ICOS and OpenSwitch guide for OpenStack ML2 integration

ICOS and OpenSwitch guide for OpenStack ML2 integration

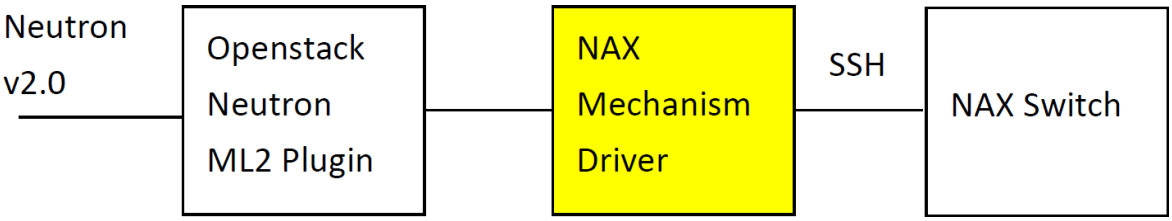
Table of Contents

Chapter 1. OpenStack Neutron ML2 Driver

1.1. Overview

The Modular Layer 2 (ML2) plugin is a framework allowing OpenStack Networking to simultaneously utilize the variety of layer 2 networking technologies found in complex real-world data centers.

NAX ML2 driver implements ML2 Driver API and supports configuring of NAX switches. NAX ML2 driver uses SSH to configure the NAX switch.



Initial version of this driver only support VLAN for OpenStack Newton.

MLAG feature does not work with OpenSwitch

Refer to https://wiki.openstack.org/wiki/Neutron/ML2 for OpenStack Neutron ML2 details.

1.2. Install the NAX ML2 plugin

Before you begin starting the NAX ML2 plugin, ensure that:

- The Neutron server is running the ML2 plugin.
- The Open vSwitch mechanism driver is configured in the ML2 configuration.

The NAX ML2 mechanism driver code is located in the following directory: *<neutron_install_dir>/networking_nax/plugins/ml2/drivers/nax*

The NAX ML2 mechanism configuration template is located at: */etc/neutron/plugins/ml2/ml2_conf_nax.ini*

<neutron_install_dir> is the directory where the Neutron project is installed.

<neutron_install_dir>is located at/usr/lib/python2.7/site-packages

NAX ML2 driver is not default build-in driver in OpenStack. OpenStack can't find specific configuration files for NAX ML2 driver. In order to let OpenStack load the correct configuration files, you have to modify files on Neutron server node.

The following section will show you how to install NAX ML2 driver on OpenStack Newton.

1.2.1. NAX ML2 Driver

1. Download NAX ML2 driver
2. Copy networking-nax-master directory to /home/root directory
3. Install NAX ML2 Plug-in

```
# cd networking-nax-master/
# python setup.py install
running install
[pbr] Generating AUTHORS
[pbr] AUTHORS complete (0.0s)
running build
running build_py
creating build
creating build/lib
creating build/lib/networking_nax
….. Skip display ……
running install_scripts
No handlers could be found for logger "oslo_config.cfg"
INFO [alembic.runtime.migration] Context implMySQLImpl.
INFO [alembic.runtime.migration] Will assume non-transactional DDL.
Running upgrade for neutron ...
INFO [alembic.runtime.migration] Context implMySQLImpl.
INFO [alembic.runtime.migration] Will assume non-transactional DDL.
```

```
OK
INFO [alembic.runtime.migration] Context implMySQLImpl.
INFO [alembic.runtime.migration] Will assume non-transactional DDL.
Running upgrade for networking-nax ...
INFO [alembic.runtime.migration] Context implMySQLImpl.
INFO [alembic.runtime.migration] Will assume non-transactional DDL.
OK
#
```

1.2.2. ML2 Configuration File

Configure the following keys in addition to an ML2 configuration. Specify OpenvSwitch and NAX mechanism driver.

