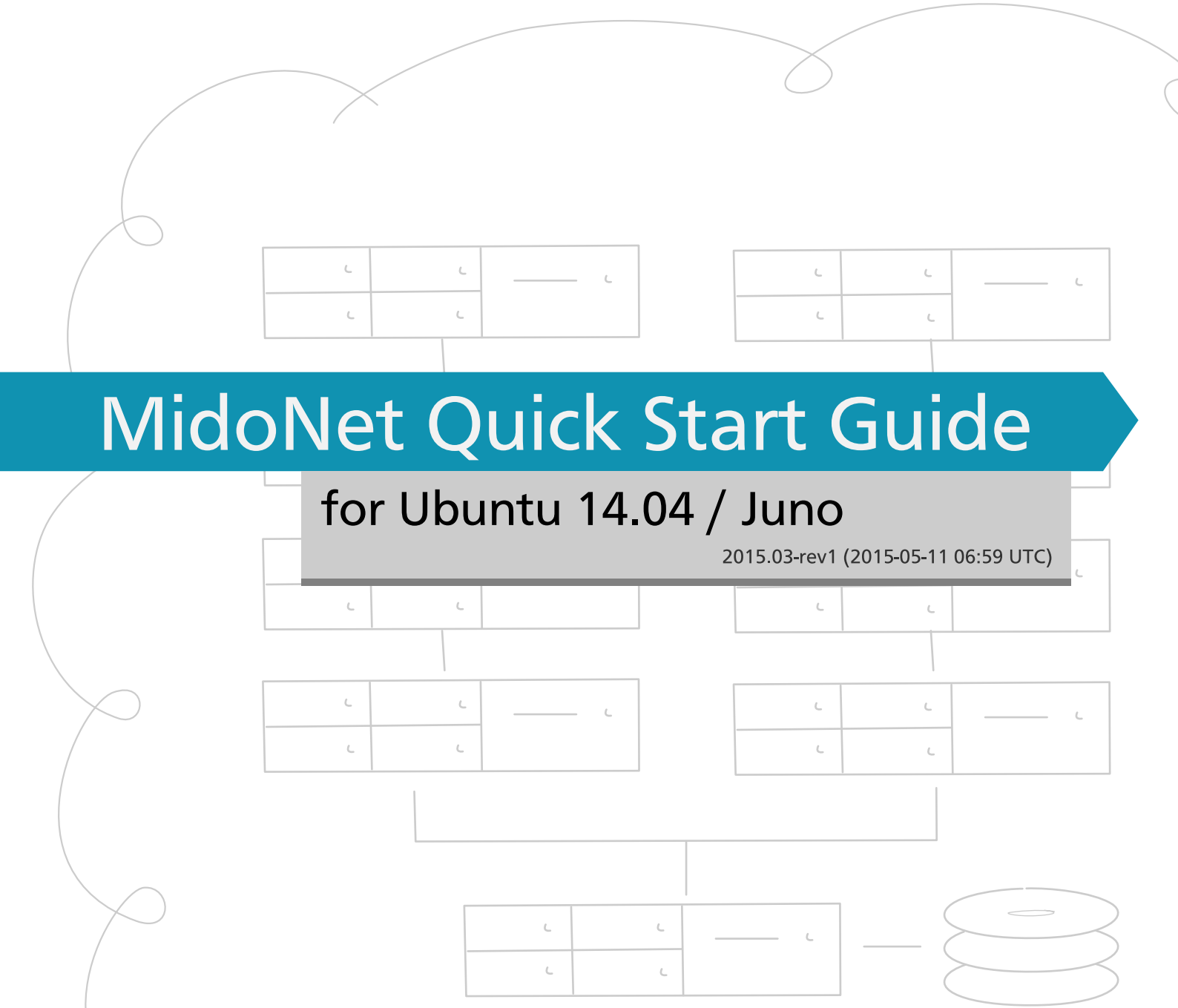

MidoNet Quick Start Guide

for Ubuntu 14.04 / Juno

2015.03-rev1 (2015-05-11 06:59 UTC)

midonet

docs.midonet.org

MidoNet Quick Start Guide for Ubuntu 14.04 / Juno

2015.03-rev1 (2015-05-11 06:59 UTC)

MidoNet is a network virtualization software for Infrastructure-as-a-Service (IaaS) clouds.

