ipinfusion™

OcNOS®

Open Compute Network Operating System for Data Centers

Ansible

Version 7.0.1

July 2026

IP Infusion Inc. Proprietary

©2026 IP Infusion Inc. All Rights Reserved.

This documentation is subject to change without notice. The software described in this document and this documentation are furnished under a license agreement or nondisclosure agreement. The software and documentation may be used or copied only in accordance with the terms of the applicable agreement. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or any means electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's internal use without the written permission of IP Infusion Inc.

IP Infusion Inc.

3979 Freedom Circle, Suite 900

Santa Clara, CA 95054

+1 408-400-1900

http://www.ipinfusion.com/

For support, questions, or comments via E-mail, contact:

support@ipinfusion.com

Trademarks:

IP Infusion and are trademarks or registered trademarks of IP Infusion. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Use of certain software included in this equipment is subject to the IP Infusion, Inc. End User License Agreement at http://www.ipinfusion.com/license. By using the equipment, you accept the terms of the End User License Agreement.

CONTENTS

Contents ... **3**

Preface ... **6**
About this Guide ... 6
Audience ... 6
Conventions ... 6
IP Infusion Product Release Version ... 6
Related Documentation ... 7
Feature Availability ... 7
Migration Guide ... 7
IP Maestro Support ... 7
Technical Support ... 7
Technical Sales ... 7
Technical Documentation ... 7
Documentation Disclaimer ... 8
Comments ... 8
Command Line Interface ... 9
Overview ... 9
Chapter Organization ... 9
Command Line Interface Help ... 9
Command Completion ... 10
Command Abbreviations ... 10
Command Line Errors ... 11
Command Negation ... 11
Syntax Conventions ... 11
Variable Placeholders ... 12
Command Description Format ... 13
Keyboard Operations ... 13
Show Command Modifiers ... 14
Begin Modifier ... 14
Include Modifier ... 15
Exclude Modifier ... 15
Redirect Modifier ... 16
Last Modifier ... 16
String Parameters ... 16
Command Modes ... 17
Command Mode Tree ... 18
Transaction-based Command-line Interface ... 18

Overview ... **20**
Installing the OcNOS Collection from Ansible Galaxy ... 20
Set-up Ansible Files ... 21
Group Variables File ... 21

© 2026 IP Infusion Inc. Proprietary

Inventory File 21
Supported Modules 22
ipinfusion.ocnos.ocnos_facts 22
Sample Playbook 22
Sample Output 22
Return Values 23
cli_command 26
Parameters 26
Return Values 27
Samples 27
ipinfusion.ocnos.ocnos_command 28
Parameters 29
Return Values 32
Samples 32
cli_config 32
Parameters 32
Return Values 34
Samples 34
ipinfusion-ocnos-ocnos-config 35
Parameters 36
Return Values 38
Samples 39
ipinfusion.ocnos.ocnos_ping or net_ping 44
Parameters 44
Return Values 44
Sample 45
ipinfusion.ocnos.ocnos_bgp_facts 45
Sample Playbook 45
Sample Output 46
ipinfusion.ocnos.ocnos_isis_facts 47
Sample Playbook 47
Sample Output 47

Ansible Configuration 49

Jinja2 Templates for Configuring OcNOS 49
BGP Configuration 49
Template File for BGP 49
Parameter File for BGP 50
QoS Configuration 51
Template File for QOS 51
Parameter File for QOS 52
Route Map Configuration 54
Template File for Route Map 54
Sample Parameter File for Route Map 54
Prefix List Configuration 55
Template File for Prefix List 55
Sample Parameter File for Prefix List 55

© 2026 IP Infusion Inc. Proprietary

ACL Configuration 55
Template File for ACL 55
Sample Parameter File for ACL 56
SNMP Configuration 56
Template File for SNMP 56
Parameter File for SNMP 56
IS-IS Configuration 57
Template File for ISIS 57
Parameter File for ISIS 58
Interface Configuration 59
Template File for Interface Configuration 59
Parameter File for Interface configuration 60
BFD Configuration 61
Template File for BFD 61
Parameter File for BFD 61
Hardware Profile Configuration 62
Template File for Hardware Profile 62
Parameter File for Hardware Profile 62
NTP Configuration 63
Template File for NTP 63
Parameter File for NTP 63
VLAN Configuration 63
Template File for VLAN 63
Parameter File for VLAN 63
LLDP Configuration 64
Template File for LLDP 64
Parameter File for LLDP 64
Diff_ignore_lines Configuration 66

© 2026 IP Infusion Inc. Proprietary

PREFACE

About this Guide

This guide describes how to configure Ansible in OcNOS.

Audience

This guide is intended for network administrators and other engineering professionals who configure OcNOS.

