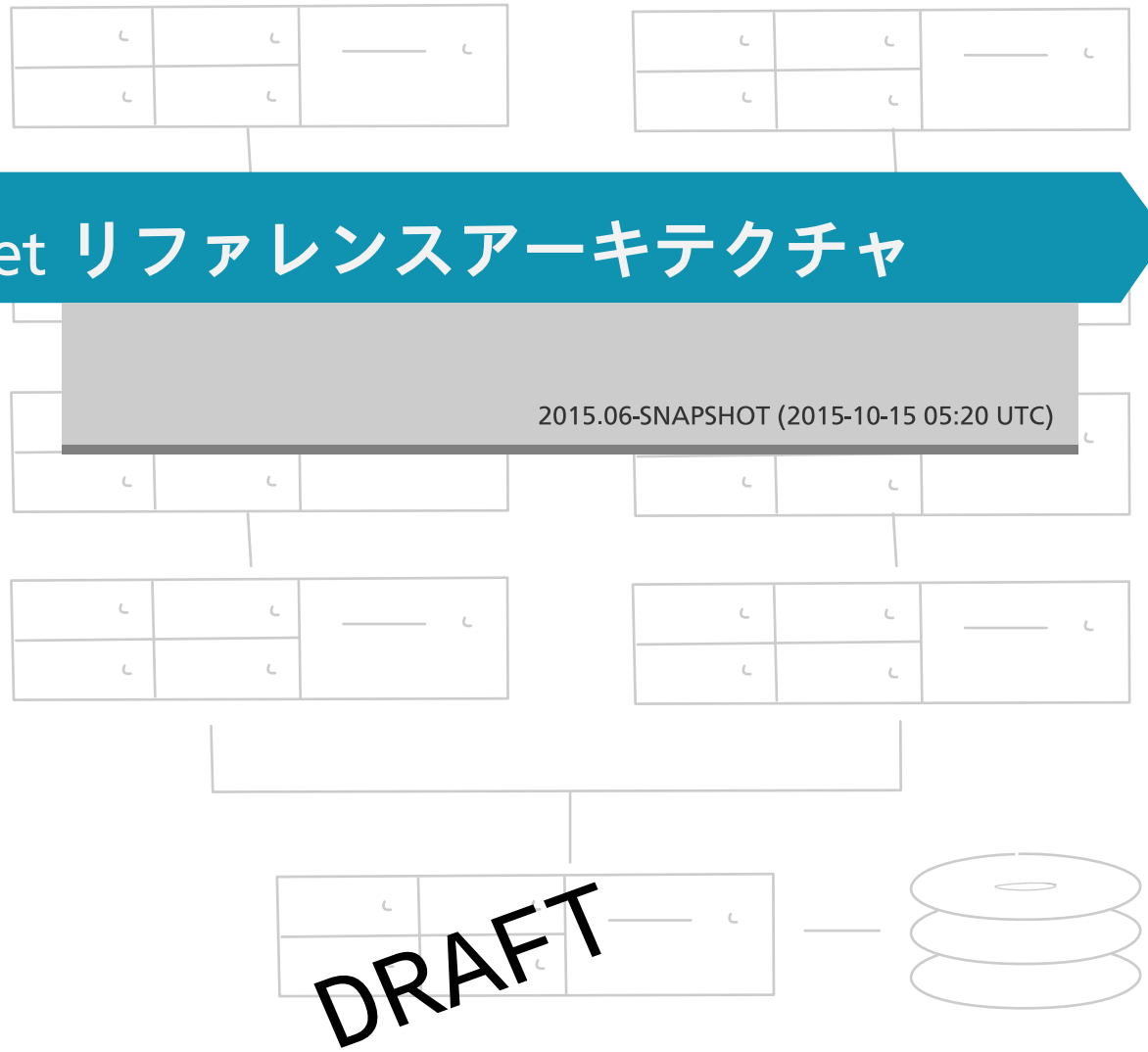
MidoNet リファレンスアーキテクチャ

2015.06-SNAPSHOT (2015-10-15 05:20 UTC)



midonet

docs.midonet.org

MidoNet リファレンスアーキテクチャ

2015.06-SNAPSHOT (2015-10-15 05:20 UTC)

概要

MidoNetは、Infrastructure-as-a-Service (IaaS)のためのネットワーク仮想化ソフトウェアです。

これにより、ネットワークハードウェアとIaaSクラウドを切り離すことができ、ホストと物理ネットワークの間に、インテリジェントなソフトウェア抽象レイヤーを作成することができます。