It decouples your IaaS cloud from your network hardware, creating an intelligent software abstraction layer between your end hosts and your physical network.

This guide walks through the minimum installation and configuration steps neccessary to use MidoNet with OpenStack.

Note

Please consult the MidoNet Mailing Lists or Chat if you need assistance.

Table of Contents

Preface

Conventions

The MidoNet documentation uses several typesetting conventions.

Notices

Notices take these forms:

Note

A handy tip or reminder.

Important

Something you must be aware of before proceeding.

Warning

Critical information about the risk of data loss or security issues.

Command prompts

$ prompt

Any user, including the root user, can run commands that are prefixed with the $ prompt.

 # prompt

The root user must run commands that are prefixed with the # prompt. You can also prefix these commands with the **sudo** command, if available, to run them.

1. Architecture

Table of Contents

Important

This guide assumes the following system architecture, based on Figure 1.3. Minimal architecture example with OpenStack Networking (neutron) of the OpenStack Documentation. This architecture consists of three hosts:

- Controller Node (**controller**)
- Gateway Node (**network**)
- Compute Node (**compute1**)

The *MidoNet Network State Database (NSDB)* uses ZooKeeper and Cassandra to store network topology and state information. The NSDB components can be installed on separate hosts, but this guide assumes them to be installed on all three nodes (**controller**, **network**, **compute1**).

The *MidoNet Agent (Midolman)* has to be installed on all nodes where traffic enters or leaves the virtual topology, in this guide this are the **network** and **compute1** nodes.

The *Midonet API* can be installed on a separate host, but this guide assumes it to be installed on the **controller** node.

The *Midonet Command Line Interface (CLI)* can be installed on a separate host, but this guide assumes it to be installed on the **controller** node.

The *Midonet Neutron Plugin* replaces the ML2 Plugin and has to be installed on all three nodes (**controller**, **network**, **compute1**).

Hosts and Services

Controller Node (controller)

- General
  - Database (MariaDB)
  - Message Broker (RabbitMQ)
- OpenStack
  - Identity Service (Keystone)
  - Image Service (Glance)
  - Compute (Nova)
  - Networking (Neutron)

    - Dashboard (Horizon)
- MidoNet
    - API
    - CLI
    - Neutron Plugin
    - Network State Database (NSDB)
        - Network Topology (ZooKeeper)
        - Network State Information (Cassandra)

Gateway Node (network)

- OpenStack
    - Networking (Neutron)
        - DHCP Agent
        - Metadata Agent
- MidoNet
    - Agent (Midolman)
    - Neutron Plugin
    - Network State Database (NSDB)
        - Network Topology (ZooKeeper)
        - Network State Information (Cassandra)

Compute Node (compute1)

- OpenStack
    - Compute (Nova)
    - Networking (Neutron)
- MidoNet
    - Agent (Midolman)
    - Neutron Plugin
    - Network State Database (NSDB)
        - Network Topology (ZooKeeper)
        - Network State Information (Cassandra)

2. Basic Environment Configuration

Table of Contents

Networking Configuration

Important

All hostnames must be resolvable, either via DNS or locally.

This guide assumes that you follow the instructions in OpenStack Networking (neutron) of the OpenStack Documentation.

Repository Configuration

Configure necessary software repositories and update installed packages.