Conventions

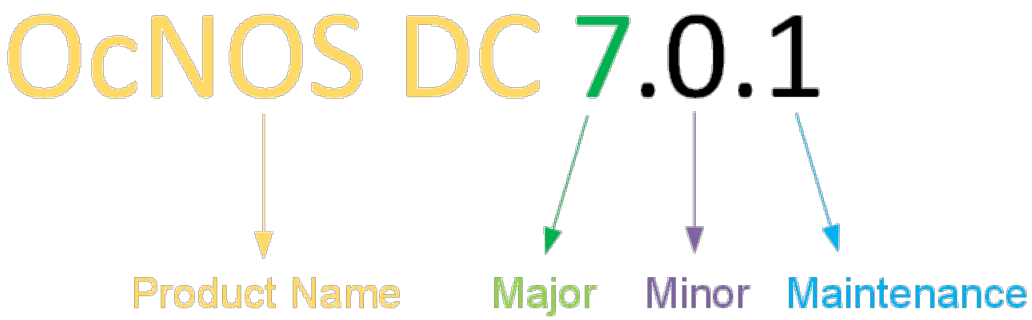
The Table 1 table shows the conventions used in this guide.

Table 1. Conventions

Convention	Description
Italics	Emphasized terms; titles of books
Note:	Special instructions, suggestions, or warnings
`monospaced type`	Code elements such as commands, parameters, files, and directories

IP Infusion Product Release Version

Each integer in release numbers indicates Major, Minor, and Maintenance release versions. Build numbers that follow the release numbers are for internal tracking and verification of the software build process and are visible to customers as part of the software version number.



Product Name: IP Infusion Product Family

Major Version: New customer-facing functionality that represents a significant change to the code base; including a significant marketing change or direction in the product.

Minor Version: Enhancements or extensions to existing features, changes to address external needs, or internal improvements to satisfy new sales regions or marketing initiatives.

Maintenance Version: A collection of product bugs or issues usually scheduled every 30 or 60 days, based on the number of issues.

Related Documentation

For information about installing OcNOS, see the *Installation Guide* for your platform.

Feature Availability

Each OcNOS SKU contains a set of supported features. For a list of available features based on the SKU that you purchased, refer to the *Feature Matrix* .

Migration Guide

Check the *Migration Guide* for necessary configuration changes before migrating from one version of OcNOS to another.

IP Maestro Support

Monitor devices running OcNOS Release 6.3.4-70 and above using IP Maestro software.

Technical Support

IP Infusion maintains an online technical support site that provides a variety of technical support programs for licensed OcNOS customers at the Technical Assistance Center.

Customers and partners enjoy full access to the support website. The site allows customers and partners to open technical support calls, update open calls with new information, and review the status of open or closed calls. The password-protected site includes technical documentation, Release Notes, and descriptions of service offerings.

Technical Sales

Contact the IP Infusion sales representative for more information about the OcNOS solution.

Technical Documentation

For core commands and configuration procedures, visit: Product Documentation.

For training videos, visit: OcNOS Free Training Videos.

For a list of supported platforms and SKUs of OcNOS features, refer to the OcNOS Feature Matrix.

© 2026 IP Infusion Inc. Proprietary

Documentation Disclaimer

The global documentation site is evolving to provide an enhanced website user experience for select topics included in this release. Some guides are now available outside the existing documentation library and can be accessed directly from custom documentation landing pages. These guides offer robust in-built search functionality.

For the latest documentation, visit the product-specific documentation landing page and select the relevant guide.

Comments

If you have comments, or need to report a problem with the content, contact techpubs@ipinfusion.com.

© 2026 IP Infusion Inc. Proprietary

Command Line Interface

This chapter introduces the OcNOS Command Line Interface (CLI) and how to use its features.

Overview

You use the CLI to configure, monitor, and maintain OcNOS devices. The CLI is text-based and each command is usually associated with a specific task.

You can give the commands described in this manual locally from the console of a device running OcNOS or remotely from a terminal emulator such as putty or xterm. You can also use the commands in scripts to automate configuration tasks.

Chapter Organization

The chapters in command references are organized as described in Command Description Format (page 13).

The chapters in configuration guides are organized into these major sections:

- An overview that explains a configuration in words
- Topology with a diagram that shows the devices and connections used in the configuration
- Configuration steps in a table for each device where the left-hand side shows the commands you enter and the right-hand side explains the actions that the commands perform
- Validation which shows commands and their output that verify the configuration

Command Line Interface Help

You access the CLI help by entering a full or partial command string and a question mark “?”. The CLI displays the command keywords or parameters along with a short description. For example, at the CLI command prompt, type:

```
> show ?
```

The CLI displays this keyword list with short descriptions for each keyword:

```
show ?
  application-priority            Application Priority
  arp                             Internet Protocol (IP)
  bfd                             Bidirectional Forwarding Detection (BFD)
  bgp                             Border Gateway Protocol (BGP)
  bi-lsp                          Bi-directional lsp status and configuration
  bridge                          Bridge group commands
  ce-vlan                         COS Preservation for Customer Edge VLAN
  class-map                       Class map entry
  cli                             Show CLI tree of current mode
  clns                            Connectionless-Mode Network Service (CLNS)
  control-adjacency               Control Adjacency status and configuration
  control-channel                 Control Channel status and configuration
  cspf                            CSPF Information
  customer                        Display Customer spanning-tree
  cvlan                           Display CVLAN information
  debugging                       Debugging functions
  etherchannel                    LACP etherchannel
```

```
  ethernet                        Layer-2
...
```