この文書には、推奨されるハードウェアを含め、MidoNetのネットワーク仮想化インフラストラクチャを準備するのに便利な情報が含まれています。特に、この文書では次について説明します。

- MidoNetの概要。
- OpenStack®およびその他のクラウドコントローラに対してMidoNetネットワーク仮想化を構成するのに必要なハードウェアとオペレーティングシステムソフトウェアの概要。
- ボーダーゲートウェイプロトコル（BGP）の設定とMidoNetネットワークアーキテクチャの一般的な概要。

注意

この文書はドラフトです。それは、関連する情報が欠落しているか、テストされていない情報が含まれていることができる。 ご自身の責任でそれを使用してください。

注記

援助を必要とする場合は、 MidoNetメーリングリストやチャット までご連絡ください。

目次

図の一覧

表の一覧

はじめに

表記規則

MidoNet のドキュメントは、いくつかの植字の表記方法を採用しています。

注意

注意には以下の種類があります。

注記

簡単なヒントや備忘録です。

重要

続行する前に注意する必要があるものです。

警告

データ損失やセキュリティ問題のリスクに関する致命的な情報です。

コマンドプロンプト

$ プロンプト

root ユーザーを含むすべてのユーザーが、$ プロンプトから始まるコマンドを実行できます。

プロンプト

root ユーザーは、# プロンプトから始まるコマンドを実行する必要があります。利用可能ならば、 これらを実行するために、sudo コマンドを使用できます。

第1章 MidoNetの概要

目次

MidoNetはIaaS（Infrastructures-as-a-Service/サービスとしてのインフラストラクチャ）のための、分散/非集中型のソフトウェア定義仮想ネットワークプラットフォームです。

MidoNetは、OpenStackなどのIaaS製品向けのネットワーク機能を完全に仮想化し、機能性、堅牢性、拡張性に優れた安全なネットワークを提供します。 MidoNetは標準のx86サーバー上でソフトウェアを実行するオーバーレイネットワークです。拡張可能なネットワークアンダーレイ（物理サーバーとスイッチなど）の上位に位置し、ソフトウェアによってインテリジェントなネットワーク機能をネットワークエッジに提供します。

MidoNetでは、仮想ネットワークのトラフィックはエッジ間に作成されたトンネルを通して送信されます。このトンネルでは仮想マシン（VM）と外部ポートからのパケットがカプセル化され、物理ネットワークからの仮想ネットワークトラフィックと分離されます。 このモデルでは、仮想ネットワークの変更、たとえば新しい仮想マシンを作成した場合でも、基盤となるネットワークの状態に影響しません。また、仮想と物理のネットワークが明確に分離されるため、管理者によるIaaSプラットフォームの管理が簡略化されます。

MidoNetの主な特長

MidoNetの主な特長は次のとおりです。

- レイヤ2からレイヤ4までの完全な仮想化ネットワーク
- VLANによらないVLAN - 従来のVLANの制限を実質的に排除した仮想L2分散アイソレーションとスイッチング
- 単一障害点のない完全な分散アーキテクチャ
- 仮想L3分散ルーティング
- 分散型のロードバランス機能とファイアウォールサービス
- ステートフルおよびステートレスNAT
- アクセスコントロールリスト（ACL）
- レストフルAPI
- ネットワークサービスの監視
- VXLANのサポート： VXLANトンネルゾーン、VXLAN L2ゲートウェイ
- ゼロディレイ（遅延なし）のNAT接続トラッキング

推奨されるハードウェア

このセクションでは、MidoNetの導入に推奨されるハードウェアの情報を提供します。

表1.1 推奨される導入ハードウェア

ハードウェア	要件
ネットワークステイトデータベース、APIおよびエージェントノード	CPU： 64-ビット x86、クアッドコア以上 メモリ： 32GB以上（RAM） HDD： 30GB以上（空き容量） NIC： 1Gbit以上x 2
GWノードx 2	CPU： 64-ビット x86、クアッドコア以上 メモリ： 32GB以上（RAM） HDD： 30GB以上 ネットワーク： 1Gbit以上x 3
NICカード：	高性能データネットワーク: 複数のキューとMSI - XをサポートするNICを使用
トップオブラックスイッチ	ジャンプフレームをサポートするノンブロッキングマルチレイヤースイッチ（L2/L3）
ハードィスク	理想的には、ZooKeeperのトランザクションログとCassandraデータファイルは、ホスト上の他のサービスの追加が可能な専用ディスクが必要です。ただし、小規模なPOCSやデプロイメントの場合は、Cassandraディスクを他のサービスと共有し、ZooKeeperのトランザクションログを専用にするだけでも構いません。