Edit */etc/neutron/plugins/ml2/ml2_conf.ini*

```
[ml2]
#
# From neutron.ml2
#
type_drivers = flat,vlan
tenant_network_types = vlan
mechanism_drivers = openvswitch,networking-nax
extension_drivers = port_security
```

1.2.3. NAX ML2 driver configuration

To configure the NAX ML2 mechanism driver, do the following:

- Modify the NAX switch information to /etc/neutron/plugins/ml2/ml2_conf_nax.ini file including the following information (see the example below):
  - The IP address of the switch
  - The hostname and port of the node that is connected to the switch
  - The NAX switch credential username and password

```
[NAX_SWITCH:172.16.0.123]
controller=0/1
username=test
password=12345678
[NAX_SWITCH:172.16.0.124]
computer3=0/15
computer4=0/16
```

For a more detailed description, please see the Section 1.4, “NAX ML2 driver configuration”.

1.2.4. Systemd service configuration

```
# vi /usr/lib/systemd/system/neutron-server.service
ExecStart=/usr/bin/neutron-server --config-file /usr/share/neutron/
```

```
neutron-dist.conf
--config-dir /usr/share/neutron/server --config-file /etc/neutron/
neutron.conf --config-file
/etc/neutron/plugin.ini --config-dir /etc/neutron/conf.d/common --config-dir
/etc/neutron/conf.d/neutron-server --log-file /var/log/neutron/
server.log --config-file
/etc/neutron/plugins/ml2/ml2_conf_nax.ini
#
```

1.3. Restart the NAX ML2 plugin

If you add brand new servers or switch to your OpenStack cluster, you have to edit *ml2_conf_nax.ini* and to restart Neutron to let new servers or switch to be controlled by NAX ML2 plugin.

To restart the services, use:

```
# systemctl restart neutron-server
```

1.4. NAX ML2 driver configuration

To configure the NAX ML2 mechanism driver, do the following:

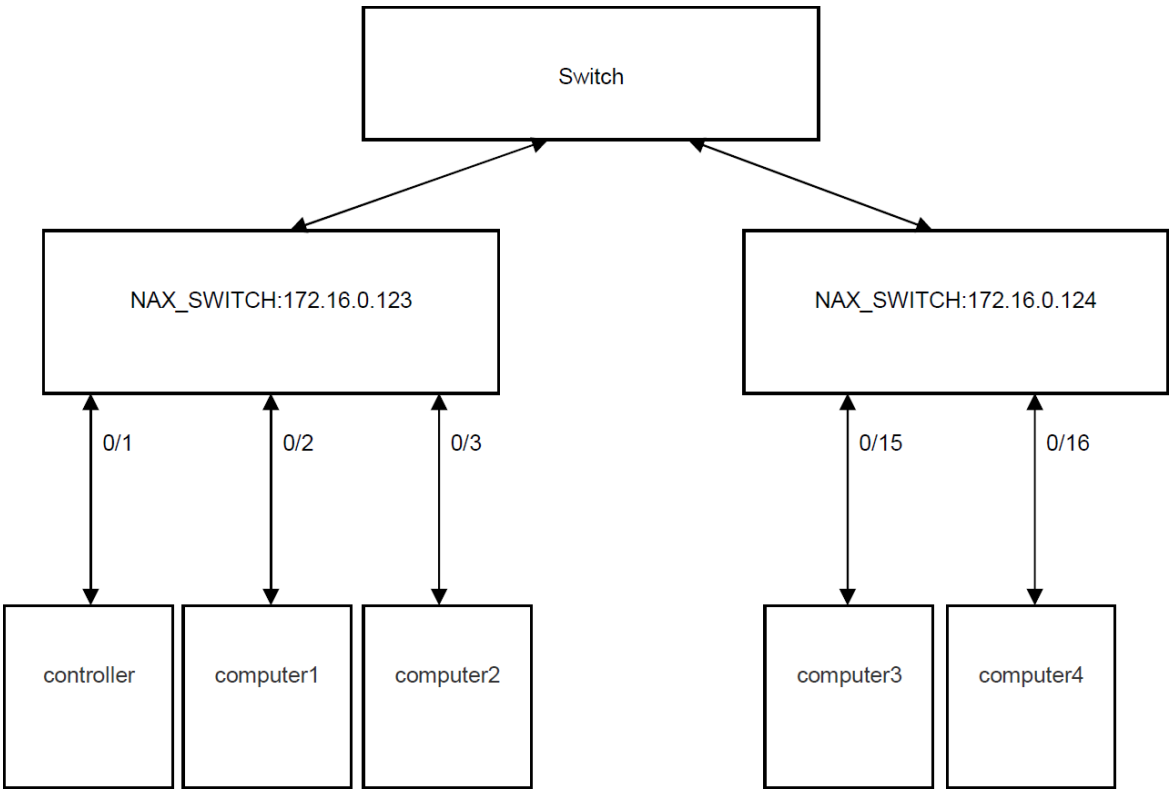
- Copy *ml2_conf_nax.ini* of NAX ML2 driver to */etc/neutron/plugins/ml2/*
- Modify the NAX switch information to *ml2_conf_nax.ini* file include the following information.

