
Dell EMC Networking Saltstack Integration Documentation

Release 1.0

Dell EMC Networking Team

Sep 07, 2018

Table of Contents

1	Introduction	1
1.1	Salt	1
1.2	Dell EMC Networking Salt integration	1
2	Installation	3
2.1	Install Salt	3
2.2	Install NAPALM	3
2.3	Configure salt-proxy	4
2.4	Configure salt-minion	4
2.5	Configure connection with device	4
2.6	Start Salt Services	5
3	Dell EMC Networking OS10 SaltStack modules	7
3.1	Network module	7
3.2	BGP module	7
3.3	Route module	8
3.4	SNMP module	8
4	Provision CLOS fabric using Dell EMC Networking Saltstack modules example	9
4.1	Step 1	9
4.2	Step 2	9
4.3	Step 3	11
4.4	Step 4	11
4.5	Step 5	11
4.6	Step 6	11
5	Install or upgrade devices running Dell EMC Networking OS10 using SaltStack	13
5.1	Step 1	13
5.2	Step 2	13
5.3	Step 3	13
5.4	Step 4	14
6	Release notes	15
6.1	Release 1.0.0	15
7	Support	17
7.1	Contact	17

This information explains the usage of SaltStack (Salt) for deploying the configuration into Dell EMC Networking OS10 switches.

1.1 Salt

Salt is a configuration management system, capable of maintaining remote nodes in defined states (for example, ensuring that a specific configuration is maintained in the switch).

Salt and NAPALM together represent a vendor-agnostic way of a configuration management system (see [more information about saltstack](#)).

1.2 Dell EMC Networking Salt integration

Dell EMC Networking OS10 switches can be managed and automated using Salt in a vendor-neutral way using the `napalm-proxy`.

This information contains instructions to install SaltStack (Salt) with the [napalm_dellos10](#) proxy. Included are instructions for the setup environment for managing Dell EMC Networking OS10 switches

2.1 Install Salt

Installation of Salt contains three components:

- salt-master
- salt-minion
- salt-proxy

The simplest way to install Salt is via [salt bootstrap](#).

```
wget -O bootstrap-salt.sh https://bootstrap.saltstack.com/develop
sudo sh bootstrap-salt.sh -M
```

See the [platform-specific instructions](#) from the official Saltstack documentation for more information. Be aware to install the master distribution from the PPA repo, as the local server will run as Master and control the devices as proxy-minions.

See the [CentOS documentation](#) for more information.

2.2 Install NAPALM

Install the Dell EMC Networking OS10 NAPALM driver:

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

(continues on next page)

(continued from previous page)

```
sudo pip install --upgrade cffi
sudo pip install napalm-dellos10
```

You can also install NAPALM using `napalm-install` Saltstack formula. See `napalm-install-formula` for a more detailed usage example.

2.3 Configure salt-proxy

The salt-proxy configuration is shown, and the default location of the salt-proxy configuration file is `/etc/salt/proxy`.

```
master: localhost
multiprocessing: false # turn off multiprocessing
mine_enabled: true # not required, but nice to have
pki_dir: /etc/salt/pki/proxy # not required - this separates the proxy keys into a
↳different directory
```

2.4 Configure salt-minion

The salt-minion configuration is shown, and the default location of the salt-minion configuration file is `/etc/salt/minion`.

```
master: localhost
```

2.5 Configure connection with device

In salt-napalm, all switch-specific information such as switch IP address and credentials are configured in the pillar file.

2.5.1 Step 1

The default pillar data file location is `/srv/pillar`. You must create this directory as it will not be available by default (you can change the location later). See `pillar-roots` for complete information.

```
mkdir -p /srv/pillar
```

2.5.2 Step 2

Create a `top.sls` file in that directory, which tells the salt-master which minions receive which pillar.

Create and edit the `/srv/pillar/top.sls` file and match the example:

```
base: # Default value, configurable at /etc/salt/master
  LEAF_1: # ``DEVICE_ID``, used to interact with the device, from Salt CLI
    - leaf_1_pillar # ``DEVICE_SLS_FILENAME``, Name of the file containing the
↳specifications of the device
```

(continues on next page)

(continued from previous page)

```
LEAF_2:
- leaf_2_pillar
```

Note:

DEVICE_ID, given in the above configuration shall be used in following places,

1. Run the salt-proxy `salt-proxy --proxyid=[DEVICE_ID] -l debug`
2. Connect to device using Salt CLI `salt 'LEAF_1' test.ping`

2.5.3 Step 2a

Create a `DEVICE_SLS_FILENAME` file (mentioned in Step 2) in `/srv/pillar/leaf_1_pillar.sls`:

```
proxy:
  proxytype: napalm
  driver: dellos10
  host: 192.168.128.128
  username: my_username
  passwd: my_password
  optional_args:
    global_delay_factor: 3 # This is optional value, increase value in case device_
    ↳response is slow
```

The `passwd` is in plain-text and is used for encrypting the password (see [salt-renderers.gpg](#)).

