが、これらのサービスをリンクするサービスチェーンを作成する機能がより重要になります。これにより、クラウド内に従来のルーティングとレイヤー 2 のスイッチだけでなく、レイヤー 2 と 3 ベースの基準を超えるフローをプログラムするソフトウェア定義ネットワーク (SDN) も必要です。

**ビデオ:** Netflix、Amazon Video などのケーブルオペレーター (MSO) およびその他の Over The Top (OTT) プロバイダーは多くのサーバーとネットワークデバイスを使用します。このようなネットワークは、コンテンツ配信ネットワーク (CDN) と総称されます。これは、専用のハードウェアではなく仮想化されたデバイスのクラウドインスタンスを使用して運用コストを大幅に削減できるもう 1 つの領域です。このような仮想化 CDN は、配信ネットワークにより近いデータセンターに配置することができます。これにより、追加の容量が必要となった場合のスケールアップ、冗長化、災害復旧が簡単になります。



### 注記

Red Hat では、NFV 機能を数多くの製品に組み込み、通信事業者の期待に応えるための開発に取り組んでいます。このアプローチをキャリアグレードのオープンソース NFV ソリューションと呼んでいます。Red Hat では、NFV/電気通信会社向けに別の製品を開発する予定はありません。

共通のユニークなプラットフォームがあることによってサービスプロバイダーにもたらされる利点は、IT とネットワーク収束です。NFV アーキテクチャーに使用できる Red Hat 製品は、以下のとおりです。

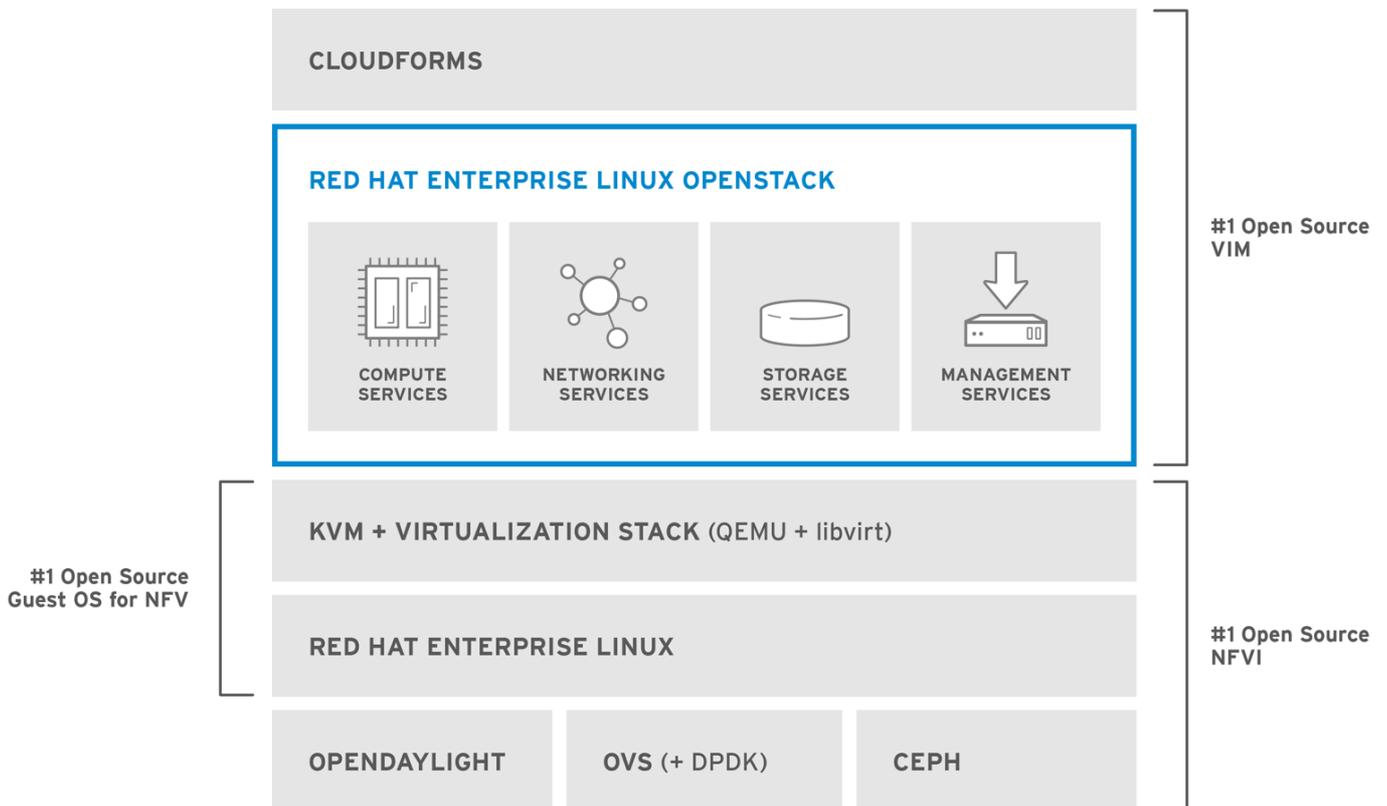
- **Red Hat OpenStack Platform:** サービスプロバイダーは、VIM として IT および NFV のワークロードを実行することが可能です。