インストールの要件

オペレーティングシステム

MidoNetは次のオペレーティングシステムの64ビットバージョンに対応しています。

- Ubuntu 14.04 LTS
- Red Hat Enterprise Linux 7
- CentOS 7

BGPの設定要件

GWノードにBGPを設定するには、次のハードウェアと情報が必要です。

- ボーダールーターに接続したGWノード2台。一般的には、ロードバランスのために、各GWノードは異なるボーダールーターに接続します。
- 各GWノードには少なくとも2つの物理ネットワークインターフェイスが必要とされ、1つは内部ネットワーク、もう1つは上流のボーダールーターに接続します。
- 使用しているローカル（プライベート）ネットワークの自律システム（AS）番号
- リモート（パブリック）ネットワーク数（インターネットサービスプロバイダ（ISP）またはデータセンターなど）

- 1つ以上のボーダールーターのインターフェイスのIPアドレス
- GWノードの仮想ポートのIPアドレス
- 各GWノードに対するアドバタイジングルートのIPアドレス

OpenStackの統合

MidoNetはOpenStackとの連係に優れ、レイヤー2のネットワーク隔離、レイヤー3のルーティング、セキュリティグループ、フオローティングIPなど、OpenStackの現在のネットワーク機能のほとんどすべてに対応しています。

MidoNetは、OpenStack ネットワークプラグインとOpenStack ネットワークドライバによってOpenStackと統合できます。 エンドユーザーは同じOpenStack API、GUI、CLIコマンドを使用できます。MidoNetでは既存のAPIコールを使用して、すべてのネットワークコールをシームレスに処理できます。

また、OpenStackとMidoNetのソフトウェアをホストするすべてのサーバーを接続して、管理ネットワークを構成できます。 このネットワークでは、クラウドソフトウェアのアウトオブバンドを管理できます。

OpenStackは*コントローラノード*と*コンピュートノード*の2種類のノードに導入できます。

- コントローラノードは通常、nova-api、nova-cert、nova-conductor、nova-scheduler、nova-consoleauth、neutronなどのサービスをホストします。
- コンピュートノードは通常、nova-computeとハイパーバイザサービスをホストします。 nova-computeはコントローラノードにインストールしないでください。

注記

RHELディストリビューションでは、OpenStackパッケージ名の前に「openstack」が付いていることがあります（openstack-nova-apiなど）。

