

MidoNet トラブルシューティング ガイド

2015.06-SNAPSHOT (2015-10-19 10:08 UTC)

DRAFT



MidoNet トラブルシューティング ガイド

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。

このドキュメントは、MidoNetやOpenStackのトラブルシューティングに関する有用な情報が含まれています。



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.

目次

はじめに	iv
表記規則	iv
1. 全体アプローチ	1
アンダーレイネットワーク	1
オーバーレイネットワーク	3
トポロジージュミレーション	5
仮想トポロジ	5
2. 共通トピック	6
MidoNet エージェント	6
MidoNet API	6
ボーダーゲートウェイプロトコル (BGP)	6
ZooKeeper	8
VM の相互接続	8
3. ツールとコマンド	10
midonet-cli	10
mm-dpctl	10
mm-trace	10
ip	11
4. ディレクトリーとファイル	12
Cassandra	12
MidoNet Agent	12
MidoNet API	12
Quagga (BGPD)	12
ZooKeeper	12
5. プロセス	14

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 全体アプローチ

目次

アンダーレイネットワーク	1
オーバーレイネットワーク	3
トポロジータリシユミレーシヨソ	5
仮想トポロジータ	5

MidoNetの環境でトラブルシューティングをする時は、複数のレイヤーをチェックする必要があります。

- アンダーレイネットワーク
- オーバーレイネットワーク
- 仮想ネットワークトポロジースイミレーション
- 仮想ネットワークトポロジ

レイヤー	コンポーネント
仮想ネットワークポロジ	Neutron, MidoNet NSDB
仮想ネットワークポロジシュミレーション	MidoNet エージェント
オーバーレイネットワ	トンネル、データパス
アンダーレイネットワー	物理環境, オペレーティングシステム

潜在的な問題を避けるために、これらのレイヤーはボトムからトップにかけてチェックしてください。

アンダーレイネットワーク

アンダーレイネットワーク、つまり物理ネットワークは、接続に関する問題があった時に一番最初に確認するポイントです。

- ハードウェア / ケーブル
- ルーティング
- ファイヤーウォール
- アクセスコントロール
- リナックスカーネル / オープン vSwitch
- 時間の同期化

ハードウェア/ ケーブル

ハードウェアが正常に機能しているか確認してください。

1. 物理リンクは確立されていますか？

```
# ethtool eth0
Settings for eth0:
```

```
# ip link
[...]
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT
group default qlen 1000
    link/ether aa:bb:cc:dd:ee:ff brd ff:ff:ff:ff:ff:ff
```

ルーティング

ルーティングが適切に設定されているか確認してください。そして、ping コマンドを使ってホスト間の接続性を確認してください。

```
# netstat -nr
Destination    Gateway        Genmask         Flags    MSS  Window  irtt  Iface
0.0.0.0        192.168.0.1   0.0.0.0         UG       0    0        0     eth0
192.168.0.0    0.0.0.0       255.255.255.0   U        0    0        0     eth0
```

```
# ip route
default via 192.168.0.1 dev eth0 proto static
192.168.0.0/24 dev eth0 proto kernel scope link src 192.168.96.100 metric 1
```

