

MidoNet クイック スタート ガイド

RHEL 7 / Juno (OSP)

2015.06-SNAPSHOT (2015-11-26 07:36 UTC)

DRAFT



midonet

docs.midonet.org

目次

はじめに	iv
表記規則	iv
1. アーキテクチャ	1
ホストとサービス	2
2. 環境の基本構成	4
ネットワークの構成	4
SELinuxの構成	4
レポジトリの構成	4
3. OpenStackのインストール	6
アイデンティティサービス (Keystone)	6
コンピュートサービス (Nova)	6
ネットワーキングサービス (Neutron)	9
4. MidoNetのインストール	15
NSDBノード	15
コントローラノード	18
Midolman のインストール	20
MidoNetのホストの登録	21
5. ネットワークの初期設定	23
6. BGP アップリンク構成	24
7. 高度な手順	28

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。

1

T - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT

T - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT

T - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT

- T - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT

T - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT

- T - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT

T - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT

- T - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT

ゲートウェイノード (gateway1, gateway2)

- BGP Daemon (Quagga)
- MidoNet
 - エージェント (Midolman)

4

T - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT

T - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT

T - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT

T - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT

T - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT

T - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT

T - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT

T - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT - DRAFT

6


```
[neutron]
...
service_metadata_proxy = true
metadata_proxy_shared_secret = METADATA_SECRET
```



注記

[Metadata Proxy configuration](#)と同じ METADATA_SECRET を使用しま
す。

Compute APIサービスを再開します。

```
# systemctl restart openstack-nova-api
```

コンピュータノード



重要

Red Hatの文書の [8.3. Install a Compute Node](#) の指示に従ってください。
ただし、次の相違事項と追加事項に注意してください。

1. 8.3.1. Compute Service Databaseを作成する

適用*しないで*ください。 コントローラ ノードでは実行済みです。

2. 8.3.2. Compute Serviceの認証を構成する

適用*しないで*ください。 コントローラ ノードでは実行済みです。

3. 8.3.3. Compute Serviceパッケージをインストールする

そのまま適用*しないで*ください。

代わりに、次のパッケージのみインストールします。

```
# yum install openstack-nova-compute openstack-utils
```

4. 8.3.4. Compute ServiceのSSLの使用を構成する

このまま適用します。

5. 8.3.5. Compute Serviceを構成する

次のトピックを除き、そのまま適用します。

a. 8.3.5.6.3. L2エージェントを構成する

適用*しないで*ください。

b. 8.3.5.6.4. 仮想インターフェイスのプラグインを構成する

適用*しないで*ください。

6. 8.3.6. Compute Service Databaseを追加する

適用*しないで*ください。 コントローラ ノードでは実行済みです。

7. 8.3.7. コンピュータサービスを開始する

a. 1. Message Bus Serviceの開始

このまま適用します。

b. 2. Libvirt Serviceの開始

このまま適用します。

c. 3. API Serviceの開始

適用*しないで*ください。 コントローラノードでのみ必要です。

d. 4. スケジューラの開始

適用*しないで*ください。 コントローラノードでのみ必要です。

e. 5. コンダクタの開始

適用*しないで*ください。 コントローラノードでのみ必要です。

f. 6. コンピュータサービスの開始

このまま適用します。

8. 代わりに、次の手順を実行します

a. libvirtを構成する

/etc/libvirt/qemu.conf ファイルを修正して次を含めます。

```
user = "root"  
group = "root"  
  
cgroup_device_acl = [  
    "/dev/null", "/dev/full", "/dev/zero",  
    "/dev/random", "/dev/urandom",  
    "/dev/ptmx", "/dev/kvm", "/dev/kqemu",  
    "/dev/rtc", "/dev/hpet", "/dev/vfio/vfio",  
    "/dev/net/tun"  
]
```

b. libvirtdサービスを再開する

```
# systemctl restart libvirtd.service
```

c. nova-rootwrap ネットワークフィルタをインストールする

```
# yum install openstack-nova-network
# systemctl disable openstack-nova-network.service
```

d. Computeサービスを再開する

```
# systemctl restart openstack-nova-compute.service
```

ネットワークサービス (Neutron)



重要

Red Hatの文書の [Chapter 7. OpenStack Networking Service Installation](#) の指示に従ってください。ただし、次の相違点に注意してください。

このまま適用します。

コントローラノード



重要

Red Hatの文書の [7.4. Configure the Networking Service](#) の指示に従ってください。ただし、次の相違点に注意してください。

1. 7.4.1. OpenStack Networking Serviceの認証を構成する

このまま適用します。

2. 7.4.2. Networking ServiceのためにRabbitMQ Message Brokerを設定する

このまま適用します。

3. 7.4.3. OpenStack Networking Service Plug-inを設定する

適用*しないで*ください。代わりに、次の手順を実行します。

- a. /etc/neutron/neutron.conf ファイルを変更して、次のキーを [DEFAULT] セクションに追加します。

```
[DEFAULT]
...
core_plugin = midonet.neutron.plugin.MidonetPluginV2
allow_overlapping_ips = True
```