Configuration option = Default value	**Description**
GENERAL_SETTINGS	Configure default value
physical_network=<physical network>	The <physical network> is physical network name. Allows valid physical network to create/delete a network. This is an optional field.
baremetal_only=true	Only allow to create/delete bare metal machine. This is an optional field.
NAX_SWITCH:<ipaddr>	The <ipaddr> is IP address of service port on the NAX switch. This is a required field. If not set, all communications to NAX switch will be failed.
<hostname>=<switch port>	The <hostname>is hostname of the server. The <switch port> is port of NAX switch connected to the server. This is a required field. For MLAG asymmetric topology must configure NC port of NAX switch is not connected to the server. NC means: Not Connected
username=<username>	The NAX switch credential username. This is required field.
password=<password>	The NAX switch credential password. This is an optional field. If not set, an empty password is assumed.
ssh_port=<ssh_port>	The NAX switch is connected to ssh port. This is an optional field. If not set, a value of 22 is assumed.

In this section, we will use three examples to demonstrate how to configure the NAX ML2 configuration file.

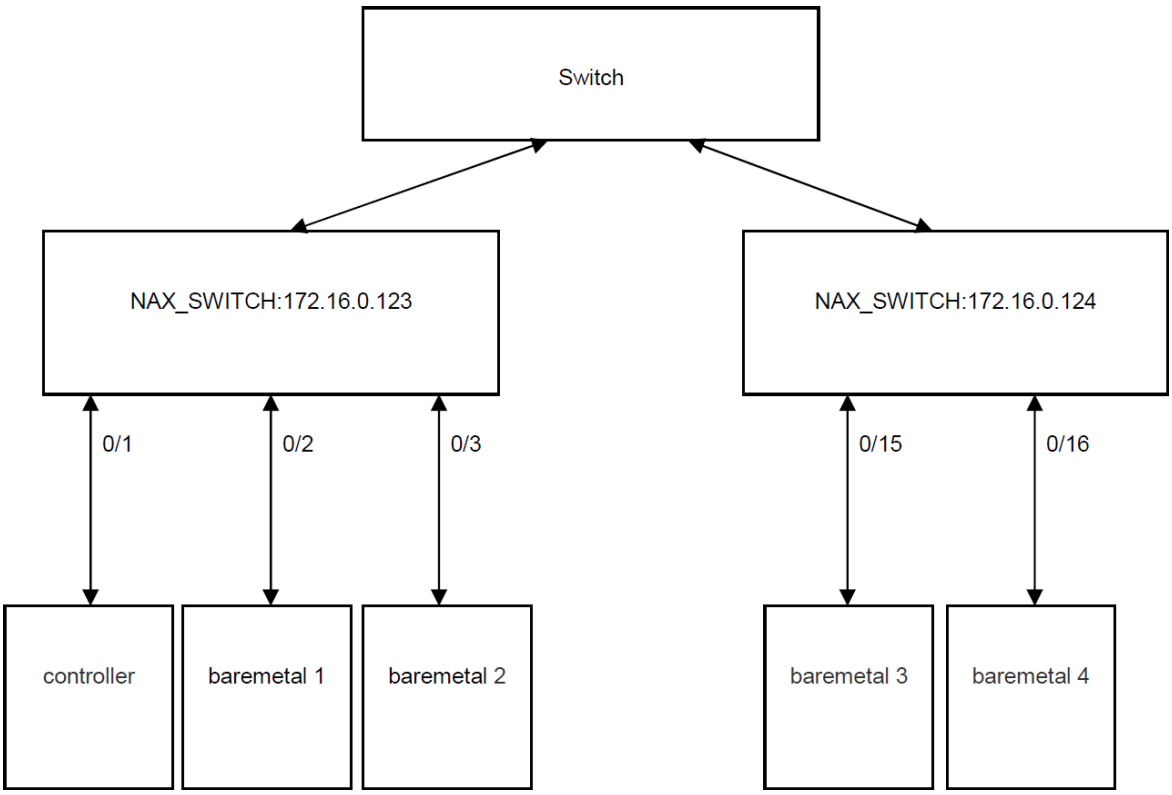
1.4.1. Configure NAX ML2 driver for ToR switch

