OcNOS®

Open Compute Network Operating System Version 6.4.2

Ansible Guide

December 2023

IP Infusion Inc.
3965 Freedom Circle, Suite 200
Santa Clara, CA 95054
+1 408-400-1900
http://www.ipinfusion.com/

For support, questions, or comments via E-mail, contact:
support@ipinfusion.com

Contents

CHAPTER 1 Getting Started

Overview

This guide demonstrates how Ansible can be used to manage OcNOS devices using a common generic framework of platform agnostic Ansible networking modules.

This guide shows:

- Managing OcNOS devices through Ansible playbooks
- Sample configuration jinja2 templates for protocols
- Limitations

Install from Ansible Galaxy

The OcNOS Ansible module is installed from Ansible Galaxy.

1. Ensure the installed Ansible version is 2.9 or later.

Here is the example with Ansible version 2.9.6:

```
# ansible --version
ansible 2.9.6
config file = /etc/ansible/ansible.cfg
configured module search path =
[u'/home/ <yourhome> /.ansible/plugins/modules',
u'/usr/share/ansible/plugins/modules']
ansible python module location = /usr/lib/python2.7/site-packages/ansible
executable location = /usr/bin/ansible
python version = 2.7.5 (default, Aug 7 2019, 00:51:29) [GCC 4.8.5 20150623
(Red Hat 4.8.5-39)]
```

Here is the example with Ansible version 2.15.2:

```
# ansible --version
ansible [core 2.15.2]
  config file = None
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/
ansible/plugins/modules']
  ansible python module location = /root/ansible-8.1.0/lib/python3.9/site-packages/
ansible
  ansible collection location = /root/.ansible/collections:/usr/share/ansible/
collections
  executable location = /root/ansible-8.1.0/bin/ansible
  python version = 3.9.17 (main, Jun  9 2023, 02:31:12) [GCC 10.3.1 20211027] (/root/
ansible-8.1.0/bin/python)
  jinja version = 3.1.2
  libyaml = False
```

2. You might also need to install an SSH plugin such as Paramiko or Ansible-Pylibssh.

```
$ pip install paramiko
If your ansible.netcommon module version is 1.1.0 or later, libssh for ssh channel can
be used.
$ pip install ansible-pylibssh
```

3. Install from Galaxy:

```
$ ansible-galaxy collection install ipinfusion.ocnos
```

4. When the standalone package is delivered, use the following command to install it on your system:

```
$ ansible-galaxy collection install ipinfusion-ocnos-x.x.x.tar.gz
```

Set up Ansible Files

```
$ cat group_vars/ocnos.yml
ansible_connection: network_cli
ansible_network_os: ipinfusion.ocnos.ocnos
ansible_become: yes
ansible_become_method: enable
ansible_ssh_user: ocnos
ansible_ssh_pass: ocnos
```

Note: The following inventory file is an example. Change the address and name for your site.

```
$ cat inventory/inventory.ini
[ocnosvm]
OcNOS-VM1 ansible_host=192.168.122.180 interface1=eth2
[ocnossw]
OcNOS-SW1 ansible_host=10.5.178.3 interface1=xe1/2
[ocnos:children]
ocnosvm
ocnossw
```

Modules

The `ipinfusion.ocnos.ocnos_xxx` is a prefix of OcNOS Ansible methods.xxx. Currently, the following methods are supported:

- `ipinfusion.ocnos.ocnos_fact`
- `ipinfusion.ocnos.ocnos_command`
- `ipinfusion.ocnos.ocnos_config`
- `ipinfusion.ocnos.ocnos_ping`
- `ipinfusion.ocnos.ocnos_bgp_facts`
- `ipinfusion.ocnos.ocnos_isis_facts`

For platform-agnostic modules like `gather_facts`, `cli_command`,`cli_config`, and `net_ping`, you do not need to change the module name since the platform is specified by `ansible_network_os`.

ipinfusion.ocnos.ocnos_facts

ocnos_facts collects facts from devices running OcNOS. Its result will be returned via the `ansible_net_xxx` variable.

Sample Playbook

```
---
- hosts: ocnos

  tasks:
  - name: Test OcNOS Facts
    ipinfusion.ocnos.ocnos_facts:
      gather_subset: all
    register: result

  - name: Show Facts
    debug:
      msg: The version is {{ ansible_net_version }}. HW model is {{ ansible_net_model
}}, its serial is {{ ansible_net_serialnum }}
```

Sample Output

```
$ ansible-playbook -i inventory/inventory.ini fact-playbook.yml -l OcNOS-SW1

PLAY [ocnos] ***************************************************************

TASK [Gathering Facts] *****************************************************
ok: [OcNOS-SW1]

TASK [Test OcNOS Facts] ****************************************************
ok: [OcNOS-SW1]

TASK [Show Facts] **********************************************************
ok: [OcNOS-SW1] => {
    "msg": "The version is DELL_S6000-ON-OcNOS-1.3.8.44a-DC_IPBASE-S0-P0. HW model is
DELL S6000-ON, its serial is CN07VJDK282985730184"
}

PLAY RECAP *****************************************************************
OcNOS-SW1                  : ok=3    changed=0    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0
```

Return Values

Table 1-1: Return values

Key	Returned	Description
`ansible_net_all_ipv4_addresses list`	when interfaces are configured	All IPv4 addresses configured on the device
`ansible_net_all_ipv6_addresses list`	when interfaces are configured	All IPv6 addresses configured on the device
`ansible_net_config string`	when config is configured	The current active config from the device
`ansible_net_gather_subset list`	always	The list of fact subsets collected from the device
`ansible_net_hostname string`	always	The configured hostname of the device
`ansible_net_image string`	always	The image file the device is running
`ansible_net_interfaces dict`	when interfaces are configured	A hash of all interfaces running on the system
`ansible_net_memfree_mb int`	when hardware is configured	The available free memory on the remote device in Mb
`ansible_net_memtotal_mb int`	when hardware is configured	The total memory on the remote device in Mb
`ansible_net_model string`	always	The model name returned from the device
`ansible_net_neighbors dict`	when interfaces is configured	The list of LLDP neighbors from the remote device
`ansible_net_serialnum string`	always	The serial number of the remote device
`ansible_net_version string`	always	The operating system version running on the remote device

The parameters are not supported:

- `ansible_net_filesystem`
- `ansible_net_api`

The example below shows the parameters in the table and the playbook to get them.

playbook: fact-all-playbook.yml

```
---
```

```
- hosts: ocnos

  tasks:
  - name: Test OcNOS Facts
    ipinfusion.ocnos.ocnos_facts:
      gather_subset: all
    register: result

  - debug: var=ansible_net_all_ipv4_addresses
  - debug: var=ansible_net_all_ipv6_addresses
  - debug: var=ansible_net_gather_subset
  - debug: var=ansible_net_hostname
  - debug: var=ansible_net_image
  - debug: var=ansible_net_interfaces
  - debug: var=ansible_net_memfree_mb
  - debug: var=ansible_net_memtotal_mb
  - debug: var=ansible_net_model
  - debug: var=ansible_net_neighbors
  - debug: var=ansible_net_serialnum
  - debug: var=ansible_net_version
```

Operation

```
$ ansible-playbook -i inventory/inventory.ini fact-all-playbook.yml -l OcNOS-SW1

PLAY [ocnos] ***********************************************************************

TASK [Gathering Facts] *************************************************************
ok: [OcNOS-SW1]

TASK [Test OcNOS Facts]*************************************************************
ok: [OcNOS-SW1]

TASK [debug]************************************************************************
ok: [OcNOS-SW1] => {
    "ansible_net_all_ipv4_addresses": [
        "127.0.0.1",
        "127.0.0.1",
        "10.5.178.3"
    ]
}

TASK [debug]************************************************************************
ok: [OcNOS-SW1] => {
    "ansible_net_all_ipv6_addresses": [
        "fe80::eef4:bbff:fe3e:c0ec",
        "fe80::eef4:bbff:fe3e:c0ec",
        "::1",
        "::1",
        "fe80::eef4:bbff:fefe:2beb"
```

```
    ]
}

TASK [debug]*******************************************************************
ok: [OcNOS-SW1] => {
    "ansible_net_gather_subset": [
        "hardware",
        "default",
        "interfaces",
        "config"
    ]
}

TASK [debug]*******************************************************************
ok: [OcNOS-SW1] => {
    "ansible_net_hostname": "OcNOS-SW1-1"
}

TASK [debug]*******************************************************************
ok: [OcNOS-SW1] => {
    "ansible_net_image": "DELL_S6000_ON-OcNOS-1.3.8.44a-DC_IPBASE-S0-P0-installer"
}

TASK [debug]*******************************************************************
ok: [OcNOS-SW1] => {
    "ansible_net_interfaces": {
        "eth0": {
            "bandwidth": "1g(auto)",
            "description": null,
            "duplex": "full",
            "ipv4": {
                "address": "10.5.178.3",
                "masklen": "22"
            },
            "ipv6": {
                "address": "fe80::eef4:bbff:fefe:2beb",
                "masklen": "64"
            },
            "lineprotocol": "up",
            "macaddress": "ecf4.bbfe.2beb",
            "mediatype": "METH",
            "mtu": "1500"
        },
        "lo": {
            "bandwidth": null,
            "description": null,
            "duplex": null,
            "ipv4": {
                "address": "127.0.0.1",
                "masklen": "8"
```

```
        },
        "ipv6": {
            "address": "::1",
            "masklen": "128"
        },
        "lineprotocol": "up",
        "macaddress": null,
        "mediatype": "LB",
        "mtu": null
    },
    "lo.management": {
        "bandwidth": null,
        "description": null,
        "duplex": null,
        "ipv4": {
            "address": "127.0.0.1",
            "masklen": "8"
        },
        "ipv6": {
            "address": "::1",
            "masklen": "128"
        },
        "lineprotocol": "up",
        "macaddress": null,
        "mediatype": "LB",
        "mtu": null
    },
    "vlan1.1": {
        "bandwidth": null,
        "description": null,
        "duplex": null,
        "ipv4": null,
        "ipv6": null,
        "lineprotocol": "down",
        "macaddress": "ecf4.bb3e.c0ec",
        "mediatype": "SVI",
        "mtu": null
    },
    "vlan1.10": {
        "bandwidth": null,
        "description": null,
        "duplex": null,
        "ipv4": null,
        "ipv6": null,
        "lineprotocol": "down",
        "macaddress": "ecf4.bb3e.c0ec",
        "mediatype": "SVI",
        "mtu": null
    },
    "xe1/2": {
```

```
            "bandwidth": null,
            "description": "test interface set by ansible 9th",
            "duplex": null,
            "ipv4": null,
            "ipv6": null,
            "lineprotocol": "down",
            "macaddress": "ecf4.bb3e.c0ec",
            "mediatype": "ETH",
            "mtu": null
        },

----- Snipped ----

         "xe9/2": {
            "bandwidth": null,
            "description": null,
            "duplex": null,
            "ipv4": null,
            "ipv6": null,
            "lineprotocol": "down",
            "macaddress": "ecf4.bb3e.c0ec",
            "mediatype": "ETH",
            "mtu": null
        }
    }
}

TASK [debug]*******************************************************************************
ok: [OcNOS-SW1] => {
    "ansible_net_memfree_mb": 7554
}

TASK [debug]*******************************************************************************
ok: [OcNOS-SW1] => {
    "ansible_net_memtotal_mb": 7988
}

TASK [debug]*******************************************************************************
ok: [OcNOS-SW1] => {
    "ansible_net_model": "DELL S6000-ON"
}

TASK [debug]*******************************************************************************
ok: [OcNOS-SW1] => {
    "ansible_net_neighbors": {}
}

TASK [debug]*******************************************************************************
ok: [OcNOS-SW1] => {
    "ansible_net_serialnum": "CN07VJDK282985730184"
```

```
}

TASK [debug]*******************************************************************************
ok: [OcNOS-SW1] => {
    "ansible_net_version": "1.3.8.44a"
}

PLAY RECAP *********************************************************************************
OcNOS-SW1                  : ok=14   changed=0    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0
```

cli_command

cli_command is platform agnostic and it can also use OcNOS.

https://docs.ansible.com/ansible/latest/modules/cli_command_module.html

Parameters

Standard cli_command parameters are supported:

Table 1-2: Supported parameters

Parameter	Choices/Defaults	Comments
`answer` `list`		The answer to reply with if prompt is matched. The value can be a single answer or a list of answer for multiple prompts. In case the command execution results in multiple prompts the sequence of the prompt and excepted answer should be in same order.
`check_all` `boolean`	Choices: • no (default) • yes	By default if any one of the prompts mentioned in prompt option is matched it won't check for other prompts. This boolean flag, that when set to True will check for all the prompts mentioned in prompt option in the given order. If the option is set to True all the prompts should be received from remote host if not it will result in timeout.
`command` `- / required`		The command to send to the remote network device. The resulting output from the command is returned, unless send only is set.
`newline` `boolean`	Choices: • no • yes (default)	The boolean value, that when set to false will send answer to the device without a trailing newline.
`prompt` `list`		A single regex pattern or a sequence of patterns to evaluate the expected prompt from command.
`sendonly` `boolean`	Choices: • no (default) • yes	The boolean value, that when set to true will send command to the device but not wait for a result.

