Supermicro DataCenter SONiC

Configuration Guide

Revision 1.5

Manual Revision 1.5

Release Date: 09-Nov-2023

Document Revision History

Date	Revision	Description
09/30/2022	1.0	First release.
01/09/2023	1.1	Added firmware upgrade, ZTP, syslog sections. Added network, redistribute connected, and redistribute static sections in BGP. Added boot-up options, warm-boot, loopback interface, interface naming mode, MAC, and IPv6 interface for management interface sections.
03/23/2023	1.2	Added route reflector, route reflector configuration and save the BGP configuration sections. Updated NTP section with IPv6 configurations. Added ACL and MCLAG section.
04/07/2023	1.3	Added IPv6 ACL section. Updated default Auto-negotiation and MCLAG show commands sections.
06/15/2023	1.4	Add Annex 1 for T7132
11/06/2023	1.5	Annex 1 is removed and it's contents are merged with appropriate sections. Updated MCLAG with L2 and IPv6 configurations. Added port mirroring section.

Contents

1 Introduction

This document explains the switch configuration for Supermicro switch model SSE-G3748.

Software for **O**pen **N**etworking **i**n the **C**loud (SONiC) is a Linux based open-source network operating system that runs on different hardware platforms to meet the requirements of cloud data center. SONiC has various modules implemented as containers that interact with each other.

SONiC switches can be configured from Command Line Interface (CLI). This CLI can be used to configure as well as to display the configuration state and status. The CLI is accessible through a RS232 console port and SSH connections.

The configuration commands need root privileges to execute them and the commands are case-sensitive. The show commands can be executed by all users without the root privileges. Root privileges can be obtained either by prefixing "sudo" keyword to all config commands, or by invoking the root prompt using the command "sudo -i".

1.1 Switch initial configuration

By default, all the show and config commands support '-?', '-h' and '--help' options, which help to understand the command and it's usage.

Parameter	**Default Value**
Management IP	DHCP
Login username	admin
Password	YourPaSsWoRd
Serial Baud rate	115200

1.1.1 Console Port

The SSE-G3748 have an RJ45 connector for the RS232 console port.

Use the serial cable provided with the switch to connect the RS232 port to any computer.

The computer COM port settings should be as follows:

Baudrate: 115200
Data: 8 bit
Parity: none

Stop: 1 bit
Flow Control: none

1.1.2 Initial Switch Access

Switch prompts the user to change the default password on first login attempt as shown below.

sonic login: admin
Password: YourPaSsWoRd
You are required to change your password immediately (administrator enforced)
Changing password for admin.
Current password: YourPaSsWoRd
New password: <new-password>
Retype new password: <new-password>

The management port (eth0) is configured in DHCP mode to obtain an IP address automatically from a DHCP server. To configure a static IP address, refer to Static IP Address Configuration in this document. The management IP address of the switch can be viewed as shown below.

```
admin@sonic: ~$ show ip interfaces |grep eth0
eth0              192.168.86.10/24      up/up       N/A         N/A
admin@sonic: ~$
```

For further information on Management Access refer to Management Access of this user guide.

The G3748 switch has 54 data ports to service the data - 48 1G ports with RJ45 connectors and 6 SFP28 ports. It also has 1 management port, through which user can SSH into the switch. The management ethernet port doesn't participate in the switching functionalities. The switch has console port, which is set at 115200 baud rate.

1.1.3 Simple L2 Switch Configuration

All the data ports are configured as layer-3 routed port by default and no Layer 2 VLANs are enabled. To configure all the ports as layer-2 switch ports, run the script 'setup_all_ports_l2.py' as shown below. This script creates VLAN 10 and assigns all the data ports as a untagged member of VLAN 10. The script runs for few minutes and that is normal.

```
admin@sonic:~$ setup_all_ports_l2.py
[14516.022613] 8021q: 802.1Q VLAN Support v1.8
[14516.039604] IPv6: ADDRCONF(NETDEV_UP): Vlan10: link is not ready
[14522.696275] Bridge: port 2(Ethernet0) entered blocking state
[14522.702100] Bridge: port 2(Ethernet0) entered disabled state
[14522.729906] device Ethernet0 entered promiscuous mode
admin@sonic:~$
```

```
admin@sonic:~$ show vlan config
Name     VID  Member     Mode
------  -----  ----------  --------
Vlan10    10  Ethernet0   untagged
Vlan10    10  Ethernet1   untagged
Vlan10    10  Ethernet2   untagged
Vlan10    10  Ethernet3   untagged
Vlan10    10  Ethernet4   untagged
Vlan10    10  Ethernet5   untagged
Vlan10    10  Ethernet6   untagged
Vlan10    10  Ethernet7   untagged
Vlan10    10  Ethernet8   untagged
Vlan10    10  Ethernet9   untagged
Vlan10    10  Ethernet10  untagged
Vlan10    10  Ethernet11  untagged
Vlan10    10  Ethernet12  untagged
Vlan10    10  Ethernet13  untagged
Vlan10    10  Ethernet14  untagged
Vlan10    10  Ethernet15  untagged
Vlan10    10  Ethernet16  untagged
Vlan10    10  Ethernet17  untagged
Vlan10    10  Ethernet18  untagged
Vlan10    10  Ethernet19  untagged
Vlan10    10  Ethernet20  untagged
Vlan10    10  Ethernet21  untagged
Vlan10    10  Ethernet22  untagged
Vlan10    10  Ethernet23  untagged
Vlan10    10  Ethernet24  untagged
Vlan10    10  Ethernet25  untagged
Vlan10    10  Ethernet26  untagged
Vlan10    10  Ethernet27  untagged
Vlan10    10  Ethernet28  untagged
Vlan10    10  Ethernet29  untagged
Vlan10    10  Ethernet30  untagged
Vlan10    10  Ethernet31  untagged
Vlan10    10  Ethernet32  untagged
Vlan10    10  Ethernet33  untagged
Vlan10    10  Ethernet34  untagged
Vlan10    10  Ethernet35  untagged
Vlan10    10  Ethernet36  untagged
Vlan10    10  Ethernet37  untagged
Vlan10    10  Ethernet38  untagged
Vlan10    10  Ethernet39  untagged
Vlan10    10  Ethernet40  untagged
Vlan10    10  Ethernet41  untagged
Vlan10    10  Ethernet42  untagged
Vlan10    10  Ethernet43  untagged
Vlan10    10  Ethernet44  untagged
```

```
Vlan10    10  Ethernet45  untagged
Vlan10    10  Ethernet46  untagged
Vlan10    10  Ethernet47  untagged
Vlan10    10  Ethernet48  untagged
Vlan10    10  Ethernet49  untagged
Vlan10    10  Ethernet50  untagged
Vlan10    10  Ethernet51  untagged
Vlan10    10  Ethernet52  untagged
Vlan10    10  Ethernet53  untagged
admin@sonic:~$
```

Though this script configures all the data port with VLAN 10, user can change it to any other preferred VLAN using VLAN configuration commands. Refer to VLAN section to change the access VLAN or to configure the port in trunk mode.

Before enabling the layer-2 on the ports, make sure there is no loop in network topology and other switches/bridges in the network.

1.2 Definitions and Acronyms

DHCP - Dynamic Host Configuration Protocol

IP - Internet Protocol

MTU - Maximum Transmission Unit

NTP - Network Time Protocol

UDP - User Datagram Protocol

TTL - Time to live

DSCP - Differentiated Services Code Point

TLV - Type Length Value

TACACS - Terminal Access Controller Access Control System

SNMP - Simple Network Management Protocol

VLAN - Virtual LAN

LAN - Local Area Network

PVID - Port VLAN ID

LA - Link Aggregation

LACP - Link Aggregation Control Protocol

LLDP - Link Layer Discovery Protocol

MIB - Management Information Base

TCP - Transmission Control Protocol

ARP - Address Resolution Protocol

MAC - Media Access Control

1.3 Introduction about Switch Models

Supermicro has different switch models that support SONiC OS. This section gives brief overview about the models.

1.3.1 SSE-G3748

The SSE-G3748 switch has 54 data ports to service the data – 47 RJ45 ports and 6 25G SFP+ ports. The SFP+ ports can be configured at 10G speed.

SSE-G3748 also has an out-of-band 1G management port, through which user can SSH into the switch. The switch has one console port, which is set at 115200 baud rate.

