

MidoNet クイック スタート ガイド

RHEL 7 / Juno (RDO)

2015.06-SNAPSHOT (2015-10-31 16:43 JST)

DRAFT



mido**net**

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)	7
4. MidoNetのインストール	11
NSDBノード	11
コントローラノード	14
Midolman のインストール	16
MidoNetのホストの登録	17
5. BGP アップリンク構成	19
6. 高度な手順	23

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。

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

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

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

第2章 環境の基本構成

目次

ネットワークの構成	4
SELinuxの構成	4
レポジトリの構成	4

ネットワークの構成



重要

すべてのホスト名はDNSまたはローカルで解決できる必要があります。

SELinuxの構成



重要

このガイドはSELinux（インストール済みの場合）が`permissive`または`disabled`のいずれかのモードであることが前提となります。

モードを変更する場合は、次のコマンドを実行します。

```
# setenforce Permissive
```

SELinuxの構成を恒久的に変更する場合は、`/etc/selinux/config` ファイルを適宜変更してください。

```
SELINUX=permissive
```

レポジトリの構成

必要なソフトウェアレポジトリを構成して、インストールしたパッケージを更新します。

1. Red Hatベースレポジトリを有効にする

```
# subscription-manager repos --enable=rhel-7-server-rpms
```

2. 追加のRed Hatレポジトリを有効にする

```
# subscription-manager repos --enable=rhel-7-server-extras-rpms  
# subscription-manager repos --enable=rhel-7-server-optional-rpms
```

3. レポジトリの優先順位付けを有効にする

```
# yum install yum-plugin-priorities
```

4. EPELレポジトリを有効にする

```
# yum install http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-5.noarch.  
rpm
```

5. RD0レポジトリを有効にする

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

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

コンピュータノード



重要

OpenStackの文書の [Install and configure a compute node](#) 指示に従います。ただし、次の追加点に注意してください。

- ## 1. libvirtを構成する

/etc/libvirt/qemu.conf ファイルを変更して以下を含めます。

```
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"
]
```

- ## 2. libvirtdサービスを再開する

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

- ### 3. 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
```

- #### 4. 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

OpenStackの文書の [Chapter 6. OpenStack Networking \(neutron\)](#) の指示に従います。ただし、次の相違点に注意してください。

コントローラノード



重要

OpenStackの文書の [Install and configure controller node](#) 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. 前提条件を設定する場合

このまま適用します。

- ## 2. ネットワーキングのコンポーネントをインストールする場合

適用*しないで*ください。

代わりに、次のパッケージをインストールします。

```
# yum install openstack-neutron python-neutron-plugin-midonet
```


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







```
...

Addresses of hosts that are deemed contact points.
seed_provider:
 - class_name: org.apache.cassandra.locator.SimpleSeedProvider
 parameters:
 - seeds: "nsdb1,nsdb2,nsdb3"
```

## b. ノード固有の構成

## i. NSDB ノード 1

/etc/cassandra/conf/cassandra.yaml ファイルを編集して、次のものを含めます。

```
Address to bind to and tell other Cassandra nodes to connect to.
listen_address: nsdb1

...

The address to bind the Thrift RPC service.
rpc_address: nsdb1
```

## ii.NSDB ノード 2

/etc/cassandra/conf/cassandra.yaml ファイルを編集して、次のものを含めます。

```
Address to bind to and tell other Cassandra nodes to connect to.
listen_address: nsdb2

...

The address to bind the Thrift RPC service.
rpc_address: nsdb2
```

## ii NSDB ノード 3

/etc/cassandra/conf/cassandra.yaml ファイルを編集して、次のものを含めます。

```
Address to bind to and tell other Cassandra nodes to connect to.
listen_address: nsdb3

...

The address to bind the Thrift RPC service.
rpc_address: nsdb3
```

### 3. サービスの init スクリプトを編集する

インストール時に `/var/run/cassandra` ディレクトリが作成されますが、一時的なファイルシステムに配置されるため、システムのリブート後に消失します。その結果、Cassandra サービスの停止や再開ができなくなります。

これを回避するには、`/etc/init.d/cassandra` ファイルを編集して、サービスの開始時にディレクトリを作成します。

```
[...]
case "$1" in
 start)
 # Cassandra startup
 echo -n "Starting Cassandra: "
```



```
<param-name>keystone-admin_token</param-name>
 <param-value>ADMIN_TOKEN</param-value>
</context-param>
```

```
<context-param>
 <param-name>zookeeper-zookeeper_hosts</param-name>
 <param-value>nsdb1:2181, nsdb2:2181, nsdb3:2181</param-value>
</context-param>
```

```
<context-param>
 <param-name>midocluster-properties_file</param-name>
 <param-value>/var/lib/tomcat/webapps/host_uid.properties</param-value>
</context-param>
```

### 3. Tomcatパッケージをインストールする

```
yum install tomcat
```

#### 4. TomcatHTTPヘッダの最大サイズを構成する

/etc/tomcat/server.xml ファイルを変更して、HTTPコネクタの最大サイズを調整します。

```
<Connector port="8080" protocol="HTTP/1.1"
 connectionTimeout="20000"
 URIEncoding="UTF-8"
 redirectPort="8443"
 maxHttpHeaderSize="65536" />
```

## 5. MidoNet APIのコンテキストを構成する

/etc/tomcat/Catalina/localhost/midonet-api.xml ファイルを作成し、修正して次を含めます。

```
<Context
 path="/midonet-api"
 docBase="/usr/share/midonet-api"
 antiResourceLocking="false"
 privileged="true"
/>
```

## 6. Tomcatを開始する

```
systemctl enable tomcat.service
systemctl start tomcat.service
```

## MidoNet CLIのインストール

## 1. MidoNet CLIパッケージをインストールする

```
yum install python-midonetclient
```

## 2. MidoNet CLIを構成する

~/midonetrc ファイルを作成し、修正して次を含めます。

```
[cli]
api_url = http://controller:8080/midonet-api
username = admin
password = ADMIN_PASS
project_id = admin
```





```
zone tzone0 host host1 address ip_address_of_host1

midonet> tunnel-zone tzone0 add member host host2 address ip_address_of_host2
zone tzone0 host host2 address ip_address_of_host2

midonet> tunnel-zone tzone0 add member host host3 address ip_address_of_host3
zone tzone0 host host3 address ip_address_of_host3
```

## 第5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
  - ・ 192.0.2.0/24/24
- ・ 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
```





```
midonet> port-group pgroup0 list member
port-group pgroup0 port router0:port0
port-group pgroup0 port router0:port1
```

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