2.6 Start Salt Services

```
sudo systemctl start salt-master
sudo systemctl restart salt-minion
```

2.6.1 Start proxy-minion for device

Test the proxy-minion:

```
sudo salt-proxy --proxyid=[DEVICE_ID] -l debug
```

On the first connection attempt, the minion cannot talk and is stuck with an error message:

```
[ERROR   ] The Salt Master has cached the public key for this node, this salt minion_
↳will wait for 10 seconds before attempting to re-authenticate
[INFO    ] Waiting 10 seconds before retry.
```

This is normal and is due to the salt key from the minion not being accepted by the master. Quit the minion with `CTRL + C` and run `sudo salt-key`.

```
[root@master ~]# salt-key -L
Unaccepted Keys:
LEAF_1
```

(continues on next page)

(continued from previous page)

```
LEAF_2
Accepted Keys:
```

This example shows that the salt-master is aware of four salt-minions, but none of the keys has been accepted. To accept the keys and allow the Minions to be controlled by the salt-master, use the `salt-key` command:

```
[root@master ~]# salt-key -A
[root@master ~]# salt-key -L
Unaccepted Keys:
Accepted Keys:
LEAF_1
LEAF_2
```

The `salt-key` command allows for signing keys individually or in bulk. The example shows using `-A` bulk-accepts all pending keys. To accept keys individually, use the lowercase of the same option (`-a`).

Start the proxy again.

2.6.2 Test your configuration

Once the key has been accepted, restart the proxy in debug mode and start a separate terminal session:

```
sudo salt 'LEAF_1' test.ping
```

To test for all leaf devices:

```
sudo salt 'LEAF_*' test.ping
```

It should return True if there are no problems. If everything checks out, hit CTRL + C and restart `salt-proxy` as a daemon.

```
sudo salt-proxy --proxyid=[DEVICE_ID] -d
```

Example:

```
sudo salt-proxy --proxyid=LEAF_1 -d
sudo salt-proxy --proxyid=LEAF_2 -d
```

Finally, sync your packages:

```
sudo salt '*' saltutil.sync_all
```

Dell EMC Networking OS10 SaltStack modules

3.1 Network module

The `network module` facilitates the basic methods for interaction with the network device through the virtual proxy ‘napalm’.

- `cli`: returns a dictionary with the raw output of all commands passed as arguments
- `commit`: commits the NAPALM’s candidate configuration changes made on the network device
- `compare_config`: returns the difference between the running configuration and the candidate configuration
- `config`: returns the whole configuration of the network device
- `facts`: returns characteristics of the network device
- `interfaces`: returns details of the interfaces on the device
- `lldp`: returns a detailed view of the LLDP neighbors
- `load_config`: applies configuration changes on the device; it can be loaded from a file or from inline string; if you send both a filename and a string containing the configuration, the file has higher precedence
- `load_template`: renders a configuration template (default: Jinja) and loads the result on the device
- `ping`: executes a ping on the network device and returns a dictionary as a result

3.2 BGP module

The `bgp module` manages BGP configuration on network devices and provides statistics.

- `config`: returns the BGP configuration on the device
- `neighbors`: returns details regarding the BGP sessions configured on the network device

3.3 Route module

The `route module` retrieves route details from network devices.

- `show`: returns all details for a certain route learned via a specific protocol; if the protocol is not specified, all possible routes display

3.4 SNMP module

The `SNMP module` Manages SNMP on network devices.

- `config`: returns the SNMP configuration

Provision CLOS fabric using Dell EMC Networking Saltstack modules example

This example describes how to use SaltStack 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

Create a salt-proxy for the OS10 switches (see [Configure the connection with a device](#)). Switches used in the topology are named as spine1, spine2, leaf1, leaf2, leaf3, leaf4.