If you type the ? in the middle of a keyword, the CLI displays help for that keyword only.

```
> show de?
debugging  Debugging functions
```

If you type the ? in the middle of a keyword, but the incomplete keyword matches several other keywords, OcNOS displays help for all matching keywords.

```
> show i? (CLI does not display the question mark).
interface  Interface status and configuration
ip         IP information
isis       ISIS information
```

Command Completion

The CLI can complete the spelling of a command or a parameter. Begin typing the command or parameter and then press the tab key. For example, at the CLI command prompt type **sh**:

```
> sh
```

Press the tab key. The CLI displays:

```
> show
```

If the spelling of a command or parameter is ambiguous, the CLI displays the choices that match the abbreviation. Type **show i** and press the tab key. The CLI displays:

```
> show i
  interface  ip         ipv6       isis
> show i
```

The CLI displays the **interface** and **ip** keywords. Type **n** to select **interface** and press the tab key. The CLI displays:

```
> show in
> show interface
```

Type ? and the CLI displays the list of parameters for the **show interface** command.

```
> show interface
  IFNAME  Interface name
  |       Output modifiers
  >       Output redirection
  <cr>
```

The CLI displays the only parameter associated with this command, the **IFNAME** parameter.

Command Abbreviations

The CLI accepts abbreviations that uniquely identify a keyword in commands. For example:

```
> sh int xe0
```

is an abbreviation for:

```
> show interface xe0
```

© 2026 IP Infusion Inc. Proprietary

Command Line Errors

Any unknown spelling causes the CLI to display the error **Unrecognized command** in response to the ?. The CLI displays the command again as last entered.

```
> show dd?
% Unrecognized command
> show dd
```

When you press the Enter key after typing an invalid command, the CLI displays:

```
(config)#router ospf here
                        ^
% Invalid input detected at '^' marker.
```

where the ^ points to the first character in error in the command.

If a command is incomplete, the CLI displays the following message:

```
> show
% Incomplete command.
```

Some commands are too long for the display line and can wrap mid-parameter or mid-keyword, as shown below. This does *not* cause an error and the command performs as expected:

```
area 10.10.0.18 virtual-link 10.10.0.19 authentication-key 57393
```

Command Negation

Many commands have a **no** form that resets a feature to its default value or disables the feature. For example:

- The **ip address** command assigns an IPv4 address to an interface
- The **no ip address** command removes an IPv4 address from an interface

Syntax Conventions

Table 2 describes the conventions used to represent command syntax in this reference.

Table 2. Syntax conventions

Convention	Description	Example
monospaced font	Command strings entered on a command line	**show ip ospf**
lowercase	Keywords that you enter exactly as shown in the command syntax.	show ip ospf
UPPERCASE	See Variable Placeholders (page 12)	IFNAME
()	Optional parameters, from which you must select one. Vertical bars delimit the selections. Do not enter the parentheses or vertical bars as part of the command.	(A.B.C.D|<0-4294967295>)
()	Optional parameters, from which you select one or none. Vertical bars delimit the	(A.B.C.D|<0-4294967295>|)

Table 2. Syntax conventions (continued)

Convention	Description	Example
	selections. Do not enter the parentheses or vertical bars as part of the command.	
`()`	Optional parameter which you can specify or omit. Do not enter the parentheses or vertical bar as part of the command.	`(IFNAME|)`
`{}`	Optional parameters, from which you must select one or more. Vertical bars delimit the selections. Do not enter the braces or vertical bars as part of the command.	`{intra-area <1-255>|inter-area <1-255>|external <1-255>}`
`[]`	Optional parameters, from which you select zero or more. Vertical bars delimit the selections. Do not enter the brackets or vertical bars as part of the command.	`[<1-65535>|AA:NN|internet|local-AS|no-advertise|no-export]`
`?`	Nonrepeatable parameter. The parameter that follows a question mark can only appear once in a command string. Do not enter the question mark as part of the command.	`?route-map WORD`
`.`	Repeatable parameter. The parameter that follows a period can be repeated more than once. Do not enter the period as part of the command.	`set as-path prepend .<1-65535>`

Variable Placeholders

Table 3 shows the tokens used in command syntax use to represent variables for which you supply a value.