- Red Hat Enterprise Linux および Red Hat Enterprise Linux Atomic Host: VNF として仮想マシンやコンテナを作成することができます。
- Red Hat JBoss および OpenShift Enterprise: OSS/BSS 層として機能するように変更することができます。
- Red Hat CloudForms: VNF マネージャーとして機能し、VIM や NFVI などの複数のソースのデータを統合された画面に表示することができます。

## 1.2. 設計環境

Red Hat NFV ソリューションは、Red Hat ポートフォリオの複数の製品で構成されており、以下のような数多くのテクノロジーを統合します。

- Red Hat Enterprise Linux (RHEL)
- KVM および Open vSwitch を実装した Red Hat OpenStack Platform
- Red Hat CloudForms
- Red Hat Ceph Storage



OPENSTACK\_391225\_0216

Red Hat NFV ソリューションは、以下の層で構成されます。

- インフラストラクチャー層
- 仮想化層
- OpenStack 層
- 管理層

### 1.2.1. インフラストラクチャー層

インフラストラクチャー層は、Red Hat NFV ソリューションの基盤となります。主要なコンポーネントは以下のとおりです。

- Red Hat Enterprise Linux (RHEL)
- Ceph を使用したソフトウェア定義のストレージ
- OpenDaylight または認定済みのサードパーティー製のコントローラーを使用したソフトウェア定義ネットワーク (SDN)
- スタンドアロンまたは Data Plane Development Kit (DPDK) で高速化された Open vSwitch (OVS)

キーポイント:

- NFV ユースケースにおける RHEL のキーポイントは、高可用性フレーバーと Real Time フレーバーです。
- RHEL Atomic Host は、Docker 形式で Linux のコンテナを実行するように最適化された Red Hat Enterprise Linux 7 の派生製品です。この製品は、SELinux を使用して、マルチテナント環境における強力な保護を提供します。また、アトミックなアップグレード/ロールバックを実行する機能を提供して、メンテナンスを迅速化/容易化し、ダウンタイムを短縮することができます。RHEL Atomic は、VNF をコンテナとして実行するのに非常に適しています。
- Ceph は、パフォーマンス、スケール、信頼性などのストレージの特性を提供します。
- OpenDayLight は、オープンソースの SDN コントローラーです。Red Hat NFV ソリューションは、Nuage、Contrail など複数の SDN のビジネスオフリングでも認定済みです。
- RHEL は、DPDK およびデバイス直接割り当てのテクノロジー (SR-IOV、PCI パススルー) をサポートします。DPDK は、Red Hat OpenStack Platform 9 リリースの「Extra」チャンネルの一部でテクノロジープレビューとして提供されています。

### 1.2.2. 仮想化層

仮想化層は、Red Hat Enterprise Virtualization Hypervisor により提供され、主に以下の要素で構成されます。

- Kernel-based Virtual Machine (KVM) および Quick Emulator (QEMU)
- Libvirt

キーポイント:

- リアルタイム KVM は、高速なパフォーマンスではなく、平均値の低い、予測可能なレイテンシーがより重要な場合に仮想 Radio Access network (vRAN) などで使用されます。

### 1.2.3. OpenStack 層

OpenStack 層は、主に Red Hat OpenStack Platform で構成されます。OpenStack サービスの一部のみが NFV のユースケースに関連します。Red Hat は、OpenStack の開発で最も貢献度が高いコントリビューターとして、Intel、Cisco、Huawei、NEC など、NFV 機能の分野で活躍する業界の他のリーダーとアップストリームの OpenStack コミュニティーでコラボレーションを展開しています。NFV の特別機能を必要とする主なサービスは以下のとおりです。

- Compute (nova)

- OpenStack Networking (neutron)
- Block Storage (cinder)
- Telemetry (ceilometer)

キーポイント:

- 仮想 CPU ピニング、ラージページ、NUMA 認識などの Compute の NFV 機能
- IPv6、SR-IOV および DPDK-Accelerated Open vSwitch などの OpenStack Networking の NFV 機能
- Block Storage サービスは NFV 関連の主要なストレージサービスで、バックエンドとして Ceph を使用してキャリアグレードのストレージ要件に対応します。
- OpenStack の計測サービス Telemetry は専用のデータベースでイベントを収集します。収集されたイベントは次に管理ツールに送信され、さらに使用されます。FCAPS とは、Fault (障害管理)、Configuration (構成管理)、Accounting (課金管理)、Performance (性能管理)、Security (機密管理) の略で、ISO モデルで、ネットワーク管理タスクを定義する管理カテゴリーです。