1. **Configure Ubuntu repositories**

   Edit the `/etc/apt/sources.list` file to contain the following:

```
# Ubuntu Main Archive
deb http://archive.ubuntu.com/ubuntu/ trusty main
deb http://security.ubuntu.com/ubuntu trusty-security main

# Ubuntu Universe Archive
deb http://archive.ubuntu.com/ubuntu/ trusty universe
deb http://security.ubuntu.com/ubuntu trusty-security universe
```

2. **Configure Ubuntu Cloud Archive repository**

   Create the `/etc/apt/sources.list.d/cloudarchive-juno.list` file and edit it to contain the following:

```
# Ubuntu Cloud Archive
deb http://ubuntu-cloud.archive.canonical.com/ubuntu trusty-updates/juno
 main
```

   Install the repository's key:

```
# apt-get update
# apt-get install ubuntu-cloud-keyring
```

3. **Configure DataStax repository**

   Create the `/etc/apt/sources.list.d/datastax.list` file and edit it to contain the following:

```
# DataStax (Apache Cassandra)
deb http://debian.datastax.com/community stable main
```

   Download and install the repository's key:

```
# curl -L http://debian.datastax.com/debian/repo_key | apt-key add -
```

4. **Configure MidoNet repositories**

   Create the `/etc/apt/sources.list.d/midonet.list` file and edit it to contain the following:

```
# MidoNet
deb http://repo.midonet.org/midonet/v2015.03 stable main

# MidoNet OpenStack Integration
deb http://repo.midonet.org/openstack-juno stable main

# MidoNet 3rd Party Tools and Libraries
deb http://repo.midonet.org/misc stable main
```

   Download and install the repositories' key:

```
# curl -L http://repo.midonet.org/packages.midokura.key | apt-key add -
```

5. **Install available updates**

```
# apt-get update
# apt-get dist-upgrade
```

6. **If necessary, reboot the system**

```
# reboot
```

3. OpenStack Installation

Table of Contents

Important

Follow the OpenStack Juno Installation Guide for Ubuntu 14.04 (LTS) , but **note the following differences**.

Identity Service (Keystone)

Important

Follow the OpenStack documentation's Chapter 3. Add the Identity service instructions, but **note the following additions**.

1. **Create MidoNet API Service**

   As Keystone admin, execute the following command:

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

2. **Create MidoNet Administrative User**

   As Keystone admin, execute the following commands:

```
$ keystone user-create --name midonet --pass MIDONET_PASS --tenant
 service
$ keystone user-role-add --user midonet --role admin --tenant service
```

Compute Services (Nova)

Important

Follow the OpenStack documentation's Chapter 5. Add the Compute service instructions, but **note the following differences**.

Controller Node

Note

Follow the OpenStack documentation's Install and configure controller node instructions as is.

Compute Node

Important

Follow the OpenStack documentation's Install and configure a compute node instructions, but **note the following additions**.

1. **Configure libvirt**

   Edit the `/etc/libvirt/qemu.conf` file to contain the following:

```
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. **Restart the libvirt service**

```
# service libvirt-bin restart
```

1. **Install nova-rootwrap network filters**

```
# apt-get install nova-network
```

2. **Restart the Compute service**

```
# service nova-compute restart
```

Networking Service (Neutron)

Important

Follow the OpenStack documentation's Chapter 6. OpenStack Networking (neutron) instructions, but **note the following differences**.

Controller Node

Important

Follow the OpenStack documentation's Install and configure controller node instructions, but **note the following differences**.

1. **To configure prerequisites**

   Apply as is.

2. **To install the Networking components**

   Do **not** apply.

   a. Instead, install the following packages:

```
# apt-get install neutron-server python-neutron-plugin-midonet
```

3. **To configure the Networking server component**

   Do **not** apply step 'd. Enable the Modular Layer 2 (ML2) plug-in, router service, and overlapping IP addresses'.

   a. Instead, edit the `/etc/neutron/neutron.conf` file and add the following key to the `[DEFAULT]` section:

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

   

   ### Note

   Make sure to not leave any space at the starting of lines in any configuration file (this applies to all configuration files).

4. **To configure the Modular Layer 2 (ML2) plug-in**

   Do **not** apply.

   Instead, perform the following steps.

   a. Create the directory for the MidoNet plugin:

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

   b. Create the `/etc/neutron/plugins/midonet/midonet.ini` file and edit it to contain the following:

   ```
   [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. Edit the `/etc/default/neutron-server` file to contain the following:

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

5. **To configure Compute to use Networking**

   Apply as is.