1.3.2 SSE-T7132

SSE-T7132 switch has 34 data ports to service the data – 32 QSFP-DD ports and 2 10G SFP+ ports. The 32 QSFP-DD connectors can be configured to operate at different speeds or can be split to provide more number of logical ports. By default the switch comes with 400G x 32 ports + 10G x 2 ports in layer 3 configuration.

Default Settings

Interface Name	Interface Numbers	Speed	MTU	Autoneg	FEC
400 Gigabit ethernet	Ethernet0 – Ethernet248	400G	9100	Disabled	RS
10 Gigabit ethernet	Ethernet256 – Ethernet257	10G only	9100	Disabled	None

In show interfaces commands, if the naming mode is default, then use the interface name. For example, show interfaces status Ethernet0.
If the naming mode is set to alias, then use the alias name. For example, show interfaces status Eth0.

SSE-T7132 also has an out-of-band 1G management port, through which user can SSH into the switch. The switch has one console port, which is set at 115200 baud rate.

2 System Configuration

This Section describes the System features supported in SONiC.

2.1 Management IP

The management interface (eth0) in SONiC, by default, is configured in DHCP mode. Connect the management interface to the same network to which your DHCP server is connected. The IP address received from DHCP server can be viewed using the 'show ip interfaces |grep eth0' command.

Static IP address can be used as an alternate if there is no DHCP server in your network. In DHCP mode, switch gets the default gateway address from the DHCP server. If the switch is configured with a static IP address, then the gateway should be configured manually.

2.1.1 Interface IP Address Configuration

Follow the steps below to manually configure management interface IP address.

Step	Command	Description
Step 1	**config interface ip add <interface_name> <ip_addr> <default gateway IP address>**	Configure the management interface IP address manually. Interface name - may be any of the following: Ethernet0 – Ethernet53 and eth0 <ip_addr> - An IPv4 Address to be configured on the interface. <default gateway IP address> - IPv4 Address of the default gateway.
Step 2	**show ip interfaces |grep eth0**	Displays the management interface IP address.
Step 3	**sudo config save –y**	Optional step - Saves the running configuration to be part of startup configuration.

The following example shows the commands used to manually configure the management interface with IP address 192.168.86.10 and default gateway 192.168.86.100.

```
admin@sonic: ~$ sudo config interface ip add eth0 192.168.86.10/24 192.168.86.100/24
```

The example below shows how to view the management IP address.

```
admin@sonic: ~$ show ip interfaces |grep eth0
```

```
eth0            192.168.86.10/24    up/up       N/A         N/A
admin@sonic: ~$
```

Zero Touch Provisioning (ZTP) is enabled by default. The ZTP restarts the network discovery to get ZTP parameters until the ZTP files are found. This might cause the switch unreachable intermittently. **If ZTP is not used, then keep the ZTP disabled and save the config by using the below commands.**

ztp disable -y
config save -y

2.1.2 Gateway Configuration

Follow the steps below to manually configure the gateway.

admin@sonic: ~$ **sudo config route add prefix 0.0.0.0/0 nexthop 192.168.86.1**

The example below shows how to view the default gateway.

admin@sonic: ~$ **show ip route |grep 0.0.0.0/0**
S>* 0.0.0.0/0 [1/0] via **192.168.86.1**, eth0, weight 1, 00:09:22

2.2 Management Access

SONiC switches enable access control of the switch by user name and password.

2.2.1 Defaults

Parameter	**Default Value**
User Name/Password	admin/YourPaSsWoRd

2.2.2 Configure User

The user admin is available by default. Additional users can be created to access the switch. Each user id have it's own password against which the users are authenticated at the time of login to the switch.

Follow the steps below to create User and Password.

Step	Command	Description
Step 1	**useradd [options] LOGIN** **useradd -D [options]**	Useradd - Create new user. -D – Default (print or change default useradd configuration) LOGIN - new value of the login name NOTE: Refer Linux manual for options related to create user.
Step 2	**passwd [options] [LOGIN]**	Passwd - Configure password LOGIN - new value of the login name NOTE: Refer Linux manual for options related to create user.
Step 3	**sudo users**	Display the users currently logged in to the switch.
Step 4	**show users**	List of users currently logged in to the switch along with the IP address of the login session.

The example below shows the commands used to configure users.

```
admin@sonic: ~$ sudo useradd supermicro
admin@sonic: ~$ sudo passwd supermicro
New password:
Retype new password:
passwd: password updated successfully
admin@sonic: ~$ sudo users
admin supermicro
admin@sonic: ~$
(Note: Please logout and log back in new user for changes take effect.)
admin@sonic: $ show users
admin    ttyS0       2021-07-21 22:26
supermicro pts/0       2021-07-21 22:34 (192.168.86.38)
admin@sonic: $
```

2.2.3 Modify User

Follow the steps below to modify the User.

Step	Command	Description
Step 1	**usermod [options] [LOGIN]**	Usermod - Modify user LOGIN - new value of the login name NOTE: Refer Linux manual for options related to modify user.
Step 2	**sudo users**	Display the users currently logged in to the switch.
Step3	**show users**	List of users currently logged in to the switch along with the IP address of the login session.

The example below shows the commands used to modify users.

```
admin@sonic: ~$ sudo usermod supermicro -l supermicro_test
admin@sonic: ~$ sudo users
admin   supermicro_test
admin@sonic: ~$
admin@sonic: ~$ sudo passwd supermicro_test
New password:
Retype new password:
passwd: password updated successfully
(Note: Please logout and log back in modified user for changes take effect.)
admin@sonic: ~$ show users
admin          ttyS0       2021-07-21 22:26
supermicro_test    pts/0       2021-07-21 22:34 (192.168.86.38)
admin@sonic: ~$
```

2.2.4 Remove User

Follow the steps below to remove the User.

Step	Command	Description
Step 1	**userdel [options] LOGIN**	Userdel - Delete user

		LOGIN - The login name of the user to be removed. NOTE: Refer Linux manual for options related to delete user.
Step 2	**sudo users**	Display current user
Step3	**show users**	List of users currently logged in to the device

The example below shows the commands used to remove users.

```
admin@sonic: ~$ sudo userdel supermicro_test
admin@sonic: ~$ sudo users
admin
admin@sonic: ~$ show users
admin    ttyS0      2021-07-21 22:26
admin    pts/0      2021-07-21 22:35 (192.168.86.38)
admin@sonic: ~$
```

2.3 Interface Properties

SONiC switches support various types of interfaces – physical interfaces, port channel interfaces and VLAN interfaces. Each interface has different characteristics, some of which are configurable.
The switch has two types of physical interfaces – 48 ports with 1G speed and six ports with 25G speed.

25G Ports (10G Ports)
The switch has six 25G speed capable SFP28 ethernet ports. These ports can also be configured to operate at 10G speed with SFP+ cables and transceivers and also can operate at 1G speed. These ports are named from Ethernet48 to Ethernet53

1 Gigabit Ethernet Ports
The switch has 48 Gigabit ethernet ports and they operate at 1G speed. These ports can be configured to operate at 100M speed.

2.3.1 Defaults

Interface Name	Interface Numbers	Speed	MTU	Autoneg	FEC
Gigabit ethernet	Ethernet0 - Ethernet47	1G default Can operate in 100Mb	9100	Enabled	N/A
Fx-ethernet	Ethernet48 – Ethernet53	25G default Can operate in 10G/1G	9100	Enabled	None

In show interfaces commands, if the naming mode is default, then use the interface name. For example, show interfaces status Ethernet1.
If the naming mode is set to alias, then use the alias name. For example, show interfaces status Gi0/2.

2.3.2 Description

Follow the steps below to display interface description string.

Step	Command	Description
Step 1	**show interfaces description [interface_name]**	Displays the interface description configuration. Interface name - may be any of the following: Ethernet0 Ethernet53

The example below shows the commands used to display interface description.