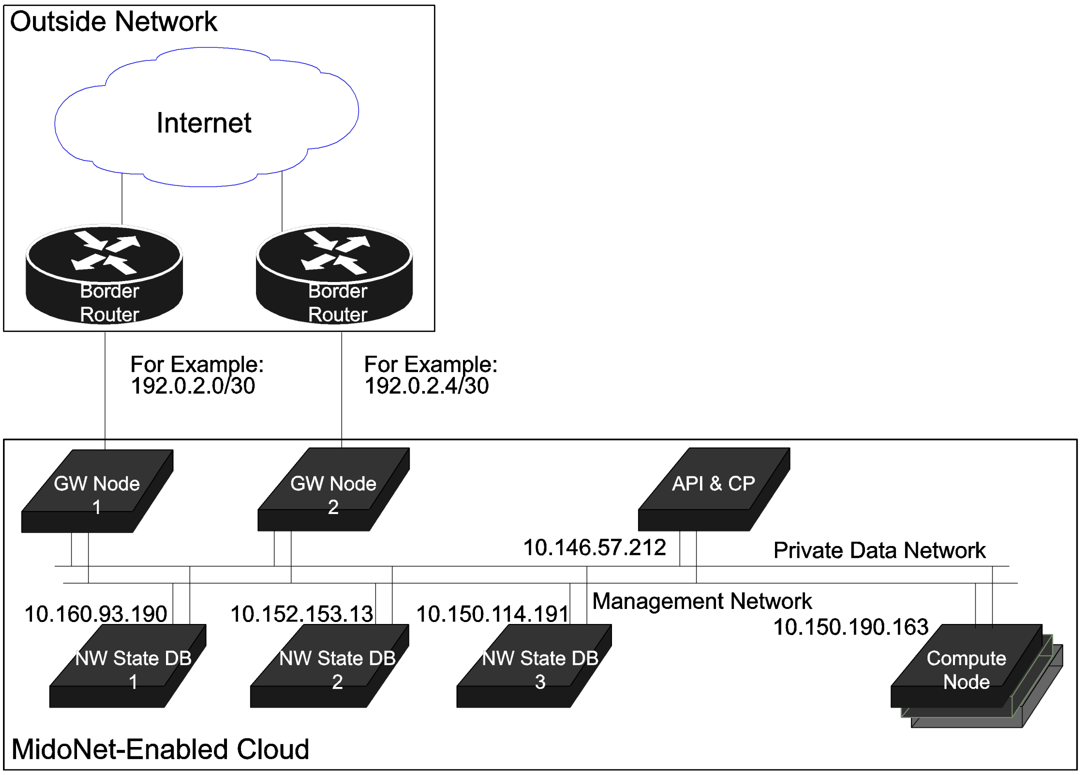
第2章 MidoNetのネットワークアーキテクチャ

目次

MidoNetでは、既存の物理ネットワークの上位のオーバーレイネットワークを運用します。 MidoNetエージェントはGW（ゲートウェイ）ノードとコンピュートホストのネットワークエッジのソフトウェアで実行します。

MidoNetエージェントに加え、ステイト管理システムによって分散されたエージェントそれぞれとも連係させます。一元管理のために、MidoNetではレストフルAPIサーバーも提供されます。

図2.1 MidoNetのトポロジの例



内部と上位のネットワーク

MidoNetは、すべてのサーバー間のIP到達能力に依存しています。

そのため、グローバルIPアドレスを使用しないIPネットワークをお勧めします（http://tools.ietf.org/html/rfc1918[RFC 1918]参照）。ラック1つまたはラックなしの小規模な導入では単一障害点になってしまいますが、1つのイーサネットスイッチを内部のネットワークに使用できます。大規模な導入では階層型IPルーティングネットワークが適切です。可能であればECMP（Equal Cost Multi-Path）を使用してください。

下層ネットワーク

下層ネットワークは、MidoNetソフトウェアをホストする物理ネットワークです。

GREおよびVXLANトンネル

MidoNetでは、下層の物理ホスト間の通信にトンネルを使用します。 MidoNetは次の2つのトンネルプロトコルをサポートしています。

- GRE（General Routing Encapsulation）プロトコル（MidoNetのデフォルトのトンネルプロトコル）。GREのラッパーサイズは46バイトに固定されています。
- VxLAN（Virtual Extensible LAN）プロトコル。 VXLANでは50バイトのオーバーヘッドが追加されます。

フラグメンテーションとリアセンブルを回避するため、適切なMTU（最大転送単位）を設定してこのオーバーヘッドを許可します。

下層ネットワークに対するMTUサイズの注意事項

オーバーヘッドを許可するために、ネットワークのアップリンクと仮想マシンはMTUのデフォルトサイズ（1500）に、GREトンネルを使用する物理ネットワークデバイスのMTUは1546にします。VxLANトラフィックが機能するためには、物理ネットワークデバイスのMTUのサイズを1550にする必要があります。

重要

仮想マシンのMTUは、ボーダーゲートウェイのアップリンクインターフェイスのMTUを上回らないようにします。

上層ネットワークに対するMTUのサイズの注意事項

最適なデータリンクの設定は個々の環境によって異なります。MidoNetはイーサネットのジャンボフレームをサポートしています。ジャンボフレームのサポートを設定する場合、MidoNetネットワークのネットワークインターフェイスのMTUは、下層（物理）ネットワークのMTUのサイズを少なくとも46バイトまたは50バイト下回る必要があります（GREとVXLANをそれぞれカプセル化するため）。仮想ネットワークのトラフィックパスで発生するMTUの不一致に注意してください。このような不一致によって、IPのフラグメンテーション/デフラグメンテーションが起こり、ネットワークパフォーマンスに悪影響を及ぼす可能性があります。

下層ネットワークで1500バイト以上のイーサネットフレームがサポートされない場合、MidoNetネットワークのMTUを1454または1450バイトに設定します（GREとXVLANをそれぞれカプセル化するため）。この設定によって、仮想マシンのネットワークインターフェイスに適切なMTUのサイズを設定することができます。

L3ゲートウェイアップリンクのNICのオフロード

NICへのオフロードは仲介ホストではなく、エンドホストに対して実行されます。ゲートウェイアップリンクのNICはルーターのNICと同様に扱う必要があります。

L3ゲートウェイアップリンクのNICでLROが有効な場合、NICは受信したTCPパケットと結合して、宛先のMTUより大きいパケットをMidoNetに渡します。MidoNetではサイズの大きいセグメントはオフロード（LSO-転送前にサイズの大きいTCPを分割）されず、IPフラグメンテーションもサポートされないため、パケットはドロップします。そのため、L3ゲートウェイアップリンクのNICへのオフロードを無効にする必要があります。アップリンクのNICで次を実行してください。

```
# ethtool -K p2p1 lro off
```