Return Values

As well as the standard cli_command, common return values are supported. JSON is not supported.

Table 1-3: Return values

Key	Returned	Description
stdout string	when sendonly is false	The response from the command. Sample: Software version: DELL_S6000-ON-OcNOS-1.3.8.44a-DC_IPBASE-S0-P0 [...]

Samples

The following is an example of `show version`.

Playbook:

```
---
- hosts: ocnos

  tasks:
  - name: Test OcNOS command
    cli_command:
      command: show version
    register: result

  - name: debug
    debug:
      msg: "{{ result.stdout_lines }}"
```

Output:

```
$ ansible-playbook -i inventory/inventory.ini clicommand-playbook.yml -l OcNOS-SW1

PLAY [ocnos] ************************************************************

TASK [Gathering Facts] **************************************************
ok: [OcNOS-SW1]

TASK [Test OcNOS command] ***********************************************
ok: [OcNOS-SW1]

TASK [debug] ************************************************************
ok: [OcNOS-SW1] => {
    "msg": [
        "Software version: DELL_S6000-ON-OcNOS-1.3.8.44a-DC_IPBASE-S0-P0 09/28/2019
21:41:50",
        " Copyright (C) 2019 IP Infusion. All rights reserved",
        "",
        " Software Product: OcNOS, Version: 1.3.8.44a",
        " Hardware Model: DELL S6000-ON",
        " Software Feature Code: DC-IPBASE",
```

```
        " System Configuration Code: S0",
        " Package Configuration Code: P0",
        " Software Baseline Version: 1.3.8.44a",
        "",
        "Installation Information:",
        " Image Filename: DELL_S6000_ON-OcNOS-1.3.8.44a-DC_IPBASE-S0-P0-installer",
        " Install method: tftp",
        " ONIE SysInfo: x86_64-dell_s6000_s1220-r0"
    ]
}

PLAY RECAP *********************************************************************
OcNOS-SW1                  : ok=3    changed=0    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0
```

This example shows `reload` used with multiple prompts:

```
---
- hosts: ocnos

  tasks:
  - name: multiple prompt, multiple answer (mandatory check for all prompts)
    cli_command:
      command: reload
      check_all: True
      prompt:
         - "Would you like to save them now?"
         - "Are you sure you would like to reset the system?"
      answer:
         - 'y'
         - 'y'
```

ipinfusion.ocnos.ocnos_command

cli_command can execute only one command per task since it doesn't support multiple commands parameters. Unlike cli_command, ocnos_command supports multiple commands.

Parameters

Table 1-4: Parameters

Parameter	Choices/Defaults	Comments
commands - / required		List of commands to send to the remote device over the configured provider. The resulting output from the command is returned. If the wait_for argument is provided, the module is not returned until the condition is satisfied or the number of retries as expired.
interval –	Default: 1	Configures the interval in seconds to wait between retries of the command. If the command does not pass the specified conditions, the interval indicates how long to wait before trying the command again.
match –	Choices: • any • all (default)	The match argument is used in conjunction with the wait_for argument to specify the match policy. Valid values are all or any. If the value is set to all then all conditionals in the wait_for must be satisfied. If the value is set to any then only one of the values must be satisfied.
provider dictionary		A dict object containing connection details.
auth_pass string		Specifies the password to use if required to enter privileged mode on the remote device. If authorize is false, then this argument does nothing. If the value is not specified in the task, the value of environment variable ANSIBLE_NET_AUTH_PASS will be used instead.
authorize boolean	Choices: • no (default) • yes	Instructs the module to enter privileged mode on the remote device before sending any commands. If not specified, the device will attempt to execute all commands in non-privileged mode. If the value is not specified in the task, the value of environment variable ANSIBLE_NET_AUTHORIZE will be used instead.
host string / required		Specifies the DNS host name or address for connecting to the remote device over the specified transport. The value of host is used as the destination address for the transport.
password string		Specifies the password to use to authenticate the connection to the remote device. This value is used to authenticate the SSH session. If the value is not specified in the task, the value of environment variable ANSIBLE_NET_PASSWORD will be used instead.
port integer	Default: 22	Specifies the port to use when building the connection to the remote device.
ssh_keyfile path		Specifies the SSH key to use to authenticate the connection to the remote device. This value is the path to the key used to authenticate the SSH session. If the value is not specified in the task, the value of environment variable ANSIBLE_NET_SSH_KEYFILE will be used instead.

Table 1-4: Parameters (Continued)

Parameter	Choices/Defaults	Comments
timeout integer	Default: 10	Specifies the timeout in seconds for communicating with the network device for either connecting or sending commands. If the timeout is exceeded before the operation is completed, the module will error.
username string		Configures the username to use to authenticate the connection to the remote device. This value is used to authenticate the SSH session. If the value is not specified in the task, the value of environment variable ANSIBLE_NET_USERNAME will be used instead.
retries -	Default: 10	Specifies the number of retries a command should be tried before it is considered failed. The command is run on the target device every retry and evaluated against the wait_for conditions.
wait_for -		List of conditions to evaluate against the output of the command. The task will wait for each condition to be true before moving forward. If the conditional is not true within the configured number of retries, the task fails.

Return Values

Table 1-5: Return values

Key	Returned	Description
stdout list	always	the set of responses from the commands.
stdout_lines list	always	The value of stdout split into a list

Samples

The example below shows that three show commands can be specified in a task.

Playbook

```
---
- hosts: ocnos

  tasks:
  - name: Test OcNOS command
    ipinfusion.ocnos.ocnos_command:
      commands:
        - show version
        - show hardware-information memory
        - show interface br
    register: result
```

```
- name: Show Result
  debug:
    msg: "{{ result.stdout_lines }}"
```

cli_config

Parameters

Only the parameters below in standard Ansible cli_config are supported.

Table 1-6: Parameters

Parameter	Choices/Defaults	Comments
backup boolean	Choices: • no (default) • yes	This argument will cause the module to create a full backup of the current running config from the remote device before any changes are made. If the backup_options value is not given, the backup file is written to the backup folder in the playbook root directory or role root directory, if playbook is part of an ansible role. If the directory does not exist, it is created.
backup_options dictionary		This is a dict object containing configurable options related to backup file path. The value of this option is read only when backup is set to yes, if backup is set to no this option will be silently ignored.
dir_path path		This option provides the path ending with directory name in which the backup configuration file will be stored. If the directory does not exist it will be first created and the filename is either the value of filename or default filename as described in filename options description. If the path value is not given in that case a backup directory will be created in the current working directory and backup configuration will be copied in filename within backup directory.
filename -		The filename to be used to store the backup configuration. If the the filename is not given it will be generated based on the hostname, current time and date in format defined by <hostname>_config.<current-date>@<current-time>
config string		The config to be pushed to the network device. This argument is mutually exclusive with rollback and either one of the option should be given as input. The config should have indentation that the device uses.

Return Values

Table 1-7: Return values

Key	Returned	Description
backup_path string	when backup is yes	The full path to the backup file Sample: /playbooks/ansible/backup/hostname_config.2016-07-16@22:28:34
commands list	always	The set of commands that will be pushed to the remote device Sample: ['hostname foobar_by_cliconfig']

Samples

Playbook:

```
---
- hosts: ocnos

  tasks:
  - name: multiline config
    cli_config:
      config: |
        hostname foo
        bridge 1 protocol mstp
        vlan 2-10 bridge 1
```

Output:

```
$ ansible-playbook -i inventory/inventory.ini cliconfig-playbook.yaml -l OcNOS-SW1
PLAY [ocnos] ************************************************************

TASK [Gathering Facts]***************************************************************
ok: [OcNOS-SW1]

TASK [multiline config]**************************************************************
changed: [OcNOS-SW1]

PLAY RECAP ***************************************************************
OcNOS-SW1                : ok=2    changed=1    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0
```

Validation:

Note: The **bold lines** in `show run` indicate the configuration that was changed by this playbook.

```
$ ssh -l ocnos 10.5.178.3
ocnos@10.5.178.3's password:
Last login: Fri Dec 13 16:59:17 2019 from 10.5.176.106

OcNOS version DELL_S6000-ON-OcNOS-1.3.8.44a-DC_IPBASE-S0-P0  09/28/2019 21:41:50
foo>en
foo#show spanning-tree mst detail
% 1: Bridge up - Spanning Tree Enabled
% 1: CIST Root Path Cost 0 - CIST Root Port 0 -  CIST Bridge Priority 32768
% 1: Forward Delay 15 - Hello Time 2 - Max Age 20 - Transmit Hold Count 6 - Max-hops 20
% 1: CIST Root Id 8000000000000000
% 1: CIST Reg Root Id 8000000000000000
% 1: CIST Bridge Id 8000000000000000
% 1: 0 topology change(s)  - last topology change Thu Jan  1 00:00:00 1970

% 1: portfast bpdu-filter disabled
% 1: portfast bpdu-guard disabled
foo#show run
!
! Software version: DELL_S6000-ON-OcNOS-1.3.8.44a-DC_IPBASE-S0-P0 09/28/2019 21:41:50
!
!Last configuration change at 16:59:18 UTC Fri Dec 13 2019 by ocnos
!
no service password-encryption
!
logging monitor 7
!
ip vrf management
!
forwarding profile l2-profile-three
!
hostname foo
ip domain-lookup vrf management
no ip domain-lookup
bridge 1 protocol mstp
data-center-bridging enable bridge 1
feature telnet vrf management
feature ssh vrf management
snmp-server enable snmp vrf management
snmp-server view all .1 included vrf management
snmp-server community public group network-operator vrf management
feature ntp vrf management
ntp enable vrf management
username ocnos role network-admin password encrypted $1$we7czZA/$kGreh592N7ohrMdsGQUj5.
feature rsyslog vrf management
!
```



```
vlan database
 vlan 2-10 bridge 1 state enable
!
spanning-tree mst configuration
!
interface eth0
 ip vrf forwarding management
 ip address dhcp
!
```

ipinfusion.ocnos.ocnos_config

This is a module equivalent in functionality with the platform-agnostic cli_config module.

