Ansible Module aviznetworks.sonic

# Day-2...day-n configuration difference

In the Ansible script tasks, the **aviznetworks.sonic** module will check and compare the configuration of each task with the existing configuration on the devices. The module will merge the configurations if a similar configuration supports the protocol rules, or the new configurations will override it.

Example:

**Day-1 Configuration**

- name: SONiC port configuration

sonic\_interfaces:

config:

- interfaces: ["Ethernet32", "Ethernet4"]

mtu: 9000

enable: True

**Day-2…Day-n Configuration**

- name: SONiC port configuration

sonic\_interfaces:

config:

- interfaces: ["Ethernet32"]

mtu: 9100

enable: True

description: "fmcli description\_eth32"

In the example above, on day-2...days-n configuration, updating the MTU and configure the port description. In the log, **Diff** shows, removing the old mtu value and adding the new value.

Only “description” getting added, because in the old configuration description was not provided.

log:

| "diff": {  "interfaces": {  "interface ethernet Ethernet32": [  "- mtu 9000",  "+ mtu 9100",  "+ description fmcli description\_eth32",  ]  }  } |
| --- |

# State: Merge / Delete

Two State supports will merge/delete the configuration on day-0…day-n task operation.

## Merge

A default operation, will be used in case of merge/add/update/override configurations.

# day-0

- name: SONiC port configuration

sonic\_interfaces:

config:

- interfaces: ["Ethernet32"]

mtu: 9000

enable: True

description: "fmcli description\_eth32"

- interfaces: ["Ethernet34"]

mtu: 9000

description: "fmcli description\_eth34"

state: merge # default

# day-n

- name: SONiC port configuration

sonic\_interfaces:

config:

- interfaces: ["Ethernet32"]

mtu: 9000

ip\_address: 40.0.0.10/31 # new configuration

description: "fmcli description\_eth32"

state: merge # default

In this configuration, it will check if any configuration exists it will get overridden and if it does not exist it will get added as a new configuration.

log:

| **Day-0**  "commands": [  "config terminal",  "interface ethernet Ethernet32",  "mtu 9000",  "no shutdown",  "description fmcli description\_eth32",  "end",  "save",  "config terminal",  "interface ethernet Ethernet34",  "mtu 9000",  "description fmcli description\_eth32",  "end",  "save"  ] | **Day-n**  "commands": [  "config terminal",  "interface ethernet Ethernet32",  "ip address 40.0.0.2/31",  "end",  "save"  ] |
| --- | --- |

## Delete

It will delete the configuration if it already exists.

### Delete inner configs

# day-0

- name: SONiC port configuration

sonic\_interfaces:

config:

- interfaces: ["Ethernet32", "Ethernet36"]

mtu: 9000

enable: True

description: "fmcli description"

state: merge # default

# day-n delete some config. Ex: delete the description for the interfaces

- name: SONiC port configuration

sonic\_interfaces:

config:

- interfaces: ["Ethernet32", "Ethernet36"]

description: "fmcli description"

state: delete

Here, day-n task will delete the inner configuration only mentioned under the interface.

log:

| **Day-0**  "commands": [  "config terminal",  "interface ethernet Ethernet32",  "mtu 9000",  "no shutdown",  "description fmcli description",  "end",  "save",  "config terminal",  "interface ethernet Ethernet36",  "mtu 9000",  "no shutdown",  "description fmcli description",  "end",  "save"  ] | **Day-n delete config**  "commands": [  "config terminal",  "interface ethernet Ethernet32",  "no description fmcli description",  "end"  "save",  "config terminal",  "interface ethernet Ethernet36",  "no description fmcli description",  "end"  "save"  ] |
| --- | --- |

### Delete whole configs

# day-0, day-1, ...

- name: SONiC port configuration

sonic\_interfaces:

config:

- interfaces: ["Ethernet32", "Ethernet36"]

mtu: 9000

enable: True

description: "fmcli description"

# day-n delete whole config

- name: SONiC port configuration

sonic\_interfaces:

config:

- interfaces: ["Ethernet32", "Ethernet36"]

state: delete

Here, the day-n task will delete the complete configuration for the listed interfaces

log:

| **Day-0**  "commands": [  "config terminal",  "interface ethernet Ethernet32",  "mtu 9000",  "no shutdown",  "description fmcli description",  "end",  "save",  "config terminal",  "interface ethernet Ethernet36",  "mtu 9000",  "no shutdown",  "description fmcli description",  "end",  "save"  ] | **Day-n delete config**  "commands": [  "config terminal",  "interface ethernet Ethernet32",  "no mtu 9000",  "shutdown",  "no description fmcli description",  "end"  "save",  "config terminal",  "no interface ethernet Ethernet32",  "end"  "save",  "config terminal",  "interface ethernet Ethernet36",  "no mtu 9000",  "shutdown",  "no description fmcli description",  "end"  "save",  "config terminal",  "no interface ethernet Ethernet36",  "end"  "save"  ] |
| --- | --- |

# 

# Sonic\_commands

This module supports the execution of list of commands inside fmcli prompt.

Executes the task with the module **sonic\_commands**. Provide the configuration data as a list under the “**config**”key. Here **“commands”** is the primary key.

Supported keys/arguments - value:

| **commands** | - a primary key argument  - list of commands Ex: **["config", "no mlag domain-id 1", "end", "save"]** |
| --- | --- |

Sample playbook task

- name: SONiC port configuration with wait\_for and match=any

sonic\_commands:

config:

- commands: ["config", "no mlag domain-id 1", "end", "save"]

- commands: ["config", "no interface vlan 10", "end", "save", "show run"]

# 

# 

# Ports/Interface

Configuring and deleting multiple interfaces.

Executes the task with the module **sonic\_interfaces**. Provide the configuration data as a list under the “**config**”key. Here **“interfaces”** is the primary key.

Supported keys/arguments - value:

| **interfaces** | - a primary key argument  - list of interface names. Ex: **["Ethernet32", "Ethernet4"]** |
| --- | --- |
| **mtu** | mtu value, **int** type. Ex: 9100 |
| **enable** | To enable or disable the ports, boolean type. Ex: true/false  **true**: enable/bring-up the ports  **false**: disable/bring-down the ports |
| **ip\_address** | - Configure the IP(v4) address to a port.  - provide the IP address with a mask. Ex: **10.4.4.4/24**  - make sure, use a single interface name for the argument **interface** |
| **description** | Description of interfaces |
| **fec** | Choose a value from the list: **["rs", "fc", "none"]** |
| **speed** | Choose a value from the list:  ["1G", "10G", "25G", "40G", "50G", "100G", "400G"] |

Sample playbook task

- name: SONiC port configuration

sonic\_interfaces:

config:

- interfaces: ["Ethernet32", "Ethernet4"]

mtu: 9000

enable: true

description: "fmcli description\_port"

- interfaces: 'Ethernet36'

mtu: 9000

fec: rs

enable: false

ip\_address: 10.4.4.4/23

description: "fmcli description"

# 

# Loopback

A loopback interface is a virtual Layer 3 interface. A loopback interface emulates a physical interface and is always in the UP state.

To configure and delete multiple loopbacks, execute the task with the module **sonic\_loopback**. Provide the configuration data as a list under the “**config**”key. Here **“loopback\_id”** is the primary key.

Supported keys/arguments - value:

| **loopback\_id** | - a primary key argument  - int type. Ex: 1 |
| --- | --- |
| **ip\_address** | - Configure the IP(v4) address to an Interface.  - provide the IP address with a mask. Ex: **10.4.4.4/24** |

Sample playbook task

# task 1: add or merge the loopback configuration

- name: SONiC loopback configuration merge

sonic\_loopback:

config:

- loopback: 1

ip\_address: 10.4.5.8/23

- loopback: 2

ip\_address: 10.4.5.7/23

# task 2: delete the loopback 1 configuration

- name: SONiC loopback configuration delete

sonic\_loopback:

config:

- loopback: 1

state: delete # delete loopback 1

# task 3: delete the ip\_address from loopback 2 configuration

- name: SONiC loopback configuration delete

sonic\_loopback:

config:

- loopback: 2

ip\_address: 10.4.5.7/23

state: delete # delete ip\_address from loopback 2

# task 4: delete the loopback 1 and loopback 2 configuration

- name: SONiC loopback configuration delete

sonic\_loopback:

config:

- loopback: 1

- loopback: 2

ip\_address: 10.4.5.7/23

state: delete # delete loopback 1 and ip\_address from loopback 2

# Vlan

**Configure and delete multiple vlans:**

The switch supports up to 4094 VLANs. Each can be identified with a number between 2 and 4094.