Output from SSE-G3748:

```
admin@sonic: ~$ show interfaces description
   Interface    Oper    Admin       Alias    Description
-----------       ------     -------     -------------  -------------
  Ethernet0     up      up    Gi0/1
```

```
 Ethernet1   down      up   Gi0/2
 Ethernet2   down      up   Gi0/3
 Ethernet3   down      up   Gi0/4
 Ethernet4   down      up   Gi0/5
 Ethernet5   down      up   Gi0/6
 Ethernet6   down      up   Gi0/7
 Ethernet7   down      up   Gi0/8
 Ethernet8   down      up   Gi0/9
 Ethernet9   down      up   Gi0/10
Ethernet10   down      up   Gi0/11
Ethernet11   down      up   Gi0/12
Ethernet12   down      up   Gi0/13
Ethernet13   down      up   Gi0/14
Ethernet14   down      up   Gi0/15
Ethernet15   down      up   Gi0/16
Ethernet16   down      up   Gi0/17
Ethernet17   down      up   Gi0/18
Ethernet18   down      up   Gi0/19
Ethernet19   down      up   Gi0/20
Ethernet20   down      up   Gi0/21
Ethernet21   down      up   Gi0/22
Ethernet22   down      up   Gi0/23
Ethernet23   down      up   Gi0/24
Ethernet24   down      up   Gi0/25
Ethernet25   down      up   Gi0/26
Ethernet26   down      up   Gi0/27
Ethernet27   down      up   Gi0/28
Ethernet28   down      up   Gi0/29
Ethernet29   down      up   Gi0/30
Ethernet30   down      up   Gi0/31
Ethernet31   down      up   Gi0/32
Ethernet32   down      up   Gi0/33
Ethernet33   down      up   Gi0/34
Ethernet34   down      up   Gi0/35
Ethernet35   down      up   Gi0/36
Ethernet36   down      up   Gi0/37
Ethernet37   down      up   Gi0/38
Ethernet38   down      up   Gi0/39
Ethernet39   down      up   Gi0/40
Ethernet40   down      up   Gi0/41
Ethernet41   down      up   Gi0/42
```

```
 Ethernet42    down      up   Gi0/43
 Ethernet43    down      up   Gi0/44
 Ethernet44    down      up   Gi0/45
 Ethernet45    down      up   Gi0/46
 Ethernet46    down      up   Gi0/47
 Ethernet47    down      up   Gi0/48
 Ethernet48    down      up    Fx0/1
 Ethernet49    down      up    Fx0/2
 Ethernet50    down      up    Fx0/3
 Ethernet51    down      up    Fx0/4
 Ethernet52    down      up    Fx0/5
 Ethernet53    down      up    Fx0/6
admin@sonic: ~$
admin@sonic: ~$ show interfaces description Ethernet33
  Interface    Oper    Admin     Alias    Description
-----------          ------      -------  ---------  -------------
Ethernet33    down      up   Gi0/34
```

Output from SSE-T7132:

```
admin@sonic:~$ show interface des
  Interface    Oper    Admin    Alias    Description
-----------  ------  -------  -------  -------------
  Ethernet0    down      up    Eth1
  Ethernet8    down      up    Eth2
 Ethernet16    down      up    Eth3
 Ethernet24    down      up    Eth4
 Ethernet32    down      up    Eth5
 Ethernet40    down      up    Eth6
 Ethernet48    down      up    Eth7
 Ethernet56    down      up    Eth8
 Ethernet64    down      up    Eth9
 Ethernet72    down      up   Eth10
 Ethernet80    down      up   Eth11
 Ethernet88    down      up   Eth12
 Ethernet96      up      up   Eth13
Ethernet104    down      up   Eth14
Ethernet112    down      up   Eth15
Ethernet120    down      up   Eth16
Ethernet128    down      up   Eth17
Ethernet136    down      up   Eth18
```

```
Ethernet144   down     up   Eth19
Ethernet152   down     up   Eth20
Ethernet160    up      up   Eth21
Ethernet168   down     up   Eth22
Ethernet176   down     up   Eth23
Ethernet184   down     up   Eth24
Ethernet192   down     up   Eth25
Ethernet200   down     up   Eth26
Ethernet208   down     up   Eth27
Ethernet216   down     up   Eth28
Ethernet224   down     up   Eth29
Ethernet232   down     up   Eth30
Ethernet240   down     up   Eth31
Ethernet248   down     up   Eth32
Ethernet256   down     up   Eth33
Ethernet257   down     up   Eth34
admin@sonic:~$ show interface des Ethernet96
  Interface    Oper    Admin    Alias    Description
-----------  ------  -------  -------  -------------
 Ethernet96     up      up    Eth13
```

2.3.3 Auto-negotiation

Interface speed is negotiated between the connected devices, if both ends support negotiation.

Auto negotiation is enabled by default on all the Gi ports and all the Fx ports.

Step	Command	Description
Step 1	**config interface autoneg <interface_name> <mode>**	Turn on/off the auto-negotiation. Interface name - may be any of the following interfaces: Ethernet0 – Ethernet53 <mode> - Enabled/disabled.
Step 2	**show interface autoneg status**	Displays the auto-negotiation status for all interfaces.

Step 3	**sudo config save –y**	Optional step - Saves this current configuration to be part of startup configuration.

The example below shows the commands used to configure Interface Negotiation.

```
admin@sonic:~$ sudo config interface autoneg Ethernet0 enabled
admin@sonic:~$ show interface autoneg status |grep Ethernet0
  Ethernet0       enabled     1G        N/A   N/A       N/A    up     up

admin@sonic:~$ sudo config interface autoneg Ethernet50 disabled
admin@sonic:~$ show interface autoneg status |grep Ethernet50
 Ethernet50      disabled    25G       N/A   N/A       N/A   down    up
admin@sonic:~$
```

2.3.3.1 Auto-negotiation in SSE-T7132

Interface speed is negotiated between the connected devices, if both ends support negotiation. The auto-negotiation in high speed interfaces includes FEC and link training. Currently, auto-negotiation cannot be enabled or disabled from SONiC CLI. SSE-T7132S does not support 400G auto-negotiation.

Auto-negotiation can be enabled while creating the devport. Link training must be enabled with Auto-negotiation and base-page technology abilities must be set appropriately based on the number of lanes assigned to the port.

Auto negotiation is disabled by default on all ports. The switch supports the following modes for the ports.

- 10GBASE-KR
- 40GBASE-KR4
- 40GBASE-CR4
- 100GBASE-KR4
- 100GBASE-CR4
- 25GBASE-KR-S/CR-S
- 25GBASE-KR/CR
- 50GBASE-KR/CR
- 100GBASE-KR2/CR2
- 200GBASE-KR4/CR4

The speed and FEC type is not mandatory when auto-negotiation is enabled.

In the corresponding configuration file for each sku, there are devports and their properties like speed, physical lane number, and number of lanes defined. The first devport with eth type in the yaml config file is mapped to the 1st interface in SONiC such as Ethernet0, 2nd devport with eth type is mapped to Ethernet8 if it is a 400G speed interface. The index number of SONiC Ethernet interface is determined by

the SerDes lane index which starts from zero in SONiC. For example, in the default 400G SKU configuration, each Ethernet interface takes 8 serdes lanes, so the logical interface index is 0, 8, 16, 24….

The example below shows a devport definition in the config file, config_16x400G_64x100G_sse_t7132s.yaml under /usr/share/sonic/device/x86_64-supermicro_sse_t7132s-r0/Supermicro_sse_t7132s_16x400_64x100_habana.

```
devports:
-    id: “0”    ← Devport ID
     sysport: “1000”  ←System-port associated with this devport
  type: “cpu”      ←Devport type: to CPU
-    fec: “KPFEC”   ←FEC type for devport 241
     id: “241”      ←Devport ID
     lanes: “0:8”    ←SerDes lanes associated with this devport
     serdes_group: “30”  ← Innovium Serdes Group associated with this devport
     speed: “400G”     ←Speed
     sysport: “241”     ←system-port associated with devport
     type: “eth”       ←Devport type
```

The speed and FEC type is not mandatory when auto-negotiation is enabled.

The example below shows the config file change to enable auto-negotiation on devport id 241.

```
devports:
-    id: “0”
     sysport: “1000”
  type: “cpu”
-    fec: “KPFEC”   ←Not mandatory when auto-nego is true
     id: “241”      ←Devport ID
     lanes: “0:8”
     serdes_group: “30”
     speed: “400G”     ←Not mandatory when auto-nego is true
     auto_neg: “true”
     link_training: “true”  ← Must be true when auto_nego is true
     bp_tech_ability: “200GBASE_KR4_CR4, 100GBASE_KR2_CR2, 50GBASE_KR_CR”  ← Base page tech ability
     sysport: “241”
     type: “eth”
```

2.3.4 Forward Error Correction (FEC) Mode

The switch allows user to enable/disable FEC mode on the Fx-ethernet ports. FEC mode would be useful in noisy link where a errors in transmission cause retransmission. Switch supports Reed-solmon (RS), and Fire-code (FC) FECs. By default on all Fx-ethernet ports the FEC is set RS.