In the figure below, it shows the VM topology used for VLAN configuration.




```
[NAX_SWITCH:172.16.0.123]
controller=0/1
computer1=0/2
computer2=0/3
[NAX_SWITCH:172.16.0.124]
computer3=0/15
computer4=0/16
```

In the figure below, it shows the bare metal topology used for VLAN configuration.




```
[NAX_SWITCH:172.16.0.123]
controller=0/1
```

neutron-port configurations:

To notify neutron about bare metal ports, Ironic uses its own mechanisms to inspect the hardware, and forward that information as part of neutron-port configuration.

For that new fields introduced in neutron lport:

local_link_information - that field located in the lport binding-profile and used for inform neutron how the port is connected the TOR switch. It includes 3 parameters:

switch_id - identifier of the switch that the port connected to. It can be the switch MAC address.

port_id - a physical port-identifier in the switch.

switch_info - other information about the switch (optional param).

The NAX mechanism driver should use that information while binding the lport.

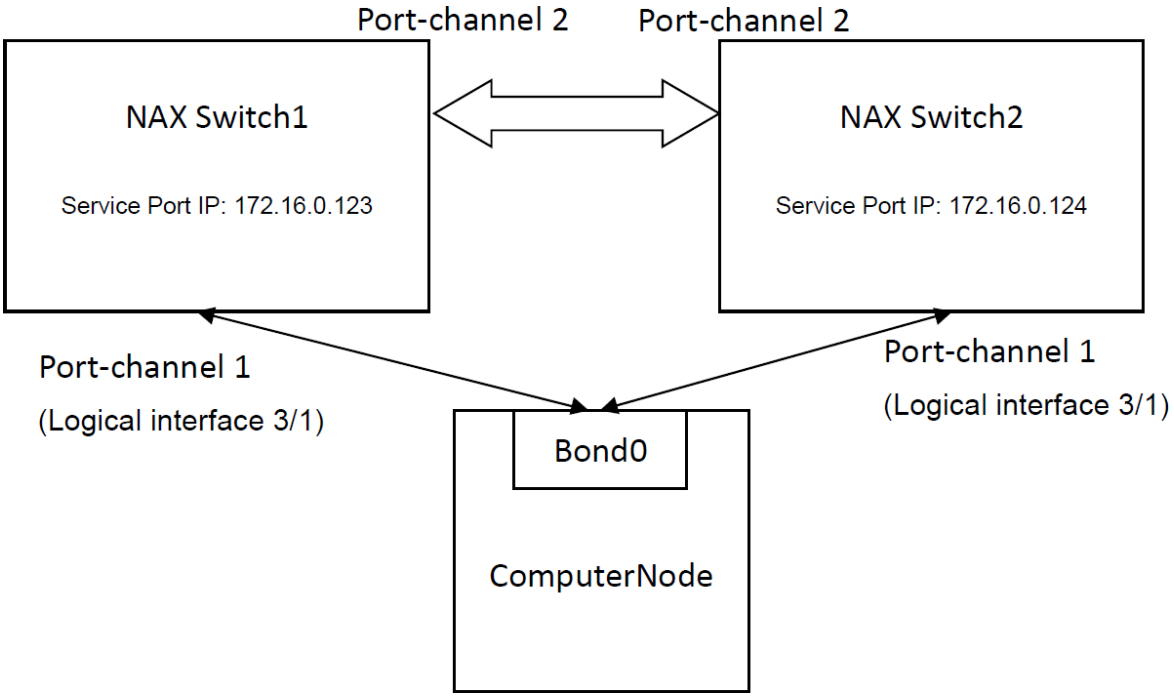
A JSON example of binding:profile with local_link_information reuse:

```
"binding:profile": {
'local_link_information': [
{
'switch_info': '172.16.0.123',
```

```
'port_id': '0/2'
'switch_id': 'aa:bb:cc:dd:ee:ff'
},
],
}
```