6. **To finalize installation**

   Apply as is.

Gateway Node

Important

Follow the OpenStack documentation's Install and configure network node instructions, but **note the following differences**.

1. **To configure prerequisites**

   Apply as is.

2. **To install the Networking components**

   Do **not** apply.

   a. Instead, install the following packages:

   ```
   # apt-get install neutron-dhcp-agent neutron-metadata-agent python-
   neutron-plugin-midonet
   ```

3. **To configure the Networking common components**

   Do **not** apply step 'd. Enable the Modular Layer 2 (ML2) plug-in, router service, and overlapping IP addresses'.

   a. Instead, edit the `/etc/neutron/neutron.conf` file and add the following key to the `[DEFAULT]` section:

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

4. **To configure the Modular Layer 2 (ML2) plug-in**

   Do **not** apply.

5. **To configure the Layer-3 (L3) agent**

   Do **not** apply.

6. **To configure the DHCP agent**

   Do **not** apply.

   a. Instead, 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
   ```

7. **To configure the metadata agent**

   Apply as is.

8. **To configure the Open vSwitch (OVS) service**

   Do **not** apply.

9. **To finalize the installation**

   Do **not** apply.

   a. Instead, restart the following services:

   ```
   # service neutron-dhcp-agent restart
   # service neutron-metadata-agent restart
   ```

Compute Node

Important

Follow the OpenStack documentation's Install and configure compute node instructions, but **note the following differences**.

1. **To configure prerequisites**

   Apply as is.

2. **To install the Networking components**

   Do **not** apply.

   a. Instead, install the following package:

   ```
   # apt-get install neutron-common
   ```

3. **To configure the Networking common components**

   Do **not** apply step 'd. Enable the Modular Layer 2 (ML2) plug-in, router service, and overlapping IP addresses'.

4. **To configure the Modular Layer 2 (ML2) plug-in**

   Do **not** apply.

5. **To configure the Open vSwitch (OVS) service**

   Do **not** apply.

6. **To configure Compute to use Networking**

   Apply as is.

7. **To finalize the installation**

   Do **not** apply.

   a. Instead, restart the following service:

   ```
   # service nova-compute restart
   ```

4. MidoNet Installation

Table of Contents

NSDB Nodes

ZooKeeper Installation

1. **Install ZooKeeper packages**

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

2. **Configure ZooKeeper**

   a. **Common Configuration**

   Edit the `/etc/zookeeper/conf/zoo.cfg` file to contain the following:

```
server.1=controller:2888:3888
server.2=network:2888:3888
server.3=compute1:2888:3888
```

   b. **Node-specific Configuration**

      i. **Controller Node**

      Create the `/var/lib/zookeeper/myid` file and edit it to contain the host's ID:

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

      ii. **Gateway Node**

      Create the `/var/lib/zookeeper/myid` file and edit it to contain the host's ID:

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

      iii. **Compute Node**

      Create the `/var/lib/zookeeper/myid` file and edit it to contain the host's ID:

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

3. **Restart ZooKeeper**

```
# service zookeeper restart
```

4. **Verify ZooKeeper Operation**

After installation of all nodes has been completed, verify that ZooKeeper is operating properly.

A basic check can be done by executing the `ruok` (Are you ok?) command on all nodes. This will reply with `imok` (I am ok.) if the server is running in a non-error state:

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

More detailed information can be requested with the `stat` command, which lists statistics about performance and connected clients:

```
$ 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 Installation

1. **Install Cassandra packages**

```
# apt-get install openjdk-7-jre-headless
# apt-get install dsc20=2.0.10-1 cassandra=2.0.10
# apt-mark hold dsc20 cassandra
```

2. **Configure Cassandra**

   a. **Common Configuration**

   Edit the `/etc/cassandra/cassandra.yaml` file to contain the following:

```
# 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: "controller,network,compute1"
```

   b. **Node-specific Configuration**

      i. **Controller Node**

      Edit the `/etc/cassandra/cassandra.yaml` file to contain the following:

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

...
```

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

ii. **Gateway Node**

Edit the `/etc/cassandra/cassandra.yaml` file to contain the following:

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

...

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

iii. **Compute Node**

Edit the `/etc/cassandra/cassandra.yaml` file to contain the following:

```
# Address to bind to and tell other Cassandra nodes to connect to.
listen_address: compute1

...

