



ш.  $\forall$ 

 $\alpha$ 

ш.  $\forall$ 

 $\alpha$ 

1

 $\forall$  $\alpha$ 

ш.  $\forall$  $\simeq$ 

ш

 $\forall$  $\simeq$ 

 $\forall$  $\alpha$ 

1

ш.  $\triangleleft$  $\alpha$ 

1

ш.  $\triangleleft$  $\alpha$ 

ш  $\triangleleft$  $\alpha$ 

Τ

< $\alpha$ 

T

2015.06-SNAPSHOT (2015-10-19 10:08 UTC) 製作著作 © 2015 Midokura SARL All rights reserved.

#### 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。

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

DRAFT

1

DRAFT

I

DRAFT

1

DRAFT

1

DRAFT

1

DRAF

1

DRAFT

1

DRAFT

1

DRAFT

1

DRAFT

1

# 目次

は	じめに	νi
	表記規則	٧i
1.	MidoNetの概要	1
	MidoNetの主な特長	
	推奨されるハードウェア	2
	インストールの要件	2
	OpenStackの統合	3
2.	MidoNetのネットワークアーキテクチャ	. 4
	内部と上位のネットワーク	
	下層ネットワーク	
	BGPの設定とレイヤー3のトポロジ	6
	仮想ルーター	7
	プロバイダルータ	7
	コンピュートホストエージェント	8
	ブリッジ	8
	メタデータサーバー	8
3.	ソリューションコンポーネント	9
	状態管理	9
4.	MidoNetゲートウェイノード	12
	ゲートウェイノードの設定条件	12
	ゲートウェイノードの接続	12
5.	Midolman	13
	推奨するインストールノード	13
	構成ガイドライン	13
	アクセスに関する注意事項	13
6.	MidoNet API	15
	推奨されるインストールノード	15
	耐障害性の構成ガイドライン	15
	アクセスに関する注意事項	15
7.	MidoNetコマンドラインインターフェイス	16

DRAFT

I

DRAFT

1

DRAF

1

DRAFT

1

 $\vdash$ 

DRAF

1

DRAFT

1

DRAFT

1

DRAFT

1

DRAFT

1

# 図の一覧

2.1.	MidoNetのトポロジの例	4
2.2.	レイヤー3のトポロジ(物理下層ネットワーク)	6
2 3	レイヤー3のトポロジ(仮想上層ネットワーク)	7

DRAFT

-

DRAFT

1

DRAFT

1

DRAF

1

DRAFT

1

 $\vdash$ 

DRAF

1

DRAFT

1

DRAFT

1

DRAFT

1

DRAFT

ı

クチャ

# 表の一覧

1.1. 推奨される導入ハードウェア ...... 2

느  $\triangleleft$  $\alpha$ 

1

<u>.</u>  $\triangleleft$  $\alpha$ 

1

 $\forall$ DR/

Τ

 $\forall$  $\alpha$ 

Τ

# 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。

ш.  $\forall$  $\alpha$ 

 $\forall$  $\alpha$ 

 $\forall$ 

 $\alpha$ 

 $\forall$ 

 $\alpha$ 

 $\forall$ 

 $\alpha$ 

 $\triangleleft$  $\alpha$ 

ш.

 $\triangleleft$ 

ш.

<0

ш

 $\triangleleft$  $\alpha$ 

1

ш <0

T

# 第1章 MidoNetの概要

## 目次

MidoNetの主な特長	1
推奨されるハードウェア	
インストールの要件	2
OpenStackの統合	3

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

Τ

RAF

1

RA

RA

RA

RA

RA

1

AF

 $\alpha$ 

AF

 $\alpha$ 

1

ш

2

Τ

RA

Τ

# 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)

ш.  $\forall$ 

 $\alpha$ 

1

ш.  $\forall$ 

 $\alpha$ 

 $\forall$ 

 $\alpha$ 

1

 $\forall$  $\alpha$ 

 $\forall$ 

 $\simeq$ 

 $\forall$  $\alpha$ 

ш.  $\triangleleft$  $\alpha$ 

1

ш.  $\triangleleft$  $\alpha$ 

ш  $\triangleleft$  $\alpha$ 

Τ

. L < $\alpha$ 

Τ

- 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、novascheduler、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