Step	Command	Description
Step 1	**config interface fec <interface_name> <interface_fec>**	Configure FEC on the interface. <interface_name> - may be any of the following interfaces: Ethernet48 – Ethernet53 <interface_fec> - rs, fc, and none.
Step 2	**show interface status**	Displays the interface status for all interfaces with the current FEC.
Step 3	**sudo config save –y**	Optional step - Saves this current configuration to be part of startup configuration.

Follow the steps below to enable FEC mode on interface.

The example below shows the commands used to configure the FEC for Fx-ethernet interface.

```
admin@sonic:~$ sudo config interface fec Ethernet53 rs
admin@sonic:~$ show interface status |grep -E "Ethernet53|--|Interface"
  Interface    Lanes    Speed    MTU    FEC    Alias    Vlan    Oper    Admin          Type    Asym PFC
-----------  -------  -------  -----  -----  -------  ------  ------  -------  --------------  ----------
 Ethernet53      53      25G   9100     rs   Fx0/6  routed      up      up  SFP/SFP+/SFP28       N/A
admin@sonic:~$
```

2.3.4.1 FEC in SSE-T7132

There are 8 SerDes lanes in each QSFP-DD which can support different lane speed (10G/25G/50G) and form different MAC speeds of 400GbE, 200GbE, 100GbE. The following table shows the combinations of MAC speed, FEC, and Lanes.

MAC Speed	PCS-FEC	Lanes	Start Lane	Restriction
400GbE	**CL119 PCS with KP-FEC**	8	0	None

200GbE	**CL119 PCS with KP-FEC**	8	0	None
200GbE	**CL119 PCS with KP-FEC**	4	0, 4	None
100GbE	**CL91 PCS with KP-FEC/KR-FEC**	2	0,2,4,6	0,2: no KR-FEC mixed with KP-FEC 4,6: no KR-FEC mixed with KP-FEC
100GbE	**CL91 PCS with KP-FEC/KR-FEC**	4	0,4	None
100GbE	**CL82 PCS with no FEC**	4	0,4	None
50GbE	**CL133 PCS with KP-FEC/KR-FEC**	1	0,1,2, …7	0,1,2,3: no KR-FEC mixed with KP-FEC 4,5,6,7: no KR-FEC mixed with KP-FEC
50GbE	**CL133PCS with KP-FEC/KR-FEC/no FEC**	2	0,2,4,6	0,2: no KR-FEC mixed with KP-FEC 4,6: no KR-FEC mixed with KP-FEC
25GbE	**CL107/25GEC PCS with KR-FEC/FC-FEC/no FEC**	1	0,1,2, …7	0,1,2,3: no KR-FEC mixed with KP-FEC 4,5,6,7: no KR-FEC mixed with KP-FEC
40GbE	**CL82 PCS with no FEC**	4	0,4	None
10GbE	**CL49 PCS with no FEC**	1	0,1,2, …7	None

Follow the steps below to enable FEC mode on interface.

The example below shows the field in config file used to configure the FEC for devport 241, interface Ethernet0.

```
devports:
- id: “0”  ← Devport ID
  sysport: “1000” ←System-port associated with this devport
  type: “cpu”  ←Devport type: to CPU
```

```
- fec: “NONE”  ←FEC type for devport 241
  id: “241”
  lanes: “0:8”
  serdes_group: “30”
  speed: “400G”
  sysport: “241”
  type: “eth”
```

The example below shows the commands used to check the FEC setting

```
admin@sonic:~$ show interface status Ethernet0
  Interface                  Lanes   Speed   MTU   FEC   Alias   Vlan   Oper   Admin   Type   Asym PFC
-----------  -----------------------------  -------  -----  -----  -------  ------  ------  -------  ------  ----------
  Ethernet0  241,242,243,244,245,246,247,248   400G  9100  none    Eth1  routed  down    up     N/A
N/A
```

2.3.5 Speed

The Gigabit ethernet interfaces Ethernet0 to Ethernet47 auto-negotiate to operate at 1G or 100Mb by default. The 25G-Ethernet interfaces Ethernet48 to Ethernet53 operate at 25G by default. This default speed can be changed.

25G Ethernet ports can be configured to operate in speed 10G or 1G. FEC and negotiation has to be disabled before setting the 25G-ethernet ports to 10G.

1G Gigabit ethernet ports can be configured to operate at 100Mb. The auto-negotiation has to be disabled before setting the Gi-Ethernet ports to 100Mb.

Step	Command	Description
Step 1	**config interface speed <interface_name> <interface_speed>**	Configure the speed for the interface. Interface name - may be any of the following interfaces: Ethernet0 – Ethernet53 <speed> - The interface speed in Mbps.
Step 2	**show interface status**	Displays the interface status for all interfaces with the current speed.
Step 3	**sudo config save –y**	Optional step - Saves this current configuration to be part of startup configuration.

Follow the steps below to configure Interface speed.

The example below shows the commands used to configure the speed for Gi-ethernet interface.

```
admin@sonic:~$ sudo config interface autoneg Ethernet0 disabled
admin@sonic:~$ sudo config interface speed Ethernet0 100
admin@sonic:~$ show interface status |grep -E "Ethernet0|--|Interface"
  Interface    Lanes    Speed    MTU    FEC    Alias    Vlan    Oper    Admin          Type    Asym PFC
-----------  -------  -------  -----  -----  -------  ------  ------  -------  --------------  ----------
  Ethernet0        0     100M   9100   none    Gi0/1  routed    down       up             N/A        N/A
admin@sonic:~$
```

The example below shows the commands used to configure the speed for Fx-ethernet interface.

```
admin@sonic:~$ sudo config interface autoneg Ethernet53 disabled
admin@sonic:~$ sudo config interface fec Ethernet53 none
admin@sonic:~$ sudo config interface speed Ethernet53 10000
admin@sonic:~$ show interface status |grep -E "Ethernet53|--|Interface"
  Interface    Lanes    Speed    MTU    FEC    Alias    Vlan    Oper    Admin          Type    Asym PFC
-----------  -------  -------  -----  -----  -------  ------  ------  -------  --------------  ----------
 Ethernet53       53      10G   9100   none    Fx0/6  routed      up       up  SFP/SFP+/SFP28        N/A
admin@sonic:~$
```

2.3.5.1 Speed in SSE-T7132

The Ethernet256 and Ethernet257 are fixed at 10G without auto negotiation and cannot be changed.

The speed of other Ethernet interfaces depends on the devport setting in the SKU config file. The switch does not support the speed commands.

2.3.6 Shutdown / Startup

The admin status of interfaces are set to up by default. Follow the steps below to shutdown or startup (no shutdown) the interface.

Step	Command	Description

Step 1	**config interface startup <interface_name>**	Change the admin state of the interface to up. Interface name - may be any of the following interfaces: Ethernet0 – Ethernet53
Step 2	**config interface shutdown <interface_name>**	Change the admin state of the interface to down. Interface name - may be any of the following interfaces: Ethernet0 – Ethernet53
Step 3	**show interface status**	Displays the admin state and operational state of the interfaces.
Step 4	**sudo config save –y**	Optional step - Saves this current configuration to be part of startup configuration.

The example below shows the commands used to shutdown the interface.

```
admin@sonic:~$ sudo config interface shutdown Ethernet0
admin@sonic:~$ show interface status |grep -E "Ethernet0|--|Interface"
  Interface    Lanes    Speed    MTU    FEC    Alias    Vlan    Oper    Admin          Type    Asym PFC
-----------  -------  -------  -----  -----  -------  ------  ------  -------  --------------  ----------
  Ethernet0        0       1G   9100   none    Gi0/1  routed    down    down           N/A        N/A
admin@sonic:~$
```

The example below shows the commands used to startup (no shutdown) the interface.

```
admin@sonic:~$ sudo config interface startup Ethernet0
admin@sonic:~$ show interface status |grep -E "Ethernet0|--|Interface"
  Interface    Lanes    Speed    MTU    FEC    Alias    Vlan    Oper    Admin          Type    Asym PFC
-----------  -------  -------  -----  -----  -------  ------  ------  -------  --------------  ----------
  Ethernet0        0       1G   9100   none    Gi0/1  routed      up      up           N/A        N/A
admin@sonic:~$
```

2.3.7 MTU

The Maximum Transmission Unit (MTU) size is the maximum size of the frame that can be switched through the interface. The MTU value for an interface can be changed.