#### 1.2.4. 管理層

管理層は主に以下の要素で構成されます。

- Red Hat CloudForms
- Red Hat OpenStack Platform director
- FCAPS 運用ツール

キーポイント:

- Red Hat CloudForms は、仮想インフラストラクチャマネージャー (VIM) やネットワーク機能仮想化インフラストラクチャ (NFVI) などの複数のソースから取得したデータを、統合された画面で表示するクラウド管理ツールです。このツールにより、ハイブリッドの VIM シナリオ (レガシーの仮想化プラットフォームで実行される VNF および OpenStack プラットフォーム上で実行される NFV) を管理して FCAPS を統合することができます。
- Red Hat OpenStack Platform director では、NFV インフラストラクチャ (NFVI) をプロビジョニングして、VNF など幅広い電気通信向けのアプリケーションをサポートすることができます。
- 運用ツールは、障害管理、構成管理、課金管理、性能管理、機密管理 (FCAPS) を提供します。これらのツールを使用して、NFVI の正常性を監視することができます。これらは、より大きな規模でクラウドプラットフォームを管理するために、クラウドコミュニティが使用しているオープンソースのツールです。

## 2. テクノロジーの詳細

Red Hat NFV ソリューションの構成には、以下の機能が採用されています。

### 2.1. Compute

Compute (nova) は、必要に応じて仮想マシンを構築、スケジューリング、無効化して OpenStack 環境でインスタンスを管理します。Compute は、NFV のユースケースに適しています。たとえば、スケジューラーは、パフォーマンスや耐障害性に極めて重要です。これは、インスタンスを正しく設置することで実現でき

ます。また、初回起動時と、障害検出時の対応の際にはいずれも、迅速に新規インスタンスを起動する必要があります。Compute には、以下のような主要な機能があります。

### 2.1.1. Non Uniform Memory Access (NUMA) トポロジーの対応

NUMA では、システムメモリーは、特定の CPU またはソケットに割り当てられるノードと呼ばれるゾーンに分割されます。CPU のローカルにあるメモリーには、そのシステムのリモートの CPU に接続されているメモリーにアクセスするよりも高速です。通常、NUMA システム上のソケットにはそれぞれローカルのメモリーノードがあり、別の CPU のローカルにあるノードのメモリーや、全 CPU で共有されるバス上のメモリーよりもコンテンツに早くアクセスできます。各 CPU ソケットには、仮想化を目的とする個別の CPU として扱われる複数の CPU コアを配置することができます。

OpenStack Compute (nova) は、インスタンスの起動時にスマートなスケジューリングや配置の意思決定を行います。管理者はこのような機能を活用すると、NFV や高性能コンピューティング (HPC) などの特化されたワークロードを対象にするカスタマイズされたパフォーマンスフレーバーを構築することができます。詳しい情報は「[NUMA](#)」を参照してください。

### 2.1.2. 仮想 CPU ピニング

CPU ピニングとは、特定のホストにある特定の物理 CPU で、特定の仮想マシンの仮想 CPU を実行する機能のことです。

OpenStack Compute 環境は、専用の物理 CPU コアへの仮想マシンインスタンスのピンニングをサポートします。この設定は、以下の手順に従って行います。

- ホストプロセス用にコンピュータホスト上の専用コアを確保します。
- 仮想マシンインスタンス自体のためにコンピュータホスト上の専用コアを確保します。
- 必要なスケジューラーフィルターを有効化します。
- CPU ピニング用に設定されたホストをすべて追加するためのホストアグリゲートを作成します。
- このホストアグリゲートを対象にするパフォーマンスを重視したフレーバーを作成します。
- CPU ピニング用に設定されていない全ホストためのホストアグリゲートを作成して、そのホストアグリゲートにその他のフレーバーすべてをマッピングします。
- CPU ピニングが設定されたインスタンスを起動します。

詳しい情報は、「[NUMA ノードを使用する CPU ピニングの設定](#)」を参照してください。

### 2.1.3. ラージページ (および Huge Page) のサポート

物理メモリーは、ページと呼ばれる連続した一連のリージョンに分割されます。効率性を向上させるには、システムはメモリーの個別のバイトにアクセスする代わりに、ページ全体にアクセスしてメモリーを取得します。ページにはそれぞれ、ページサイズと呼ばれるバイト数が含まれています。ページにアクセスするには、システムはまず仮想アドレスを物理アドレスに変換して、どのページに要求したメモリーが含まれているかを判断します。

基礎的なレベルでは、メモリーを大量に使用するプロセスまたはメモリー集約型のプロセスは、通常ラージページおよび Huge Page と呼ばれる、大きいサイズのページによってメリットがもたらされる場合があります。Red Hat Enterprise Linux 7.1 システムは、2 MB と 1 GB の Huge Page をサポートします。これらは、ランタイム時または起動時に割り当てることができます。詳しい情報は「[Huge Page および Transparent Huge Page \(THP\)](#)」を参照してください。