Table 3. Variable placeholders

Token	Description
`WORD`	A contiguous text string (excluding spaces)
`LINE`	A text string, including spaces; no other parameters can follow this parameter
`IFNAME`	Interface name whose format varies depending on the platform; examples are: **`eth0`**, **`Ethernet0`**, **`ethernet0`**, **`xe0`**
`A.B.C.D`	IPv4 address
`A.B.C.D/M`	IPv4 address and mask/prefix
`X:X::X:X`	IPv6 address
`X:X::X:X/M`	IPv6 address and mask/prefix
`HH:MM:SS`	Time format
`AA:NN`	BGP community value

© 2026 IP Infusion Inc. Proprietary

Table 3. Variable placeholders (continued)

Token	Description
`XX:XX:XX:XX:XX:XX`	MAC address

`<1-5>`
`<1-65535>`
`<0-2147483647>`
`<0-4294967295>`
	Numeric range

Command Description Format

The Table 4 table explains the sections used to describe each command in this reference.

Table 4. Command descriptions

Section	Description
Command Name	The name of the command, followed by what the command does and when should it be used
Command Syntax	The syntax of the command
Parameters	Parameters and options for the command
Default	The state before the command is executed
Command Mode	The mode in which the command runs; see Command Modes (page 17)
Applicability	The command introduced in a specific release version and modified or updated in subsequent versions.
Example	An example of the command being executed

Keyboard Operations

The Table 5 table lists the operations you can perform from the keyboard.

Table 5. Keyboard operations

Key combination	Operation
Left arrow or Ctrl+b	Moves one character to the left. When a command extends beyond a single line, you can press left arrow or Ctrl+b repeatedly to scroll toward the beginning of the line, or you can press Ctrl+a to go directly to the beginning of the line.
Right arrow or Ctrl-f	Moves one character to the right. When a command extends beyond a single line, you can press right arrow or Ctrl+f repeatedly to scroll toward the end of the line, or you can press Ctrl+e to go directly to the end of the line.
Esc, b	Moves back one word
Esc, f	Moves forward one word
Ctrl+e	Moves to end of the line

© 2026 IP Infusion Inc. Proprietary

Table 5. Keyboard operations (continued)

Key combination	Operation
Ctrl+a	Moves to the beginning of the line
Ctrl+u	Deletes the line
Ctrl+w	Deletes from the cursor to the previous whitespace
Alt+d	Deletes the current word
Ctrl+k	Deletes from the cursor to the end of line
Ctrl+y	Pastes text previously deleted with Ctrl+k, Alt+d, Ctrl+w, or Ctrl+u at the cursor
Ctrl+t	Transposes the current character with the previous character
Ctrl+c	Ignores the current line and redisplays the command prompt
Ctrl+z	Ends configuration mode and returns to exec mode
Ctrl+l	Clears the screen
Up Arrow or Ctrl+p	Scroll backward through command history
Down Arrow or Ctrl+n	Scroll forward through command history

Show Command Modifiers

Note: The show command output included in the guides is for illustration purposes only. Based on the combination of features enabled and ongoing enhancements made to the commands, the output for these commands may vary. For instance, the actual command output may differ depending on the software version, configuration, and platform. Field names, values, and formats are subject to change.

You can use two tokens to modify the output of a **`show`** command. Enter a question mark to display these tokens:

```
# show users ?
  | Output modifiers
  > Output redirection
```

You can type the | (vertical bar character) to use output modifiers. For example:

```
> show rsvp | ?
  begin     Begin with the line that matches
  exclude   Exclude lines that match
  include   Include lines that match
  last      Last few lines
  redirect  Redirect output
```

Begin Modifier

The **`begin`** modifier displays the output beginning with the first line that contains the input string (everything typed after the **`begin`** keyword). For example:

```
# show running-config | begin xe1
...skipping
interface xe1
ipv6 address fe80::204:75ff:fee6:5393/64
!
interface xe2
```

© 2026 IP Infusion Inc. Proprietary

```
ipv6 address fe80::20d:56ff:fe96:725a/64
!
line con 0
login
!
end
```

You can specify a regular expression after the **begin** keyword, This example begins the output at a line with either “xe2” or “xe4”:

```
# show running-config | begin xe[2-4]

...skipping
interface xe2
 shutdown
!
interface xe4
 shutdown
!
interface svlan0.1
 no shutdown
!
route-map myroute permit 2
!
route-map mymap1 permit 10
!
route-map rmap1 permit 2
!
line con 0
 login
line vty 0 4
 login
!
end
```

Include Modifier

The **include** modifier includes only those lines of output that contain the input string. In the output below, all lines containing the word “input” are included:

```
# show interface xe1 | include input
    input packets 80434552, bytes 2147483647, dropped 0, multicast packets 0
    input errors 0, length 0, overrun 0, CRC 0, frame 0, fifo 1, missed 0
```

You can specify a regular expression after the **include** keyword. This examples includes all lines with “input” or “output”:

```
#show interface xe0 | include (in|out)put
    input packets 597058, bytes 338081476, dropped 0, multicast packets 0
    input errors 0, length 0, overrun 0, CRC 0, frame 0, fifo 0, missed 0
    output packets 613147, bytes 126055987, dropped 0
    output errors 0, aborted 0, carrier 0, fifo 0, heartbeat 0, window 0
```