注記

構成ファイルの行の開始にスペースを残さないでください（これはすべての構成ファイルに適用されます）。

- b. MidoNetプラグインのディレクトリを作成します。

```
mkdir /etc/neutron/plugins/midonet
```

/etc/neutron/plugins/midonet/midonet.ini ファイルを作成し、修正して次を含めます。

+

```
[DATABASE]
sql_connection = mysql://neutron:NEUTRON_DBPASS@controller/neutron

[MIDONET]
# MidoNet API URL
midonet_uri = http://controller:8080/midonet-api
# MidoNet administrative user in Keystone
username = midonet
password = MIDONET_PASS
# MidoNet administrative user's tenant
project_id = services
```

- a. NeutronをMidoNetの構成に転送するためのシンボリックリンクを作成します。

```
# ln -s /etc/neutron/plugins/midonet/midonet.ini /etc/neutron/plugin.ini
```

4. 7.4.4. VXLANおよびGREトンネル


```
[DEFAULT]
interface_driver = neutron.agent.linux.interface.MidonetInterfaceDriver
dhcp_driver = midonet.neutron.agent.midonet_driver.DhcpNoOpDriver
use_namespaces = True
enable_isolated_metadata = True

[MIDONET]
# MidoNet API URL
midonet_uri = http://controller:8080/midonet-api
# MidoNet administrative user in Keystone
username = midonet
password = MIDONET_PASS
# MidoNet administrative user's tenant
project_id = services
```

3. Starting the DHCP Agent

Apply as is.

Metadata Agent



注記

Since MidoNet does not have the concept of a Network Node like with the default OpenStack networking plugin, the Metadata Agent is going to be installed on the Controller Node.



重要

Follow the Red Hat documentation's [7.8. Configure the L3 Agent](#) instructions, but note the following differences.

1. Configuring Authentication

Apply as is.

2. Configuring the Interface Driver

Do not apply.

3. Configuring External Network Access

Do not apply.

4. Starting the L3 Agent

Do not apply.

5. Starting the Metadata Agent

Apply as is.

6. Enable leastrouter scheduling

Do not apply.

1. Additional changes

Edit the `/etc/neutron/metadata agent.ini` file to contain the following:

```
[DEFAULT]  
[...]  
nova_metadata_ip = controller  
metadata_proxy_shared_secret = METADATA_SECRET
```



注記

Use the same METADATA_SECRET as in the [Nova configuration](#).

Restart the Metadata Agent:

```
# systemctl restart neutron-metadata-agent.service
```

T - DRAFT


```
mkdir -p /var/run/cassandra
chown cassandra:cassandra /var/run/cassandra
su $CASSANDRA_OWNr -c "$CASSANDRA_PROG -p $pid_file" > $log_file 2>&1
retval=$?
```

[...]

4. Cassandraを有効にして開始する

```
# systemctl enable cassandra.service
# systemctl start cassandra.service
```

5. Cassandraの動作を確認する

すべてのノードのインストールが完了したら、Cassandraが適切に動作するか確認します。



重要

Cassandraデーモンの実行が失敗する場合、ログで「buffer overflow」のエラーメッセージが出たら、`/etc/hosts` ファイル内の `127.0.0.1` アドレスからホスト名へのマッピング・エントリを設定してみて（`hostname -i` の戻り値が `127.0.0.1` になりますように）、実行エラーの解決が出来るかもしれません。

基本的な検査は、`nodetool status` コマンドを実行して行えます。サーバーがエラーのない状態で稼働している場合は、最初の列に UN (Up/Normal) と返されます。

```
$ nodedtool -host 127.0.0.1 status
[...]
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load          Tokens    Owns    Host ID                                     Rack
UN  192.0.2.1    123.45 KB     256      33.3%   11111111-2222-3333-4444-555555555555   rack1
UN  192.0.2.2    234.56 KB     256      33.3%   22222222-3333-4444-5555-666666666666   rack1
UN  192.0.2.3    345.67 KB     256      33.4%   33333333-4444-5555-6666-777777777777   rack1
```

コントローラノード

MidoNet APIのインストール

1. MidoNet APIパッケージをインストールする

```
# yum install midonet-api
```

2. MidoNet APIを構成する

/usr/share/midonet-api/WEB-INF/web.xml ファイルを変更して以下を含めます。

```
<context-param>
  <param-name>rest_api-base_uri</param-name>
  <param-value>http://controller:8080/midonet-api</param-value>
</context-param>
```

```
<context-param>
  <param-name>keystone-service_host</param-name>
  <param-value>controller</param-value>
</context-param>
```