Follow the steps below to configure Interface MTU.

Step	Command	Description
Step 1	**config interface mtu <interface_name> <mtu_value>**	Configures interface mtu Interface name – may be any of the following: Ethernet0 Ethernet53 Mtu value – maximum transmission unit
Step 2	**show interface status**	Displays the interface configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure Interface MTU.

```
admin@sonic: ~$ sudo config interface mtu Ethernet44 1500
admin@sonic: ~$ show interfaces status
  Interface      Lanes    Speed    MTU    FEC    Alias      Vlan      Oper      Admin     Type      Asym PFC
-------------  -------  --------  -------  ------  -------  ---------  --------  ---------  --------  -------------
Ethernet43      43       1G   9100   none     Gi0/44   routed    down      up        N/A       N/A
 Ethernet44      44       1G   1500   none     Gi0/45   routed    down      up        N/A       N/A
```

2.3.8 Advertised-speed

Follow the steps below to configure Interface advertised-speeds.

Step	Command	Description
Step 1	**config interface autoneg [OPTIONS] <interface_name> <mode>** **config interface advertised-speeds <interface_name> <speed_list>**	Configures interface advertised-speeds Interface name – may be any of the following: Ethernet0 - Ethernet53

		Mode – autoneg enable or disable Speed list – Valid speeds: 1000, 100, 10000, 0, all
Step 2	**show interface autoneg status**	Displays the interface autoneg configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

To configure Advertised speed for an interface, the Auto negotiation has to be enabled on that interface.
e.g: sudo config interface autoneg Ethernet4 enabled

The example below shows the commands used to configure Interface advertised-speed. The 'Rmt Adv Speeds' displayed is the value advertised by the peer device and may vary depending on the peer device's capability.

```
admin@sonic:~$ sudo config interface autoneg Ethernet4 enabled
admin@sonic:~$ sudo config interface advertised-speeds Ethernet4 all
admin@sonic:~$ show interface autoneg status Ethernet4
  Interface   Auto-Neg Mode   Speed   Adv Speeds   Rmt Adv Speeds   Type   Adv Types   Oper   Admin
-----------  ---------------  -------  ------------  ----------------  ------  -----------  ------  -------
  Ethernet4       enabled   1000M        all      100M,1000M    N/A       N/A     up      up

admin@sonic:~$ sudo config interface advertised-speeds Ethernet4 1000
admin@sonic:~$ show interface autoneg status Ethernet4
  Interface   Auto-Neg Mode   Speed   Adv Speeds   Rmt Adv Speeds   Type   Adv Types   Oper   Admin
-----------  ---------------  -------  ------------  ----------------  ------  -----------  ------  -------
  Ethernet4       enabled   1000M       1000M      100M,1000M    N/A       N/A     up      up
admin@sonic:~$
```

2.3.9 Advertised-type

Follow the steps below to configure Interface advertised-type.

Step	Command	Description
Step 1	**config interface autoneg [OPTIONS] <interface_name> <mode>**	Configures interface advertised-type

	config interface advertised-type <interface_name> <interface_type_list>	Interface name – may be any of the following: Ethernet0 - Ethernet53 Mode – autoneg enable or disable Interface type list – Valid interface types:KR, XGMII, KR4, SR4, LR, CR2, XLAUI, XFI, SR, SR2, CR, LR4, none, GMII, SFI, CR4, CAUI, XAUI, CAUI4, KR2, all
Step 2	**show interface autoneg status**	Displays the interface autoneg configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

To configure Advertised type for an interface, the Auto negotiation has to be enabled.
e.g: sudo config interface autoneg Ethernet4 enabled

The example below shows the commands used to configure Interface advertised-types.

```
admin@sonic:~$ sudo config interface autoneg Ethernet10 enabled
admin@sonic:~$ sudo config interface advertised-types Ethernet10 all
admin@sonic:~$ show interface autoneg status Ethernet10
  Interface    Auto-Neg Mode    Speed    Adv Speeds    Type    Adv Types    Oper    Admin
-----------  ---------------  -------  ------------  ------  -----------  ------  -------
 Ethernet10        enabled        1G          N/A       N/A         all      down      up
admin@sonic:~$
admin@sonic:~$ sudo config interface advertised-types Ethernet10 CR
admin@sonic:~$ show interface autoneg status Ethernet10
  Interface    Auto-Neg Mode    Speed    Adv Speeds    Type    Adv Types    Oper    Admin
-----------  ---------------  -------  ------------  ------  -----------  ------  -------
 Ethernet10        enabled        1G          N/A       N/A          CR      down      up
admin@sonic:~$
```

This command will accept only the supported advertised-types for the given platform and given port; the supported advertised-types values will vary based on the platform and port.

2.3.10 Configure IPv4 address

Follow the steps below to configure IPv4 address for an interface .

Step	Command	Description
Step 1	**config interface ip add <interface_name> <ip_addr> <default gateway IP address>**	Configures interface ip Interface name – may be any of the following: Ethernet0 - Ethernet53 Ip addr – A Valid IPv4 address Gateway Ip addr – A Valid IPv4 address
Step 2	**show interface status**	Displays the interface configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure IP address for an interface.

```
admin@sonic:~$ sudo config interface ip add Ethernet3 192.168.80.13/24
admin@sonic:~$ show ip interfaces (Note: Truncated output is added here)
Interface    Master    IPv4 address/mask    Admin/Oper    BGP Neighbor    Neighbor IP
-----------  --------  -------------------  ------------  --------------  -------------
Ethernet3              192.168.80.13/24     up/up         N/A             N/A
docker0                240.127.1.1/24       up/down       N/A             N/A
eth0                   172.18.0.154/24      up/up         N/A             N/A
lo                     127.0.0.1/16         up/up         N/A             N/A
admin@sonic:~$ sudo config interface ip add Ethernet1 192.168.12.13/24 192.168.12.254
admin@sonic:~$ show ip interfaces
Interface    Master    IPv4 address/mask    Admin/Oper    BGP Neighbor    Neighbor IP
-----------  --------  -------------------  ------------  --------------  -------------
Ethernet1              192.168.12.13/24     up/up         N/A             N/A
Ethernet3              192.168.80.13/24     up/up         N/A             N/A
docker0                240.127.1.1/24       up/down       N/A             N/A
eth0                   172.18.0.154/24      up/up         N/A             N/A
```

lo	127.0.0.1/16	up/up	N/A	N/A

2.3.11 Remove IPv4 address

Follow the steps below to remove IPv4 address from an interface.

Step	Command	Description
Step 1	**config interface ip remove <interface_name> <ip_addr>**	Configures interface ip Interface name – may be any of the following: Ethernet0 - Ethernet53 Ip addr – A Valid IPv4 address
Step 2	**show interface status**	Displays the interface configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to remove the IPv4 address from an interface.

```
admin@sonic:~$ sudo config interface ip remove Ethernet1 192.168.12.13/24
admin@sonic:~$ show ip interfaces
  Interface    Master    IPv4 address/mask    Admin/Oper    BGP Neighbor    Neighbor IP
  -----------  --------  -------------------  ------------  --------------  -------------
   Ethernet0              10.0.0.0/31          up/up         BGPNeigh01      10.0.0.1
   Ethernet1              10.0.0.2/31          up/up         BGPNeigh02      10.0.0.3
   Ethernet2              10.0.0.4/31          up/up         BGPNeigh03      10.0.0.5
   Ethernet3              10.0.0.6/31          up/up         BGPNeigh04      10.0.0.7
192.168.80.13/24            N/A                N/A
   Ethernet4              10.0.0.8/31          up/up         BGPNeigh05      10.0.0.9
   Ethernet5              10.0.0.10/31         up/up         BGPNeigh06      10.0.0.11
```

2.3.12Configure IPv6 address

Follow the steps below to configure IPv6 address for an interface .