ファイヤーウォール

ファイアーウォールが必要なプロトコル、ホスト、ポートをブロックしていないか確認してください。

もし確信の無い場合は、ファイヤーウォールを解除して、接続に関する問題が残っていないか検証してください。

```
# iptables -L
```

Access Control

SELinux もしくは AppArmorなどのアクセスコントロールシステムが、必要な機能をブロックしていないかを確認してください。

もし確信が無い場合は、ACLシステムを解除して、まだ問題があるかどうかを検証してください。

Linuxカーネル / オープンvSwitch

オープンvSwitchカーネルモジュールがロードされていて、走っているカーネルのバージョンと合っているか確認してください。

```
# modinfo openvswitch
filename:      /lib/modules/kernel_version/kernel/net/openvswitch/openvswitch.ko
license:      GPL
description:   Open vSwitch switching datapath
depends:       libcrc32c,vxlan,gre
intree:       Y
```

```
# lsmod | grep openvswitch
openvswitch          70743  0
vxlan                37584  1 openvswitch
gre                 13808  1 openvswitch
libcrc32c           12644  2 xfs,openvswitch
```



```
Datapath name      : midonet
Datapath index    : 11
Datapath Stats:
  Flows :1340066
  Hits  :1111802509
  Lost  :0
  Misses:17302163
Port #0 "midonet" Internal Stats{rxPackets=0, txPackets=0, rxBytes=0, txBytes=0,
rxErrors=0, txErrors=0, rxDropped=0, txDropped=0}
Port #1 "tngre-overlay" Gre Stats{rxPackets=508157678, txPackets=398704120,
rxBytes=291245619484, txBytes=318474308439, rxErrors=0, txErrors=0, rxDropped=0,
txDropped=0}
Port #2 "tnvxlan-overlay" VXLAN Stats{rxPackets=0, txPackets=0, rxBytes=0, txBytes=0,
rxErrors=0, txErrors=0, rxDropped=0, txDropped=0}
Port #3 "tnvxlan-vtep" VXLAN Stats{rxPackets=0, txPackets=0, rxBytes=0, txBytes=0,
rxErrors=0, txErrors=0, rxDropped=0, txDropped=0}
Port #4 "tapa0164c42-dd" NetDev Stats{rxPackets=389426272, txPackets=342761506,
rxBytes=1128206548338, txBytes=241007949600, rxErrors=0, txErrors=0, rxDropped=0,
txDropped=0}
Port #5 "tap19ccc069-f1" NetDev Stats{rxPackets=0, txPackets=54640, rxBytes=0,
txBytes=2347034, rxErrors=0, txErrors=0, rxDropped=0, txDropped=0}
Port #6 "tape3055fc6-cc" NetDev Stats{rxPackets=21375, txPackets=42911, rxBytes=3573207,
txBytes=4607633, rxErrors=0, txErrors=0, rxDropped=0, txDropped=0}
```

```
# mm-dpctl --dump-dp midonet
1340149 flows
  Flow:
    match keys:
      Tunnel{tun_id=4360, ipv4_src=10.11.0.16, ipv4_dst=10.11.0.15, tun_flag=0,
      ipv4_tos=0, ipv4_ttl=-3}
      InPort{1}
      Ethernet{src=02:13:38:97:08:f3, dst=fa:16:3f:92:53:60}
      EtherType{0x800}
      KeyIPv4{src=8.8.8.8, dst=10.17.3.14, proto=17, tos=0, ttl=55, frag=0}
      UDP{src=53, dst=56975}
    actions:
      Output{port=21}
[...]
```

```
# mm-ctl --list-hosts
Host: id=17ef018f-de8b-431b-89f0-b5472f176769
  name=hostname
  isAlive=true
  addresses:
  vport-host-if-bindings:
    VirtualPortMapping{virtualPortId=ac0c2557-9fa0-4009-9e18-dc62ea65052a,
localDeviceName=' tapac0c2557-9f'}
    VirtualPortMapping{virtualPortId=c37d8bf2-d008-464e-a688-0627f2da342f,
localDeviceName=' f58b0880_MN_dp'}
    VirtualPortMapping{virtualPortId=7aa08012-d06c-4c78-ae8-1fff7c063fed,
localDeviceName=' tap7aa08012-d0'}
    VirtualPortMapping{virtualPortId=5aa6a752-57f2-4749-b160-9e632e0a16bb,
localDeviceName=' f58b0880_MN_dp'}
[...]
```

MTU

アンダーレイネットワークのフラグメンテーションを避けるために、VMインスタンスのMTUは、トンネルプロトコルのオーバーヘッドに対応する必要があります。

この調整されたMTUは、DHCPを通じて、MidoNetに酔って自動的にアドバタイズされます。しかし、これは、VMで使われているオペレーティングシステムによっては、適応されない場合があります。

BGPネームスペースを見つけてください。

```
# ip netns list
mbgpX_ns
```

BGPネームスペースのインターフェースをみる。

```
# ip netns exec <mbgpX_ns> ip link show
```

BGPピアがエンドにある場合でもリーチできることを確認ください。

```
# ip netns exec <mbgpX_ns> ip neigh show
```