### 2.1.4. サポート (PCI パススルー)

PCI パススルーにより、ゲストは PCI デバイスに排他的にアクセスして、さまざまなタスクを実行することができます。標準のパススルーにより、仮想マシンによる PCI デバイスの排他的アクセスが可能となり、また PCI デバイスはゲスト OS に物理的に接続されているかのように表示および動作することができます。ネットワークの場合には、PCI パススルーを利用して、ネットワークデバイス全体 (例: ネットワークアダプター上の物理ポート) を仮想マシン内で実行されているゲスト OS の専用にすることが可能です。

Single Root I/O Virtualization (SR-IOV) とは、単一の PCI デバイスが、さまざまな PCI ハードウェア機能の中の複数のリソースへのアクセスを分離できるようにする仕様です。

詳しくは、「[PCI デバイスの NUMA ノード局所性](#)」を参照してください。

## 2.2. ネットワーク

パフォーマンス要件は、主として高性能のパケット処理 (ネットワークからパケットを取得して、仮想マシンに送り、迅速に処理してネットワークに戻す方法) が対象です。利用可能なテクニックの1つに、仮想マシンが SR-IOV を介してネットワークに直接物理的にアクセスするという方法があります。OpenStack Networking は、特定の構成で IPv6 テナントのサブネットをサポートしています。また、仮想化スイッチを加速してパケット処理を高速にする DPDK-Accelerated Open vSwitch も、Red Hat OpenStack Platform 9 リリースの OpenStack Networking ではテクノロジープレビュー機能としてサポートされています。

### 2.2.1. Single Root I/O Virtualization (SR-IOV)

SR-IOV により、単一ルートの機能 (例: 単一のイーサネットポート) を複数の別個の物理デバイスのように提示することができます。SR-IOV 対応の物理デバイスは、PCI 設定領域で複数の機能として提示するように設定することができます。各デバイスには、Base Address Registers (BAR) を備えた独自の設定領域があります。SR-IOV は 2 つの PCI 機能を使用します。

- Physical Function (PF) は、SR-IOV 機能を含む完全な PCIe デバイスです。PF は、通常の PCI デバイスとして検出、管理、設定されます。PF は、Virtual Function (VF) を割り当てることで SR-IOV の機能性を設定し、管理します。
- Virtual Function (VF) は、I/O のみを処理する単純な PCIe 機能です。それぞれの VF は、PF に由来しています。デバイス 1 台に備わる VF の数は、デバイスのハードウェアによって制限されます。物理デバイスであるシングルイーサネットポートは、仮想マシンで共有可能な多くの VF を呼び出すことができます。

ハイパーバイザーは 1 つまたは複数の VF を 1 台の仮想マシンにマッピングすることができます。VF の設定領域は、ゲストに提示される設定領域にマッピングされます。

OpenStack Networking は、仮想ブリッジの以前の要件を無視して、物理 NIC の機能をインスタンスに直接拡張することができます。詳しい情報は「[仮想ネットワークの SR-IOV サポート](#)」を参照してください。

### 2.2.2. ポートセキュリティの拡張機能

以前のリリースでは、セキュリティ対策として、OpenStack Networking のセキュリティーグループを実装することにより、デフォルトのルールが常に適用され、IP アドレススプーフィング攻撃を阻止して、仮想マシンがそのセキュリティーグループのポートに所属しない MAC または IP アドレスでトラフィックを送受信しないようにしていました。

ネットワーク機能が仮想マシン内で実行される NFV のユースケースでは、デフォルトのスプーフィング対策ルールをオフにするオプションが必要です。この拡張機能により、テナント管理者がトポロジーにファイアウォールが必要かどうか、またどこに必要なかを決定できるように、管理者は「port-security-enabled」属性を使用してセキュリティーグループ機能をポートごとに有効化/無効化できるようになります。

### 2.2.3. IPv6 のサポート

OpenStack Networking は、プロジェクトが、Stateless Address Autoconfiguration (SLAAC) または DHCPv6 を使用して仮想マシンに動的に IPv6 アドレスを割り当てることができるように、二重スタック構成での IPv6 テナントのサブネットをサポートします。また OpenStack Networking は、仮想マシンが IPv6 アドレスを既存のインフラストラクチャーから取得できるように、物理ルーターで SLAAC を統合することができます。詳しい情報は「[IPv6 を使用したテナントネットワーク](#)」を参照してください。