1.4.2. Configure NAX ML2 driver for MLAG symmetric topology

A typical MLAG setup is illustrated in the following diagram.



Configure MLAG in the plugin with multiple connections per host. For example, ComputerNode is connected to two NAX switches 172.16.0.123 and 172.16.0.124 over port-channel1.

```
[NAX_SWITCH:172.16.0.123]
ComputerNode=3/1
[NAX_SWITCH:172.16.0.124]
ComputerNode=3/1
```

Port-channel 1 is assigned logical interface ID3/1.

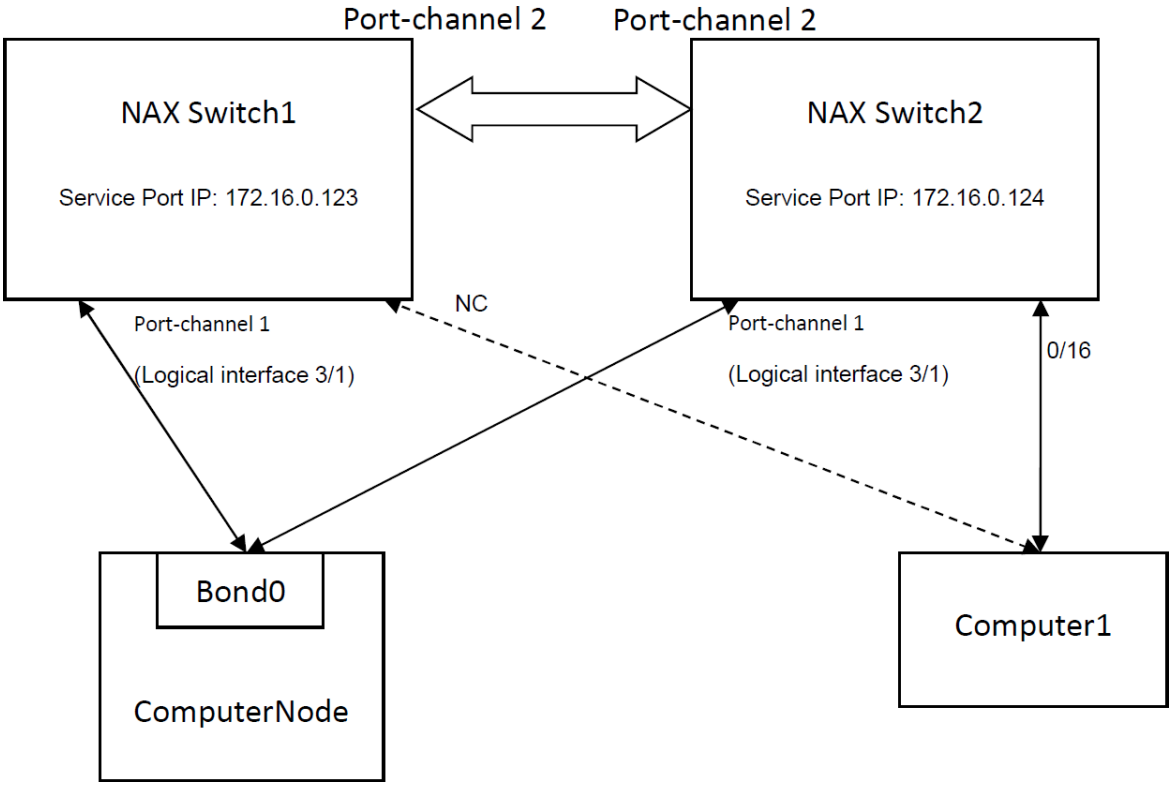
Use the **show port-channel all** ICOS command to show the logical interface IDs on NAX switch.

```
(Routing) #show port-channel all
Log.    Channel                                    Adm.
Intf    Name                        Min      Link   Mode          Type
------- --------------- ---- ------ ---- ---- ------ ------------- ----
3/1      ch1                        1        Down     En.         Stat
3/2      ch2                        1        Down     En.         Stat
```

```
3/3       ch3                    1          Down       En.          Stat
3/4       ch4                    1          Down       En.          Stat
3/5       ch5                    1          Down       En.          Stat
3/6       ch6                    1          Down       En.          Stat
3/7       ch7                    1          Down       En.          Stat
3/8       ch8                    1          Down       En.          Stat
3/9       ch9                    1          Down       En.          Stat
```