Parameters

Table 1-8: Supported parameters

Parameter	Choices/Defaults	Comments
after -		The ordered set of commands to append to the end of the command stack if a change needs to be made. Just like with before this allows the playbook designer to append a set of commands to be executed after the command set.
backup -	Choices: • no (default) • yes	This argument will cause the module to create a full backup of the current running-config from the remote device before any changes are made. If the backup_options value is not given, the backup file is written to the backup folder in the playbook root directory or role root directory, if playbook is part of an ansible role. If the directory does not exist, it is created.
backup_options dictionary		This is a dict object containing configurable options related to backup file path. The value of this option is read only when backup is set to yes, if backup is set to no this option will be silently ignored.
dir_path path		This option provides the path ending with directory name in which the backup configuration file will be stored. If the directory does not exist it will be first created and the filename is either the value of filename or default filename as described in filename options description. If the path value is not given in that case a backup directory will be created in the current working directory and backup configuration will be copied in filename within backup directory.
filename -		The filename to be used to store the backup configuration. If the the filename is not given it will be generated based on the hostname, current time and date in format defined by <hostname>_config.<current-date>@<current-time>
before -		The ordered set of commands to push on to the command stack if a change needs to be made. This allows the playbook designer the opportunity to perform configuration commands prior to pushing any changes without affecting how the set of commands are matched against the system.

Table 1-8: Supported parameters (Continued)

Parameter	Choices/Defaults	Comments
diff_ignore_lines –		Use this argument to specify one or more lines that should be ignored during the diff. This is used for lines in the configuration that are automatically updated by the system. This argument takes a list of regular expressions or exact line matches.
lines –		The ordered set of commands that should be configured in the section. The commands must be the exact same commands as found in the device running-config. Be sure to note the configuration command syntax as some commands are automatically modified by the device config parser. aliases: commands
match –	Choices: • line (default) • strict • exact • none	Instructs the module on the way to perform the matching of the set of commands against the current device config. If match is set to line, commands are matched line by line. If match is set to strict, command lines are matched with respect to position. If match is set to exact, command lines must be an equal match. Finally, if match is set to none, the module will not attempt to compare the source configuration with the running configuration on the remote device.
parents –		The ordered set of parents that uniquely identify the section or hierarchy the commands should be checked against. If the parents argument is omitted, the commands are checked against the set of top level or global commands.
replace –	Choices: • line (default) • block • config	Instructs the module on the way to perform the configuration on the device. If the replace argument is set to line then the modified lines are pushed to the device in configuration mode. If the replace argument is set to block then the entire command block is pushed to the device in configuration mode if any line is not correct.
running_config string		The module, by default, will connect to the remote device and retrieve the current running-config to use as a base for comparing against the contents of source. There are times when it is not desirable to have the task get the current running-config for every task in a playbook. The running_config argument allows the implementer to pass in the configuration to use as the base config for this module. aliases: config

Table 1-8: Supported parameters (Continued)

Parameter	Choices/Defaults	Comments
save_when –	Choices: • always • never (default) • modified • changed	When changes are made to the device running-configuration, the changes are not copied to non-volatile storage by default. Using this argument will change that before. If the argument is set to always, then the running-config will always be copied to the startup-config and the modified flag will always be set to True. If the argument is set to modified, then the running-config will only be copied to the startup-config if it has changed since the last save to startup-config. If the argument is set to never, the running-config will never be copied to the startup-config. If the argument is set to changed, then the running-config will only be copied to the startup-config if the task has made a change.
src –		The src argument provides a path to the configuration file to load into the remote system. The path can either be a full system path to the configuration file if the value starts with / or relative to the root of the implemented role or playbook. This argument is mutually exclusive with the lines and *parents* arguments. It can be a Jinja2 template as well. The src file must have same indentation as a live switch config.

Return Values

Table 1-9: Return values

Key	Returned	Description
backup_path string	When backup is yes	The full path to the backup file Sample: /home/somewhere/ansible/backup/OcNOS-SW1_config.2020-03-17@05:33:06
commands list	Always	The set of commands that will be pushed to the remote device. Sample: ['hostname OcNOS-SW1-20']
date string	When backup is yes	The date extracted from the backup file name Sample: 2020-03-17
filename string	When backup is yes and filename is not specified in backup options	The name of the backup file Sample: OcNOS-SW1_config.2020-03-17@05:33:06

Table 1-9: Return values (Continued)

Key	Returned	Description
shortname string	When backup is yes and filename is not specified in backup options	The full path to the backup file excluding the timestamp Sample: /home/somewhere/ansible/backup/OcNOS-SW1_config
time string	when backup is yes	The time extracted from the backup file name Sample: 05:33:06

Samples

Playbook:

```
---
- hosts: ocnos
  gather_facts: false

  tasks:
  - name: Test OcNOS configs
    ipinfusion.ocnos.ocnos_config:
      lines: "hostname {{ inventory_hostname }}-1 "

  - name: configure interface settings
    ipinfusion.ocnos.ocnos_config:
      lines:
        - description test interface set by ansible
        - ip address 172.16.101.5/24
      parents: interface {{ interface1 }}

  - name: configurable backup path
    ipinfusion.ocnos.ocnos_config:
      backup: yes
      backup_options:
        filename: backup-{{ inventory_hostname }}.cfg
        dir_path: /home/momose/ansible/backup
```

Output:

```
$ ansible-playbook -i inventory/inventory.ini config-playbook.yaml -l OcNOS-SW1

PLAY [ocnos] ***********************************************************

TASK [Test OcNOS configs]***********************************************
ok: [OcNOS-SW1]
```

```
TASK [configure interface settings] ************************************
changed: [OcNOS-SW1]

TASK [configurable backup path] ****************************************
ok: [OcNOS-SW1]

PLAY RECAP *************************************************************
OcNOS-SW1                  : ok=3    changed=1    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0
```

Validation:

```
$ ssh -l ocnos 10.5.178.3
ocnos@10.5.178.3's password:
Last login: Fri Dec  6 15:04:18 2019 from 10.5.176.106

OcNOS version DELL_S6000-ON-OcNOS-1.3.8.44a-DC_IPBASE-S0-P0  09/28/2019 21:41:50
OcNOS-SW1-1>show run
!
! Software version: DELL_S6000-ON-OcNOS-1.3.8.44a-DC_IPBASE-S0-P0 09/28/2019 21:41:50
!
!Last configuration change at 15:04:22 UTC Fri Dec 06 2019 by ocnos
!
no service password-encryption
!
logging monitor 7
!
ip vrf management
!
forwarding profile l2-profile-three
!
hostname OcNOS-SW1-1
ip domain-lookup vrf management
no ip domain-lookup
feature telnet vrf management
feature ssh vrf management
snmp-server enable snmp vrf management
snmp-server view all .1 included vrf management
snmp-server community public group network-operator vrf management
feature ntp vrf management
ntp enable vrf management
username ocnos role network-admin password encrypted $1$we7czZA/$kGreh592N7ohrMdsGQUj5.
feature rsyslog vrf management
!
interface eth0
 ip vrf forwarding management
 ip address dhcp
!
interface lo
 ip address 127.0.0.1/8
```

```
 ipv6 address ::1/128
!
interface lo.management
 ip vrf forwarding management
 ip address 127.0.0.1/8
 ipv6 address ::1/128
!
interface xe1/1
!
interface xe1/2
 description test interface set by ansible
 ip address 172.16.100.5/24
!
interface xe1/3
!
interface xe1/4
!
interface xe2
!
interface xe3/1
 port breakout enable
!
interface xe3/2
 switchport
!
interface xe3/3
!
interface xe3/4
!
interface xe4
!
interface xe5/1
!
interface xe5/2
!
interface xe5/3
!
interface xe5/4

OcNOS-SW1-1>show int xe1/2
Interface xe1/2
  Scope: both
  Flexport: Non Control Port (InActive)
  Hardware is ETH  Current HW addr: ecf4.bb3e.c0ec
  Physical:ecf4.bb3e.c0ee  Logical:(not set)
  Description: test interface set by ansible
  Port Mode is Router
  Interface index: 10002
  Metric 1 mtu 1500
  <UP,BROADCAST,MULTICAST>
```

```
  VRF Binding: Not bound
  DHCP client is disabled.
  Last Flapped: Never
  Statistics last cleared: Never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  RX
    unicast packets 0 multicast packets 0 broadcast packets 0
    input packets 0 bytes 0
    jumbo packets 0
    undersize 0 oversize 0 CRC 0 fragments 0 jabbers 0
    input error 0
    input with dribble 0 input discard 0
    Rx pause 0
  TX
    unicast packets 0 multicast packets 0 broadcast packets 0
    output packets 0 bytes 0
    jumbo packets 0
    output errors 0 collision 0 deferred 0 late collision 0
    output discard 0
    Tx pause 0
OcNOS-SW1-1>exit
```

Verify the backup file was created:

```
$ ls -lsa backup/
total 8
0 drwxrwxrwx 2 momose momose   33 Dec 17 07:33 .
4 drwxrwxr-x 7 momose momose 4096 Dec 17 07:32 ..
4 -rw-rw-r-- 1 momose momose 3144 Dec 17 07:33 backup-OcNOS-SW1.cfg
```

ipinfusion.ocnos.ocnos_ping / net_ping

These modules are similar. net_ping uses ocnos_ping when ansible_network_os is set to ipinfusion.ocnos.ocnos.

Parameters

Table 1-10: Supported Parameters

Parameter	Choices/Defaults	Comments
count -	Default: 5	Number of packets to send.
dest / required		The IP Address or hostname (resolvable by switch) of the remote node.

Table 1-10: Supported Parameters

state -	Choices: absent present (default)	Determines if the expected result is success or fail.
vrf -	Default: "management"	The VRF to use for forwarding.

Return Values

Table 1-11: Return Values

Key	Returned	Description
commands list	always	Show the command sent. Sample: ['ping\nip\n \n192.168.122.1\n3\n64\n1\n100\n2\n0\nn\nn\n']
packet_loss string	always	Percentage of packets lost. Sample: 0%
packets_rx integer	always	Packets successfully received. Sample: 3
packets_tx integer	always	Packets successfully transmitted. Sample: 3
rtt dictionary	always	Show RTT stats. Sample: {"avg": 0.115, "max": 0.135, "min": 0.079}

Sample

Playbook:

```
---
- hosts: ocnos

  tasks:
  - name: Test OcNOS Ping
    ipinfusion.ocnos.ocnos_ping:
      dest: 192.168.122.1
      interface: eth0
      count: 3
      vrf: " "
    register: result
```

Output:

```
$ ansible-playbook -i inventory/inventory.ini ocnos_ping.yml -l OcNOS-VM1
```

```
PLAY
[ocnos]*********************************************************************

TASK [Test OcNOS Ping]**********************************************************
ok: [OcNOS-VM1]

TASK [Show Result]**************************************************************
ok: [OcNOS-VM1] => {
    "msg": {
        "ansible_facts": {
            "discovered_interpreter_python": "/usr/bin/python"
        },
        "changed": false,
        "commands": "ping\nip\n \n192.168.122.1\n3\n64\n1\n100\n2\n0\nn\nn\n",
        "failed": false,
        "packet_loss": "0%",
        "packets_rx": 3,
        "packets_tx": 3,
        "rtt": {
            "avg": 0.119,
            "max": 0.122,
            "min": 0.115
        }
    }
}

PLAY RECAP *********************************************************************
OcNOS-VM1                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0
```

ipinfusion.ocnos.ocnos_bgp_facts

This module provides BGP related information. Currently, this supports only bgp neighbors.