または、下記をネットワークスクリプトファイルに追加してください。

```
ETHTOOL_OPTS="lro off"
```

BGPの設定とレイヤー3のトポロジ

このセクションでは、BGPの設定とレイヤー3のトポロジに関する図と情報を提供します。

図2.2「レイヤー3のトポロジ（物理下層ネットワーク）」 [6] は、基盤となるネットワークインフラストラクチャの例です。

図2.3「レイヤー3のトポロジ（仮想上層ネットワーク）」 [7] は、MidoNet仮想化ネットワークを基盤となるネットワークアーキテクチャの上位に配置した例と、BGPルートアドバタイズメントの情報です。

図2.2 レイヤー3のトポロジ（物理下層ネットワーク）

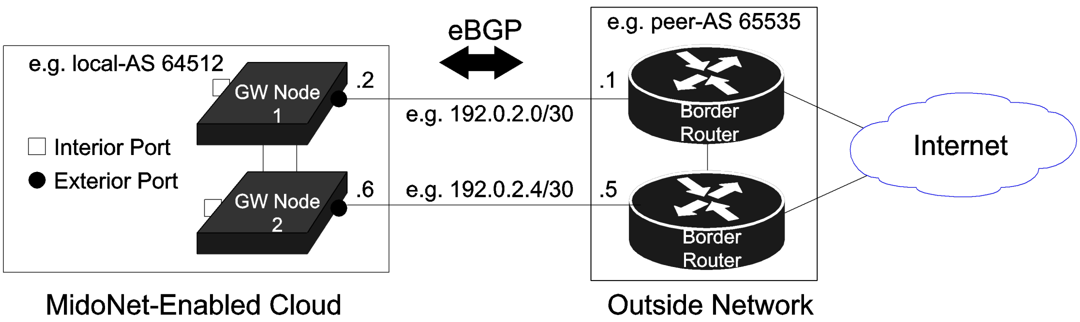
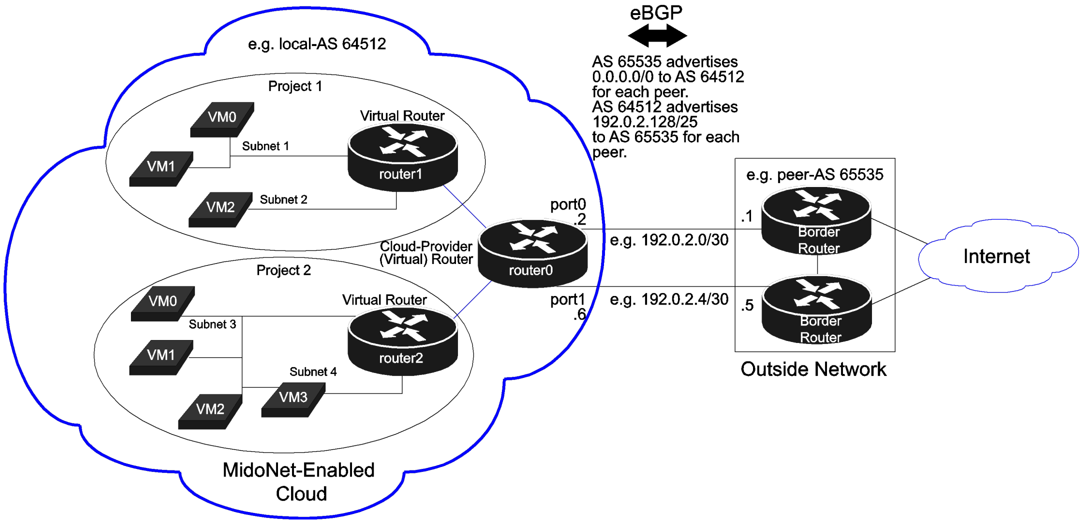


図2.3 レイヤー3のトポロジ（仮想上層ネットワーク）



仮想ルーター

仮想ルーターは物理レイヤー3（L3）ルーターの抽象化で、MidoNetのL3転送エレメントです。

物理ルーターと同様に、仮想ルーターでもネットワークインターフェイス（ポート）、ネットワークアップリンク（上流デバイス、通常、物理ルーターとブリッジに接続する物理イーサネットインタフェイスに接続するポート）を設定できます。また、仮想ルーターは他のルーターとブリッジに接続できます。

プロバイダルータ

プロバイダルータはMidoNetの内部のルータで、クラウドのゲートウェイルータとして機能します。

外部ネットワークがNeutronで作成されると、プロバイダルータも作成されます。 外部ネットワークが複数あっても、プロバイダルータは常にシステム内に1つしかありません。