To configure Vlans execute the task with the module **sonic\_vlan**. Provide the configuration data as a list under the “**config**”key. Here **“vlan\_ids”** is the primary key.

Supported keys/arguments - value:

| **vlan\_ids** | - a primary key argument  - list of Vlan id. Ex: **["100", "200", "221-223"]**  - Vlan id can be in range. Ex: **["221-223"]**  **#** It will create the VLAN ids 221, 222, 223 |
| --- | --- |
| **vlan\_id** | - a primary key argument  - a single Vlan id. Ex: **"100"**  **Note:** use **vlan\_ids** or **vlan\_id** |
| **interfaces** | - list of interface names and or list of portchannel id. Ex: **["Ethernet32", "Ethernet4", "portchannel10", "pch20"]** |
| **vlan\_mode** | - Vlan mode could be **trunk** or **access**  - Default vlan\_mode is access mode  - When using access mode, the task will be executed only with the first **interface** from the list |
| **anycast\_gateway** | - The command sequence configures an anycast gateway IP address for a specific VLAN interface  - provide the IP address with a mask. Ex: **10.4.4.4/24**  - Task will configure the IP address only on the first **vlan\_ids** from the list or can be used **vlan\_id** |
| **vrf\_name** | - name of vrf member. Ex: testVRF |
| **ip\_address** | - SVI(Switch Virtual Interface)  - Configure the IP address to the Vlan.  - provide the IP address with a mask. Ex: **10.4.4.4/24**  - Task will configure the IP address only on the first **vlan\_ids** from the list or can be used **vlan\_id** |
| **name** |  |

Sample playbook

- name: "vlan trunk mode on interfaces"

sonic\_vlan:

config:

- vlan\_ids: [20, 30, 35-37] # [20, 30, 35, 36, 37]

vlan\_mode: "trunk"

interfaces: ["Ethernet200", "Ethernet208"]

- name: "vlan access mode on interfaces"

sonic\_vlan:

config:

- vlan\_id: 50

vlan\_mode: "access"

interfaces: ["Ethernet204"]

- name: "vlan trunk mode on portchannel"

sonic\_vlan:

config:

- vlan\_ids: [20, 30, 40] # [20, 30, 35-37]

vlan\_mode: "trunk"

interfaces: ["portchannel300", "pch200"]

- name: "vlan ip\_address configuration"

sonic\_vlan:

config:

- vlan\_ids: [20] #make sure provide one vlan id for SVI

ip\_address: "100.10.0.5/25"

# 

# Portchannel

**Configure and delete multiple port-channel.**

To configure port-channel, execute the task with the module **sonic\_port\_channel**. Provide the configuration data as a list under the “**config**”key. Here **“pch\_id”** is the primary key.

Supported keys/arguments - value:

| **pch\_id** | - a primary key argument  - portchannel id. Ex: **"100"** |
| --- | --- |
| **interfaces** | - list of member interface names. Ex: **["Ethernet32", "Ethernet4"]** |
| **description** | - portchannel description |
| **mode** | - only supported value for the mode is **"active"**  - with active mode, it will configure the channel group or add members to the portchannel  - if mode argument not passed, it will not configure the members to the portchannel or remove the members from the portchannel |

- name: "create a port-channel"

sonic\_port\_channel:

config:

- pch\_id: 100

- name: "Add member port to port-channel"

sonic\_port\_channel:

config:

- pch\_id: 200

interfaces: ["Ethernet216"]

description: "pch description 100"

mode: "active"

# BGP

## BGP router

To configure the BGP router, execute the task with the module **sonic\_bgp**. Provide the configuration data as a list under the “**config**”key. Here **“bgp\_asn”** is the primary key.

A BGP configuration data list must be set under the key **“bgp”**.