Sample Playbook

fact-bgp.yml

```
---
hosts: ocnos
 gather_facts: no

  tasks:
  - name: Test OcNOS Facts
    ipinfusion.ocnos.ocnos_bgp_facts:
      gather_subset: neighbor
    register: result

  - name: Show Facts
    debug:
       msg: "{{ ansible_facts }}"
```

Sample Output

$ ansible-playbook -i inventory/inventory.ini fact-bgp.yml -l OcNOS-VM1

```
PLAY [ocnos]*******************************************************************************

TASK [Test OcNOS Facts]********************************************************************
ok: [OcNOS-VM1] => {
    "msg": {
        "discovered_interpreter_python": "/usr/bin/python",
        "ansible_net_bgp_neighbor": {
            "10.10.10.10": {
                "Received": {
                    "InQueue": 0,
                    "messages": 0,
                    "notifications": 0
                },
                "Sent": {
                    "InQueue": 0,
                    "messages": 799,
                    "notifications": 0
                },
                "addressFamily": {
                    "IPv4 Unicast": {
                        "BGPtableVer": 1,
                        "acceptedPrefixes": 0,
                        "announcedPrefixes": 0,
                        "index": 1,
                        "mask": "0x2",
                        "neighborVer": 0,
                        "offset": 0
                    },
                    "VPNv4 Unicast": {
                        "BGPtableVer": 1,
                        "acceptedPrefixes": 0,
                        "announcedPrefixes": 0,
                        "index": 1,
                        "mask": "0x2",
                        "neighborVer": 0,
                        "offset": 0
                    }
                },
                "connections": {
                    "dropped": 0,
                    "established": 0
                },
                "holdTime": 90,
                "keepAlive": 30,
                "lastRead": "Never",
                "localAS": "100",
```

```
            "minTimeBetweenAdv": 5,
            "remoteAS": "100",
            "routeRefreshRequest": {
                "received": 0,
                "sent": 0
            },
            "state": "Active"
        },
        "2001:500:602::101": {
            "Received": {
                "InQueue": 0,
                "messages": 0,
                "notifications": 0
            },
            "Sent": {
                "InQueue": 0,
                "messages": 0,
                "notifications": 0
            },
            "addressFamily": {
                "IPv4 Unicast": {
                    "BGPtableVer": 1,
                    "acceptedPrefixes": 0,
                    "announcedPrefixes": 0,
                    "index": 2,
                    "mask": "0x4",
                    "neighborVer": 0,
                    "offset": 0
                }
            },
            "connections": {
                "dropped": 0,
                "established": 0
            },
            "holdTime": 90,
            "keepAlive": 30,
            "lastRead": "Never",
            "localAS": "100",
            "minTimeBetweenAdv": 5,
            "remoteAS": "100",
            "routeRefreshRequest": {
                "received": 0,
                "sent": 0
            },
            "state": "Connect"
        }
    },
    "ansible_net_gather_subset": [
        "neighbor"
    ]
```

```
    }
}

PLAY RECAP *********************************************************************
OcNOS-VM1                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0
```

ipinfusion.ocnos.ocnos_isis_facts

This module provides IS-IS related information. Currently, this supports only IS-IS neighbors.

net_isis_neighbor reads the output of ‘show clns neighbors’ , analyzes and converts it to ansible output format.

Sample Playbook

fact-isis.yml

```
---
- hosts: ocnos
  gather_facts: no

  tasks:
  - name: Test OcNOS ISIS facts
    ipinfusion.ocnos.ocnos_isis_facts:
      gather_subset: neighbor
    register: result


  - name: Show ISIS Facts
    debug:
      msg: "{{ result }}"
```

Sample Output

$ ansible-playbook -i inventroy/inventory.ini fact-isis.yml -l OcNOS-VM1

```
PLAY [ocnos]*******************************************************************

TASK [Test OcNOS Facts]********************************************************
ok: [OcNOS-VM1]

TASK [Show Facts]**************************************************************
ok: [OcNOS-VM1] => {
    "msg": {
        "discovered_interpreter_python": "/usr/bin/python",
        "net_gather_subset": [
            "neighbor"
        ],
        "net_isis_neighbor": {
            "0000.0000.0001": {
                "Holdtime": "21",
                "Interface": "eth1",
```

```
                "Protocol": "IS-IS",
                "SNPA": "5254.0027.4096",
                "State": "Up",
                "Type": "L2"
            }
        }
    }
}

PLAY RECAP *********************************************************************
OcNOS-VM1                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0
```

Steps to use an Ansible Playbook

In the below example, we will show steps to use an Ansible Playbook using Jinja2 template and sample parameter file. This example show how to configure LDP protocol:

ocnos_ldp.j2

```
osboxes@osboxes:~/playbooks$ cat templates/ocnos_ldp.j2
{%if ldp is defined%}
router ldp
 {% for peer in ldp.peers -%}
 targeted-peer ipv4 {{ peer.address }}
  exit
 {% endfor %}
{% if  ldp.advertise_label_prefix is defined %}
 advertise-labels for only_lo to any
{%endif%}

{% for interface in ldp.interfaces -%}
interface {{ interface.name }}
 enable-ldp {{ interface.protocol }}
 label-switching
 exit
{% endfor %}
{%endif%}
```

We will provide the LDP configuration details in the appropriate host_vars file.

sw2.yml

```
osboxes@osboxes:~/playbooks$ cat host_vars/sw2.yml
ldp:
 peers:
   - address: 1.1.1.1
   - address: 3.3.3.3
 advertise_label_prefix: only_lo
ldp_interfaces:
   - { ldp_interface: eth1, ldp_protocol: ipv4 }
   - { ldp_interface: eth2, ldp_protocol: ipv4 }
```

Here is the overall directory structure of the Ansible Playbook and associated files.

```
osboxes@osboxes:~/playbooks$ tree
|___ansible.cfg
|___backup
```

```
|___group_vars
    |___ocnos.yml
|___hosts-net
|___host_vars
    |___sw2.yml
|___ldp-playbook.yml
|___showldp-playbook.yml
|___templates
    |___ocnos_ldp.j2
```

The following is the content of ansible.cfg file which points to hosts-net inventory file.

ansible.cfg

```
osboxes@osboxes:~/playbooks$ cat ansible.cfg
[defaults]
inventory = hosts-net
host_key_checking = False
retry_files_enabled = False
interpreter_python = auto
osboxes@osboxes:~/playbooks
```

Following is the content of the hosts-net inventory file. Currently this has details of only one device.

hosts-net

```
osboxes@osboxes:~/playbooks$ cat hosts-net
[ocnos]
sw2 ansible_host=10.12.9.105
osboxes@osboxes:~/playbooks$
```

Following is the content of ocnos.yml in group_vars folder.

ocnos.yml

```
osboxes@osboxes:~/playbooks$ cat group_vars/ocnos.yml
ansible_connection: network_cli
ansible_network_os: ipinfusion.ocnos.ocnos
ansible_become: yes
ansible_become_method: enable
ansible_ssh_user: ocnos
ansible_ssh_pass: ocnos
osboxes@osboxes:
```

The below playbook pushes the ldp configuration created using the template file 'ocnos_ldp.j2' for all the ocnos hosts using the appropriate host_vars file 'sw2.yml':

ldp-playbook.yml

```
(ansible) osboxes@osboxes:~/playbooks$ cat ldp-playbook.yml
---

  - hosts: ocnos
```

```
    gather_facts: no

    tasks:

      - name: configure LDP config on OcNOS
        cli_config:
          config: "{{ lookup('template', 'templates/{{  ansible_network_os }}_ldp.j2')
}}"
```

Configuration on the OcNOS device before executing the Ansible Playbook:

```
#show running-config ldp
!
!
#
```

Now we can execute the Ansible playbook and below are the logs that will be seen.

```
(ansible) osboxes@osboxes:~/playbooks$ ansible-playbook ldp-playbook.yml

PLAY [ocnos]*********************************************************************

TASK [configure LDP config on OcNOS]*********************************************
changed: [sw2]

PLAY RECAP***********************************************************************
sw2                        : ok=1    changed=1    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0
```

Now check the configs on the OcNOS device. which should show the LDP configurations.

```
#show running-config ldp
!
router ldp
 targeted-peer ipv4 1.1.1.1
  exit-targeted-peer-mode
 targeted-peer ipv4 3.3.3.3
  exit-targeted-peer-mode
 advertise-labels for only_lo to any
!
!
interface eth2
 enable-ldp ipv4
!
interface eth1
 enable-ldp ipv4
!
#
```

The below playbook shows how to check the runtime status of LDP protocol through the 'show ldp session' command and prints its output. It is assumed that the neighboring nodes are configured accordingly to get the LDP session up and running.

showldp-playbook.yml

```
osboxes@osboxes:~/playbooks$ cat showldp-playbook.yml
---

  - hosts: ocnos
    gather_facts: no

    tasks:
     - name: show LDP config ocnos
       cli_command:
         command: show ldp session
       register: result

     - name: debug
       debug:
         msg: "{{ result.stdout_lines }}"
osboxes@osboxes:~/playbooks$
```

When you run this playbook, the following will be its output. Parsing of the show command output needs to be done to determine if the runtime status of the protocol is fine.

```
(ansible) osboxes@osboxes:~/playbooks$ ansible-playbook showldp-playbook.yml

PLAY [ocnos]*********************************************************************

TASK [show LDP config ocnos]*****************************************************
ok: [sw2]

TASK [debug]*********************************************************************
ok: [sw2] => {
    "msg": [
        "Peer IP Address           IF Name    My Role    State      KeepAlive  UpTime",
        "3.3.3.3                   eth2       Passive   OPERATIONAL   30    03:58:20",
        "1.1.1.1                   eth1       Active    OPERATIONAL   30    03:58:20"
    ]
}

PLAY RECAP***********************************************************************
sw2                        : ok=2    changed=0    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0

(ansible) osboxes@osboxes:~/playbooks$
```

The below playbook is to unconfigure the LDP configuration on the OcNOS device.

```
osboxes@osboxes:~/playbooks$ cat unconfigureldp-playbook.yml
---
 - hosts: ocnos
   gather_facts: no

   tasks:
```

```
  - name: give "no router ldp" command
    ipinfusion.ocnos.ocnos_config:
      lines:
        - no router ldp

  - name: show LDP config ocnos
    cli_command:
      command: show running-config ldp
    register: result

  - name: debug
    debug:
      msg: "{{ result.stdout_lines }}"
```

Jinja2 Templates for configuring OcNOS

In this section, we will provide a few Jinja2 templates which can be used to configure a few protocols in OcNOS. Also a sample yaml parameter file is provided for each j2 template with explanations of the parameters. While creating the Jinja2 template, only the commonly used configuration is considered. Customers can use these templates as such, if it meets their configuration needs or can modify them accordingly.

Template File for LDP

ocnos_ldp.j2

```
{%if ldp is defined%}
router ldp
 {% for peer in ldp.peers -%}
 targeted-peer ipv4 {{ peer.address }}
  exit
 {% endfor %}
{% if  ldp.advertise_label_prefix is defined %}
 advertise-labels for only_lo to any
{%endif%}

{% for interface in ldp.interfaces -%}
interface {{ interface.name }}
 enable-ldp {{ interface.protocol }}
 label-switching
 exit
{% endfor %}
{%endif%}
```

Sample Parameter File for LDP

sw2.yml

`ldp:`	Router LDP configurations
`  peers:`	Peer Details
`    - address: 1.1.1.1`	Configures the targeted-peer IPv4 address as <1.1.1.1> under router LDP
`    - address: 3.3.3.3`	Configures the targeted-peer IPv4 address as <3.3.3.3> under router LDP
`  advertise_label_prefix: only_lo`	If defined it will Configure the advertise label command for "ony_lo" prefix_list under router ldp
`  interfaces:`	LDP Interface configurations
`    - interface: eth1`	Enables LDP on interface <eth1> for protocol <ipv4>
`      protocol: ipv4`	Enables LDP for ipv4 protocol under interface eth1