1.4.3. Configure NAX ML2 driver for MLAG asymmetric topology

A typical MLAG setup is illustrated in the following diagram.



Configure MLAG in the plugin with multiple connections per host. For example, ComputerNode is connected to two NAX switch1 (IP address: 172.16.0.123) and NAX switch2 (IP address: 172.16.0.124) over port-channel 1. Computer1 is connected to port16 of NAX Switch2. You need to add the NC port of NAX Switch1 to connect Computer1.

```
[NAX_SWITCH:172.16.0.123]
ComputerNode=3/1
Computer1=NC
[NAX_SWITCH:172.16.0.124]
ComputerNode=3/1
Computer1=0/16
```

Port-channel 1 is assigned logical interface ID3/1.

Use the **show port-channel all** ICOS command to show the logical interface IDs on NAX switch.

```
(Routing) #show port-channel all
Log.    Channel                                    Adm.
Intf    Name                          Min      Link   Mode          Type
------- --------------- ---- ------ ---- ---- ------ ------------- ----
3/1      ch1                          1        Down     En.         Stat
3/2      ch2                          1        Down     En.         Stat
3/3      ch3                          1        Down     En.         Stat
3/4      ch4                          1        Down     En.         Stat
3/5      ch5                          1        Down     En.         Stat
3/6      ch6                          1        Down     En.         Stat
3/7      ch7                          1        Down     En.         Stat
3/8      ch8                          1        Down     En.         Stat
3/9      ch9                          1        Down     En.         Stat
```

1.5. Theory of Operation

1.5.1. NAX ML2 Mechanism Event

A mechanism driver is called on the creation, update, and deletion of networks and ports. For every event, there are two methods that get called - one within the database transaction (method suffix of _precommit), one right afterward (method suffix of _postcommit).

Event	Action
Initialize	1. Read ml2_conf_nax.ini configuration file 2. Build up nax_info data structure: [device_ip,item_key, value]
create_network_precommit	Check network type. Only network type vlan is supported.
create_network_postcommit	Add tenant_id, network_id, vlan_id to database (nax_switch_network)
update_network_precommit	X
update_network_postcommit	X
delete_network_precommit	X
delete_network_postcommit	Delete tenant_id, network_id, vlan_id from database (nax_switch_network)
create_port_precommit	1. Check the connection status of switch. If the switch can be accessible, the ML2 driver will add the entries to nax_switch_vm database. 2. Add entry in database (nax_switch_vm): vm_id, hostname, tenant_id, network_id: provided by ML2 mechanism event. vm_switch_ip, switch_port: Use the hostname to search nax_info data structure (Retrieve two entries for MLAG topology). vlan_id : Use network_id to search nax_switch_network database.
create_port_postcommit	1. Use network_id and hostname to search nax_switch_vm database and get vm_switch_ip 2. Use vm_switch_ip, vlan_id and hostname to search nax_switch_vm database. If count >1, don’t care If count=1, Get switch_port and vlan_id from search entry. Connecting to NAX switch using SSH, create vlan_id and set switch_port to be a member of vlan_id