プロバイダルータはOpenStackで外部ネットワークを作成すると、自動的に作成されます。

プロバイダルータは単一の仮想エンティティであってもその状態はMidoNetに配信され、外部ネットワークに接続を提供する複数のゲートウェイノードがある場合は単一障害点にはなりません。

実際の導入では、MidoNetプロバイダルータに接続するポートの大半はアップリンクポートで、通常は複数のアップリンクポートがあります （上記のネットワークの例では、ネットワークはアップリンクポートに接続していません）。 MidoNetの外部に接続する他の仮想ポートと同様に、各ポートはデバイス（イーサネットインターフェイス）とホストにバインドされています。 ホストは、サービスプロバイダと通信する上流のルータに物理的に接続するゲートウェイノードです。

外部ネットワークの作成の詳細については、次のリンクを参照してください：
http://docs.openstack.org/havana/install-guide/install/apt/content/install-neutron.configure-networks.html

コンピュートホストエージェント

MidoNetは、VMを実行する各コンピュートホストにエージェントを1つインストールする必要があります。

コンピュートホスト上のMidoNetエージェントは、クラウドの横方向のトラフィックの大半と上位の送信トラフィックを処理します。現在はKMVハイパーバイザをサポートしています。

ブリッジ

ブリッジはMidoNetのL2転送エレメントです。

ブリッジには仮想ポートを作成できます。 また、ブリッジの仮想ポートにVMを接続できます。 同じポートに接続しているすべてのVMにはL2の接続を通してアクセスできます。 仮想ブリッジのポートを他の仮想デバイスや物理マシンに接続することはできません。

ブリッジには、受信したフレームを送信するネットワークデバイスのMACアドレスとそれを受信するブリッジポート間のマッピングが保存されます。

ブリッジによって、ソースのMACアドレスとブリッジポートのテーブルが自動的に作成されます。 ブリッジではこのテーブルを使用して、フレームがネットワークデバイスから適切なブリッジポートに送信されます。 MACテーブルは消去できます。

メタデータサーバー

メタデータサーバーはインスタンスVMの構成情報、たとえば、認証情報やVMのカスタマイズスクリプトなどを保存するのに使用されます。

メタデータサーバーにはVMの構成の値のテーブルが保存されます。

第3章 ソリューションコンポーネント

目次

MidoNetソリューションは次のコンポーネントで構成されています。

- MidoNet Network State Databaseノード
- MidoNetゲートウェイノード
- MidoNet APIで実行しているサーバー、MidoNetエージェント（Midolman）およびコマンド ラインインターフェイス（CLI）
- NovaおよびOpenStackネットワーキングサービスをホストするOpenStackコントローラノード
- OpenStackコンピュートノード

状態管理

MidoNetネットワークステイトデータベースは、MidoNetの構成、ランタイムの状態および統計データを保存するサーバークラスタです。

MidoNetでは構成の状態に関する情報は2つの異なるシステムに保存されます。また、Apache# Zookeeper#とCassandra#を使用してMidoNetエージェント間を連係させ、ネットワークの構成と状態（ネットワークステイトデータベースノード）を保存します。

このクラスタはMidoNetネットワークステイトデータベースを実行する専用サーバーで構成するため、これらのサーバーで他のソフトウェアをホストしないでください。3台の専用サーバーの使用をお勧めします。

ZooKeeper

MidoNetではApache ZooKeeperを使用して、仮想および物理ネットワークトポロジに関する重要なパスデータを保存します。

保存されるデータには 仮想マシン（VM）、ブリッジ、ルーター間の相互接続、アドレス解決プロトコル（ARP）とNDテーブル、ホストUUID（Universally Unique Identifier）およびインターネットプロトコル（IP）のアドレス登録が含まれます。MidoNetエージェントとMidoNet APIサーバーはZooKeeperのスキーマを管理します。ZooKeeperに保存される情報の性質から、このスキーマは速度よりもクラスタ内のデータの整合性と一貫性を重視して最適化します。

必要なソフトウェア

MidoNetにはZooKeeperバージョン3.4.5が必要です（MidoNetレポジトリで提供されます）。

また、ZooKeeperはApacheソフトウェア財団からも入手できます。

ZooKeeperソフトウェアに加え、Java® Runtime Environmentも必要です。 多くのLinuxディストリビューションで提供されるOpenJDK# 7をお勧めします。