Exclude Modifier

The **exclude** modifier excludes all lines of output that contain the input string. In the following output example, all lines containing the word “input” are excluded:

```
# show interface xe1 | exclude input
Interface xe1
  Scope: both
  Hardware is Ethernet, address is 0004.75e6.5393
  index 3 metric 1 mtu 1500 <UP,BROADCAST,RUNNING,MULTICAST>
```

© 2026 IP Infusion Inc. Proprietary

```
  VRF Binding: Not bound
  Administrative Group(s): None
  DSTE Bandwidth Constraint Mode is MAM
  inet6 fe80::204:75ff:fee6:5393/64
    output packets 4438, bytes 394940, dropped 0
    output errors 0, aborted 0, carrier 0, fifo 0, heartbeat 0, window 0
    collisions 0
```

You can specify a regular expression after the **exclude** keyword. This example excludes lines with “output” or “input”:

```
show interface xe0 | exclude (in|out)put
Interface xe0
  Scope: both
  Hardware is Ethernet  Current HW addr: 001b.2139.6c4a
  Physical:001b.2139.6c4a  Logical:(not set)
  index 2 metric 1 mtu 1500 duplex-full arp ageing timeout 3000
  <UP,BROADCAST,RUNNING,MULTICAST>
  VRF Binding: Not bound
  Bandwidth 100m
  DHCP client is disabled.
  inet 10.1.2.173/24 broadcast 10.1.2.255
  VRRP Master of :  VRRP is not configured on this interface.
  inet6 fe80::21b:21ff:fe39:6c4a/64
    collisions 0
```

Redirect Modifier

The **redirect** modifier writes the output into a file. The output is not displayed.

```
# show cli history | redirect /var/frame.txt
```

The output redirection token (>) does the same thing:

```
# show cli history >/var/frame.txt
```

Last Modifier

The **last** modifier displays the output of last few number of lines (As per the user input). The last number ranges from 1 to 9999.

For example:

```
#show running-config | last 10
```

String Parameters

The restrictions in Table 6 apply for all string parameters used in OcNOS commands, unless some other restrictions are noted for a particular command.

Table 6. String parameter restrictions

Restriction	Description
Input length	1965 characters or less
Restricted special characters	“?”, “,”, “>”, “|”, and “=” The “|” character is allowed only for the **description** command in interface mode.

Command Modes

Commands are grouped into modes arranged in a hierarchy. Each mode has its own set of commands. The table below lists the command modes common to all protocols.

Table 7. Common Command Modes

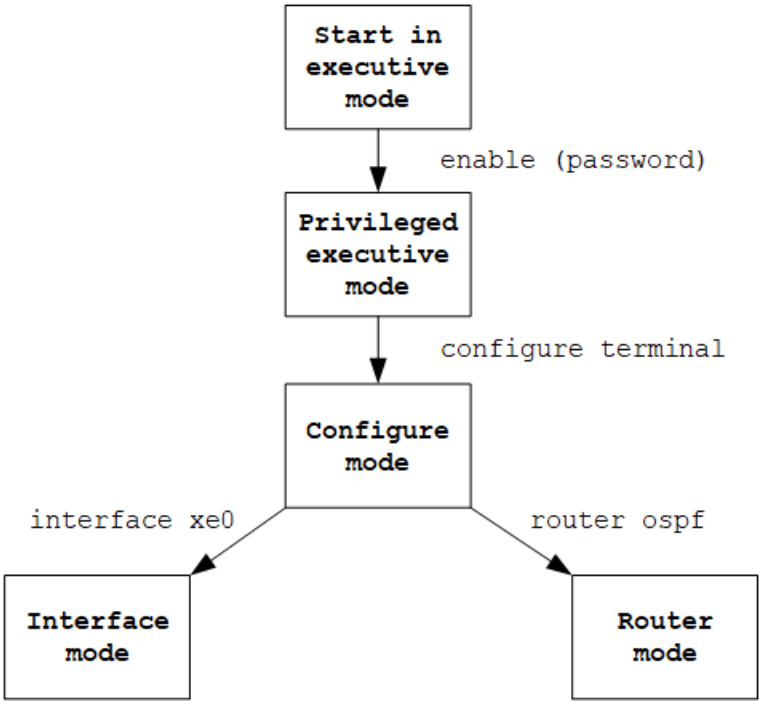
Name	Description
`Execution mode`	Also called *view* mode, this is the first mode to appear after you start the CLI. It is a base mode from where you can perform basic commands such as show, exit, quit, help, and enable.
Privileged execution mode	Also called *enable* mode, in this mode you can run additional basic commands such as debug, write, and show.
Configure mode	Also called *configure terminal* mode, in this mode you can run configuration commands and go into other modes such as interface, router, route map, key chain, and address family. Configure mode is single user. Only one user at a time can be in configure mode.
Interface mode	In this mode you can configure protocol-specific settings for a particular interface. Any setting you configure in this mode overrides a setting configured in router mode.
Router mode	This mode is used to configure router-specific settings for a protocol such as BGP or OSPF.

© 2026 IP Infusion Inc. Proprietary

Command Mode Tree

The diagram below shows the common command mode hierarchy.

