
Dell EMC Networking Napalm Integration Documentation

Release 1.0

Dell EMC Networking Team

Apr 22, 2020

Table of Contents

1	Introduction	1
1.1	NAPALM	1
1.2	Dell EMC Networking NAPALM integration	1
2	Installation	3
2.1	Install NAPALM	3
2.2	Test your setup	3
3	Dell EMC Networking OS10 NAPALM APIs	5
3.1	Implemented APIs	5
3.2	Missing APIs	6
4	Provision CLOS fabric using Dell EMC Networking NAPALM APIs example	7
4.1	Step 1	7
4.2	Step 2	7
5	Install or upgrade devices running Dell EMC Networking OS10 using NAPALM	11
5.1	Step 1	11
5.2	Step 2	11
5.3	Step 3	11
6	Release notes	13
6.1	Release 1.0.0	13
7	Support	15
7.1	Contact	15
8	License	17

This information explains the usage of NAPALM for deploying the configuration into Dell EMC Networking OS10 switches.

1.1 NAPALM

Network automation and programmability abstraction layer with multivendor support (NAPALM) is a Python library that implements a set of functions to interact with different network device operating systems using a unified API.

NAPALM tries to provide a common interface and mechanisms to push configuration and retrieve state data from network devices. This method is very useful in combination with tools such as Ansible/SaltStack, which allow you to manage a set of devices independent of their network OS.

See [NAPALM](#) for complete information.

1.2 Dell EMC Networking NAPALM integration

Dell EMC Networking OS10 switches can be managed and automated using NAPALM unified API to manipulate configurations or to retrieve data.

This information contains instructions to install `napalm_dellos10`. Included is information on the setup environment for managing Dell EMC Networking OS10 switches using Python.

2.1 Install NAPALM

Install the Dell Networking OS10 NAPALM driver:

```
sudo apt-get install libffi-dev libssl-dev python-dev python-cffi libxslt1-dev python-  
↳pip  
pip install --upgrade pip  
sudo pip install --upgrade cffi  
sudo pip install napalm-dellos10
```

2.2 Test your setup

Open the Python interpreter terminal using “python”:

```
>>> from napalm.base import get_network_driver  
>>> d = get_network_driver('dellos10')  
>>> e = d('<HOSTNAME/IP_ADDRESS>', '<USERNAME>', '<PASSWORD>', optional_args={'global_  
↳delay_factor': 3})  
>>> e.open()  
>>> e.get_facts()  
>>> e.close()
```

Dell EMC Networking OS10 NAPALM APIs

NAPALM connects to the devices to manipulate configurations, or to retrieve data using APIs. See [NAPALM](#) for more information about each API.

3.1 Implemented APIs

- `load_merge_candidate`
- `compare_config`
- `commit_config`
- `discard_config`
- `get_facts`
- `get_config`
- `ping`
- `get_snmp_information`
- `get_interfaces`
- `get_route_to`
- `get_interfaces_ip`
- `get_bgp_config`
- `get_bgp_neighbors_detail`
- `get_bgp_neighbors`
- `get_lldp_neighbors`
- `get_lldp_neighbors_detail`
- `get_lldp_neighbors_interface_detail`

- `install_switch_image`
- `upgrade_switch_image`
- `get_image_status`

3.2 Missing APIs

- `rollback` is not supported
- `load_replace_candidate` not supported

Provision CLOS fabric using Dell EMC Networking NAPALM APIs example

This information describes how to use NAPALM to build a CLOS fabric with Dell EMC Networking OS10 switches. The sample topology is a two-tier CLOS fabric with two spines and four leafs connected as mesh. EBGp is running between the two tiers.

All switches in spine have the same AS number, and each leaf switch has a unique AS number. All AS number used are private. For application load-balancing purposes, the same prefix is advertised from multiple leaf switches and uses *BGP multipath relax* feature.

4.1 Step 1

Download the CLOS fabric configuration:

```
spine1 spine2 leaf1 leaf2 leaf3 leaf4
```

Copy all the downloaded configuration files in the the `/root/napalm/clos_configuration` directory.