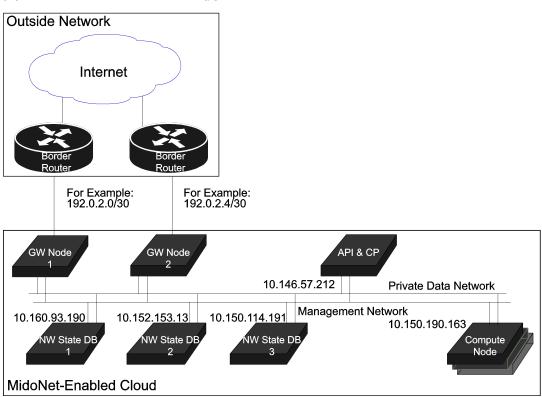
### 目次

内部と上位のネットワーク	4
下層ネットワーク	5
BGPの設定とレイヤー3のトポロジ	6
仮想ルーター	7
プロバイダルータ	
コンピュートホストエージェント	
ブリッジ	8
メタデータサーバー	8

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。

ш.

 $\forall$ 

 $\alpha$ 

ш.

DRA

RA

1

RA

RAF

О -

ш.

 $\forall$ 

 $\simeq$ 

ш.

RA

1

ш.

<

1

ш

RA

Τ

RAF

L

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。

I

RAF

RA

 $\forall$ 

DR

RA

RA

RA

 $\triangleleft$ 

 $\alpha$ 

1

RAF

1

 $\alpha$ 

Τ

RA

ī

### 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"

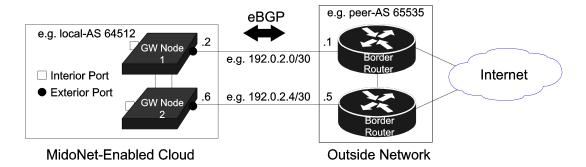
## BGPの設定とレイヤー3のトポロジ

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)



ш.  $\forall$  $\alpha$ 

1

ш.

 $\forall$  $\alpha$ 

I

 $\forall$  $\alpha$ 

 $\forall$  $\alpha$ 

ш.

 $\forall$ 

 $\simeq$ 

 $\forall$ 

 $\alpha$ 

ш.  $\triangleleft$ 

ш.

< $\alpha$ 

ш

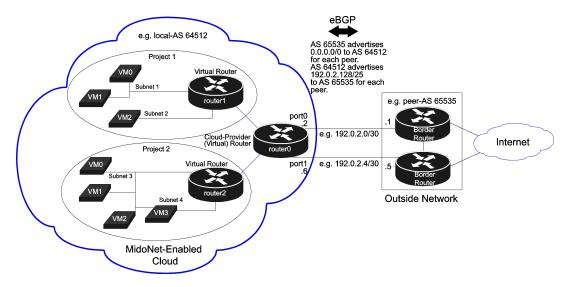
 $\triangleleft$  $\alpha$ 

L

ш < $\alpha$ 

ī

#### 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/installneutron, configure-networks, html

ш.  $\forall$  $\alpha$ 

ш.  $\forall$ 

 $\alpha$ 

 $\forall$ 

 $\alpha$ 

1

 $\forall$ 

 $\Delta$ 

ш.  $\triangleleft$ 

 $\simeq$ 

 $\forall$  $\simeq$ 

ш.  $\triangleleft$  $\alpha$ 

1

ш.  $\triangleleft$  $\alpha$ 

ш  $\triangleleft$  $\alpha$ 

Τ

بيا < $\alpha$ 

Τ

# 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。

ш.  $\forall$  $\alpha$ 1

ш.  $\forall$  $\alpha$ 

 $\forall$  $\alpha$ 

 $\forall$  $\Delta$ 

 $\forall$  $\Delta$ 

ш.