BGP Configuration

Template File for BGP

ocnos_bgp.j2

```
{%if bgp is defined%}
router bgp {{ bgp.asn }}
{% if bgp.router is defined %}
{%else%}
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 no bgp inbound-route-filter
{% endif %}
 {% for network in bgp.networks -%}
  network {{ network.network_id }}
{%if network.network_id =='36.0.0.3/32'%}
 max-paths ibgp 2
{%endif%}
 {% endfor -%}
 {% for neighbor in bgp.neighbors -%}
 neighbor {{ neighbor.neighbor_id }} remote-as {{ neighbor.remoteas }}
 neighbor {{ neighbor.neighbor_id }}  {{ neighbor.detection }} bfd multihop
 neighbor {{ neighbor.neighbor_id }} update-source {{ neighbor.updatesource }}
 {% endfor %}
allocate-label all
 !
{% for address_family in bgp.address_family -%}
{% if address_family.address_family_type  == 'vpnv4' -%}
 address-family {{ address_family.address_family_type }} unicast
```

```
 {% endif %}
 {% if address_family.address_family_type == 'labeled-unicast' -%}
 address-family ipv4 {{ address_family.address_family_type }}
 {% endif %}
{% if address_family.address_family_type == 'vrf' -%}
{% for vrf in address_family.vrfs -%}
address-family ipv4 {{address_family.address_family_type }} {{ vrf.vrf_name }}
{% if vrf.protocol is defined %}
 redistribute {{vrf.protocol}}
{% endif %}
 redistribute connected
 exit-address-family
 {% endfor %}
{% endif %}
{% if address_family.neighbors is defined %}
{% for neighbor in address_family.neighbors -%}
neighbor {{neighbor.neighbor_id}} activate
{% if neighbor.route_reflector_type is defined %}
 neighbor {{neighbor.neighbor_id}} {{ neighbor.route_reflector_type }}
{% endif %}
{% if neighbor.next_hop_type is defined %}
 neighbor {{ neighbor.neighbor_id}} {{ neighbor.next_hop_type }}
{% endif %}
 {% endfor %}
exit-address-family
!
{% endif %}
 {% endfor %}
{%endif%}
```

Parameter File for BGP

sw2.yml

`bgp:`	Router BGP configurations
`asn: 65001`	Autonomous system number
`networks:`	Network command
`-network_id: 36.0.0.2/32`	Configures the network IPv4 address as <36.0.0.2> under router BGP 65001
`neighbors:`	Neighbor command
`- neighbor_id: 10.0.1.14`	Identifies the neighbor
`remoteas: 65001`	configure remote-as 65001 for neighbor 10.0.1.14 command under router bgp
`detection: fall-over`	Configure detection type as <fall-over bfd multihop>
`command under router bgp`	

`updatesource: lo`	Configure update-source lo> for <neighbor 36.0.0.1 > under router bgp
`address_family:`	Address-family configuration
`- address_family_type: labeled-unicast`	Address family type label-unicast will be configured under router bgp
`neighbors:`	Neighbor configuration under address family
`- neighbor_id: 10.0.1.14`	Activate neighbor 10.0.1.14 for address family label-unicast under bgp
`next_hop_type: next-hop-self`	If next_hop_type is defined then it will configure the neighbor 10.0.1.14 next-hop-self command will be configured for address-family label-unicast
`route_reflector_type: route-reflector-client`	If route_reflector_type is defined then neighbor 10.0.1.14 will be configured as route reflector client for address-family label-unicast
`- address_family_type: vpnv4`	Adress family type vpnv4 will be configured under router bgp
`neighbors:`	Neighbor configuration under address family
`- neighbor_id: 10.0.1.14`	Activate neighbor 10.0.1.14 for address family vpnv4 under bgp
`route_reflector_type:route-reflector-client`	If route_reflector_type is defined then neighbor 10.0.1.14 will be configured as route reflector client for address-family vpnv4
`- address_family_type: vrf`	Adress family type vrf will be configured under router bgp
`vrfs:`	Multiple Vrf 's name will be define under this
`- vrf_name: 1001`	Address family type vrf with name <1001> will be configured under router bgp

RSVP Configuration

Template File for RSVP

ocnos_rsvp.j2

```
{%if rsvp is defined %}
router rsvp
{% if rsvp.rsvppath is defined %}
{% for path in rsvp.rsvppath -%}
rsvp-path {{ path.name }} mpls
 {% for hop in path.hops -%}
 {{ hop }} strict
 {% endfor %}
!
{% endfor %}
{% endif %}
{% if rsvp.interfaces is defined %}
{% for interface in rsvp.interfaces -%}
interface {{ interface.name }}
```

```
 {{ interface.command }}
!
{% endfor %}
{% endif %}
{% if rsvp.trunks is defined %}
{% for trunk in rsvp.trunks -%}
rsvp-trunk {{ trunk.name }} ipv4
 {{trunk.FRR}}
{% if trunk.FRR_TYPE is defined %}
 {{ trunk.FRR_TYPE }}
{% endif %}
 primary path {{ trunk.primary_path }}
 primary label-record
{% if trunk.secondary_path is defined %}
 secondary path {{ trunk.secondary_path }}
{% endif %}
 from {{trunk.ingress}}
 to {{ trunk.egress }}
!
{% endfor %}
{% endif %}

{% if rsvp.bypass is defined %}
{% for bypass in rsvp.bypass -%}
rsvp-bypass {{ bypass.name }}
 from {{bypass.ingress}}
 to {{ bypass.egress }}
 label-record
 path {{ bypass.path }}
 exit
{% endfor %}
{% endif %}
{%endif%}
```

Parameter File for RSVP

sw2.yml

`rsvp:`	Router RSVP configurations
`trunks:`	Rsvp trunk Details
`- name: TO_AR-1`	Configures the rsvp-trunk with name TO_AR-1
`ingress: 36.0.0.2`	Configures the starting point of the trunk as 36.0.0.2
`egress: 10.0.1.14`	Configures the end point of the trunk as 10.0.1.14
`FRR: primary fast-reroute protection facility`	Configures the FRR as FACILITY
`FRR_TYPE: primary fast-reroute node-protection`	Configure the type of FRR type as <node-protection>

`primary_path: TO_AR-1`	Configures the trunk with a primary path TO_AR-1
`secondary_path: TO_AR-1_Sec`	Configures the trunk with a secondary path TO_AR-1_Sec
`bypass:`	Bypass configuration
`- name: TO_AR-1_BKUP`	Configures the rsvp-bypass with name TO_AR-1_BKUP
`ingress: 36.0.0.2`	Configures the starting point of the bypass as 36.0.0.2
`egress: 10.0.1.14`	Configures the end point of the trunk as 10.0.1.14
`path: TO_AR-1_BKUP`	Configures the bypass with path TO_AR-1_BKUP
`rsvppath:`	Rsvp path configuration
`- name: TO_AR-2_BKUP`	Configures the RSVP path with name TO_AR-2_BKUP
`hops:`	Hops configuration under path TO_AR-2_BKUP
`- 10.110.140.110`	Configures 10.110.140.110 as a strict hop under rsvp-path TO_AR-2_BKUP
`- 101.1.1.2`	Configures 101.1.1.2 as a strict hop under rsvp-path TO_AR-2_BKUP
`- 101.3.1.2`	Configures 101.3.1.2 as a strict hop under rsvp-path TO_AR-2_BKUP
`- 111.2.1.2`	Configures 111.2.1.2 as a strict hop under rsvp-path TO_AR-2_BKUP
`- 10.0.1.15`	Configures 10.0.1.15 as a strict hop under rsvp-path TO_AR-2_BKUP
`interfaces:`	Interface configuration for rsvp
`- name: xe1`	Configures Interface xe1 command
`command: enable-rsvp`	Configures enable-rsvp command under interface xe1

QoS Configuration

Template File for QOS

ocnos_qos.j2

```
{%if QOS is defined %}
qos enable
qos statistics
!
{% for classmap in  QOS.classmap -%}
{% if classmap.protocol == "dscp" %}
class-map {{ classmap.matchtype }} {{ classmap.name }}
 match {{ classmap.protocol }} {{ classmap.dscptype }}
{% endif %}
{% if classmap.protocol == "queuing" %}
class-map {{ classmap.matchtype }} {{ classmap.protocol }} {{ classmap.que_name }}
 match {{ classmap.classification }} {{ classmap.name }}
{% endif %}
{% if classmap.protocol == "vlan" %}
class-map {{ classmap.matchtype }} {{ classmap.name }}
```

```
 match {{ classmap.protocol }} {{ classmap.dscptype }}
{% endif %}
!
{% endfor %}
{% for policymap in  QOS.policymap -%}
{% if policymap.qos_name is defined %}
{% for param in policymap.params %}
{% if param.matchtype is defined %}
{%if param.val is defined %}
{% if param.val==1%}
policy-map {{ param.matchtype }} {{ param.protocol }} {{ policymap.qos_name }}
{% endif %}
{% endif %}
{% endif %}
 class {{ param.name }}
{% if param.cosvalue is defined %}
  set queue {{param.cosvalue}}
{% endif %}
  exit
{% endfor %}
!
{% endif %}
{% if policymap.que_name is defined %}
{% for param in policymap.params %}
{%if  param.val is defined %}
{%if  param.val ==1 %}
policy-map {{ param.matchtype }} {{ param.protocol }} {{ policymap.que_name }}
{% endif %}
{% endif %}
 class type  {{ param.protocol }}  {{param.classmap_name}}
  shape {{param.shape_rate}}
  exit
{% endfor %}
{% endif %}
{% endfor %}
!
{% for interface in QOS.interfaces -%}
interface {{interface.name}}
{% for policy in interface.policy_type -%}
{% if policy.type == "qos" %}
 service-policy type {{policy.type}} input {{policy.policy_name}}
{% endif %}
{% if policy.type == "queuing" %}
 service-policy type {{policy.type}} output {{policy.policy_name}}
{% endif %}
{% endfor %}
{% endfor %}
{%endif%}
```

Parameter File for QOS

sw2.yml

QOS:	QOS configuration
classmap:	Class-map configuration
- matchtype: match-all	Configures the Logical-AND of all match statements under this class-map
name: DSCP-AF11	Specify a class-map name (Max Size 32)
protocol: dscp	Configures the protocol type dscp under class-mapp DSCP-AF11
dscptype: af11	Configures Match type of dscp as af11 under DSCP-AF1
- matchtype: match-all	Configures the Logical-AND of all match statements under this class-map
name: VLAN100	Specify a class-map name (Max Size 32)
protocol: vlan	Configure te protocol type vlan under class-map VLAN100
dscptype: 500	Configures the vlan id under the class-map VLAN100
- matchtype: type	Configures the type of match statements under this class-map
protocol: queuing	Configure the protocol as queuing
que_name: defaultq	Configure the class-map defaultq name for protocol queuing
classification: service-template	Configures the classification type as service-template under defaultq
name: vpws	Configures the name of the service-template
- matchtype: type	Configures the type of match statements under this class-map
protocol: queuing	Configure the protocol as queuing under class-map
que_name: matchall	Configure the class-map matchall name for protocol queuing
classification: vlan	Configures the classification type as vlan
name: 1001	Configures the if of the vlan as 1001
policymap:	Policymap configuration
- qos_name: ALL-VLANs	Configure the name of the policy-map as ALL-VLANs
params:	Configure the parameter for policy-map
- matchtype: type	Configures the type command for policy-map
protocol: qos	Configure the type of protocol as qos for policy-map
name: VLAN100	Configure the class name as VLAN100 under policy-map ALL-VLANs
val: 1	Define the occurrence of under the policy-map
- qos_name: DSCP-ALL	Configure the name of the policy-map as DSCP-ALL
params:	Configure the parameter for policy-map
- matchtype: type	Configures the type command for policy-map
protocol: qos	Configure the type of protocol as qos for policy-map
name: DSCP-AF11	Configure the class name as DSCP-AF11 under policy-map DSCP-ALL
cosvalue: 1	Configures the queue value to be taken for matched traffic

Under class DSCP-AF11	
val:1	Define the occurrence of under the policy-map
- matchtype: type	Configures the type command for policy-map
protocol: qos	Configure the type of protocol as qos for policy-map
name: DSCP-AF12	Configure the class name as DSCP-AF12 under policy-map DSCP-ALL
cosvalue: 1	Configures the queue value to be taken for matched traffic
Under class DSCP-AF12	
val:2	Define the occurrence of under the policy-map
- que_name: shaper	Configure the name of the queue as shaper
params:	Configure the parameter for policy-map
- matchtype: type	Configures the type command for policy-map
protocol: queuing	Configure the type of protocol as queuing for policy-map
name: DSCP-EF	Configure the class name as DSCP-EF under policy-map shaper
val: 1	Define the occurrence of under the policy-map
classmap_name: defaultq	Configures the class-map name defaultq
shape_rate: 10 gbps	Configures the shape rate as 10 gbps under under class DSCP-EF
interfaces:	Interface configuration
- name: eth3	Configure the interface eth3 command
policy_type:	Policy configuration under interface
- type: qos	Configures the service policy type as qos
policy_name: ALL-VLANs	Configures the input policy name as ALL-VLANs for policy-type qos
- type: queuing	Configures the service policy type as queuing
policy_name: shaper	Configures the output policy name as shaper for policy-type queuing