BGPピアリングにおけるスニッフ

```
# ip netns exec <mbgpX_ns> tcpdump -i <ns_itf>
```

Quagga' s VTYシェルを入力ください。

```
# ip netns exec <mbgpX ns> vtysh
```

QuaggaのVTYシェル中の既存の設定をみます。

```
# show run
Building configuration...

Current configuration:
!
hostname bgpd
log file /var/log/quagga/bgpd.2609.log
!
password zebra_password
!
router bgp 65535
  bgp router-id 192.168.107.29
  network 42.159.202.0/24
  neighbor 192.168.107.30 remote-as 65534
  neighbor 192.168.107.30 timers 5 15
  neighbor 192.168.107.30 timers connect 10
!
line vty
!
end
```

QuaggaのVTYシェル中のBGPサマリーをみます。

```
# show ip bgp summary
BGP router identifier 192.168.107.29, local AS number 65535
RIB entries 19, using 2128 bytes of memory
Peers 1, using 4560 bytes of memory
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
192.168.107.30	4	65534	5512	5523	0	0	0	07:40:09	10

Quagga' s VTYシェル中のBGPルーティング情報を見ます。

```
# show ip bgp
BGP table version is 0, local router ID is 192.168.107.29
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	0.0.0.0	192.168.107.30	0		65534	?
*>	1.1.1.0/30	192.168.107.30	0		65534	?

8

VM のポート UUID を特定するには、Horizon で VM のネットワークに移動し、ポートリストで VM の内部 IP を検索します。そこから、次の例のように TAP インターフェイス名を作成します。

ポート UUID: 7aa08012-d06c-4c78-ae8-1fff7c063fed

TAP インターフェイス: tap7aa08012-d0

インターフェースでトラフィックを検証する

コンピューティングホストの VM の TAP インターフェイスで tcpdump を使用すれば、ゲストホストにログインせずに VM の仮想 NIC でトラフィックが確認されているかどうかを検証できます。

```
# tcpdump -n -i tap7aa08012-d0
```

TAP インターフェイスでパケットカウンターを監視します。

```
# watch -d ip -s link show tap7aa08012-d0
```

第3章 ツールとコマンド

目次

midonet-cli	10
mm-dpctl	10
mm-trace	10
ip	11

このセクションは、便利なツールとコマンドに関する概要を提供します。

midonet-cli

midonet-cli コマンドは、MidoNet APIに接続し、python-midonet-client パッケージをインストールしているホスト上で走らせることができます。

mm-dpctl

mm-dpctl コマンドは、どのMidoNetエージェントでも走らせることができ、現在のフローなどデータパス情報を表示することができます。

利用可能なオプション:

```
$ mm-dpctl
usage: mm-dpctl
  --add-dp <arg>      新しいデータパスを追加
  --delete-dp <arg>   データパスを削除
  --dump-dp <arg>     与えられたデータパスにインストールされている全てのフローを表示
  --list-dps           インストールされているデータパスをリスト化
  --show-dp <arg>     与えられたのデータパスに関連する情報を表示
  --timeout <arg>     何秒でタイムアウトするかを特定。特定した時間内にプログラムが結果が帰ってこない場合は、停止してエラーコードを戻します。
```

例題:

```
$ mm-dpctl --show-dp midonet # shows datapath and interfaces
$ mm-dpctl --dump-dp midonet # shows current flows
```

mm-trace

a mm-trace コマンドは、MidoManエージェントがある特定のトラフィックフローをキャプチャして、シミュレーションの各ステージのログを取ることができます。

この設定は、MidoMan再起動の際に、常に同じではありません。

アウトプットは /var/log/midolman/mm-trace.log ファイルに書き込まれます。

利用可能なオプション:

```
$ mm-trace --help
-h, --host <arg>   Host (default = localhost)
-p, --port <arg>   JMX port (default = 7200)
--help             ヘルプメッセージを表示
```