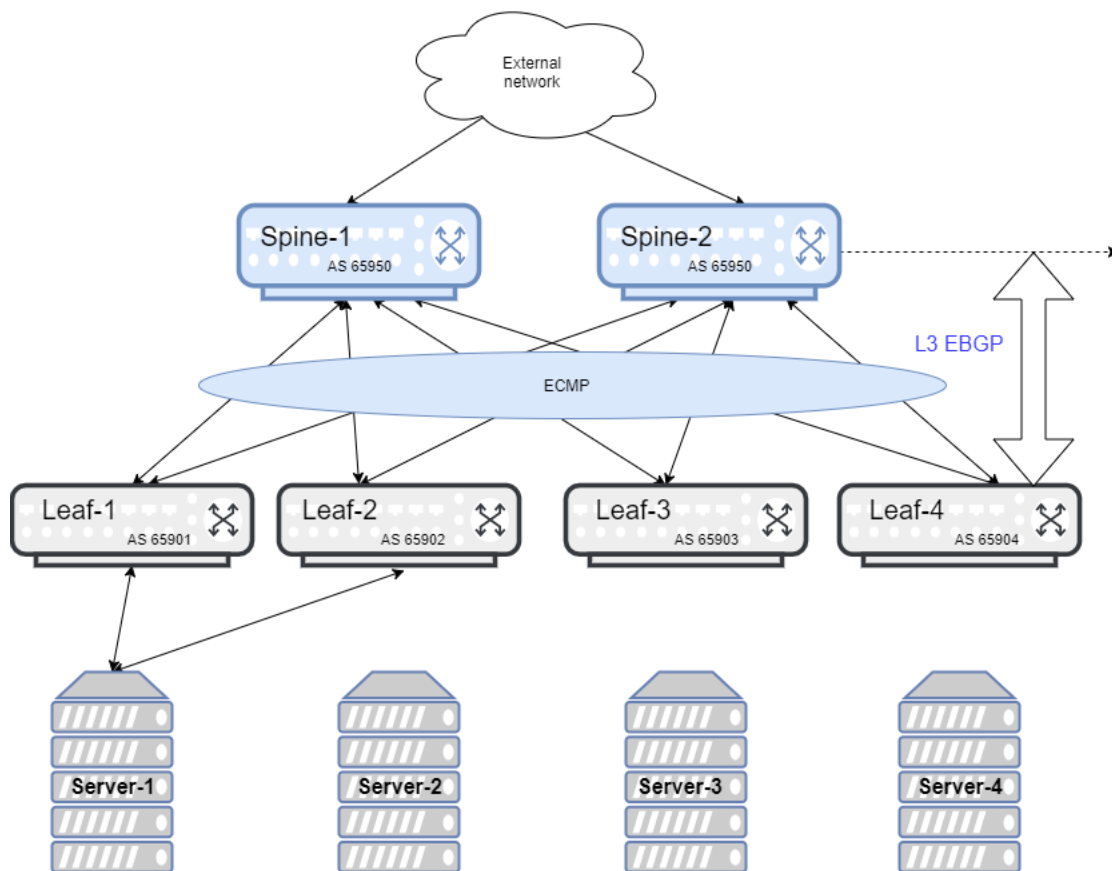
In a new terminal session, test your switch to confirm it is configured correctly:

```
sudo salt '*' test.ping
```

4.2 Step 2

Edit the salt-master configuration and append the `file_roots` value. The default salt-master configuration location is `/etc/salt/master`:

```
file_roots:
  base:
    - /srv/salt/states
    - /srv/salt/pillar
    - /srv/salt/states/configuration
```



4.3 Step 3

Download the CLOS fabric configuration.

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

Copy all the downloaded configuration files in the salt-server directory `/srv/salt/states/configuration`.

4.4 Step 4

Edit the pillar configuration and append for each switch:

```
proxy:
  proxytype: napalm
  driver: dellos10
  host: 192.168.128.128
  username: my_username
  passwd: my_password
  optional_args:
    global_delay_factor: 3 # This is optional value, increase value in case device_
    ↳response is slow
  config_file: leaf1.cfg
```

Repeat for all switches in the pillar data.

4.5 Step 5

Create a states file in `/srv/salt/states` named `clos_load_config.sls`:

```
clos_config:
  netconfig.managed:
    - template_name: salt://{{ pillar.get('config_file') }}
```

4.6 Step 6

Push the CLOS configuration into all switches:

```
salt "*" state.sls clos_load_config
```

Install or upgrade devices running Dell EMC Networking OS10 using SaltStack

This example describes how to use SaltStack to install or upgrade the software image on a device running Dell EMC Networking OS10.

5.1 Step 1

Create a salt-proxy for the OS10 switch you want to upgrade or install an image (see *Configure the connection with a device*).

In a new terminal session, test that your switch is configured correctly:

```
sudo salt '<DEVICE_ID>' test.ping
```

5.2 Step 2

Upload the image that you need to install to an TFTP/FTP/SCP/SFTP/HTTP server.

5.3 Step 3

Install or upgrade the image on the switch. The example image file path should look like image file path: /root/PKGS_OS10-Enterprise-10.4.0E.R2.30-installer-x86_64.bin.

SCP server details: 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

The image_file_url should look like 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".

Upgrade the switch image:

```
sudo salt '<DEVICE_ID>' napalm.call upgrade_switch_image 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"
```

Install the switch image:

```
sudo salt '<DEVICE_ID>' napalm.call install_switch_image 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"
```

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

- 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)
-

5.4 Step 4

Find the status of switch image install or upgrade:

```
sudo salt '<DEVICE_ID>' napalm.call get_image_status
```

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

6.1 Release 1.0.0

Initial SaltStack support for Dell EMC Networking OS10 switches.

New modules:

- `napalm_network`
- `napalm_bgp`
- `napalm_snmp`
- `napalm_route`

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

3. 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.