Step	Command	Description

Step 1	**config interface ip add <interface_name> <ip_addr> <default gateway IP address>**	Configures interface ip Interface name – may be any of the following: Ethernet0 - Ethernet53 Ip addr – A Valid IPv6 address Gateway Ip addr – A Valid IPv6 address. Gateway IP address is optional.
Step 2	**show ipv6 interface**	Displays the interface IPv6 addresses.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure IP address for the interface.

```
admin@sonic:~$ sudo config interface ip add Ethernet0 2::2/64
admin@sonic:~$ show ipv6 interfaces (Note: Truncated output is added here)
Interface    Master    IPv4 address/mask                    Admin/Oper    BGP Neighbor    Neighbor IP
-----------  --------  -----------------------------------  ------------  --------------  -------------
Bridge                 fe80::886e:d3ff:fe7c:5551%Bridge/64  up/down       N/A             N/A
Ethernet0              2::2/64                              up/up         N/A             N/A
                       fe80::ec4:7aff:fe2e:17bd%Ethernet0/64                N/A             N/A
Ethernet1              fe80::ec4:7aff:fe2e:17bd%Ethernet1/64  up/up       N/A             N/A
Ethernet2              fe80::ec4:7aff:fe2e:17bd%Ethernet2/64  up/up       N/A             N/A
Ethernet3              fe80::ec4:7aff:fe2e:17bd%Ethernet3/64  up/up       N/A             N/A
Ethernet4              fe80::ec4:7aff:fe2e:17bd%Ethernet4/64  up/up       N/A             N/A
Ethernet5              fe80::ec4:7aff:fe2e:17bd%Ethernet5/64  up/up       N/A             N/A
```

Below example shows connecting to the switch using the IPv6 address.

```
root@Ubuntu-20:/home/G3748/build# ssh -6 admin@2::2%ens18
The authenticity of host 2::2%ens18 (2::2%ens18)' can't be established.
ECDSA key fingerprint is SHA256:CsAp9CFqVpli4lLz4Liexf1AzzXiUs4HZZuLpfXqJzU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 2::2%ens18' (ECDSA) to the list of known hosts.
admin@2::2%ens18's password:
Linux sonic 4.19.0-12-2-arm64 #1 SMP Debian 4.19.152-1 (2020-10-18) aarch64
You are on
```

```
 ____   ___ _   _ ____
/ ___| / _ \| \ | (_)/ ___|
\___ \| | | |  \| | | |
 ___) | |_| | |\  | | |___
|____/ \___/|_| \_|_|\____|

-- Software for Open Networking in the Cloud --

Unauthorized access and/or use are prohibited.
All access and/or use are subject to monitoring.

Help:    http://azure.github.io/SONiC/

Last login: Fri Mar 24 05:53:23 2023 from 10.13.65.43
admin@sonic:~$
```

2.3.13 Remove IPv6 address

Follow the steps below to remove IPv6 address from an interface.

Step	Command	Description
Step 1	**config interface ip add <interface_name> <ip_addr> <default gateway IP address>**	Configures interface ip Interface name – may be any of the following: Ethernet0 - Ethernet53 Ip addr – A Valid IPv6 address Gateway Ip addr – A Valid IPv6 address
Step 2	**show ipv6 interface**	Displays the interface IPv6 addresses.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to remove the IPv6 address from an interface.

```
admin@sonic:~$ sudo config interface ip remove Ethernet0 2::2/64
admin@sonic:~$ show ipv6 interfaces (Note: Truncated output is added here)
Interface    Master    IPv4 address/mask                Admin/Oper    BGP Neighbor    Neighbor IP
```

```
----------- -------- ------------------------------------ ------------ -------------- -------------
Bridge          fe80::886e:d3ff:fe7c:5551%Bridge/64    up/down      N/A          N/A
Ethernet0         fe80::ec4:7aff:fe2e:17bd%Ethernet0/64  up/up        N/A          N/A
Ethernet1         fe80::ec4:7aff:fe2e:17bd%Ethernet1/64  up/up        N/A          N/A
Ethernet2         fe80::ec4:7aff:fe2e:17bd%Ethernet2/64  up/up        N/A          N/A
Ethernet3         fe80::ec4:7aff:fe2e:17bd%Ethernet3/64  up/up        N/A          N/A
Ethernet4         fe80::ec4:7aff:fe2e:17bd%Ethernet4/64  up/up        N/A          N/A
Ethernet5         fe80::ec4:7aff:fe2e:17bd%Ethernet5/64  up/up        N/A          N/A
```

2.3.14Configure IPv6 address for Management Interface

Follow the steps below to configure IPv6 address for management Interface.

Step	Command	Description
Step 1	**config interface ip add <interface_name> <ip_addr> <default gateway IP address>**	Configures interface IPv6 Interface name – may be any of the following: Ethernet0 - Ethernet53 Ip addr – A Valid IPv6 address Gateway Ip addr – A Valid IPv6 address
Step 2	**show ipv6 interfaces**	Displays IPv6 Address of all interfaces.
Step 3	**show management_interface address**	Displays the management interface IP configuration.
Step 4	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure management Interface IPv6.

SWITCH A:

```
admin@sonic:~$ sudo config interface ip add eth0 2001::1/64
[ 5730.458254] mvneta 7f020000.ethernet eth0: Link is Down
[ 5734.212789] mvneta 7f020000.ethernet eth0: PHY [7f022004.mdio-mii:00] driver [Marvell 88E1510]
[ 5734.221825] mvneta 7f020000.ethernet eth0: configuring for phy/sgmii link mode
[ 5734.229268] mvneta 7f020000.ethernet eth0: Link is Up - 1Gbps/Full - flow control off
[ 5734.232555] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
[ 5734.243201] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
```

```
[ 5735.334464] mvneta 7f020000.ethernet eth0: Link is Down
[ 5738.403390] mvneta 7f020000.ethernet eth0: Link is Up - 1Gbps/Full - flow control off
admin@sonic:~$
admin@sonic:~$ show management_interface address
Management IP address = 2001::1/64
admin@sonic:~$
admin@sonic:~$ show ipv6 interfaces (Note: Truncated output is added here)
Interface    Master    IPv4 address/mask                          Admin/Oper    BGP Neighbor    Neighbor IP
-----------  --------  -----------------------------------------  ------------  --------------  -------------
eth0                   2001::1/64                                 up/up         N/A             N/A
                       fe80::ec4:7aff:fe2e:1635%eth0/64                         N/A             N/A
lo                     ::1/128                                    up/up         N/A             N/A
admin@sonic:~$
```

SWITCH B:

```
admin@sonic:~$ sudo config interface ip add eth0 2001::2/64
[ 5739.430346] mvneta 7f020000.ethernet eth0: Link is Down
[ 5744.164528] mvneta 7f020000.ethernet eth0: PHY [7f022004.mdio-mii:00] driver [Marvell 88E1510]
[ 5744.173756] mvneta 7f020000.ethernet eth0: configuring for phy/sgmii link mode
[ 5744.182417] mvneta 7f020000.ethernet eth0: Link is Up - 1Gbps/Full - flow control off
[ 5744.197762] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
[ 5744.204056] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
[ 5745.286659] mvneta 7f020000.ethernet eth0: Link is Down
[ 5748.355337] mvneta 7f020000.ethernet eth0: Link is Up - 1Gbps/Full - flow control off
admin@sonic:~$
admin@sonic:~$ show management_interface address
Management IP address = 2001::2/64
admin@sonic:~$
admin@sonic:~$ show ipv6 interfaces (Note: Truncated output is added here)
Interface    Master    IPv4 address/mask                          Admin/Oper    BGP Neighbor    Neighbor IP
-----------  --------  -----------------------------------------  ------------  --------------  -------------
eth0                   2001::2/64                                 up/up         N/A             N/A
                       fe80::ec4:7aff:fe2e:6769%eth0/64                         N/A             N/A
lo                     ::1/128                                    up/up         N/A             N/A
admin@sonic:~$
```

Test the connectivity between the switches over IPv6.