<context-param>

DIT

第5章 ネットワークの初期設定



重要

OpenStack文書の [Create initial networks](#) 指示に従います。ただし、次の相違点に注意してください。

1. 外部ネットワークの作成

外部ネットワークは下記のコマンドで作成します。

```
$ neutron net-create ext-net --router:external
```



注記

OpenStackの外部ネットワークが作成したとき、MidoNetは自動的に Midonet Provider Router を作成します。これはMidoNet内部ルータであり、クラウドのゲートウェイルータとして機能します。複数の外部ネットワークが含まれている場合でも、常に1台のルータのみがあります。

第6章 BGP アップリンク構成

MidoNet では、外部接続にボーダーゲートウェイプロトコル (BGP) を利用します。

BGP にはスケーラビリティと冗長性があるため、実稼動環境では BGP を使用することを強くお勧めします。

デモ環境や POC 環境では、代わりに静的ルーティングを使用できます。詳しくは、[操作ガイド](#)を参照してください。

こちらの手順では、次のサンプル環境を想定しています。

- フローティング IP ネットワーク 1 個
 - 192.0.2.0/24
- MidoNet ゲートウェイノード 2 個
 - gateway1、bgp1 に eth1 で接続
 - gateway2、bgp2 に eth1 で接続
- リモート BGP ピア 2 個
 - bgp1、198.51.100.1、AS 64513
 - bgp2、203.0.113.1、AS 64513
- 対応する MidoNet BGP ピア
 - 198.51.100.2、AS 64512
 - 203.0.113.2、AS 64512

次の手順に従って、GBP アップリンクを構成してください。

1. Keystone admin テナント ID を特定する

keystone コマンドを使用して、Keystone admin テナント ID を特定します。

\$ keystone tenant-list		
id	name	enabled
12345678901234567890123456789012	admin	True

2. MidoNet CLI を起動し、MidoNet プロバイダルーターを検索する

```
$ midonet-cli
midonet-cli>
```

MidoNet プロバイダルーターはテナントと関連付けられていないため、最初にアクティブテナントをクリア (cleart) する必要があります。

```
midonet-cli> cclear

midonet-cli> router list
router router0 name MidoNet Provider Router state up
router router1 name Tenant Router state up infiltrer chain0 outfilter chain1
```

この例の場合、MidoNet プロバイダルーターは router0 です。

3. admin テナントをロードする

構成をさらに続ける前に、admin テナントを設定 (sett) する必要があります。上記の Keystone から取得した ID を使用してください。

```
midonet-cli> sett 12345678901234567890123456789012
tenant_id: 12345678901234567890123456789012
```

4. BGP セッション用の仮想ポートを作成する

リモート BGP ピアごとに、BGP 通信に使用するポートを MidoNet プロバイダルーター上に作成します。

```
midonet> router router0 add port address 198.51.100.2 net 198.51.100.0/30
router0:port0

midonet> router router0 add port address 203.0.113.2 net 203.0.113.0/30
router0:port1

midonet> router router0 port list
port port0 device router0 state up mac ac:ca:ba:11:11:11 address 198.51.100.2 net
198.51.100.0/30
port port1 device router0 state up mac ac:ca:ba:22:22:22 address 203.0.113.1 net
203.0.113.0/30
[...]
```

この例で作成されたポートは、port0 と port1 です。

5. 仮想ポートで BGP を構成する

```
midonet> router router0 port port0 add bgp local-AS 64512 peer-AS 64513
peer 198.51.100.1
router0:port0:bgp0

midonet> router router0 port port0 list bgp
bgp bgp0 local-AS 64512 peer-AS 64513 peer 198.51.100.1

midonet> router router0 port port1 add bgp local-AS 64512 peer-AS 64513
peer 203.0.113.1
router0:port1:bgp0

midonet> router router0 port port1 list bgp
bgp bgp0 local-AS 64512 peer-AS 64513 peer 203.0.113.1
```

6. Add routes to the remote BGP peers

In order to be able to establish connections to the remote BGP peers, corresponding routes have to be added.

```
midonet> router router0 route add src 0.0.0.0/0 dst 198.51.100.0/30 port router0:port0
type normal
router0:route0

midonet> router router0 route add src 0.0.0.0/0 dst 203.0.113.0/30 port router0:port1
type normal
router0:route1
```

7. BGPルートをアドバタイズする

ホストされている仮想マシンが外部接続できるようにするため、フローティング IP ネットワークを BGP ピアにアドバタイズする必要があります。

```
midonet> router router0 port port0 bgp bgp0 add route net 192.0.2.0/24
router0:port0:bgp0:ad-route0
```



```
port-group pgroup0 port router0:port1
```

```
midonet> port-group pgroup0 list member
```

```
port-group pgroup0 port router0:port0
```

```
port-group pgroup0 port router0:port1
```

第7章 高度な手順

OpenStackへのMidoNetのインストールと設定が完了しました。



注記

Midonetの運用に関する詳細については、「MidoNet 運用 ガイド」を参照してください。