Supported keys/arguments - value:

| **bgp\_asn** | - a primary key argument  - Configures BGP for an autonomous system (AS) number  - bgp as\_number. Ex: **"1001"** |
| --- | --- |
| **bgp:**  **..router\_id** | - 32-bit IPv4 address to establish the peering session with bgp peers  - router id. Ex: 10.10.10.2 |
| **bgp:**  ..**bestpath** | - bestpath boolean type, **true** or **false**  - handle the paths received from different autonomous systems for multipath if their AS-path lengths are the same and all other multipath conditions are met  - allows load sharing across providers with different (but equal length) AS paths  - command: **bgp bestpath as-path multipath-relax** |
| **bgp:** ..**ebgp\_requires\_policy** | - epbg\_requires\_policy boolean type, **true** or **false**  - default value is **false**  - restricts in and out policy requirements for BGP peers  - command: **bgp ebgp-requires-policy  : no bgp ebgp-requires-policy** |
| **bgp:** ..**restart\_time** | - configure bgp graceful restart with restart-time in second  - Ex: 10  - command: **bgp graceful-restart restart-time 10** |
| **bgp: ..stalepath\_time** | - configure bgp graceful restart with stalepath-time in second  - Ex: 10  - command: **bgp graceful-restart stalepath-time 10** |

Sample playbook task

- name: Sonic BGP router config

sonic\_bgp:

config:

bgp\_asn: 1001

bgp:

- router\_id: 10.10.10.2

ebgp\_requires\_policy: false

bestpath: true

- router\_id: 10.10.10.4

ebgp\_requires\_policy: false

bestpath: true

retsrat\_time: 10

## bgp\_neighbors(IPv4)

To configure the bgp\_neighbors, execute the task with the module **sonic\_bgp\_neighbor**. Provide the configuration data as a list under the “**config**”key. Here **“bgp\_asn”** is the primary key.

bgp\_neighbor ipv4 configuration data list should be under the key **“neighbor: ipv4:”**.

Supported keys/arguments - value:

| **bgp\_asn** | - a primary key argument  - Configures BGP for an autonomous system (AS) number  - bgp as\_number. Ex: **"1001"** |
| --- | --- |
| **neighbor:** ..**ipv4:**  **....ips** | - 32-bit IPv4 address to establish the peering session with bgp peers  - router id. Ex: **["10.10.10.2", "10.10.10.3"]** |
| **neighbor:** ..**ipv4:**  ....**remote\_as** | - Configure neighbor remote AS. Ex: **1002**  - Two BGP routers become peers or neighbors once you establish a TCP connection b/w them.  - command: **neighbor 10.10.10.2 remote-as 1002** |
| **neighbor:** ..**ipv4:** ....**extended\_nexthop** | - extended next-hop boolean type, **true** or **false**  - default value is **false**  - Shutdown the BGP neighbor  - command: **neighbor 10.10.10.2 capability extended-nexthop** |

Sample playbook task

- name: Sonic BGP neighbor config

sonic\_bgp\_neighbor:

config:

- bgp\_asn: 1001

neighbor:

ipv4:

ips: ['40.0.0.2','40.0.0.10']

remote\_as: 1002

extended\_nexthop: true

## bgp\_route\_maps

To configure bgp route maps, execute the task with the module **sonic\_bgp\_route\_maps**. Provide the configuration data as a list under the “**config**”key. Here **“map\_name”** is the primary key.

Supported keys/arguments - value:

| **map\_name** | - a primary key argument  - Configures BGP route map  - map\_name. Ex: **"RM\_SET\_SRC"** |
| --- | --- |
| **action** | - action should be **"permit"** with **sequence\_num**  - command: **route-map RM\_SET\_SRC permit 10** |
| **sequence\_num** | - configure route map sequence number  - Ex: **10**  - command: **route-map RM\_SET\_SRC permit 10** |
| **set:**  ..**ip:** | - set the source ip  - command: **set src 10.10.10.2** |

Sample playbook task

- name: Sonic route map config

sonic\_bgp\_route\_maps:

config:

- map\_name: "RM\_SET\_SRC"

action: permit

sequence\_num: 10

set:

ip: 10.10.10.2

## bgp\_address\_family

To configure bgp address family, execute the task with the module **sonic\_bgp\_address\_family**. Provide the configuration data as a list under the “**config**”key. Here **“bgp\_asn”** is the primary key.

bgp\_adress\_family ipv4 configuration data must be set under the key **“address\_family: ipv4:”**.