Timing (PTP) and Synchronization (SyncE) Configuration

Template File for PTP and SyncE

ocnos_ptp_synce.j2

```
{%if PTP_SYNCE is defined%}
synce
ptp clock profile g8275.1
 number-ports {{ PTP_SYNCE.numberport}}
{% if PTP_SYNCE.ptp is defined %}
 {% for ptp_params in PTP_SYNCE.ptp -%}
 clock-port  {{ ptp_params.clockport }}
 {% if ptp_params.interface is defined %}
  network-interface {{ptp_params.interface}}
```

```
 {% endif %}
  exit
 {% endfor %}
{% endif %}
!
{% if PTP_SYNCE.interfaces is defined %}
{% for interface in  PTP_SYNCE.interfaces -%}
interface {{ interface.name }}
 synce
  mode {{ interface.mode }}
{% if interface.inputsource is defined %}
  input-source {{ interface.inputsource }}
{%endif%}
{% if interface.outputsource is defined %}
  {{interface.outputsource}}
{%endif%}
{% if interface.waittorestore is defined %}
  wait-to-restore {{ interface.waittorestore }}
{% endif %}
 exit
{% endfor %}
{% endif %}
{%endif%}
```

Sample Parameter File for PTP and SyncE

sw2.yml

`PTP_SYNCE:`	PTP Synce configurations
`  numberport: 5`	Configure the numpber-port value as 1 under ptp clock profile g8275.1
`  ptp:`	Ptp configuration
`    - clockport: 1`	Configure the clock-port value as 1 under ptp clock profile g8275.1
`      interface: xe19`	Configures network-interface as xe19 under clock-port 1
`  interfaces:`	Interface configurations
`    - name: xe4`	configures interface xe4
`      mode: synchronous`	Enables mode synchronous under synce
`      inputsource: 10`	Configure input-source as 10 under interface if defined
`      outputsource: output-source`	Configure output-source under interface if defined
`      Waittorestore: 1`	Configure wait-to-restore as 1 under interface if defined

VPWS Configuration

Template File for VPWS

ocnos_vpws.j2

```
{% if VPWS.pseudowire is defined %}
{% for vpws in VPWS.pseudowire -%}
  mpls l2-circuit {{ vpws.vc_name }} {{ vpws.vc_id }} {{ vpws.peer }}
{% endfor %}
{% for template in VPWS.service_template -%}
service-template {{ template.name}}
{% if template.vlan is defined %}
 match outer-vlan {{ template.vlan }}
{% endif %}
{% if template.operation is defined %}
{% if template.operation == "pop" %}
 rewrite ingress {{ template.operation }} outgoing-tpid {{ template.tpid }}
{% endif %}
{% if template.operation == "translate" %}
 rewrite ingress {{ template.operation }} {{ template.translate_vlan }} outgoing-tpid
{{template.tpid }}
{% endif %}
{% endif %}
!
{% endfor %}
{% for interface in VPWS.interfaces -%}
interface {{ interface.name }}
 switchport
 {% for binding in interface.vpws_binding -%}
 mpls-l2-circuit {{ binding.instance}} service-template {{binding.service_template}}
 {% endfor %}
!
{% endfor %}
{% endif %}
```

Sample Parameter File for VPWS

sw2.yml

`VPWS:`	VPWS configurations
`  pseudowire:`	Pseudowire(PW) instance configuration
`    - vc_name: vpws`	Configures the name of PW as vpws
`      vc_id: 1`	Configures the PW id as 1
`      peer: 36.0.0.8`	Configures PW peer id as 36.0.0.8

`- vc_name: vpws-2`	Configures the name of PW as vpws-2
`vc_id: 2`	Configures the PW id as 2
`peer: 10.0.1.14`	Configures PW peer id as 10.0.1.14
`service_template:`	Service-template configuration
`- name: vpws`	Configure service-template name as vpws
`vlan: 555`	Configure match-outer vlan as vlan-id 555 under service-template vpws
`operation: pop`	Configure rewrite ingress operation as pop under service-template vpws
`tpid: dot1.q`	Configure outgoing tpid as dot1.q under service-template vpws
`- name: vpws-2`	Configure service-template name as vpws-2
`vlan: 600`	Configure match-outer vlan as vlan-id 600 under service-template vpws
`interfaces:`	Interface configuration
`- name: xe20`	Configure interface xe20 command
`vpws_binding:`	Configuration to bing vpws instance with service -template
`- instance: vpws`	Configure the binding of instance name vpws
`service_template: vpws`	Configure the vpws binding with service-template vpws under interface
`- instance: vpws-2`	Configure the binding of instance name vpws-2
`service_template: vpws-2`	Configure the vpws binding with service-template vpws -2 under interface

L3VPN Configuration

Template File for L3VPN

ocnos_l3vpn.j2

```
{% if L3VPN.vrfs is defined %}
{% for vrf in L3VPN.vrfs -%}
  ip vrf {{ vrf.vrf_name }}
  rd {{ vrf.rd_1 }}:{{ vrf.rd_2 }}
  route-target both {{ vrf.rt_1 }}:{{ vrf.rt_2 }}
{% endfor %}

{% for interface in L3VPN.vrf_interfaces -%}
  interface {{ interface.interface_name }}
  ip vrf forwarding {{ interface.vrf_name }}
  ip address {{ interface.address }}

{% endfor %}
{% endif %}
```

Sample Parameter File for L3VPN

sw2.yml

`L3VPN:`	L3VPN configurations
`  vrfs:`	VRF instance configuration
`    - vrf_name: 1001`	Configures the name of VRFas 1001
`      rd_1: 36.0.0.2`	Configures the ASN or IP-address value depending on the ASN:nn_or_IP-address:nn route distinguisher value used .
`      rd_2: 1001`	Configures the nn on the ASN:nn_or_IP-address:nn route distinguisher value.
`      rt_1: 65001`	Configures the ASN or IP-address value depending on the ASN:nn_or_IP-address:nn format used for route-target
`        rt_2: 1001`	Configure nn value of the route-target
`  vrf_interfaces:`	Vrf interface configuration
`    - interface_name: eth2`	Configure interface eth2
`      vrf_name: 1001`	Configure the interface as part of the vrf 1001
`      address: 19.19.19.1/24`	Configure the ip address 19.19.19.1/24on the vrf interface

Route Map Configuration

Template File for Route Map

ocnos_route_map.j2

```
{%if Route_Map is defined%}
{% for routemap in Route_Map.params -%}
route-map {{ routemap.name }} {{routemap.permission}} {{routemap.seq_no}}
{% if routemap.match=="address" %}
 {{ routemap.operation}} {{ routemap.protocol }} {{ routemap.match }} prefix-list
{{routemap.prefix_list}}
{%else%}
 {{ routemap.operation}} {{ routemap.protocol }} {{ routemap.match }}
{{routemap.prefix_list}}
{%endif%}
!
{% endfor %}
{%endif%}
```

Sample Parameter File for Route Map

sw2.yml

`Route_Map:`	Route map configurations
`  params:`	Route map parameters configuration

- name: NEXTHOP_SELF	Configures the name of the route_map
permission: permit	Configure the permission type as permit for route map
seq_no: 10	Configure the sequence no. as 10
operation: set	Configures the operation type under route-map as set
protocol: vpnv4	Configures the protocol as vpnv4
match: next-hop	Configures the match-type as next-hop under route-map NEXTHOP_SELF
prefix_list: 36.0.0.1	Configure the matching prefix as 36.0.0.1 .we can define prefix list name also if it is created .
- name: LO_RED_TO_0	Configures the name of the route_map
permission: permit	Configure the permission type as permit for route map
seq_no: 10	Configure the sequence no. as 10
operation: match	onfigures the operation type under route-map as match
protocol: ip	Configures the matching protocol as ip
match: address	Configures the match-type as addressunder route-map LO_RED_TO_0
prefix_list: LO_RED_TO_0	Configure the matching prefix list name

Prefix List Configuration

Template File for Prefix List

ocnos_prefix_list.j2

```
{%if prefix_list is defined %}
{% for prefixlist in prefix_list.params -%}
{% if prefixlist.eq is defined %}
ip prefix-list {{ prefixlist.name }}
 {{prefixlist.seq_no}} {{prefixlist.permission}}
{{prefixlist.prefix}} eq {{ prefixlist.eq }}
{% else %}
ip prefix-list {{ prefixlist.name }}
 {{prefixlist.seq_no}} {{prefixlist.permission}} {{prefixlist.prefix}}
{% endif %}
!
{% endfor %} end
{%endif%}
```

Sample Parameter File for Prefix List

sw2.yml

prefix_list:	prefix-list configurations
params:	prefix-list parameters configuration

- name: only_lo	Configures the name of the prefix-list
seq_no: seq 5	Configures the sequence no. to give the priority to the matched
prefixes	
permission: permit	Configure the permission type as permit for prefix-list
prefix: 36.0.0.1/24	Configure the prefix to matched
eq: 32	If defined it wiill confgigure the Exact prefix length to be matched as 32

ACL Configuration

Template File for ACL

ocnos_acl.j2

```
{%if ACL is defined%}
{% for acl in ACL.params -%}
  ip access-list {{ acl.name }}
  {{acl.seq_no}} {{acl.permission}} {{acl.protocol}} {{acl.prefix}} {{acl.dst}}
{% endfor %}
end
{%endif%}
```

Sample Parameter File for ACL

sw2.yml

ACL:	ACL configurations
params:	ACL parameters configuration
- name: only_lo	Configures the name of the ACL
seq_no: 10	Configures the sequence no. to give the priority to the matched prefixes
permission: permit	Configure the permission type as permit for acl
protocol: any	Configure any command to match any type of protocol packet to match
prefix: 36.0.0.0/24	Configure the prefix to matched
dst: any	Configure the destination address as any
- name: only_lo	Configures the name of the ACL
seq_no: 11	Configures the sequence no. to give the priority to the matched prefixes
permission: deny	Configure the permission type as deny for acl

SNMP Configuration

Template File for SNMP

ocnos_snmp.j2

```
snmp-server enable snmp vrf {{ snmp.vrf }}
snmp-server view {{ snmp.viewname }} {{ snmp.oid }} included vrf management
snmp-server community {{ snmp.communame }} group network-admin vrf management
{% if snmp.community is defined %}
snmp-server community {{ snmp.community }} group network-operator vrf management
{% endif %}
{% if snmp.hosttest is defined %}
snmp-server host {{ snmp.hosttest }} traps version 2c test udp-port 161 vrf management
{% endif %}
snmp-server host {{ snmp.hostpub }} traps version 2c public udp-port 162 vrf management
{% for traps in snmp.traps -%}
snmp-server enable traps {{ traps.daemon }}
{% endfor -%}
```