Subcommand: add - トレーシングマッチのパケットを追加

```
-d, --debug          デバグレレベルでのログ
--dst-port <arg>    TCP/UDP 送信先ポートをマッチ
--ethertype <arg>   イーサータイプをマッチ
--ip-dst <arg>      送信先ipアドレスをマッチ
--ip-protocol <arg> ipプロトコルフィールドをマッチ
--ip-src <arg>      ipソースアドレスをマッチ
-l, --limit <arg>   このトレースを無効化する前のマッチするパケット数
--mac-dst <arg>     送信先MACアドレスをマッチ
--mac-src <arg>     送信元MACアドレスをマッチ
--src-port <arg>    TCP/UDPソースポートをマッチ
-t, --trace         トレースレベルでのログ
--help             ヘルプメッセージを表示
Subcommand: remove - トレーシングマッチのパケットを除去
-d, --debug          デバグレレベルでのログ
--dst-port <arg>    TCP/UDP 送信先ポートをマッチ
--ethertype <arg>   イーサータイプをマッチ
--ip-dst <arg>      送信先ipアドレスをマッチ
--ip-protocol <arg> ipプロトコルフィールドをマッチ
--ip-src <arg>      ipソースアドレスをマッチ
-l, --limit <arg>   このトレースを無効化する前のマッチするパケットの数
--mac-dst <arg>     送信先MACアドレスをマッチ
--mac-src <arg>     送信元MACアドレスをマッチ
--src-port <arg>    TCP/UDPソースポートをマッチ
-t, --trace         トレースレベルでのログ
--help             ヘルプメッセージを表示
Subcommand: flush - トレーシングがマッチしているリストをクリア
-D, --dead-only     期限切れのトレーサーのみをフラッシュ
--help             ヘルプメッセージを表示
Subcommand: list -  アクティブなトレーシングマッチをリスト化
-L, --live-only     アクティブなトレーサーのみをリスト化
--help             ヘルプメッセージを表示
```

Examples:

```
$ mm-trace list
$ mm-trace add --debug --ip-dst 192.0.2.1
$ mm-trace add --trace --ip-src 192.0.2.1 --dst-port 80
$ mm-trace list
tracer: --debug --ip-dst 192.0.2.1
tracer: --trace --ip-src 192.0.2.1 --dst-port 80
$ mm-trace remove --trace --ip-src 192.0.2.1 --dst-port 80
Removed 1 tracer(s)
$ mm-trace flush
Removed 1 tracer(s)
```

ip

ip コマンドは、ルーティング、デバイス、ポリシールーティング、トンネルを表示したり、操作したりするために使用します。

詳細情報に関するメインページはこちらです: <http://linux.die.net/man/8/ip>

インターフェースをリスト化します。

```
# ip link show
```

名前空間をリスト化します。

```
# ip netns list
```

名前空間の中で、インターフェースをリスト化します。

```
# ip netns exec namespace ip link show
```

第4章 ディレクトリーとファイル

目次

Cassandra	12
MidoNet Agent	12
MidoNet API	12
Quagga (BGPD)	12
ZooKeeper	12

このセクションは、よく使う設定とログファイルの概要を提供します。

ファイル名とパスは、使っているオペレーティングシステムとOpenStackのディストリビューションによって、微妙に異なりますので注意してください。

Cassandra

File	Type
/etc/cassandra/conf/cassandra.yaml	CONF
/var/log/cassandra/cassandra.log	LOG

MidoNet Agent

File	Type
/etc/midolman/midolman-akka.conf	CONF
/etc/midolman/midolman.conf	CONF
/etc/midolman/midolman-env.sh	CONF
/var/log/midolman/midolman.event.log	LOG
/var/log/midolman/midolman.log	LOG
/var/log/midolman/mm-trace.log	LOG
/var/log/midolman/upstart-stderr.log	LOG

MidoNet API

File	Type
/usr/share/midonet-api/WEB-INF/web.xml	CONF
/var/log/tomcat/catalina.out	LOG
/var/log/tomcat/midonet-api.log	LOG

Quagga (BGPD)

File	Type
/var/log/quagga/bgpd. log	LOG

ZooKeeper

File	Type
/etc/zookeeper/zoo.cfg	CONF

第5章 プロセス

このセクションは、共通プロセスを提供します。

使っているオペレーション・システムとOpenStackディストリビューションによって、名前とファイルパスが異なります。

プログラ	プロセス
Cassandra	java [...] org.apache.cassandra.service.CassandraDaemon
MidoNet Agent	java [...] org.midonet.midolman.Midolman
MidoNet Agent (Watchdog)	/usr/bin/python /usr/bin/wdog [...] org.midonet.midolman.Midolman
MidoNet API (Tomcat)	java [...] org.apache.catalina.startup.Bootstrap
ZooKeeper	java [...] org.apache.zookeeper.server.quorum.QuorumPeerMain