Figure 1. Common command modes



To change modes:

1. Enter privileged executive mode by entering **`enable`** in Executive mode.
2. Enter configure mode by entering **`configure terminal`** in Privileged Executive mode.

The example below shows moving from executive mode to privileged executive mode to configure mode and finally to router mode:

```
> enable mypassword
# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
(config)# router ospf
(config-router)#
```

Note: Each protocol can have modes in addition to the common command modes. See the command reference for the respective protocol for details.

Transaction-based Command-line Interface

The OcNOS command line interface is transaction based:

- Any changes done in configure mode are stored in a separate *candidate* configuration that you can view with the `show transaction current` command.
- When a configuration is complete, apply the candidate configuration to the running configuration with the command.

© 2026 IP Infusion Inc. Proprietary

- If a `commit` fails, no configuration is applied as the entire transaction is considered failed. You can continue to change the candidate configuration and then retry the `commit`.
- Discard the candidate configuration with the `abort transaction` command.
- Check the last aborted transaction with the `show transaction last-aborted` command.
- Multiple configurations cannot be removed with a single . You must remove each configuration followed by a `commit`.

Note: All commands MUST be executed only in the default CML shell (**`cmlsh`**). If you log in as root and start **`imish`**, then the system configurations will go out of sync. The **`imish`** shell is not supported and should not be started manually.

OVERVIEW

This section demonstrates how to use Ansible to manage OcNOS devices with a platform-agnostic framework based on ansible networking modules. It covers:

- Managing OcNOS devices using Ansible playbooks.
- Using Jinja2 templates for protocol configurations.
- Highlighting current limitations.

Installing the OcNOS Collection from Ansible Galaxy

Follow these steps to install the OcNOS Ansible module:

1. Verify Ansible Version: Ensure that the 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 [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/sitepackages/
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. Install SSH plugin dependencies 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 the OcNOS collection from Galaxy

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

4. Install from local package: 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
```

© 2026 IP Infusion Inc. Proprietary

Set-up Ansible Files

Group Variables File

```
$ 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
```

Inventory File

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
```

© 2026 IP Infusion Inc. Proprietary

Supported Modules

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

- ocnos_facts (`ipinfusion.ocnos.ocnos_facts`)
- ocnos_command (ipinfusion.ocnos.ocnos_command)
- ocnos_config (ipinfusion.ocnos.ocnos_config)
- ocnos_ping (ipinfusion.ocnos.ocnos_ping)
- ocnos_bgp_facts (ipinfusion.ocnos.ocnos_bgp_facts)
- ocnos_isis_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

The ocnos_facts module collects operational data from OcNOS devices. The data is stored in variables prefixed with `ansible_net_xxx`. These facts can be used to validate configurations or conditionally perform tasks.

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 8. Return Value

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`	All IPv6 addresses configured on the device	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_hostnam string`	always	The configured hostname of the device
`ansible_net_imag 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
```

© 2026 IP Infusion Inc. Proprietary

```
- 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"
Getting Started
© 2025 IP Infusion Inc. Proprietary 25
]
}
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"
Getting Started
© 2025 IP Infusion Inc. Proprietary 26
},
"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": {
Getting Started
© 2025 IP Infusion Inc. Proprietary 27
```

```
"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"
Getting Started
© 2025 IP Infusion Inc. Proprietary 28
}
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 9. 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.
`newlin 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.
`sendonl 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 10. 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 11. 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

© 2026 IP Infusion Inc. Proprietary

Table 11. Parameters (continued)

Parameter	Choices/Defaults	Comments
		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

© 2026 IP Infusion Inc. Proprietary

Table 11. Parameters (continued)

Parameter	Choices/Defaults	Comments
		session. If the value is not specified in the task, the value of environment variable ANSIBLE_NET_ SSH_KEYFILE will be used instead
`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.

© 2026 IP Infusion Inc. Proprietary

Return Values

Table 12. Return values

Key	Returned	Description
stdout list	always	the set of responses from the commands.
stdout_line s 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 13. 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

Table 13. Parameters (continued)

Parameter	Choices/Defaults	Comments
		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>@<currenttime>**
`config string`		The config to be pushed to the network device. This argument is mutually

Table 13. Parameters (continued)

Parameter	Choices/Defaults	Comments
		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 14. 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 inshow 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 15. 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.
diff_ignore_lin es -		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 (plain strings).