Parameter File for SNMP

sw2.yml

`Snmp`	Set SNMP service
`Traps`	globally enable snmp traps
`- daemon: bgp`	Enable bgp notification trap in global configuration mode
`- daemon: isis`	Enable isis notification trap in global configuration mode
`- daemon: pwdelete`	Enable pwdelete notification trap in global configuration mode
`- daemon: pw`	Enable pw notification trap in global configuration mode
`- daemon: mpls`	Enable mpls notification trap in global configuration mode
`- daemon: ospf`	Enable ospf notification trap in global configuration mode
`- daemon: rsvp`	Enable rsvp notification trap in global configuration mode
`vrf: management`	Configure vrf name as < snmp-server enable snmp vrf management> to enable snmp
`viewname: all`	Globally Configure viewname as < snmp-server view all.1 included vrf management>
`oid: .1`	Specify the OID-Tree in global configs
`community: test`	Configure community name as test
`communame: public`	Configure community name as public
`hosttest: 10.12.6.247`	Configure snmp-server host 10.12.6.247 traps version 2c public udp-port 161 vrf management command globlly
`hostpub: 10.12.47.72`	Configure snmp-server host 10.12.47.72 traps version 2c public udp-port 162 vrf management command globlly

ISIS Configuration

Template File for ISIS

ocnos_isisagg.j2

```
key chain {{ key.chain }}
 key {{ key.keyid }}
  key-string encrypted {{ key.passwd }}
  exit
{% for isis in isis.proc1 -%}
router isis {{ isis.processid }}
{% if isis.istype  is defined %}
 is-type {{ isis.istype }}
{%endif%}
{% if isis.mode  is defined %}
 authentication mode {{ isis.mode }} {{ isis.level }}
 authentication key-chain isis {{ isis.level }}
{%endif%}
{% if isis.level is defined %}
 spf-interval-exp {{ isis.spfvalue }} {{isis.spfinmili }}
{%endif%}
{% if isis.level1 is defined %}
 spf-interval-exp {{ isis.level1 }} {{ isis.spfvalue }} {{isis.spfinmili }}
{%endif%}
{% if isis.trafficeng  is defined %}
 metric-style wide {{ isis.trafficeng }}
 mpls traffic-eng {{ isis.trafficeng }}
{%endif%}
{% if routerid.address is defined %}
 mpls traffic-eng router-id {{ routerid.address }}
{% endif%}
{% if isis.capability is defined %}
 capability {{ isis.capability }}
{% endif%}
{% if isis.dynamic is defined %}
 dynamic-hostname
{% endif%}
 bfd {{ isis.bfd }}
net {{ isis.net }}
{% if isis.metric  is defined %}
 redistribute isis 1 metric {{ isis.metric }} {{ isis.level }} route-map {{ isis.word }}
{%endif%}
{% if isis.passive is defined %}
 passive-interface {{ isis.passive }}
{%endif%}
 exit
{% for interface in isis.interfaces -%}
```

```
 interface {{ interface.name }}
 ip router isis {{ interface.isis }}
 {% if interface.isisnw  is defined %}
 isis network {{ interface.isisnw }}
 {%endif%}
 exit
{% endfor -%}
{% endfor -%}
```

Parameter File for ISIS

sw2.yml

`Key`	authentication key management configuratioin
`  chain: isis`	Configure key chain isis command globally
`  keyid: 1`	Configure key identifier number under authentication key management
`  passwd: 0x46ff28ed3cbff32e`	Configure key-string encrypted 0x46ff28ed3cbff32e command under key id
`Isis:`	Router isis configs
`proc1:`	ISIS router configuration details
`processid: 1`	Configure router isis process id 1
`istype: level-1`	Configure IS Level 1 for this isis routing process
`level: level-1`	Configure authentication mode md5 level as 1 under router isis 1
`spfvalue: 0`	Configure spf-interval-exp 0 0 command under router isis 1
`spfinmili: 0`	Configure SPF calculation in milliseconds in spf-interval-exp 0 0 command under router isis 1
`dynamic: dynamic-hostname`	Configure dynamic hostname
`net: 49.3600.3600.9608.00`	Configure net: 49.0002.0000.0000.0099.00 under router isis 0
`bfd: all-interfaces`	Enable BFD on all interfaces
`interfaces:`	Interfaces details
`  - name: xe4`	Configure interface xe4 command
`    isis: 1`	Configure ip router isis 1 command under interface xe4
`   network: point-to-point`	Configure isis network point-to-point command
`  - name: xe2`	Configure interface xe2 command
`    isis: 1`	Configure ip router isis 1 command under interface xe2
`    network: point-to-point`	Configure interfacevlan1.1001 command
`  - name: lo`	Configure interface lo command
`   isis: 1`	Configure ip router isis 1 command under interface loopback

Interface Configuration

Template File for Interface Configuration

ocnos_interface.j2

```
{% for interface in interfaces.ifnames -%}
interface {{ interface.ifname }}
{%if interface.loadinterval is defined %}
 load-interval {{ interface.loadinterval }}
{%endif%}
{% if "lo" in  interface.ifname  %}
 ipv6 address {{ interface.address1 }}
 bfd session {{ interface.bfdsession }} multihop
{%else%}
{%endif%}
{%if interface.switch  is defined %}
 {{ interface.switch }}
{%endif%}
{%if interface.speed  is defined %}
 speed  {{ interface.speed }}
{%endif%}
{% if interface.bridge  is defined %}
 bridge-group {{ interface.bridge }}
{%endif%}
{% if interface.mode  is defined %}
 switchport mode {{ interface.mode }}
{%endif%}
{% if interface.vlan  is defined %}
 switchport trunk allowed vlan {{ interface.vlan }}
{%endif%}
{% if interface.address  is defined %}
 ip address {{ interface.address }}
{%endif%}
{% if interface.mtu  is defined %}
 mtu {{ interface.mtu }}
{%endif%}
{% if interface.groupid  is defined %}
 channel-group {{ interface.groupid }} mode {{ interface.state }}
 exit
{% endif %}
{% endfor %}
```

Parameter File for Interface configuration

sw2.yml

interfaces:	Interface configuration
ifnames:	Interface configuration details
- ifname: xe4	Configure interface xe4
address:10.110.140.20/31	Configure ip address 10.110.140.20/31 command under xe4
mtu: 9216	Configure mtu 9216 under xe4
- ifname: xe2	Configure interface xe2
loadinterval: 30	Configure load interval 30 under interface xe4
address: 10.110.140.61/31	Configure ip address 10.110.140.61/31 command under xe2
mtu: 9216	Configure mtu 9216 under xe2
- ifname: vlan1.1001	Configure interface vlan1.1001 command
address: 192.168.21.212/24	Configure ip address 192.168.21.212/24 command under interface vlan1.1001
- ifname: vlan1.101	Configure interface vlan1.101 command
address: 101.101.101.5/30	Configure ip address 101.101.101.5/30 command under interface vlan1.101
- ifname: lo	Configure interface loopback
address: 36.0.0.8/32	Configure ip address 36.0.0.8/32 command under lopback interfaces
address1: ::1/128	Configure ipv6 address ::1/128 command under loopback interface
bfdsession: 36.0.0.8 36.0.0.1	Configure bfdsession: 36.0.0.8 36.0.0.1 command under loopback interface
- ifname: xe0	Configure interface xe0
switch: switchport	Configure switchport under xe0 interface
bridge: 1	Configure bridge-group 1 under xe0 interface
mode: trunk	Configure switch mode as trunk under xe0 interfaces
vlan: all	Configure switchport trunk allowed vlan all command under int xe0
loadinterval: 30	Configure load-interval 30 under xe0 interfaces
- ifname: xe22	Configure interface xe22 command
switch: switchport	Configure switchport command under xe22 command
bridge: 1	Configure bridge-group 1 under xe22 interface
mode: trunk	Configure switch mode as trunk under xe22 interfaces
vlan: add 101,1001	Configure switchport trunk allowed vlan add 101,1001 command under int xe22
loadinterval: 30	Configure load-interval 30 under xe22 interfaces
mtu: 9216	Configute mtu 9216 under xe22 interface
- ifname: xe6	Configure interface xe6 command
switch: switchport	Configure switchport command under xe6 command

bridge: 1	Configure bridge-group 1 under xe6 interface
mode: access	Configure switch mode as access under xe6 interfaces
- ifname: xe10	Configure interface xe10 command
speed: 1g	Configure speed 1g under interface xe10
- ifname: xe11	Configure interface xe11 command
speed: 1g	Configure speed 1g under interface xe11
- ifname: ce0	Configure interface ce0 command
speed: 40g	Configure speed 40g under interface ce0

BFD Configuration

Template File for BFD

ocnos_bfd.j2

```
bfd interval {{ bfd.interval }} minrx {{ bfd.minrx }} multiplier {{ bfd.multiplier }}
{% for bfd in bfd.multihoppeer -%}
{% if bfd.address is defined %}
 bfd multihop-peer {{ bfd.address }} interval {{ bfd.interval }} minrx {{ bfd.minrx }}
multiplier {{ bfd.multiplier }}
{% endif %}
{% endfor -%}
```

Parameter File for BFD

sw2.yml

|

bfd	Bfd configuration
interval: 3	Configure globally BFD transmit Interval BFD configuration as 3
minrx: 3	Configure bfd interval 3 minrx 3 multiplier 3 command globally
multiplier: 3	Configure bfd interval 3 minrx 3 multiplier 3 command globally
multihoppeer:	Configure multihoppeer configuration
- address: 36.0.0.1	Configure bfd multihop-peer 36.0.0.1 interval 300 minrx 300 multiplier 5 command globally
interval: 300	Configure bfd multihop-peer 36.0.0.1 interval 300 minrx 300 multiplier 5 command globally
minrx: 300	Configure bfd multihop-peer 36.0.0.1 interval 300 minrx 300 multiplier 5 command globally
multiplier: 5	Configure bfd multihop-peer 36.0.0.1 interval 300 minrx 300 multiplier 5 command globally

Hardware Profile Configuration

Template File for Hardware Profile

Ocnos_hardwareprofile.j2

```
hardware-profile filter {{ hardware.filter }} enable
{% for statistics in hardware.statistics -%}
hardware-profile statistics {{ statistics.value }} enable
{% endfor -%}
```

Parameter File for Hardware Profile

sw2.ym

l

`hardware`	Hardware configuration
`filter: qos-ext`	Configure hardware-profile filter qos-txt command under config mode
`statistics:`	Hardware statistics configuration
`- value: ingress-acl`	Configure hardware-profile statistics ingress ACL command under config mode
`- value: mpls-pwe`	Configure hardware-profile statistics mpls-pwe command under config mode

NTP Configuration

Template File for NTP

Ocnos_ntp.j2

```
feature ntp vrf management
{% for ntp in ntp.states -%}
 ntp {{ ntp.state }} vrf management
{% endfor -%}
{% for server in ntp.server -%}
 ntp server {{ server.address }} vrf management
{% endfor -%}
logging server {{ ntp.logserver }} 5 vrf management
router-id {{ ntp.routerid }}
service unsupported-transceiver
```

Parameter File for NTP

sw2.yml

Ntp	NTP configuration
states:	Configure ntp states details
-state: enable	Enable ntp
- state: logging	Configure ntp logging vrf management command
server:	Configure ntp server address
- address: 216.239.35.4	Configure ntp server 216.239.35.4 vrf management command under config mode
logserver: 10.12.47.72	Configure logging server 10.12.47.72 5 vrf management
routerid: 36.0.0.8	Configure router-id 36.0.0.8 command globally

VLAN Configuration

Template File for VLAN

Ocnos_vlan.j2

```
{% if vlan.protocol is defined %}
bridge 1 protocol {{ vlan.protocol }} vlan-bridge
{%endif %}
vlan {{ vlan.level }}
{% for range in vlan.range -%}
vlan {{ range.value }} bridge {{ vlan.bridge }} state {{ vlan.state }}
{% endfor -%}
```

Parameter File for VLAN

Sw2.yml

Vlan	Vlan configuration
level: database	Configure VLAN database
range:	Vlan range configuration
- value: 101	Configure vlan 101 bridge 1 state enable command under vlan database
- value: 1001	Configure vlan 1001 bridge 1 state enable command under vlan database
bridge: 1	Configure bridge 1 under vlan database
state: enable	Configure vlan bridge 1 state as enable under vlan database
protocol: rstp	Configure bridge 1 protocol rstp vlan-bridge command globally