# The address to bind the Thrift RPC service.
rpc_address: compute1
```

3. **Clean existing data and restart Cassandra**

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

4. **Verify Cassandra Operation**

After installation of all nodes has been completed, verify that Cassandra is operating properly.

A basic check can be done by executing the `nodetool status` command. This will reply with UN (Up / Normal) in the first column if the servers are running in a non-error state:

```
$ nodetool -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
```

Controller Node

MidoNet API Installation

1. **Install MidoNet API package**

```
# apt-get install midonet-api
```

2. **Configure MidoNet API**

   Edit the `/usr/share/midonet-api/WEB-INF/web.xml` file to contain the following:

```
<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>
    <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>controller:2181,network:2181,compute1:2181</param-
value>
</context-param>
```

3. **Install Tomcat package**

```
# apt-get install tomcat7
```

4. **Configure Tomcat's Entropy Source**

   Edit the `/usr/share/tomcat7/bin/catalina.sh` file to contain the following:

```
JAVA_OPTS="$JAVA_OPTS -Djava.security.egd=file:/dev/./urandom"
```

5. **Configure Tomcat's Maximum HTTP Header Size**

   Edit the `/etc/tomcat7/server.xml` file and adjust the maximum header size for the HTTP connector:

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

6. **Configure MidoNet API context**

   Create the `/etc/tomcat7/Catalina/localhost/midonet-api.xml` file and edit it to contain the following:

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

7. **Restart Tomcat**

```
# service tomcat7 restart
```

MidoNet CLI Installation

1. **Install MidoNet CLI package**

```
# apt-get install python-midonetclient
```

2. **Configure MidoNet CLI**

   Create the `~/.midonetrc` file and edit it to contain the following:

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

Midolman Installation

The Midolman agent shall be installed on all network and compute nodes.

1. **Install Midolman package**

```
# apt-get install midolman
```

2. **Set up mn-conf**

   Edit /etc/midolman/midolman.conf to point mn-conf to the ZooKeeper cluster:

```
[zookeeper]
zookeeper_hosts = controller:2181,network:2181,compute1:2181
```

3. **Configure access to the NSDB for all agents**

   This step needs to happen only once, it will set up access to the NSDB for all MidoNet nodes. Run the following command to set the cloud-wide values for the ZooKeeper and Cassandra server addresses:

```
$ echo << EOF | mn-conf set -t default
zookeeper {
    zookeeper_hosts = $CONTROLLER:2181,$NETWORK:2181,$COMPUTE1:2181
}

cassandra {
    servers = $CONTROLLER,$NETWORK,$COMPUTE1
}
EOF
```

4. **Start Midolman**

```
# start midolman
```

MidoNet Host Registration

1. **Launch MidoNet CLI**

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

2. **Create tunnel zone**

MidoNet supports the Virtual Extensible LAN (VXLAN) and Generic Routing Encapsulation (GRE) protocols to communicate to other hosts within a tunnel zone.

To use the VXLAN protocol, create the tunnel zone with type 'vxlan':

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

To use the GRE protocol, create the tunnel zone with type 'gre':

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

3. **Add hosts to tunnel zone**

```
midonet> list tunnel-zone
tzone tzone0 name tz type vxlan

midonet> list host
host host0 name network alive true
host host1 name compute1 alive true

midonet> tunnel-zone tzone0 add member host host0
 address ip_address_of_host0
zone tzone0 host host0 address ip_address_of_host0

midonet> tunnel-zone tzone0 add member host host1
 address ip_address_of_host1
zone tzone0 host host1 address ip_address_of_host1
```

5. Further Steps

MidoNet installation and integration into OpenStack is completed.

You can now continue with the creation of initial networks in Neutron.

Note

Consult the **Operation Guide** for further instructions on operating MidoNet.