© 2026 IP Infusion Inc. Proprietary

Table 15. Supported parameters (continued)

Parameter	Choices/Defaults	Comments
		Example For regular expressions: `diff_ignore_lines: "^tacacsserver\\s+login\\s+host\\s+\\S+\\s+vrf\\s+\\S+\\s+seq-num\\s+\\d+\\s+key\\s+\\d+\\s+\\S+"` For plain strings: `diff_ignore_lines: "tacacs-server"` For a sample configuration, refer to the `Example: diff_ignore_lines section.`
`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

Table 15. Supported parameters (continued)

Parameter	Choices/Defaults	Comments
		use as the base config for this module. aliases: config
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 startupconfig 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 runningconfig 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 16. 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
shortname string	When backup is yes and filename is not specified in	The full path to the backup file excluding the timestamp Sample:

© 2026 IP Infusion Inc. Proprietary

Table 16. Return values (continued)

Key	Returned	Description
	backup options	/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
```

© 2026 IP Infusion Inc. Proprietary

```
!
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
```

Example: diff_ignore_lines

To avoid false scenarios, use the `diff_ignore_lines` tag in playbooks. In the below example, the `diff_ignore_lines: '^tacacs-server\s+login\s+host.*'` tag in the playbook ignores lines starting with the specified string or those matching the pattern `under diff_ignore_lines`. This prevents unnecessary alerts when no changes have occurred.

Playbook

```
---
- hosts: R1
gather_facts: no
become: yes # Enable privilege escalation
become_method: sudo # Use sudo for privilege escalation (default)
tasks:
- name: Configure tacacs+
cli_config:
#lines:
config: |
feature tacacs+ vrf management
tacacs-server login host 10.2.1.7 vrf management seq-num 1 key 0 blabla
tacacs-server login host 10.2.1.8 vrf management seq-num 2 key 0 blabla
aaa group server tacacs+ TACACS vrf management
server 10.2.1.7
server 10.2.1.8
aaa authentication login default vrf management group tacacs+ local
commit
diff_ignore_lines: "^tacacs-server\s+login\s+host.*"
- name: show running config tacacs+ and aaa
```

```
cli_command:
command: show running-config tacacs+
command: show run aaa
register: result
- name: debug
debug:
msg: "{{ result.stdout_lines }}"
```

Output

The following output shows the results of running the config.yml playbook, detailing the configuration changes applied to the device and the verification commands executed.

```
#ansible-playbook config.yml -i inventory.ini -vvvv
...
PLAYBOOK: config.yml
**********************************************************************
Positional arguments: config.yml
verbosity: 4
connection: smart
timeout: 10
become_method: sudo
tags: ('all',)
inventory: ('/home/ocnos/inventory.ini',)
forks: 5
1 plays in config.yml
PLAY [R1]
*********************************************************************************
.
.
.
TASK [Configure tacacs+]
*****************************************************************
task path: /home/ocnos/config.yml:9
.
.
.
changed: [R1-VM] => {
"changed": true,
"commands": [
"feature tacacs+ vrf management",
"tacacs-server login host 10.2.1.7 vrf management seq-num 1 key 0 blabla",
"tacacs-server login host 10.2.1.8 vrf management seq-num 2 key 0 blabla",
"aaa group server tacacs+ TACACS vrf management",
"server 10.2.1.7",
"server 10.2.1.8",
"aaa authentication login default vrf management group tacacs+ local",
"commit"
],
"invocation": {
"module_args": {
"backup": false,
"backup_options": null,
"commit": null,
"commit_comment": null,
"config": "feature tacacs+ vrf management\ntacacs-server login host 10.2.1.7
vrf management seq-num 1 key 0 blabla\ntacacs-server login host 10.2.1.8 vrf management
seq-num 2 key 0 blabla\naaa group server tacacs+ TACACS vrf management\nserver
10.2.1.7\nserver 10.2.1.8\naaa authentication login default vrf management group
tacacs+ local\ncommit\n",
"defaults": false,
"diff_ignore_lines": [
"^tacacs-server\\s+login\\s+host.*"
],
"diff_match": null,
"diff_replace": null,
"multiline_delimiter": null,
```

```
"replace": null,
"rollback": null
}
}
}
redirecting (type: action) ansible.builtin.cli_command to ansible.netcommon.cli_command
TASK [show running config tacacs+ and aaa]
*******************************************
task path: /home/ocnos/config.yml:23
.
.
.
ok: [R1-VM] => {
"changed": false,
"invocation": {
"module_args": {
"answer": null,
"check_all": false,
"command": "show run aaa",
"newline": true,
"prompt": null,
"sendonly": false
}
},
"stdout": "aaa group server tacacs+ TACACS vrf management\n server 10.2.1.7\n
server 10.2.1.8\n\naaa authentication login default vrf management group tacacs+
local",
"stdout_lines": [
"aaa group server tacacs+ TACACS vrf management",
" server 10.2.1.7",
" server 10.2.1.8",
"",
"aaa authentication login default vrf management group tacacs+ local"
]
}
TASK [debug]
*********************************************************************
task path: /home/ocnos/config.yml:29
.
.
.
ok: [R1-VM] => {
"msg": [
"aaa group server tacacs+ TACACS vrf management",
" server 10.2.1.7",
" server 10.2.1.8",
"",
"aaa authentication login default vrf management group tacacs+ local"
]
}
PLAY RECAP
*********************************************************************************
R1-VM : ok=3 changed=1 unreachable=0 failed=0 skipped=0
rescued=0 ignored=0
```

Validation

The validation section confirms that the system has applied the expected configurations correctly by displaying the current running configuration for AAA and TACACS+.