LLDP Configuration

Template File for LLDP

Ocnos_lldp.j2

```
lldp run
{% for lldp in lldp.lldp1 -%}
interface {{ lldp.name }}
 {{ lldp.lagent }}
  set lldp {{ lldp.state }} {{ lldp.mode }}
  lldp tlv {{ lldp.MED }} {{ lldp.powerviamdi }} select
  set lldp {{ lldp.port }} {{ lldp.ifname}}
  set lldp management-address-tlv ip-address
  {% for tlvselect in lldp.tlvselect -%}
   lldp tlv basic-mgmt {{ tlvselect.mgmt }} select
  {% endfor -%}
  exit
{% endfor -%}
```

Parameter File for LLDP

sw2.yml

`Lldp`	lldp configuration
`lldp1:`	lldp configuration details
`- name: xe2`	Configure interface xe2 command
`lagent: lldp-agent`	Enable lldp agent under xe2 interface
`state: enable`	Configure set lldp enable txrx command under interface xe2
`mode: txrx`	Configure lldp mode as txrx under interface xe2
`MED: med`	Configure lldp tlv-select med media-capabilities command under lldp-agent
`powerviamdi: media-capabilities`	Configure extended-power-via-mdi media-capailities in lldp tlv-select med media-capabilities command under lldp-agent
`port: port-id-tlv`	Configure port-id-tlv in set lldp port-id-tlv if-name command under interface xe2
`ifname: if-name`	Configure if-name as port-id-TLV in set lldp port-id-tlv if-name command under interface xe2
`tlvselect :`	tlv select configuration
`- mgmt: port-description`	Configure lldp tlv-select basic-mgmt port-description command under interface xe2
`- mgmt: system-name`	Configure lldp tlv-select basic-mgmt system-name command under interface xe2
`- mgmt: system-capabilities`	Configure lldp tlv-select basic-mgmt system-capabilities under interface xe2

- mgmt: system-description	Configure lldp tlv-select basic-mgmt system-description under interface xe2
- mgmt: management-address	Configure lldp tlv-select basic-mgmt management-address under interface xe2
- name: xe10	Configure interface xe10 command
lagent: lldp-agent	Enable lldp agent under xe10 interface
state: enable	Configure set lldp enable txrx command under interface xe10
mode: txrx	Configure lldp mode as txrx under interface xe10
MED: med	Configure lldp tlv-select med media-capabilities command under lldp-agent
powerviamdi: media-capabilities	Configure extended-power-via-mdi media-capailities in lldp tlv-select med media-capabilities command under lldp-agent
port: port-id-tlv	Configure port-id-tlv in set lldp port-id-tlv if-name command under interface xe10
ifname: if-name	Configure if-name as port-id-TLV in set lldp port-id-tlv if-name command under interface xe10
tlvselect :	tlv select configuration
- mgmt: port-description	Configure lldp tlv-select basic-mgmt port-description command under interface xe10
- mgmt: system-name	Configure lldp tlv-select basic-mgmt system-name command under interface xe10
- mgmt: system-capabilities	Configure lldp tlv-select basic-mgmt system-capabilities under interface xe10
- mgmt: system-description	Configure lldp tlv-select basic-mgmt system-description under interface xe10
- mgmt: management-address	Configure lldp tlv-select basic-mgmt management-address under interface xe10
- name: xe11	Configure interface xe11 command
lagent: lldp-agent	Enable lldp agent under xe11 interface
state: enable	Configure set lldp enable txrx command under interface xe11
mode: txrx	Configure lldp mode as txrx under interface xe11
MED: med	Configure lldp tlv-select med media-capabilities command under lldp-agent
powerviamdi: media-capabilities	Configure extended-power-via-mdi media-capailities in lldp tlv-select med media-capabilities command under lldp-agent
port: port-id-tlv	Configure port-id-tlv in set lldp port-id-tlv if-name command under interface xe11
ifname: if-name	Configure if-name as port-id-TLV in set lldp port-id-tlv if-name command under interface xe11
tlvselect :	tlv select configuration
- mgmt: port-description	Configure lldp tlv-select basic-mgmt port-description command under interface xe11
- mgmt: system-name	Configure lldp tlv-select basic-mgmt system-name command under interface xe11
- mgmt: system-capabilities	Configure lldp tlv-select basic-mgmt system-capabilities under interface xe11
- mgmt: system-description	Configure lldp tlv-select basic-mgmt system-description under interface xe11

- mgmt: management-address	Configure lldp tlv-select basic-mgmt management-address under interface xe11
- name: xe4	Configure interface xe4 command
state: enable	Enable lldp agent under xe4 interface
lagent: lldp-agent	Configure set lldp enable txrx command under interface xe4
mode: txrx	Configure lldp mode as txrx under interface xe4
MED: med	Configure lldp tlv-select med media-capabilities command under lldp-agent
powerviamdi: media-capabilities	Configure extended-power-via-mdi media-capailities in lldp tlv-select med media-capabilities command under lldp-agent
port: port-id-tlv	Configure port-id-tlv in set lldp port-id-tlv if-name command under interface xe4
ifname: if-name	Configure if-name as port-id-TLV in set lldp port-id-tlv if-name command under interface xe4
tlvselect :	tlv select configuration
- mgmt: port-description	Configure lldp tlv-select basic-mgmt port-description command under interface xe4
- mgmt: system-name	Configure lldp tlv-select basic-mgmt system-name command under interface xe4
- mgmt: system-capabilities	Configure lldp tlv-select basic-mgmt system-capabilities under interface xe4
- mgmt: system-description	Configure lldp tlv-select basic-mgmt system-description under interface xe4
- mgmt: management-address	Configure lldp tlv-select basic-mgmt management-address under interface xe4

Limitations

The following are the current limitations while configuring OcNOS through Ansible.

1. The following commands in OcNOS require the device to be rebooted to be effective.
   - hardware-profile
   - forwarding profile
   - maximum-paths
   - copy empty-config startup-config

   Ansible returns success while configuring these commands. However, the device needs to be rebooted to make these effective.

2. By default, `ANSIBLE_PERSISTENT_COMMAND_TIMEOUT` is set to 30 (seconds). While pushing large configs through Ansible which might be taking more time than this default timeout, it is suggested that to increase the `ansible_command_timeout` to appropriate value. In `group_vars/ocnos.yml`, it is suggested to add the below line with appropriate timeout value:

   ```
   ansible_command_timeout: 1800
   ```

3. While configuring the below commands, there are certain warning messages shown to the customer. Currently Ansible treats them as failure and returns failure even though it is successful. It is suggested that the user takes appropriate action while configuring these commands.
   - no ip vrf <vrf-id>

- While re-configuring shaping as part of QoS:

```
policy-map type queuing shaper
class type  queuing  defaultq
  shape 10 gbps
 exit
```

Appendix A Configuring LDP

The example below creates an Ansible playbook to configure the LDP protocol.

ocnos_ldp.j2

First create a template for the LDP configuration:

```
osboxes@osboxes:~/playbooks$ cat templates/ocnos_ldp.j2
router ldp
  {% for peer in ldp.peers %}
  targeted-peer ipv4 {{ peer.address }}
  exit
  {% endfor %}
  advertise-labels for only_lo to any
  exit

{% for interface in ldp_interfaces %}
interface {{interface.ldp_interface}}
  enable-ldp {{interface.ldp_protocol}}
  label-switching
{% endfor %}
```

sw2.yml

Next, provide the LDP configuration details in the appropriate `host_vars` file:

```
osboxes@osboxes:~/playbooks$ cat host_vars/sw2.yml
ldp:
 peers:
   - address: 1.1.1.1
   - address: 3.3.3.3

ldp_interfaces:
   - { ldp_interface: eth1, ldp_protocol: ipv4 }
   - { ldp_interface: eth2, ldp_protocol: ipv4 }
```

Here is the overall directory structure of the Ansible playbook and associated files:

```
osboxes@osboxes:~/playbooks$ tree
.
    ansible.cfg
    backup
    group_vars
          ocnos.yml
```

```
    hosts-net
    host_vars
          sw2.yml
    ldp-playbook.yml
    showldp-playbook.yml
    templates
           ocnos_ldp.j2
```

ansible.cfg

The following is the content of `ansible.cfg` file that points to the `hosts-net` inventory file.

```
osboxes@osboxes:~/playbooks$ cat ansible.cfg
[defaults]
inventory = hosts-net
host_key_checking = False
retry_files_enabled = False
interpreter_python = auto
osboxes@osboxes:~/playbooks
```

host-net

The following is the content of the `hosts-net` inventory file.

```
osboxes@osboxes:~/playbooks$ cat hosts-net
[ocnos]
sw2 ansible_host=10.12.9.105
osboxes@osboxes:~/playbooks$
```

ocnos.yml

The following is the content of `ocnos.yml` in the `group_vars` folder.

```
osboxes@osboxes:~/playbooks$ cat group_vars/ocnos.yml
ansible_connection: network_cli
ansible_network_os: ipinfusion.ocnos.ocnos
ansible_become: yes
ansible_become_method: enable
ansible_ssh_user: ocnos
ansible_ssh_pass: ocnos
osboxes@osboxes:
```

ldp-playbook.yml

The following is the playbook to push the configuration with `cli_config` module using the template created earlier:

```
(ansible) osboxes@osboxes:~/playbooks$ cat ldp-playbook.yml
---
```

```
  - hosts: ocnos
    gather_facts: no

    tasks:

      - name: configure LDP config on OcNOS
        cli_config:
          config: "{{ lookup('template', 'templates/{{  ansible_network_os }}_ldp.j2')
}}"
```

Running the Playbook

The following is the configuration on the OcNOS device before executing the Ansible Playbook:

```
OcNOS#show running-config ldp
!
!
OcNOS#
```

Execute the Ansible Playbook. Below are the logs that display:

```
(ansible) osboxes@osboxes:~/playbooks$ ansible-playbook ldp-playbook.yml

PLAY [ocnos]*********************************************************************

TASK [configure LDP config on OcNOS]*********************************************
changed: [sw2]

PLAY RECAP***********************************************************************
sw2                        : ok=1    changed=1    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0
```

Next, check the configurations on the OcNOS device, which should show the LDP configurations.

```
OcNOS#show running-config ldp
!
router ldp
 targeted-peer ipv4 1.1.1.1
  exit-targeted-peer-mode
 targeted-peer ipv4 3.3.3.3
  exit-targeted-peer-mode
 advertise-labels for only_lo to any
!
!
interface eth2
 enable-ldp ipv4
!
interface eth1
 enable-ldp ipv4
```

```
!
OcNOS#
```

showldp-playbook.yml

The playbook below shows how to check the runtime status of the LDP protocol through the `show ldp session` command. It is assumed that the neighboring nodes are configured accordingly to get the LDP session up and running:

```
osboxes@osboxes:~/playbooks$ cat showldp-playbook.yml
---

  - hosts: ocnos
    gather_facts: no

    tasks:
     - name: show LDP config ocnos
       cli_command:
         command: show ldp session
       register: result

     - name: debug
       debug:
         msg: "{{ result.stdout_lines }}"
osboxes@osboxes:~/playbooks$
```

When you run this playbook, the example below is its output. Parsing of the `show` command output needs to be done to determine if the runtime status of the protocol is correct:

```
(ansible) osboxes@osboxes:~/playbooks$ ansible-playbook showldp-playbook.yml

PLAY [ocnos]*********************************************************************

TASK [show LDP config ocnos]*****************************************************
ok: [sw2]

TASK [debug]*********************************************************************
ok: [sw2] => {
    "msg": [
        "Peer IP Address           IF Name    My Role    State      KeepAlive  UpTime",
        "3.3.3.3                   eth2       Passive   OPERATIONAL   30    03:58:20",
        "1.1.1.1                   eth1       Active    OPERATIONAL   30    03:58:20"
    ]
}

PLAY RECAP***********************************************************************
sw2                        : ok=2    changed=0    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0

(ansible) osboxes@osboxes:~/playbooks$
```