### 2.2.4. DPDK Accelerated vSwitch

DPDK は、ライブラリーのセットとユーザー空間ドライバーで構成されます。このキットは、パケットの処理を高速化し、アプリケーションが NIC と直接やりとりして独自のパケット処理を実行できるようにします。特定のユースケースでは、最大でワイヤースピードのパフォーマンスを実現します。また、DPDK-Accelerated Open vSwitch (OVS+DPDK) により、Open vSwitch のパフォーマンスが大幅に向上するとともに、中核的な機能が維持されます。ホストの物理 NIC からゲストインスタンス内のアプリケーションへ（およびゲストインスタンス間）のパケットの切り替えがほぼすべてユーザー空間で処理できるようになります。

OpenStack Networking (neutron) OVS プラグインが更新されて、OVS+DPDK バックエンドの設定をサポートするようになりました。詳しい情報は、「[OpenStack Networking の DPDK の設定](#)」を参照してください。

DPDK-Accelerated vSwitch 機能は、Red Hat OpenStack Platform 9 リリースのテクノロジープレビューとして提供されています。

### 2.2.5. Quality of Service (QoS)

QoS は、アプリケーションプロバイダーとエンドユーザー間のサービスレベルアグリーメント (SLA) を満たすために、帯域幅、レイテンシー、信頼性などの特定のネットワーク要件を保証するリソース制御システムのことを指します。スイッチやルーターなどのネットワークデバイスは、トラフィックにマークを付けて優先順位を高くして処理し、SLA で合意した QoS 条件を満たすようにすることができます。また、VoIP (Voice over IP) やビデオストリーミングなどの特定のネットワークトラフィックは、帯域幅の制約を最小限に抑えて配信する必要がある場合があります。ネットワーク QoS を管理していないシステムでは、全トラフィックが「ベストエフォート」型で配信されるため、顧客へのサービスの提供を保証することができなくなります。

Red Hat OpenStack Platform 9 では、仮想マシンの速度を制限できるように、送信帯域幅の制限ルールがサポートされます。詳しい情報は「[Quality-of-Service \(QoS\) の設定](#)」を参照してください。

### 2.2.6. OpenDaylight

OpenDaylight は、顧客、パートナー、開発者向けに、業界全体で共通のソフトウェア定義ネットワーク (SDN) のオープンソースフレームワークとプラットフォームを提供します。SDN により、ユーザーはネットワーク層をプログラムして、データプレーンとコントロールプレーンを分離することが可能となる一方で、NFV により、必要なタイミングで必要な場所にネットワークサービスを俊敏に配置できるようになります。このようなレベルのプログラマビリティを可能にすることで、SDN と NFV は、ネットワークリソースの最適化、ネットワークの俊敏性およびサービスイノベーションの向上、サービスを市場投入するまでの時間の短縮、ビジネスインテリジェンスの抽出を可能にし、最終的には動的なサービス重視の仮想ネットワークを実現することができます。

OpenDaylight は、Red Hat OpenStack Platform 9 リリースでは、テクノロジープレビューとしてサポートされています。Red Hat の OpenDaylight のディストリビューションは、OpenStack が中心となっており、OpenStack Networking と (OVSDB NetVirt 経由の) Open vSwitch に必要なインターフェースのみが同梱されています。

## 2.3. ストレージ

OpenStack のストレージは、クラウドのデータ保護と障害復旧において重要な役割を果たします。Red Hat OpenStack Platform により、顧客およびユーザーが重要なデータをクラウドストレージにセキュアにバックアップできる上、オフサイトのバックアップに対するコンプライアンス要件を満たすこともできます。また、障害復旧のための第 2 のクラウドを維持するコストを削減する一方で、クラウドストレージの提供する低コストのメリットを享受することができます。

### 2.3.1. ボリュームの移行

OpenStack では、対象となるボリューム種別をサポートするバックエンド間でボリュームを移行することができます。ボリュームを透過的に移行すると、ボリュームの現在のバックエンドから新しいバックエンドへデータが移動します。ボリューム移行の機能を使用すると、移行の進捗状況の追加、Ceilometer ヘレポートすることによる通知システムの強化、Tempest テストや CI サポートによる移行品質の保証など、ボリューム移行のステータス管理を改善することができます。詳しい情報は「[Migrate a Volume](#)」を参照してください。

### 2.3.2. スナップショットのインポートおよびエクスポート

スナップショットのインポート/エクスポートの手段を提供します。「エクスポート」では、スナップショットを別の Block Storage ボリュームに移行することができ、「インポート」ではバックエンドデバイスにある OpenStack 以外で作成されたスナップショットを OpenStack または Block Storage ボリュームにインポートすることができます。これには以下のような利点があります。

- ボリュームテンプレートとしてスナップショットをインポートして、スナップショットからボリュームを作成することができます。
- スナップショットをインポートして効果的にボリュームのインポートを管理することができます。