Event	Action
update_port_precommit	1. Check migration 2. Check update port
update_port_postcommit	1. Check migration 2. Check update port
delete_port_precommit	1. Check the connection status of switch. If the switch can be accessible, the ML2 driver will delete the entries to nax_switch_vm database.
delete_port_postcommit	1. Use network_id and hostname to search nax_switch_vm database and get vm_switch_ip 2. Use vm_switch_ip, vlan_id and hostname to search nax_switch_vm database If count >1, don't care. If count=1, Connecting to NAX switch using SSH, remove switch_port from vlan_id If count >1, don't care If count=1, Get vlan_id from search entry • Connecting to NAX switch using SSH • Delete vlan_id 3. Delete entry in database (nax_switch_vm) vm_id, hostname, tenant_id, network_id provided by ML2 mechanism event
delete_tenant	X

1.5.2. NAX ML2 Mechanism Database

nax_switch_network table: Create Network in the mechanism specific database table to store network information.

tenant_id	network_id	vlan_id	Vnid
String[36]	String[36]	Integer	Integer

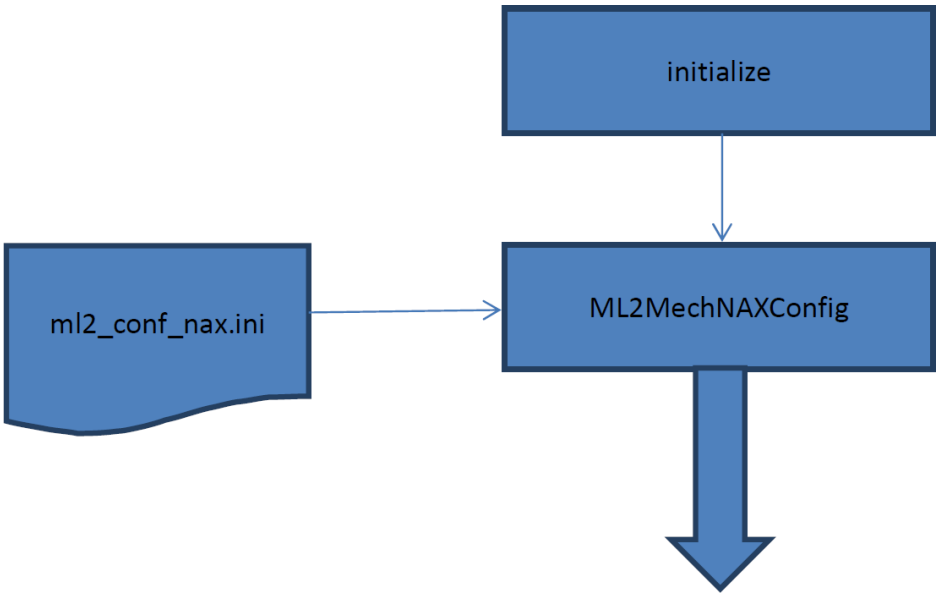
nax_switch_vm table: Create port in the mechanism specific database table to store vm information.

vm_id	hostname	switch_ip	switch_port	tenant_id	network_id	vlan_id	vnid
String[255]	String[255]	String[36]	String[36]	String[36]	String[36]	Integer	Integer

nax_switch_vxlan table: Create VXLAN tunnel specific database table to store VXLAN information.

src_swtch_ip	src_vxlan_ip	dst_swtch_ip	dst_vxlan_ip	vnid
String[36]	String[36]	String[36]	String[36]	Integer

1.5.3. Initialize Event



nax_info

switch_ip	item_key	value
172.16.0.123	controller	0/1
172.16.0.123	computer1	0/2
172.16.0.123	computer2	0/3
172.16.0.123	ssh_port	2222
172.16.0.123	username	test
172.16.0.123	password	12345678
172.16.0.123	vxlan_source	10.10.1.1
172.16.0.124	computer3	0/15
172.16.0.124	computer4	0/16
172.16.0.124	ssh_port	2222
172.16.0.124	username	test
172.16.0.124	password	12345678
172.16.0.124	vxlan_source	10.10.2.1

ml2_conf_nax.ini

```
[NAX_SWITCH:172.16.0.123]
```

```
controller=0/1
computer1=0/2
computer2=0/3
ssh_port=2222
username=test
password=12345678
vxlan_source=10.10.1.1
[NAX_SWITCH:172.16.0.124]
computer3=0/15
computer4=0/16
ssh_port=2222
username=test
password=12345678
vxlan_source=10.10.2.1
```