4.2 Step 2

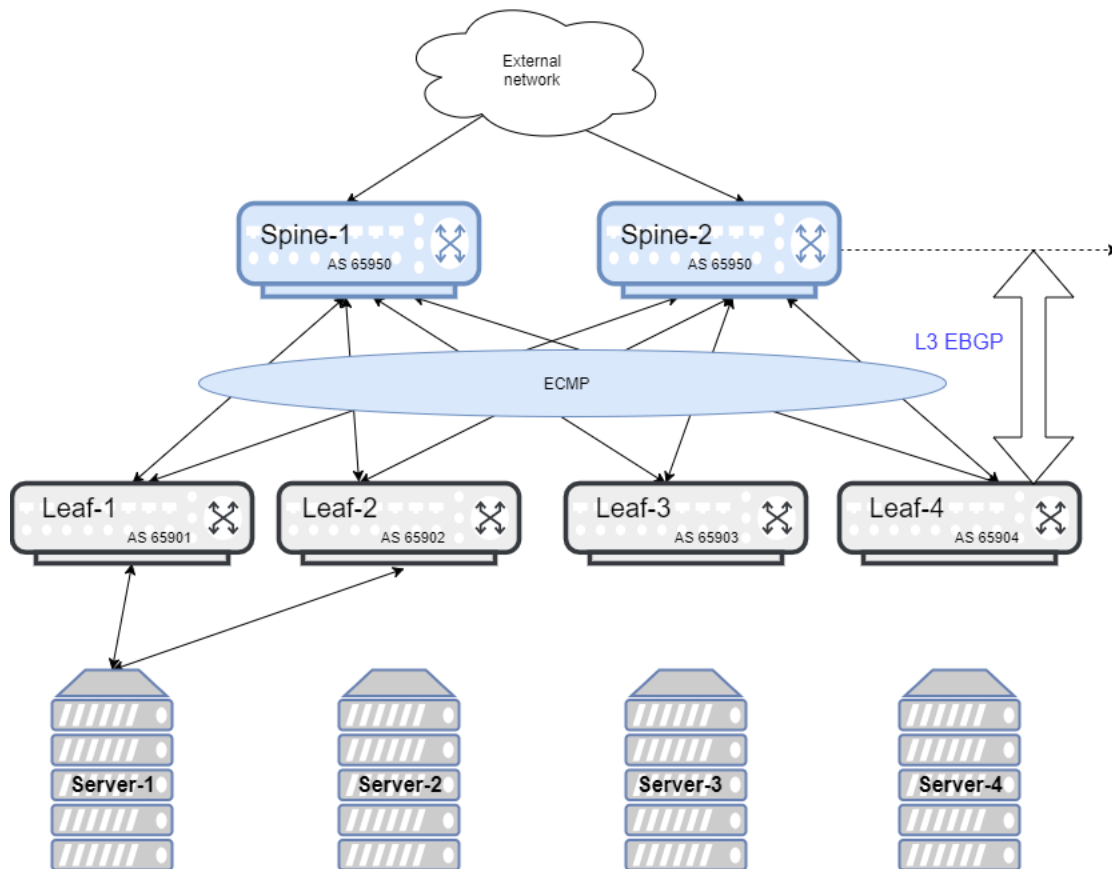
Load/merge the configuration for all the switches in the fabric.

Create a file called `load_merge.py` in a folder:

```
from napalm_base import get_network_driver

configs_dir = "/root/napalm/clos_configuration/" # add your configuration base_
↪directory path
commit_config = False # If you are happy with the changes, mark this as True
discard_config = True # Mark as True, To delete the uploaded configuration
```

(continues on next page)



(continued from previous page)

```

# Add your global switch credentials
device_creds = {
    "username": "admin",
    "password": "admin",
    "optional_args": {
        "global_delay_factor": 3
    }
}

# Add your Switch specific details
fabric_details = {
    "spine1": {
        "hostname": "1.1.1.1",
        "config_file": "spine1.cfg"
    },
    "spine2": {
        "hostname": "1.1.1.2",
        "config_file": "spine2.cfg"
    },
    "leaf1": {
        "hostname": "1.1.1.3",
        "config_file": "leaf1.cfg"
    },
    "leaf2": {
        "hostname": "1.1.1.4",
        "config_file": "leaf2.cfg"
    },
    "leaf3": {
        "hostname": "1.1.1.5",
        "config_file": "leaf3.cfg"
    },
    "leaf4": {
        "hostname": "1.1.1.6",
        "config_file": "leaf4.cfg"
    }
}

driver = get_network_driver('dellos10')

for key, value in fabric_details.iteritems():
    device_ip = value.get("hostname")
    config_file = configs_dir + value.get("config_file")

    device = driver(device_ip, **device_creds)
    device.open()
    print("{}: Started loading configuration ".format(key))
    device.load_merge_candidate(filename=config_file)
    print("Configuration difference to be applied on switch : {}".format(key))
    print(device.compare_config())

    if commit_config:
        device.commit_config()
        print("Configuration successfully loaded into switch: {}".format(key))

    if discard_config:
        device.discard_config()
        print("{}: Configuration discarded successfully".format(key))

