Migrating Cisco NDB OpenFlow to NXAPI Implementation

This chapter contains the following sections:

NDB Migration Overview

Starting with Cisco Nexus Data Broker, release 3.7, you can now migrate centralized NDB OpenFlow implementation to NXAPI implementation using the NDB migration tool. NDB migration occurs on the same virtual machine where the existing OpenFlow instance exists. The NDB migration process involves:

- Upgrading to NDB version 3.6 or later
- Exporting the device configuration in NDB 3.6
- NDB configuration cleanup
- Device conversion from OpenFlow to NXAPI by removing OpenFlow virtual service instances.
- Importing the NXAPI device configuration in NDB 3.6

NDB migration tool provides the following features:

- Single Touch Migration from OpenFlow to NXAPI devices.
- Supports NDB version from NDB 3.6.
- Supports all NDB Platform devices.
- Supports Atomic & Non-Atomic operations.
- Multiple Device Upgrade in a single Migration job.

NDB Migration Limitations

Follow these limitations and usage guidelines while migrating NDB from OpenFlow to NXAPI implementation:

- Port groups are not supported for NDB migration. If NDB has port groups, you need to manually reconfigure the port groups after migrating NDB to NXAPI.
- Port group description does not support special characters. Ensure that you remove all the special characters from port group description before starting the migration process.

Prerequisites for Migrating NDB

- You should have administrative access to mingrate NDB implementation from OpenFlow o NXAPI.
- You have following packages installed on the device:
  - Python (version 2.7)
  - Pip (version 10.0.1)
  - Open SSL
  - pexpect
  - YAML
  - Requests
  - ExScript
  - Configobj
  - paramika
  - Git

Note You need to install Python and Pip. For the rest of the packages, you can use the requirement.txt file with pip install command. For more information about package installation, see Installing Packages on Linux, on page 2.

Installing Packages on Linux

You can install required packages on Linux Ubuntu or Linux Redhat flavors:

- Installing Packages on Linux Ubuntu
- Installing Packages on Red Hat Linux, on page 3

Installing Packages on Linux Ubuntu

Complete these steps to install the following packages on Linux Ubuntu (version 10.0.1):

- Open SSL
- pexpect
- YAML
- Requests
- ExScript
- Configobj
- Paramiko

Step 1 Install Git using the **sudo** command.

Example:

```
sudo apt-get install git
```

Step 2 Install Python using the sudo command.

Example:

```
sudo apt-get install python2.7
```

Step 3 Install Pip using the sudo command.

Example:

```
Sudo apt-get install pip
```

You can also install a specific pip version using the **pip install pip==**<version> command.

Step 4 Update the `requirement.txt` file with the packages to install.

Example:

```
pexpect==4.6.0
pyyaml==3.12
Requests==2.18.4
ExScript==2.5.7
configobj==5.0.6
```

Step 5 Use the **pip install** command to install packages listed in the requirement.txt file.:

Example:

```
# pip install -r requirements.txt
```

Installing Packages on Red Hat Linux

Complete these steps to install the following packages on Red Hat Linux:

- Open SSL

- pexpect
- YAML
- Requests
- ExScript
- Configobj
- Paramiko

Step 1 Install Git using the **yum** command.

Example:

```
yum install git
```

Step 2 Install Python using the **sudo** command.

Example:

```
sudo yum install python27
```

Step 3 Install Pip using the **sudo** command.

Example:

```
Sudo yum install python-pip
```

Step 4 Install the Pur package using the pip install command, which is required if the older version of packages exist in the Red Had Linux.

Example:

```
pip install pur
```

Step 5 Update the `requirement.txt` file with the packages to install.

Example:

```
pexpect==4.6.0
pyyaml==3.12
Requests==2.18.4
ExScript==2.5.7
configobj==5.0.6
```

Step 6 Use the **pur** command to install packages listed in the requirement.txt file.:

Example:

```
# pur -r requirements.txt
```

Migrating Cisco NDB from OpenFlow to NXAPI

Complete the following steps to migrate NDB from OpenFlow to NXAPI.

Step 1 Download the migration script available at GIThub server (https://github.com/datacenter/nexus-data-broker). For example:

Example:

```
git clone http://ndb-build.cisco.com/gerrit/NDBMigration
```

The migration script is available in the `datacenter\nexus-data-broker` folder

Step 2 Open the input.yaml file and update the following fields:

Table 1:

Field Name	Description
NDB Server	
host_name/IP	Host Name or IP address of the server.
username	Username to log in to NDB Server.
password	Password to log into the NDB Server
ndb_gui_username	NDB Server GUI login username.
ndb_gui_password	NDB Server GUI login password.
old_path_ndb_build	Location of current NDB server xnc folder
new_path_ndb_build	Location where new NDB server xnc folder will be created after migrating to NXAPI.
Device Details	
host_name/IP	Host name or IP address of the switch.
username	Username to log in to the switch.
passoword	Password to log in to the switch.
mode	Switch mode for the switch after migration, ensure that it is configured to NXAPI.
tcam_ifacl	TCAM regions to create after device conversion to NXAPI.
tcam_mac-ifacl	TCAM regions to create after device conversion to NXAPI on MAC.
nxos	NXOS image to which device needs to be migrated. In case of Nexus 3000 series switches, if the current NXOS version is below u6, then first you need to upgrade the device to U6 and then to I46 or I47.