SWITCH A:

```
admin@sonic:~$ ping 2001::2
PING 2001::2(2001::2) 56 data bytes
64 bytes from 2001::2: icmp_seq=1 ttl=64 time=0.648 ms
64 bytes from 2001::2: icmp_seq=2 ttl=64 time=0.658 ms
^C
```

```
--- 2001::2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 57ms
rtt min/avg/max/mdev = 0.626/0.644/0.658/0.013 ms
admin@sonic:~$
SWITCH B:
admin@sonic:~$ ping 2001::1
PING 2001::1(2001::1) 56 data bytes
64 bytes from 2001::1: icmp_seq=6 ttl=64 time=0.658 ms
64 bytes from 2001::1: icmp_seq=7 ttl=64 time=0.632 ms
^C
--- 2001::1 ping statistics ---
10 packets transmitted, 5 received, 50% packet loss, time 217ms
rtt min/avg/max/mdev = 0.632/0.641/0.658/0.033 ms
admin@sonic:~$
```

2.3.15 Remove Management Interface IPv6

Follow the steps below to remove management Interface IPv6.

Step	Command	Description
Step 1	**config interface ip remove <interface_name> <ip_addr>**	Removes interface IPv6 Interface name – may be any of the following: Ethernet0 - Ethernet53 Ip addr – A Valid IPv6 address
Step 2	**show ipv6 interfaces**	Displays IPv6 Address of all interfaces.
Step 3	**show management_interface address**	Displays the management interface IP configuration.
Step 4	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to remove management Interface IPv6.

SWITCH A:

```
admin@sonic:~$ sudo config interface ip remove eth0 2001::1/64
[ 7425.080933] mvneta 7f020000.ethernet eth0: Link is Down
[ 7428.759705] mvneta 7f020000.ethernet eth0: PHY [7f022004.mdio-mii:00] driver [Marvell 88E1510]
```

```
[ 7428.772979] mvneta 7f020000.ethernet eth0: configuring for phy/sgmii link mode
[ 7428.780403] mvneta 7f020000.ethernet eth0: Link is Up - 1Gbps/Full - flow control off
[ 7428.792580] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
[ 7428.799610] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
[ 7429.894484] mvneta 7f020000.ethernet eth0: Link is Down
[ 7432.963349] mvneta 7f020000.ethernet eth0: Link is Up - 1Gbps/Full - flow control off
admin@sonic:~$
admin@sonic:~$ show management_interface address
admin@sonic:~$
admin@sonic:~$ show ipv6 interfaces (Note: Truncated output is added here)
Interface    Master    IPv4 address/mask                          Admin/Oper    BGP Neighbor    Neighbor IP
-----------  --------  -----------------------------------------  ------------  --------------  -------------
eth0                   fe80::ec4:7aff:fe2e:1635%eth0/64           up/up         N/A             N/A
lo                     ::1/128                                    up/up          N/A             N/A
admin@sonic:~$
```

SWITCH B:

```
admin@sonic:~$ sudo config interface ip remove eth0 2001::2/64
[ 7415.502728] mvneta 7f020000.ethernet eth0: Link is Down
[ 7419.027702] mvneta 7f020000.ethernet eth0: PHY [7f022004.mdio-mii:00] driver [Marvell 88E1510]
[ 7419.036918] mvneta 7f020000.ethernet eth0: configuring for phy/sgmii link mode
[ 7419.044560] mvneta 7f020000.ethernet eth0: Link is Up - 1Gbps/Full - flow control off
[ 7419.056490] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
[ 7419.062493] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
[ 7420.138419] mvneta 7f020000.ethernet eth0: Link is Down
[ 7423.203414] mvneta 7f020000.ethernet eth0: Link is Up - 1Gbps/Full - flow control off
admin@sonic:~$ show management_interface address
admin@sonic:~$
admin@sonic:~$ show ipv6 interfaces (Note: Truncated output is added here)
Interface    Master    IPv4 address/mask                          Admin/Oper    BGP Neighbor    Neighbor IP
-----------  --------  -----------------------------------------  ------------  --------------  -------------
eth0                   fe80::ec4:7aff:fe2e:6769%eth0/64           up/up         N/A             N/A
lo                     ::1/128                                    up/up          N/A             N/A
admin@sonic:~$
```

2.3.16 Enable IPv6 Link Local

Follow the steps below to enable Interface IPv6.

Step	Command	Description
Step 1	**config interface ipv6 enable use-link-local-only <interface_name>**	Enables interface IPv6 Interface name – may be any of the following:

		Ethernet0 - Ethernet53
Step 2	**show ipv6 link-local-mode**	Display IPv6 link-local-mode
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to enables Interface IPv6.

```
admin@sonic:~$ sudo config interface ipv6 enable use-link-local-only Ethernet5
admin@sonic:~$ sudo config interface ipv6 enable use-link-local-only PortChannel0033
admin@sonic:~$ show ipv6 link-local-mode (Note: Truncated output is added here)
+-----------------+----------+
| Interface Name  | Mode     |
+=================+==========+
+-----------------+----------+
| Ethernet5       | Enabled |
+-----------------+----------+
| Ethernet50      | Disabled |
+-----------------+----------+
| Ethernet51      | Disabled |
+-----------------+----------+
| Ethernet52      | Disabled |
+-----------------+----------+
| Ethernet53      | Disabled |
+-----------------+----------+
| Ethernet6       | Disabled |
+-----------------+----------+
| Ethernet7       | Disabled |
+-----------------+----------+
| Ethernet8       | Disabled |
+-----------------+----------+
| Ethernet9       | Disabled |
+-----------------+----------+
| PortChannel0033 | Enabled |
+-----------------+----------+
```

2.3.17 Disable IPv6 Link Local

Follow the steps below to disables Interface IPv6.

Step	Command	Description
Step 1	**config interface ipv6 disable use-link-local-only <interface_name>**	Disables interface IPv6 Interface name – may be any of the following: Ethernet0 - Ethernet53
Step 2	**show ipv6 link-local-mode**	Display IPv6 link-local-mode
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to disable Interface IPv6.

```
admin@sonic:~$ sudo config interface ipv6 disable use-link-local-only Ethernet5
admin@sonic:~$ sudo config interface ipv6 disable use-link-local-only PortChannel0033
admin@sonic:~$ show ipv6 link-local-mode (Note: Truncated output is added here)
+-----------------+----------+
| Interface Name  | Mode     |
+=================+==========+
+-----------------+----------+
| Ethernet5       | Disabled |
+-----------------+----------+
| Ethernet50      | Disabled |
+-----------------+----------+
| Ethernet51      | Disabled |
+-----------------+----------+
| Ethernet52      | Disabled |
+-----------------+----------+
| Ethernet53      | Disabled |
+-----------------+----------+
| Ethernet6       | Disabled |
+-----------------+----------+
| Ethernet7       | Disabled |
+-----------------+----------+
| Ethernet8       | Disabled |
+-----------------+----------+
| Ethernet9       | Disabled |
+-----------------+----------+
| PortChannel0033 | Disabled |
+-----------------+----------+
admin@sonic:~$
```

2.3.18MAC

Follow the steps below to configure and display MAC Address for L2 Interface.

Step	Command	Description
Step 1	**config interface ip remove vlan<vlan_id> <ip_addr>**	Remove an IP address for a VLAN. vlan_id - may be any vlan number ip_addr - ip address
Step 2	**config vlan add <vid>**	Create a VLAN. vid - May be any vlan number, Range 1 to 4094.
Step 3	**config vlan member add [-u|--untagged] <vlan_id> <member_portname>**	Add an untagged member port in the already created VLAN by using the option -u or --untagged vlan_id - may be any vlan number member_portname - any interface name which is not a router interface
Step 4	**show vlan brief**	Displays all bridge information
Step 5	**show mac [OPTIONS]**	Displays MAC Address information Options: -v, --vlan TEXT – Vlan Id -p, --port TEXT - may be any of the following: Ethernet0 - Ethernet53

