

MidoNet クイック スタート ガイド

RHEL 7 / Kilo (RDO)

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

DRAFT



docs.midonet.org

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

概要

このガイドでは、MidonetとOpenStackを使用するために必要な最小限のインストールと設定の手順を説明しています。



注記

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. 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	12
NSDBノード	12
コントローラノード	15
Midolman のインストール	17
MidoNetのホストの登録	18
5. ネットワークの初期設定	20
6. BGP アップリンク構成	21
7. 高度な手順	25

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)

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。

このガイドでは、OpenStackのドキュメントの下記にある指示に従っていることを前提としています。

[OpenStack Networking \(neutron\)](#)

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
```


第3章 OpenStackのインストール

目次

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



重要

OpenStack Kilo Installation Guide for Red Hat Enterprise Linux 7
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。

アイデンティティサービス (Keystone)



重要

OpenStack文書の [Chapter 3. Add the Identity service](#) 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. 動作の確認

ステップ1の「セキュリティ上の理由で、一時的な認証トークンメカニズムを無効にする」を適用しないでください。

MidoNet APIは、認証のためにキーストーン管理トークンを使用しますので、`admin token auth` は対応する構成セクション内に保持する必要があります。

2. MidoNet APIサーバーを作成する

Keystoneの admin として、以下のコマンドを実行します。

```
$ openstack service create --name midonet --description "MidoNet API Service" midonet
```

3. MidoNet管理ユーザーを作成する

Keystoneの admin として、以下のコマンドを実行します。

```
$ openstack user create --password-prompt midonet
$ openstack role add --project service --user midonet admin
```

コンピュータサービス (Nova)



重要

OpenStack 文書の [Chapter 5. Add the Compute 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。

コントローラノード



注記

OpenStack文書の [Install and configure controller node](#) の指示にそのまま従います。


```
[DEFAULT]
...
core_plugin = neutron.plugins.midonet.plugin.MidonetPluginV2
```

代わりに、次の手順を実行します。

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

```
[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 = service
```

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

このまま適用します。

OpenStackのインストールガイドとともに、「[Load-Balancer-as-a-Service \(LBaaS\) を構成する](#)」[9]に記載されている、Load-Balancer-as-a-Service (LBaaS) を設定してください。

代わりに、次の手順を実行します。

```
# su -s /bin/sh -c "neutron-db-manage --config-file /etc/neutron/neutron.conf --
config-file /etc/neutron/plugins/midonet/midonet.ini upgrade kilo" neutron
```

b. コンピュータサービスを再開します。

8. ダッシュボードの Project セクションで負荷分散を有効にします。

現状のまま適用します。

9. インストールをファイナライズするには

[Neutron Controller Node Installation](#) の説明に従って、インストールをファイナライズします。

DHCP Agent



注記

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

1. Configure the DHCP agent

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

```
[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 = service
```

2. Enable and start the service

```
# systemctl enable neutron-dhcp-agent.service
# systemctl start neutron-dhcp-agent.service
```

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.

1. Configure the Metadata Agent

Configure the agent according to the "To configure the metadata agent" section in the OpenStack documentation's [Install and configure network node](#) instructions.

2. Enable and start the service

```
# systemctl enable neutron-metadata-agent.service
# systemctl start neutron-metadata-agent.service
```

コンピュータノード



重要

OpenStack文書の [Install and configure compute node](#) 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. 前提条件を設定する場合
適用*しないで*ください。
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。
3. ネットワーキング共通のコンポーネントを構成する場合
適用*しないで*ください。
4. モジュラーレイヤー2 (ML2) のプラグインを構成する場合
適用*しないで*ください。
5. Open vSwitch (OVS) サービスを構成する場合
適用*しないで*ください。
6. コンピュートでネットワーキングの使用を構成する場合
このまま適用します。
7. インストールを終了する場合
適用*しないで*ください。
代わりに、次のサービスを再開します。

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

T - DRAFT

`/var/lib/zookeeper/data/myid` ファイルを作成し、ホストのIDを含めます。

```
# echo 3 > /var/lib/zookeeper/data/myid
```

3. Java Symlinkを作成する

```
# mkdir -p /usr/java/default/bin/
# ln -s /usr/lib/jvm/jre-1.7.0-openjdk/bin/java /usr/java/default/bin/java
```

4. ZooKeeperを有効にして開始する

```
# systemctl enable zookeeper.service
# systemctl start zookeeper.service
```

5. ZooKeeper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、ZooKeeperが適切に動作するか確認します。

基本的な検査は、`ruok` (Are you ok?) 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実行している場合は、最初の列に ``imok`` (I am ok) と返されます。

```
$ echo ruok | nc 127.0.0.1 2181
imok
```

詳細情報が必要な場合は、`stat`コマンドを使用すると、パフォーマンスと接続しているクライアントの統計が一覧表示されます。

```
$ echo stat | nc 127.0.0.1 2181
Zookeeper version: 3.4.5--1, built on 06/10/2013 17:26 GMT
Clients:
  /127.0.0.1:34768[0](queued=0,recved=1,sent=0)
  /192.0.2.1:49703[1](queued=0,recved=1053,sent=1053)

Latency min/avg/max: 0/4/255
Received: 1055
Sent: 1054
Connections: 2
Outstanding: 0
Zxid: 0x260000013d
Mode: follower
Node count: 3647
```

Cassandraのインストール

1. Cassandraパッケージをインストールする

```
# yum install java-1.7.0-openjdk
# yum install dsc20
```

2. Cassandraを構成する

a. 共通の構成

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

```
# The name of the cluster.
cluster name: 'midonet'
```

```
...

# 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>midoccluster-properties_file</param-name>
  <param-value>/var/lib/tomcat/webapps/host_uuid.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
```

17


```
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 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。

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
```

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。