耐障害性の構成ガイドライン

また、ZooKeeperのインスタンスは3つ以上実行することを強くお勧めします。

規模が極端に小さいテストおよび開発環境（ゲートウェイノードを含め、MidoNetエージェントが3つ以下）では、実行できるZookeeperのインスタンスは1つです。 すべての導入製品に対して、3つのインスタンスの使用を強くお勧めします。

大規模な環境では（ゲートウェイノードを含め、数十のMidoNetエージェントを実行）、ワークロードに最適化された別々のノードにCassandraとZooKeeperをインストールすることをお勧めします。また、負荷を削減し、クラスタの残りの部分により多くのリソースを提供するために、ZooKeeperとCassandraのインスタンスは5つか7つ実行することをお勧めします。

ZooKeeperとCassandraのノードはインスタンスを奇数（3、7、9など）にしてください。この定足数はノードに障害が発生したときに確保されます。 クラスタが耐えられるノードの障害数は、3ノードのクラスタは1つ、5ノードのクラスタは2つ、7ノードのクラスタは3つと、ノード数に合わせて増加します。

ZooKeeperクラスタの管理を簡略化するため、ZooKeeperにはExhibitor Supervisor Systemを使用することをお勧めします。

アクセスに関する注意事項

ZooKeeperクラスタは通常、 TCP/2181、TCP/2888、TCP/3888の3つのポートを使用します。

また、Exhibitor Supervisorは通常、TCP/8080でWebインターフェイスを実行します。

ZooKeeperクラスタはプロキシを使わず、次のMidoNetコンポーネントに直接アクセスする必要があります。

- MidoNet APIサーバー
- MidoNetエージェント（MidoNetゲートウェイノードを含む）
- 他のZooKeeperインスタンス

MidoNetコントロールメッセージのデータパスとは異なるネットワークを使用することをお勧めします。たとえば、MidoNet API、MidoNet Network State DatabaseおよびMidoNetエージェントのノード間の接続には管理ネットワークを使用します。データパスネットワーク上のMidoNetエージェント間にはポイントツーポイントのトンネルを確立する必要があります。

Exhibitorを使用している場合、Webインターフェイスがシステムオペレータにアクセスできるか確認してください。

Cassandra

MidoNetではApache Cassandraバージョン2.0を使用して、NATバインディングや接続トラッキング情報などのフローステート情報が保存され、VMの移行がサポートされます。

MidoNetではCassandraの耐久性、耐障害性、有効期限、低遅延の読み込み/書き込みなどの利点を活用していますが、Cassandraは一次データソースではなくバックアップとしてのみ使用します。

必要なソフトウェア

CassandraにはJavaランタイム環境（JRE）が必要です。

Linuxディストリビューションの多くで提供され、公式インストールガイドを使用してインストールできることから、OpenJDK 7をお勧めしています（http://openjdk.java.net/でインストールに関する情報をご覧ください）。 == Fault-tolerant configuration guidelines

推奨されるCassandraの最小構成は、3ノードのクラスタと3のレプリケーションファクタ（N）です。

MidoNetエージェント（ Midolman）では、QUORUM（N/2 + 1）を整合性ポリシーとして使用していますが、これは提案した構成では2になります。

アクセスに関する注意事項

Cassandraは2つのIPアドレスを使用します。1つはクラスタ内の通信（listen_addressパラメータ）、もう1つはリモートプロシージャコール（RPC）によるクライアントの接続です（rpc_address）。

第4章 MidoNetゲートウェイノード

目次

OpenStackとMidoNetに必要なハードウェアに加え、ネットワークエッジで実行し、OpenStackを外部ネットワークに接続するサーバーが1台以上必要です。このサーバーは「ゲートウェイノード」と呼ばれます。

ゲートウェイノードはコンピュートノードよりも多くのトラフィックを処理するため（外部ネットワークの送受信）、ゲートウェイノードのホストにはより多くのメモリを割り当てる必要があります。

ゲートウェイノードの設定条件

MidoNetは、物理接続を通して上流のルーターに直接接続します。

GWノードには少なくとも2つの物理ネットワークインターフェイスが必要とされ、1つは内部ネットワーク、もう1つは上流のルーターに接続します。使用方法と環境によって、GWノードは大量の送受信トラフィックを処理する場合があります。そのため、マルチキューーNICとMSI-Xをサポートすることを強くお勧めします（1GbpsのIntel# 82576イーサネットコントローラ、10GbpsのIntel 82599イーサネットコントローラなど）。このようなNICを使用すると、複数のキューで受信イベントを処理し、複数のコアでIRQイベントを処理できます。