 $\forall$  $\alpha$ 

ш.

 $\triangleleft$  $\alpha$ 

ш.

 $\triangleleft$  $\alpha$ 

ш

 $\triangleleft$  $\alpha$ 

ш < $\alpha$ 

Τ

# 第3章 ソリューションコンポーネント

### 目次

状態管理	C
ソノハハトローニエ	 ·

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。

ш.

 $\forall$  $\Delta$ 

ш.  $\forall$ 

 $\alpha$ 

 $\forall$ 

 $\alpha$ 

 $\forall$ 

 $\alpha$ 

ш.  $\forall$  $\simeq$ 

ш.  $\forall$  $\simeq$ 

ш.

 $\triangleleft$  $\alpha$ 

1

عبا

 $\triangleleft$ 

 $\alpha$ 

ш.  $\triangleleft$ 

2 

ш < $\alpha$ 

T

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。 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

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バーション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。

 $\forall$  $\simeq$ 

 $\vdash$  $\triangleleft$  $\alpha$ 

1

عبا  $\triangleleft$  $\alpha$ 

ш  $\triangleleft$  $\alpha$ 

I

. L < $\alpha$ 

L

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)。 == Faulttolerant 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)。

ш. Ø  $\alpha$ 

 $\forall$  $\alpha$ 

 $\forall$ 

 $\alpha$ 

 $\forall$  $\alpha$ 

 $\forall$ 

 $\alpha$ 

ш.  $\forall$  $\alpha$ 

ш.

 $\triangleleft$ 

 $\alpha$ 

1

ш. < $\alpha$ 

1

ш  $\triangleleft$  $\alpha$ 

Τ

ш < $\alpha$ 

Τ

# 第4章 MidoNetゲートウェイノード

### 目次

ゲートウェイノードの設定条件	 12
ゲートウェイノードの接続	12

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。

ш.  $\triangleleft$  $\alpha$ 

 $\forall$  $\alpha$ 

 $\forall$ 

 $\alpha$ 

 $\forall$ 

 $\Delta$ 

 $\forall$  $\alpha$ 

 $\forall$ 

 $\alpha$ 

ш.  $\triangleleft$ 

 $\alpha$ 

ш.  $\triangleleft$ 

ш  $\triangleleft$  $\alpha$ 

1

< $\alpha$ 

T

# 第5章 Midolman

### 目次

推奨するインストールノード	13
構成ガイドライン	13
アクセスに関する注音車項	13

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ション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 りにあります。

## 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)。

ш  $\forall$  $\simeq$  $\Box$ 1 RAI 1  $\triangleleft$ 2 1 RAI 1 ш.  $\triangleleft$ 2 1  $\vdash$ RA 1  $\vdash$  $\triangleleft$  $\alpha$ 1  $\vdash$ RA 1 F DRA 1 FT  $\forall$  $\alpha$ 

I

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。

ш.  $\forall$  $\alpha$ 

 $\forall$  $\alpha$ 

 $\forall$  $\alpha$ 

 $\triangleleft$ 

 $\simeq$ 

 $\forall$  $\simeq$ 

 $\forall$ 

 $\alpha$ 

ш.  $\triangleleft$  $\alpha$ 

< $\alpha$ 

Τ

< $\alpha$ 

T

# 第6章 MidoNet API

### 目次

推奨されるインストールノード	15
耐障害性の構成ガイドライン	15
アクセスに関する注音車項	15

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。

RAI

RAI

1

 $\triangleleft$ 2

1

RAI

1

 $\triangleleft$  $\simeq$ 

1

 $\triangleleft$  $\simeq$ 

I

2

1

RA

1

DRA

I

 $\forall$  $\simeq$ 

Ι

# 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。