Supported keys/arguments - value:

| **bgp\_asn** | - a primary key argument  - Configures BGP for an autonomous system (AS) number  - bgp as\_number. Ex: **"1001"** |
| --- | --- |
| **address\_family:  ..ipv4:**  ....**ips** | - 32-bit IPv4 address to establish the peering session with bgp peers  - router id. Ex: **["10.10.10.2", "10.10.10.3"]** |
| **address\_family:  ..ipv4:**  ....**neighbor:** ......**allowas\_in** | - Accept as-path with my AS present in it  - This command is used to allow the BGP peer group to receive updates from neighboring autonomous systems (AS) with a different origin AS than the local AS, providing greater flexibility in accepting BGP routes.  - value accepted for the argument **“allowas\_in”** is **origin** or b/w **1-10**  - command: **neighbor 10.10.10.2 allowas-in 1** |
| **address\_family:  ..ipv4:**  **....neighbor: ......route\_reflector\_client** | - Configure a neighbor as Route Reflector client  - boolean type, **true** or **false**  - command: **neighbor 40.0.0.11 route-reflector-client** |
| **address\_family:  ..ipv4:**  **....neighbor: ......next\_hop\_self** | - Next hop  - Set local router as next-hop for routes received from the peer group, affecting routing within the autonomous system (AS). Optional "force" modifies next-hop for eBGP-learned routes  - boolean type, **true** or **false**  - command: **neighbor 40.0.0.11 next-hop-self force** |
| **address\_family:  ..ipv4:**  **....network** | - Configures an IP v4 prefix for advertisement  - IPv4 Address - Router IP address with masklen  - Ex: **["10.10.10.2/31", "10.10.10.3/31"]**  - command: **network 10.10.10.2/31** |
| **address\_family:  ..ipv4:**  **....redistribute** | - This command enables the redistribution of connected or static routes into the BGP routing table, allowing them to be advertised to BGP peers and become part of the BGP routing decision process  - value accepted for the argument **“redistribute”** is **connected** and **static** in a list  Ex: **["connected", "static"]**  - command: **redistribute connected**  - command: **redistribute static** |
|  |  |

Sample playbook task

- name: Sonic router BGP config

sonic\_bgp\_address\_family:

config:

- bgp\_asn: 1001

address\_family: # address-family

ipv4: #address-family ipv4 unicast

neighbor:

ips: ['40.0.0.2','40.0.0.10']

allowas\_in: 1 # 1-10 or origin

network: ['40.0.0.2/31','40.0.0.10/31']

redistribute: ["connected", "static"]

# MLAG

Multi-Chassis Link Aggregation Group, executes the task with the module **sonic\_mlag**. Provide the configuration data as a list under the “**config**”key. Here **“domain\_id”** is the primary key.

Supported keys/arguments - value:

| **domain\_id** | - a primary key argument  - Ex: 1 |
| --- | --- |
| **peer\_address** | - Sets the IP address of the peer MLAG switch to establish communication and synchronisation between MLAG-enabled switches  - provide the IP address. Ex: **10.4.4.4** |
| **src\_address** | - Display LLDP neighbor information  - provide the IP address. Ex: **10.4.4.5** |
| **peer\_link** | - Provide a **port\_channel id** or an **interface** **name**  - **port\_channel:** Designates a port-channel as the communication link between MLAG switches, facilitating synchronization and control traffic. Ex: "portchannel20" / "pch10"  - **interface**: Assigns a physical Ethernet interface as the link for communication between MLAG switches, ensuring coordination and data exchange. Ex: "Ethernet4" |
| **member\_portchannels** | - Configures a port-channel as a member of the MLAG for **enhanced redundancy** and **load balancing**  - Provide the list of **port\_channel**  Ex: ["portchannel20", "pch10"] |
| **local\_interface** | - Provide a **VLAN** **id.** Ex: 100  - Configures a VLAN as the local interface for MLAG, allowing the MLAG-enabled switches to communicate and synchronize information for **enhanced redundancy** and **load balancing** within the specified VLAN |

Sample playbook task

- name: "Sonic Mlag creation "

sonic\_mlag:

config:

- domain\_id: 1

peer\_address: 192.168.0.3

peer\_link: "portchannel999"

src\_address: 192.168.0.2

member\_portchannels: ['pch201', 'pch202', 'pch501']

local\_interface: 10