## 2.4. 高可用性

高可用性とは、パフォーマンスの低下やサブシステムの障害に対応するために、設計に十分なキャパシティを追加して、長期間継続して稼働するシステムまたはコンポーネントのことを指します。可用性は、「100% 稼働」または「障害発生なし」という基準に対して相対的に測定することができます。一般的に使用されているが実現が困難なシステム/製品向けの可用性基準は、「ファイブナイン」(99.999%)の可用性と呼ばれます。高可用性システムは、システムのダウンタイムおよびデータの損失を最小限に抑えることを目的としています。

高可用性を実装するには、ハードウェアを冗長化にして各サービスのインスタンスを冗長構成で実行します。サービスのインスタンスを実行するハードウェアの1つが機能しなくなった場合は、システムはフェールオーバーして、そのサービスの別のインスタンスを使用することができます。

### 2.4.1. 高可用性 (keepalived)

Keepalived は、アクティブ Linux Virtual Servers (LVS) ルーターと、1つまたは複数のバックアップ LVS ルーター (オプション) で実行されます。アクティブ LVS ルーターには 2 つの役割があります。

- 複数の実サーバー間における負荷分散
- それぞれの実サーバー上のサービスの整合性チェック

アクティブ (マスター) ルーターは、Virtual Router Redundancy Protocol (VRRP) を使用してアクティブな状態をバックアップルーターに通知します。これにはマスタールーターが一定の間隔で広告を送信する必要があります。アクティブルーターが広告の送信を停止すると、新しいマスターが選出されます。

keepalived デーモンは、サービスまたはシステムを監視し、問題が発生した場合には自動的にスタンバ

サーバーにフェールオーバーするために使用されます。**keepalived** デーモンは、一式のチェッカーを実装し、負荷分散の対象となるサーバープールを正常性の状態に応じて動的かつ適応的に維持/管理します。また、このデーモンは、**Virtual Router Redundancy Protocol (VRRP)** の有限状態機械 (FSM) にフックセットを実装して、低レベルで高速なプロトコル対話を提供します。**keepalived** のフレームワークは、単独または全体で使用して、耐障害性の高いインフラストラクチャーを提供することができます。

#### 2.4.2. インスタンスの高可用性 (pacemaker remote)

**Pacemaker** は、Linux プラットフォーム向けの最新の高可用性および負荷分散スタックです。インスタンスの高可用性は、仮想マシンに障害が発生していないかどうかを監視して、フェンシングおよび復旧のための自動応答をトリガーすることで実現されます。フェンシングとは、障害の発生したホストを完全に切り離す操作です。リカバリーシステムは、機能しなくなったホストから仮想マシンを復旧するようにオーケストレーションします。

**Pacemaker** は、**OpenStack** のインフラストラクチャーを高可用性にする際に役立ちます。**Pacemaker** は、管理対象のアプリケーションを継承して認識せず、リソースエージェント (RA) に依存します。リソースエージェントは、クラスターが管理する各アプリケーションの起動、停止、正常性のチェックの方法に関する知識をカプセル化するスクリプトです。

クラウドや仮想化環境においてワークロードの高可用性を実現するためのソリューションには、主に 3 つの機能があります。

- 任意のコンピュータノードでエラーが発生したことを検知して、障害対応をトリガーする監視機能
- 環境から対象のコンピュータノードを除外するフェンス機能
- 障害が発生したコンピュータノードからのインスタンスの復旧をオーケストレーションするリカバリー機能

#### 2.5. モニタリング

システムを監視することで、最適なパフォーマンスを確保します。収集したデータは、顧客の請求、システムの監視または警告に使用することができます。障害の発生は避けられないことを認識した上で、**OpenStack** は耐障害性のモニタリング、障害検知、対応に重点的に取り組む必要があります。

**OpenStack** では、以下にあげるメトリックおよびイベントを監視して、デプロイメントの正常性とパフォーマンスを可視化します。

- `hypervisor_load`
- `running_vms`
- `free_disk_gb`
- `free_ram_mb`
- `queue memory`
- `queue consumers`
- `consumer_utilisation`

インフラストラクチャーおよびユースケースに特に関連するメトリックを追加で監視することができます。監視の対象は、使用する運用ツール (FCAPS) と **OpenStack** コンポーネントの両方により異なります。