ゲートウェイノードの接続

MidoNetのクラウドには、インターネットでVMにアクセスするクライアントが使用するため、グローバルIPの範囲を割り当てる必要があります。

MidoNetはスタティックルーティングを使用できますが、一般的にはフェイルオーバを迅速化するため、BGP動的ルーティングをお勧めしています。すべてのGWノードは同じIP範囲をBGP経由でアドバタイズするため、上流ネットワークはその受信フローのバランスを調整する必要があります。一般的には、プライベートの自律システム（AS）をMidoNetに割り当てて、各エッジに対してBGPセッションを設定します。

第5章 Midolman

目次

Midolman（MidoNetエージェント）は、トラフィックがMidoNetを通過するすべてのホスト上で実行するデーモンです。

Open vSwitchカーネルモジュールにネットワークトラフィックの処理方法（パケットに適用する修正およびそのトンネルする先）を指示します。

MidolmanはOpen vSwitchカーネルモジュールをインストールしたLinuxカーネル、およびユーザー空間にJava 7ランタイムが必要です（OpenJDK 7推奨）。

MidolmanはOpen vSwitchカーネルモジュールバージョン1.10.2 以降に対応しています。 カーネルモジュールをアップデートする必要がある場合、このモジュールの新しいバージョンは使用中のディストリビューションのクラウドソフトウェアレポジトリにあります。

推奨するインストールノード

Midolmanをネットワーク仮想マシンとして使用している場合、一般的にはハイパーバイザとともにホストマシンにインストールする必要があります。

BGP（ボーダーゲートウェイプロトコル）で使用する場合、BGPピアのホップがほとんどない状態（理想的には1）でノードにインストールする必要があります。 MidoNetに縦方向のトラフィックがある場合、Midolmanはトラフィックを処理するアップリンクに対して帯域幅とプロキシが十分なマシンにインストールする必要があります。

構成ガイドライン

Midolmanでは、Midolmanの設定ファイルでZooKeeperおよびCassandraサーバーのIPアドレスを指定する必要があります。

Midolmanを設定して、ZooKeeperのセッションタイムアウトとセッション猶予時間の値を減少させるとノードの障害検出が加速されます。 また、それによって、システムが回復可能な一時的な停止がノードの障害として処理されないため、その後の時間も短縮されます。 これらのタイムアウトの値を増やすと、逆の結果になります。session_gracetimeの設定値以外、デフォルトのタイムアウト値の変更はお勧めしません。

アクセスに関する注意事項

同じトンネルゾーンをホストするMidoNetエージェントは相互にIP接続する必要があります。

また、ZooKeeperとCassandraのサーバーへのTCP接続も必要です（ZooKeeperのデフォルトのTCPポート番号は2181、Cassandraは9042です）。

MidoNetエージェントではDNS (Domain Name System) を使用して、ホスト名と下層ネットワークアドレスが変換されます。 MidoNetエージェントをインストールした各サーバーのホスト名が解決可能であることを確認してください。

第6章 MidoNet API

目次

MidoNet APIはHTTP（Hypertext Transfer Protocol）で実行するレスとフルAPIです。外部のアプリケーション（クラウドコントローラを含む）と内部のMidoNetの構成の間に統合ポイントを提供します。

REST APIはステートレスなため、RESTサーバーを追加するだけでそのサービスを拡張できます。 REST APIには状態管理サーバー（Apache ZookeeperおよびCassandra）への接続が必要です。 REST APIで認証を使用する場合、OpenStack Keystoneのアイデンティティサービスを利用できます。

推奨されるインストールノード

MidoNet APIサーバーはサーブレットコンテナでホストできます。

たとえば、Apache Tomcatは比較的軽量なため、MidoNet APIサーバーをZooKeeperインスタンスまたはNeutronサーバーインスタンスとともにインストールできます。

耐障害性の構成ガイドライン

耐障害性ソリューションを提供するために、MidoNet APIの複数のインスタンスを異なるノードで実行してから、外部のロードバランサーで共通の仮想IP（VIP）アドレスを使用してAPIコールをインスタンス間に配信することをお勧めします。

特別なロードバランス機能は不要なため、任意のロードバランサーを使用できます。

アクセスに関する注意事項

Apache Tomcatサーバーはデフォルトでポート TCP/8080を使用します。

MidoNetは ZooKeeperクラスタのすべてのノードにアクセスする必要があります。ZooKeeperクラスタは通常、TCP/2181、TCP/2888、TCP/3888の3つのポートを使用します。

第7章 MidoNetコマンドラインインターフェイス

MidoNet CLIはMidoNetの仮想トポロジを検査および編集するためのコマンドラインインターフェイスです。