```

(continues on next page)

(continued from previous page)

```
device.close()
```

Install or upgrade devices running Dell EMC Networking OS10 using NAPALM

This information explains how to use NAPALM to install or upgrade the software image on a device running Dell EMC Networking OS10.

5.1 Step 1

Make sure switches are accessible through NAPALM. Open the Python interpreter terminal using “python” match the example:

```
>>> from napalm_base import get_network_driver
>>> d = get_network_driver('dellos10')
>>> e = d('<HOSTNAME/IP_ADDRESS>', '<USERNAME>', '<PASSWORD>', optional_args={'global_
↳delay_factor': 3})
>>> e.open()
>>> e.get_facts()
>>> e.close()
```

5.2 Step 2

Upload the image to install to any TFTP/FTP/SCP/SFTP/HTTP server.

5.3 Step 3

Install or upgrade the image on the switch. Example image file path is /root/PKGS_OS10-Enterprise-10.4.0E.R2.30-installer-x86_64.bin.

SCP server details are below, Server IP: 1.1.1.1 credentials: username: my_username, password: my_password image file path: /root/PKGS_OS10-Enterprise-10.4.0E.R2.30-installer-x86_64.bin

Example image_file_url:

```
image_file_url="scp://my_username:my_password@1.1.1.1/root/
PKGS_OS10-Enterprise-10.4.0E.R2.30-installer-x86_64.bin"
```

To install the switch image, create a file called `image_upgrade.py` in a folder:

```
from napalm_base import get_network_driver

# Add your global switch credentials
device_creds = {
    "username": "admin",
    "password": "admin",
    "optional_args": {
        "global_delay_factor": 3
    }
}

image_file_url="scp://my_username:my_password@1.1.1.1/root/PKGS_OS10-Enterprise-10.4.
↪0E.R2.30-installer-x86_64.bin"

driver = get_network_driver('dellos10')

device = driver("1.1.1.1", **device_creds)
device.open()
print("Switch image install started...")
device.install_switch_image(image_file_url=image_file_url)
# get the switch status using below
device.get_image_status()

device.close()
```

Note: image_file_url format for TFTP/FTP/SCP/SFTP/HTTP server are below

- ftp: Install from remote FTP server (ftp://userid:passwd@hostip/filepath)
 - http: Install from remote HTTP (http://hostip/filepath)
 - image: Install from image directory (image://filepath)
 - scp: Install from remote SCP server (scp://userid:passwd@hostip/filepath)
 - sftp: Install from remote SFTP server (sftp://userid:passwd@hostip/filepath)
 - tftp: Install from remote TFTP server (tftp://hostip/filepath)
 - usb: Install from USB directory (usb://filepath)
-

This information contains the release notes for Dell EMC Networking OS10 support.

6.1 Release 1.0.0

Initial NAPALM support for Dell EMC Networking OS10 switches.

New modules:

- `load_merge_candidate`
- `compare_config`
- `commit_config`
- `discard_config`
- `get_facts`
- `get_config`
- `ping`
- `get_snmp_information`
- `get_interfaces`
- `get_route_to`
- `get_interfaces_ip`
- `get_bgp_config`
- `get_bgp_neighbors_detail`
- `get_bgp_neighbors`
- `get_lldp_neighbors`
- `get_lldp_neighbors_detail`

- `get_lldp_neighbors_interface_detail`
- `install_switch_image`
- `upgrade_switch_image`
- `get_image_status`

You can submit issues for Dell EMC Networking OS10 Napalm driver at [NAPALM dellos10 Github Issues](#).

7.1 Contact

You can send general comments and feedback to networking_devops_tools@dell.com.

CHAPTER 8

License

(c) 2018 Dell Inc. or its subsidiaries. All Rights Reserved.

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the [License](#).

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.