FCAPS ツールは、障害、構成、課金、性能、機密上の問題を管理することができます。FCAPS 管理ツールの目的は以下のとおりです。

- 障害とは、悪影響をもたらすイベントのことです。障害管理では、ネットワーク内で発生した障害を認識、分離、修正、ログ記録します。また、傾向分析を使用してエラーを予測し、ネットワークの高可用性を維持できるようにします。
- 構成管理は、システムの構成情報を管理して、変更を監視します。構成ファイルへの変更、ソフトウェアバージョンの更新、システムハードウェアへの変更が直接の原因で多くのネットワークの問題が発生するため、構成管理は特に重要です。
- 課金管理は、ネットワークの使用状況の情報をトラッキングして、個別のユーザー、部署、事業部門に適切に請求し、会計目的で課金できるようにします。請求の発生しないネットワークでは、「会計」ではなく「アドミニストレーション」に置き換えられます。アドミニストレーションの目的は、ユーザー、パスワード、パーミッションを確立して権限のあるユーザーセットを管理すること、さらにソフトウェアのバックアップや同期の実行など設備の運用を管理することです。課金は通常、請求管理とも呼ばれます。統計を使用して、ユーザーに課金して、使用クォータを有効にすることができます。これには、ディスクの使用状況、リンクの使用、CPU の時間などが含まれます。
- 性能管理は、ネットワークパフォーマンスを許容レベルに維持することに重点を置きます。これにより、マネージャーは将来に向けてネットワークの準備を整え、設置に費やした投資などに関連した現在のネットワークの効率性を判断することができます。ネットワークパフォーマンスのモニタリングは、スループット、ネットワーク応答時間、パケット損失率、リンク使用状況、利用率、エラー発生率などが対象です。
- 機密管理は、ネットワーク内のアセットへのアクセスを制御します。主に認証や暗号化でデータのセキュリティを確保することができます。機密管理は、ネットワーク環境がセキュアであることを確認するだけでなく、収集されたセキュリティ関連の情報が定期的に分析されるようにします。機密管理の機能には、内部および外部のユーザー適切なネットワークリソースにのみアクセスできるように、ネットワーク認証、承認、監査が含まれています。他の一般的なタスクには、ネットワークファイアウォールの設定や管理、侵入検知システム、セキュリティポリシー (アクセスリストなど) が含まれます。

## 2.6. リアルタイム KVM

既存のハイパーバイザーは、NFVI の要件を満たすために設計されているだけでなく、目的が異なるので、NFV 機能を有効にするにはさらなる連携が必要です。KVM では、高パフォーマンスおよび低レイテンシーを実現できます。ユースケースによってレイテンシーの許容範囲も異なりますが、多くの場合は、レイテンシーが 20 ミクロ秒未満である必要があります。また、レイテンシーを 5-10 ミクロ秒に抑えて、パフォーマンスを保証する必要がある場合もあります。これらの数字を達成するには、システムを慎重に設定して、CPU 周波数の変更、カーネルモジュールのロードおよびアンロード、スワッピングをしないなど、レイテンシーが高くなる各種システム操作を避ける必要があります。アプリケーションは、リアルタイム以外のヘルパープログラムを除くディスクやサウンドデバイスといった速度の遅いデバイスを避けるように調節する必要があります。そのため、リアルタイム KVM をデプロイするには、タイムスタンプのカウンターが安定した状態を確保して、システムが別のクロックソースにフォールバックしないようにするなど、システムに関する深い知識が必要です。

パフォーマンスやレイテンシーの処理は、CPU (周波数、電源管理、機能など)、マイクロアーキテクチャリソース、キャッシュの階層、サイズ、メモリー (および NUMA などの階層)、速度、相互接続、I/O、I/O NUMA、デバイスなど複数の要因により異なります。

NFV 向けのリアルタイム KVM の初回の機能拡張は以下の通りです。

- データプレーン VNF の最小割り込み待ち時間
  - ラジオスケジューリングなどスケジューリング操作におけるタイミング精度を保つことにより、リアルタイム VNF のタイミング精度でタイミングの変動を最小限に抑えることができます。

- パケット処理に適用するパケットの遅延時間の変動を最小限に抑えることによって、データプレーン VNF のパケットの遅延時間の変動を最小限に抑えることができます。

割り込み遅延 (ハードウェアの割り込みが入ってから、仮想マシンの割り込みハンドラーが呼び出されるまでの時間) とは、デバイスの種別によりタイミング変動、パケット遅延、または両方となります。仮想タイマーでの割り込み遅延で、パケット処理のポーリングしか使用しない場合でもリアルタイム VNF でのタイミング精度に問題が発生する可能性があります。仮想マシン内の割り込み遅延変動を最小限に抑えるには、VNF を正しく実装する必要があります。

リアルタイム KVM は、Red Hat OpenStack Platform 9 リリース以降で、テクノロジーレビューとしてサポートされます。

### 3. 用語について

このセクションでは、本書で使用する用語について説明します。

#### 3.1. オープンソース

オープンソースは、Red Hat の DNA に浸透しています。NFV の導入を始めるにあたっての CSP の主要な要件は、迅速なイノベーションと、ベンダーロックインを防ぐことの 2 つです。オープンソースは、これらの要件を満たすことができます。オープンソースの概念は、新たな NFV 機能はいずれも、まず最初にアップストリームのコミュニティーに提出するというコミットメントに結びついています。オープンソースでは、プロプライエタリー機能のサポートに掛かるような目に見えない保守コストは発生しないことが利点です。