Step 3 Use the **python** command to run the migration script.

Example:

```
python NDBMigration.py
```

A unique jobid folder is created every time the migration script is run and contains three folders:

- Backup: Contains the old NDB zip file, export JSON file, import JSON file, and the state file. The state file contains detailed information about the status of every step involve in the migration process. Insert diagram for the state file screen shot.
- Log: Contains the migration log information
- Report: Contains information about the migration script result

Successful completion of migration process will result in NDB NXAPI implementation in the virtual machine. If the migration process fails, the resultant behavior depends on the revertFlat attribute configured in the input.yaml file.

- If revertFlag is set to 1, NDB and device configurations are reverted to old NDB version along with OF device configurations.
- If revertFlag is set to 0, the revert behavior depends on the stage where the failure occurs
  - Failure during NDB upgrade – NDB and device configurations are reverted to old NDB version along with OF device configurations.
  - Failure during NDB export – NDB and device configurations are reverted to old NDB version along with OF device configurations.
  - Failure during NDB clean up – NDB and device configurations are reverted to old NDB version along with OF device configurations.
  - Failure during device conversion – Migration script will continue to the next device and the state of the failed device is set to FAIL.
  - Failure during NDB import – Migration script will continue to the next device and the state of the failed device is set to FAIL.

Note You can rerun the migration script on failure to proceed the migration from the point of failure. Use the **python NDBMigration.py –rerun** failedjobid command to start the migration process from the point of failure. For example:

```
python NDBMigration.py –rerun  job.2018Aug07_04:49:39
```

Troubleshooting NDB Migration Issues

NDB migration script may fail during the migration process. You can look for the log file and migration report in the `jobid` folder that is created every time the migration script is run for troubleshooting information. The `jobid` folder contains three folders:

- Backup: Contains the old NDB zip file, export JSON file, import JSON file, and the state file. The `state` file contains detailed information about the status of every step involve in the migration process. Every step is represented by either of these three states:

- Pass
- Fail
- Skip

- Log: Contains the migration log information
- Report: Contains information about the migration script result.

NDB Proxy Issues

Migration process may fail due to proxy issues. To resolve any proxy issues, you need to unset the proxy.

Table 2: Proxy Issue Related Error Messages

Error Message	Command
`HTTPSConnectionPool(host='10.16.206.197', port=8443): Max retries exceeded with url: /monitor (Caused by ProxyError('Cannot connect to proxy.', error('Tunnel connection failed: 504 Gateway Timeout',)))`	`#unset http_proxy` `#unset https_proxy`

NDB Import Issues

NDB import process may fail on Cisco Nexus 3000 series switches. Check the switching mode for the Nexus 3000 switch using the **show system switch-mode** command. The switchport mode should be n3k.

```
N3K-123# show system switch-mode
       system switch-mode n3k
```

Note The Cisco Nexus N3K-3132Q-40GX and N3K-3172PQ-10GE switches support both n3k and n9k switching mode. For NDB migration, ensure that the switching mode is set to n3k.

Reverting to Previous Configuration in case of Script Failure

Follow these steps to revert to the previous configuration in case of migration script failure:

Step 1 Remove all the interface configurations from all the devices using the no feature command.

Example:

```
(config)# no feature openflow
```

Step 2 Load previous NXOS image that was loaded before running migration script on all the devices using the **install all system** command or **install all nxos** command.

Example:

```
install all system  n3000-uk9.6.0.2.U6.4a.bin kickstart n3000-uk9-kickstart.6.0.2.U6.4a.bin

install all  nxos nxos.7.0.3.I4.6.bin
```

Step 3 Install and activate openflow ova on all the devices using the **virtual-service install** command.

Example:

```
virtual-service install name ofa package bootflash:<openflow-ova>
```

Step 4 Configure open flow configuration on all the devices. Old configuration is available in the `Migration_backup` file available on each device.

Example:

```
#openflow
#switch 1
#pipeline 201
#probe-interval 5
#controller ipv4 10.16.206.136 port 6653 vrf   management security none
#of-port interface Ethernet1/1-54
```

Step 5 Configure tcam region which was present before running migration script.

Example:

```
#sh file Migration_backup_2019Feb03_01:48:52 | grep hardware
hardware access-list tcam region ifacl 256
hardware access-list tcam region openflow 256

// Current tcam configuration:
#sh run |grep  hardware
hardware access-list tcam region ifacl 256

// Command to configure Tcam region:
#config t
#hardware access-list tcam region openflow 256
```

Step 6 Stop and start the NDB to apply the new configuration.

Example:

```
./runxnc -stop
./runxnc -start
```

FAQs - NDB Migration

Q. Where should I run NDB migration script?

A. You can run the migration script from a VM where NDB is running or from a new VM (Ubuntu/Redhat). Cisco recommends that you run the migration script from a new VM.

Q. What happens if a user already has python packages with older version in Redhat?

A. You need to install the Pur package using the **pip** command and then use the **pur** command to install the packages listed in the `requirement.txt`.

Q. How to check Python version?

A. Use the **python -V** commamnd to check the current Python version..

Q. How to check Pip version?

A. Use the **pip -V** commamnd to check the current Pip version..

Q. How to check Pip packages?

A. Use the **pip list** commamnd to check the Pip packages installed along with the version.