

MidoNet クイック スタート ガイド

Ubuntu 14.04 / Juno

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

DRAFT



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
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	15
MidoNetのホストの登録	16
5. BGP アップリンク構成	18
6. 高度な手順	22

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

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

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

目次



アイデンティティサービス (Keystone)



コンピュータサービス (Nova)



コントローラノード



6

3. ネットワーキングのサーバーコンポーネントを構成する場合

ステップ' dを適用*しないで*ください。 モジュラーレイヤー2 (ML2) プラグイン、ルーターサービスおよび重複するIPアドレスを有効にします。

- a. 代わりに、`/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
```



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）。

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。

代わりに、次の手順を実行します。

- a. MidoNetプラグインのディレクトリを作成します。

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

- b. `/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 = service
```

- c. `/etc/default/neutron-server` ファイルを変更して以下を含めます。

```
NEUTRON_PLUGIN_CONFIG="/etc/neutron/plugins/midonet/midonet.ini"
```

5. 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。

6. インストールを終了する場合

このまま適用します。

DHCP エージェント



注記

MidoNet には、デフォルトの OpenStack ネットワークプラグインのよう なネットワークノードの概念がないため、DHCP エージェントをコントローラノードにインストールします。

第4章 MidoNetのインストール

目次

NSDB ノード	11
コントローラノード	14
Midolman のインストール	15
MidoNetのホストの登録	16

NSDB ノード

ZooKeeperのインストール

1. ZooKeeperパッケージをインストールする

```
# apt-get install zookeeper zookeeperd zkdump
```

2. ZooKeeperを構成する

a. 共通の構成

/etc/zookeeper/conf/zoo.cfg ファイルを修正して次を含めます。

```
server.1=nsdb1:2888:3888
server.2=nsdb2:2888:3888
server.3=nsdb3:2888:3888
```



重要

For production deployments it is recommended to configure the storage of snapshots in a different disk than the commit log. This can be set by changing the parameter `dataDir` in `zoo.cfg` to a different disk.

b. ノード固有の構成

i. NSDB ノード 1

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

```
# echo 1 > /var/lib/zookeeper/myid
```

ii. NSDB ノード 2

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

```
# echo 2 > /var/lib/zookeeper/myid
```

ii NSDB ノード 3

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

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

3. ZooKeeperを再開する


```
# 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/cassandra.yaml ファイルを変更して以下を含めます。

```
# 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/cassandra.yaml ファイルを変更して以下を含めます。

```
# 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. 既存のデータを消去してをCassandra再開する

```
# service cassandra stop
# rm -rf /var/lib/cassandra/*
# service cassandra start
```

4. 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
```

14


```
$ echo "cassandra.replication_factor : 3" | mn-conf set -t default
```

4. リソース使用の設定

リソース使用を設定するために、各エージェントホストで下記の手順を実行します。



重要

本番環境では large （大）テンプレートを強くお勧めします。

a. Midolman リソーステンプレート

Midolmanリソーステンプレートを設定するためには、次のコマンドを実行します。

```
$ mn-conf template-set -h local -t TEMPLATE NAME
```

TEMPLATE_NAME を以下のいずれかのテンプレートに置き換えます。

```
agent-compute-large
agent-compute-medium
agent-gateway-large
agent-gateway-medium
default
```

b. Java Virtual Machine (JVM) リソーステンプレート

JVMリソーステンプレートを設定するためには、デフォルトの `/etc/midolman/midolman-env.sh` ファイルを以下のいずれかに置き換えます。

```
/etc/midolman/midolman-env.sh.compute.large
/etc/midolman/midolman-env.sh.compute.medium
/etc/midolman/midolman-env.sh.gateway.large
/etc/midolman/midolman-env.sh.gateway.medium
```

5. Midolman を起動する

```
# service midolman start
```

MidoNetのホストの登録

1. MidoNet CLIを起動する

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

2. トンネルゾーンを作成する

MidoNeはVXLANVirtual (Extensible LAN) およびGRE (Generic Routing Encapsulation) プロトコルサポートしているため、トンネルゾーンで他のホストと通信できます。

VXLAN プロトコルを使用するには、「vxlan」と入力してトンネルゾーンを作成します。

```
midonet> tunnel-zone create name tz type vxlan
tzone0
```

GREプロトコルを使用するには、「gre」と入力してトンネルゾーンを作成します。

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

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
```

20


```
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6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。

これでNeutronの最初のネットワークの構築を続行できます。



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。