Open Platform for NFV (OPNFV) では、開発者やソリューションプロバイダーのコミュニティーからの投資を活用する NFV ソリューションをデプロイするオープンソースのプラットフォームを提供します。OPNFV はキャリアグレードの統合型オープンソースプラットフォームで、NFV 製品やサービスの導入を加速します。Red Hat は OPNFV プロジェクトに積極的に参加しています。

#### 3.2. キャリアグレード

長年の間、電気通信サービスプロバイダーは、「シックスナイン」の信頼性を保証できるように、幅広い機能をネットワークに組み込めるように設計してきました。シックスナインとは、1年間のダウンタイムを 32 秒以下とする、最大 99.9999% の割合でネットワークを保証するという意味です。サービスプロバイダーがそれぞれのプランを精緻化して、NFV をネットワークに段階的に導入するにあたって、仮想化された機能を組み込んだネットワークインフラストラクチャーにおけるシックスナインの信頼性に対する影響を考察しています。キャリアグレードの信頼性を実現するための要件には、可用性、セキュリティー、パフォーマンス、管理などが含まれます。

Red Hat の戦略は、お客様およびパートナーと連携し、アップストリームを優先する方針のオープンソース方式で機能を開発して、キャリアグレードのソリューションを提供することです。このため、Red Hat は製品オフラインに機能を統合する前に、オープンソースプロジェクトに統合します。OpenStack 環境に VNF をデプロイするコンテキストでは、キャリアグレードという用語は、技術が成熟するとともに再定義/推敲する必要があります。Red Hat では、ネットワークパフォーマンス、高可用性、管理可能性、デプロイメントの容易性/ユーザビリティなど複数の機能をすでに開発していますが、開発は完了していません。Red Hat は、Intel と連携して残りのギャップを解決し、運用および実稼働の準備を整えることによって、CSP がレガシーのハードウェアの場合と同様の確信を持って VNF をデプロイできるようにすることを目標としています。

#### 3.3. 統合エンジニアリング

完全なオープンソース NFV ソリューションを構築するには、極めて異なるコミュニティー (OpenStack、KVM、Linux など) からの複数のコンポーネントを統合する必要があります。Red Hat は、これらのコミュニティーの主要なコントリビューターで、すべてのコンポーネントを統合する商用ソリューションを提案しています。このソリューションは、以下の目的を達成する鍵となります。

1. 新しい NFV 機能を開発すること。たとえば、新規 NFV 機能を開発すると通常、Linux、KVM、Libvirt、OpenStack Compute、OpenStack Networking サービスに変更を加えたり、同期したりする必要があります。
2. それらの NFV 機能をサポートすること。機能の開発を担当するチームは、開発した機能のサポートも担当している場合が多くあります。これは、DevOps の概念です。キャリアグレードの SLA を保証するには、コミュニティのサポートと商業的なサポートを提供するエキスパートの両方が必要となります。

### 3.4. 欧州電気通信標準化機構 (ETSI)

ETSI は、ヨーロッパの情報通信技術 (ICT) に関する標準を開発する独立標準化グループです。欧州電気通信標準化の Industry Specification Group for Network Functions Virtualization (ISG NFV) は、電気通信ネットワークにあるさまざまな機能を仮想化するための要件やアーキテクチャーの開発を担当します。

ネットワーク機能の仮想化 (NFV) は、プロプライエタリーのハードウェアデバイスの使用に伴う問題への対処に重点を置いています。標準の IT 仮想化技術を進化させることによって、NFV はネットワーク機能をソフトウェアに実装して、業界標準の幅広いサーバーハードウェアで実行して、必要に応じてネットワーク内のさまざまなロケーションに簡単に移動できるようにします。NFV を使用すると、新しい設備を設置する必要がなくなります。ETSI ISG NFV は、仮想化された機能のメンテナンスに必要なハードウェアやソフトウェアの要件およびアーキテクチャーの仕様を設定するのに役立ちます。ETSI ISG NFV は、ネットワーク機能の開発に関する指針も管理します。

### 3.5. Management and Orchestration (MANO)

NFV の MANO アーキテクチャーは、クラウドインフラストラクチャーの管理、ホスティングされている要素または VNF および VNF で構成されるサービスをデプロイおよび接続する役割を果たします。

### 3.6. Virtual Evolved Packet Core (vEPC)

vEPC は、Long-Term Evolution (LTE) ネットワークのコアコンポーネントにプラットフォームを提供するクラウドアーキテクチャーです。このアーキテクチャーでは、ゲートウェイやモバイルエンドポイントを動的にプロビジョニングして、スマートフォンやその他のデバイスからのデータトラフィック量が増加した場合でも処理能力を維持することができます。

### 3.7. 仮想化 IMS (vIMS: Virtual IP Multimedia Subsystem)

vIMS により、無線または有線を使用した既知の全パケット通信形式の IP (インターネットプロトコル) を容易に使用できるようになります。