```
OcNOS#show running-config aaa
aaa group server tacacs+ TACACS vrf management
server 10.2.1.7
server 10.2.1.8
aaa authentication login default vrf management group tacacs+ local
OcNOS#show running-config tacacs+
feature tacacs+ vrf management
```

```
tacacs-server login host 10.2.1.7 vrf management seq-num 1 key 7 0xb7ff9eebbcc87
576
tacacs-server login host 10.2.1.8 vrf management seq-num 2 key 7 0xb7ff9eebbcc87
576
```

ipinfusion.ocnos.ocnos_ping or net_ping

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

Parameters

Table 17. 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.
`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 18. 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}

© 2026 IP Infusion Inc. Proprietary

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 }}"
```

© 2026 IP Infusion Inc. Proprietary

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
```

© 2026 IP Infusion Inc. Proprietary

ANSIBLE CONFIGURATION

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.

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 %}
```

© 2026 IP Infusion Inc. Proprietary

```
{% 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: routereflector- client`	If route_reflector_type is defined then neighbor 10.0.1.14 will be configured as route reflector client for address-family labelunicast
`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

© 2026 IP Infusion Inc. Proprietary

`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`	Address 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

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 DSCPAF11
`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

© 2026 IP Infusion Inc. Proprietary

`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-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

© 2026 IP Infusion Inc. Proprietary

`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 policytype qos
`type: queuing`	Configures the service policy type as queuing
`policy_name: shaper`	Configures the output policy name as shaper for policy-type queuing

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

© 2026 IP Infusion Inc. Proprietary

`seq_no: 10`	Configure the sequence no. as 10
`operation: match`	Configures 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: bgp`	Enable bgp 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

IS-IS 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

© 2026 IP Infusion Inc. Proprietary

`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

© 2026 IP Infusion Inc. Proprietary

	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`	Configure 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 command`	Configure interface ce0
`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.yml

`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

© 2026 IP Infusion Inc. Proprietary

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

© 2026 IP Infusion Inc. Proprietary

`Vlan`	Vlan configuration
`level: database`	Configure VLAN database
`range :`	Vlan range configuration
`- value : 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

© 2026 IP Infusion Inc. Proprietary

	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 xe11
`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

© 2026 IP Infusion Inc. Proprietary

`- name: xe11`	Configure interface xe11 command
`lagent: lldp-agent`	Enable lldp agent under xe11 interface
`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

Diff_ignore_lines Configuration

Template File for diff_ignore_lines

```
Ansible.yml
---
- hosts: ocnos
gather_facts: no
tasks:
- name: Configure tacacs+
ipinfusion.ocnos.ocnos_config:
lines:
- feature tacacs+ vrf management
- tacacs-server login host 10.2.1.7 vrf management seq-num 3 key 0 aba
- tacacs-server login host 10.2.1.8 vrf management seq-num 4 key 0 aba
- aaa group server tacacs+ TACACS vrf management
- server 10.2.1.7
- server 10.2.1.8
- aaa authentication login default vrf management group tacacs+ local
- commit
diff_ignore_lines: "aaa"
Output after Executing scripts
changed: [sw1] => {
"changed": true,
"commands": [
"feature tacacs+ vrf management",
"tacacs-server login host 10.2.1.7 vrf management seq-num 3 key 0 aba",
"tacacs-server login host 10.2.1.8 vrf management seq-num 4 key 0 aba",
"aaa group server tacacs+ TACACS vrf management",
```

```
"server 10.2.1.7",
"server 10.2.1.8",
"aaa authentication login default vrf management group tacacs+ local",
"commit"
],
"diff": {
"prepared": [
"Enter configuration commands, one per line. End with CNTL/Z.",
"",
"tacacs-server login host 10.2.1.7 vrf management seq-num 3 key 0 a a",
"tacacs-server login host 10.2.1.8 vrf management seq-num 4 key 0 a a",
"",
"",
"",
""
]
},
"invocation": {
"module_args": {
"after": null,
"backup": false,
"backup_options": null,
"before": null,
"commit": true,
"diff_ignore_lines": "aaa",
"lines": [
"feature tacacs+ vrf management",
"tacacs-server login host 10.2.1.7 vrf management seq-num 3 key 0 aba",
"tacacs-server login host 10.2.1.8 vrf management seq-num 4 key 0 aba",
"aaa group server tacacs+ TACACS vrf management",
"server 10.2.1.7",
"server 10.2.1.8",
"aaa authentication login default vrf management group tacacs+ local",
"commit"
],
"match": "line",
"parents": null,
"provider": null,
"replace": "line",
"running_config": null,
"save_when": "never",
"src": null
}
}
}
PLAY RECAP
*********************************************************************************
**************************
sw1 : ok=1 changed=1 unreachable=0 failed=0 skipped=0
rescued=0 ignored=0
```

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_timeou`**t 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
```

© 2026 IP Infusion Inc. Proprietary