The example below shows the commands used to configure and display MAC Address.

```
admin@sonic:~$ sudo config interface ip remove Ethernet0 10.0.0.0/31
admin@sonic:~$ sudo config vlan add 100
[  208.767912] 8021q: 802.1Q VLAN Support v1.8
[  208.784425] IPv6: ADDRCONF(NETDEV_UP): Vlan100: link is not ready
admin@sonic:~$ sudo config vlan member add -u 100 Ethernet0
[  217.979642] Bridge: port 2(Ethernet0) entered blocking state
```

```
[  217.985458] Bridge: port 2(Ethernet0) entered disabled state
[  217.993949] device Ethernet0 entered promiscuous mode
[  218.001930] Bridge: port 2(Ethernet0) entered blocking state
[  218.007734] Bridge: port 2(Ethernet0) entered forwarding state
[  218.015778] IPv6: ADDRCONF(NETDEV_CHANGE): Vlan100: link becomes ready
admin@sonic:~$
admin@sonic:~$ show vlan brief
+-----------+-------------+------------+---------------+------------+----------------------+
|  VLAN ID | IP Address  | Ports       | Port Tagging  | Proxy ARP  | DHCP Helper Address  |
+===========+=============+============+===============+============+======================+
|     100   |             | Ethernet0  | untagged      | disabled   |                      |
+-----------+-------------+------------+---------------+------------+----------------------+
admin@sonic:~$ show mac
No.  Vlan   MacAddress         Port         Type
-----  ------  -----------------------  -------------  -------
   1    100  0C:C4:7A:2E:67:69  Ethernet0  Dynamic
Total number of entries 1
admin@sonic:~$
admin@sonic:/$ show mac -v 100
No.  Vlan   MacAddress         Port         Type
-----  ------  -----------------------  -------------  -------
   1    100  0C:C4:7A:2E:67:69  Ethernet0  Dynamic
Total number of entries 1
admin@sonic:~$
admin@sonic:/$ show mac -p Ethernet0
 No.  Vlan   MacAddress         Port         Type
-----  ------  -----------------------  -------------  -------
   1    100  0C:C4:7A:2E:67:69  Ethernet0  Dynamic
Total number of entries 1
admin@sonic:~$
```

2.3.19Type

Follow the steps below to configure Interface type.

Step	Command	Description
Step 1	**config interface type <interface_name> <interface_type_value>**	Configures interface type Interface name – may be any of the following: Ethernet0 - Ethernet53

		Interface type value - Valid interface types: none, KR, KR4, GMII, XGMII, CR2, SR, CAUI, KR2, SR2, XAUI, XLAUI, SR4, SFI, LR4, XFI, CR4, CR, CAUI4, LR
Step 2	**show interface status**	Displays the interface configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure Interface type.

```
admin@sonic:~$ sudo config interface type Ethernet4 CR4
admin@sonic:~$ show interfaces autoneg status Ethernet4
  Interface    Auto-Neg Mode    Speed    Adv Speeds    Type    Adv Types    Oper    Admin
-----------  ---------------  -------  ------------  ------  -----------  ------  -------
  Ethernet4        disabled       1G          N/A     CR4          N/A    down       up
admin@sonic:~$
```

The interface type is configured correctly by default. It is not recommended to change this default type setting.
This command will accept only the supported interface types for the given platform and port; the supported values will vary based on the platform and port.

2.3.20 Alias

Follow the steps below to display interface alias.

Step	Command	Description
Step 1	**show interfaces alias [interface_name]**	Displays the interface alias configuration. Interface name - may be any of the following: Ethernet0 - Ethernet53

The example below shows the commands used to display interface alias.

```
admin@sonic:~$ show interfaces alias (Note: Truncated output is added here)
```

```
Name          Alias
----------    -------------
Ethernet0     Gi0/1
Ethernet1     Gi0/2
Ethernet2     Gi0/3
Ethernet3     Gi0/4
Ethernet4     Gi0/5
```

2.3.21 Configure Interface Naming Mode

Follow the steps below to configure interface naming mode.

Step	Command	Description
Step 1	**config interface_naming_mode (default | alias)**	Configures interface naming mode default – Default interface name alias – Alias interface name
Step 2	**show interfaces naming_mode**	Displays the interface naming configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure interface naming mode.

```
admin@sonic:~$ sudo config interface_naming_mode alias
Please logout and log back in for changes take effect.
admin@sonic:~$ show interfaces naming_mode
alias
admin@sonic:~$
admin@sonic:~$ sudo config interface_naming_mode default
Please logout and log back in for changes take effect.
admin@sonic:~$ show interfaces naming_mode
default
admin@sonic:~$
```

2.3.22 Counters

Follow the steps below to display interface counters.

Step	Command	Description
Step 1	**show interfaces counters [options]**	Show interface counters Options: -a, --printall -p, --period - TEXT -i, --interface - TEXT -d, --display [all] - Show internal interfaces [default: all] -n, --namespace [] - Namespace name or all

The example below shows the commands used to display interface counters.

```
admin@sonic:~$ show interfaces counters (Note: Truncated output is added here)
  IFACE   STATE   RX_OK   RX_BPS   RX_UTIL   RX_ERR   RX_DRP   RX_OVR   TX_OK   TX_BPS   TX_UTIL
TX_ERR   TX_DRP   TX_OVR
-----------  -------  -------  ---------  ---------  --------  --------  --------  -------  ---------  ---------  --------  --------  --------
 Ethernet0       D        0  0.00 B/s      0.00%         0         0         0        0  0.00 B/s      0.00%         0         0         0
 Ethernet1       D        0  0.00 B/s      0.00%         0         0         0        0  0.00 B/s      0.00%         0         0         0
 Ethernet2       D        0  0.00 B/s      0.00%         0         0         0        0  0.00 B/s      0.00%         0         0         0
```

2.3.22.1 Counters Detailed

Follow the steps below to display interface counters detailed.

Step	Command	Description
Step 1	**show interfaces counters detailed [OPTIONS] <interface_name>**	Displays the interface counters in detail Interface name - may be any of the following: Ethernet0 - Ethernet53 Options:

		-p, --period TEXT - Display statistics over a specified period (in seconds)

The example below shows the commands used to display interface counters in detail.

```
admin@sonic:~$ show interfaces counters detailed Ethernet3 (Note: Truncated output is added here)
Packets Received 64 Octets..... 0
Packets Received 65-127 Octets..... 0
Packets Received 128-255 Octets..... 0
Packets Received 256-511 Octets..... 0
Packets Received 512-1023 Octets..... 0
Packets Received 1024-1518 Octets..... 0
Packets Received 1519-2047 Octets..... N/A
Packets Received 2048-4095 Octets..... N/A
Packets Received 4096-9216 Octets..... 0
Packets Received 9217-16383 Octets..... 0

Total Packets Received Without Errors..... 0
Unicast Packets Received..... 0
```

2.3.22.2 Counters Errors

Follow the steps below to display interface counters errors.

Step	Command	Description
Step 1	**show interfaces counters errors [OPTIONS]**	Displays the interface counters errors Options: -p, --period - TEXT -d, --display [all] - Show internal interfaces [default: all] -n, --namespace [] - Namespace name or all

The example below shows the commands used to display interface counters errors.

```
admin@sonic:~$ show interfaces counters errors (Note: Truncated output is added here)

   IFACE    STATE   RX_ERR   RX_DRP   RX_OVR   TX_ERR   TX_DRP   TX_OVR
----------  -------  --------  --------  --------  --------  --------  --------
```

Ethernet0	D	0	0	0	0	0	0
Ethernet1	D	0	0	0	0	0	0

2.3.22.3 Counters Rates

Follow the steps below to display interface counters rates.

Step	Command	Description
Step 1	**show interfaces counters rates [OPTIONS]**	Displays the interface counters rates Options: -p, --period - TEXT -d, --display [all] - Show internal interfaces [default: all] -n, --namespace [] - Namespace name or all

The example below shows the commands used to display interface counters rates.

admin@sonic:~$ **show interfaces counters rates** (Note: Truncated output is added here)

IFACE	STATE	RX_OK	RX_BPS	RX_PPS	RX_UTIL	TX_OK	TX_BPS	TX_PPS	TX_UTIL
Ethernet0	D	0	0.00 B/s	0.00/s	0.00%	0	0.00 B/s	0.00/s	0.00%
Ethernet1	D	0	0.00 B/s	0.00/s	0.00%	0	0.00 B/s	0.00/s	0.00%

2.3.22.4 Counters Rif

Follow the steps below to display interface counters rif.

Step	Command	Description
Step 1	**show interfaces counters rif [OPTIONS] <interface_name>**	Displays all the interface RIFs counters Interface name - may be any of the following: Ethernet0 - Ethernet53 Options:

		-p, --period TEXT - Display statistics over a specified period (in seconds)

The example below shows the commands used to display interface counters rif.

```
admin@sonic:~$ show interfaces counters rif Ethernet4
Ethernet4
---------
    RX:
        0 packets
        0 bytes
        0 error packets
        0 error bytes
    TX:
        0 packets
        0 bytes
        0 error packets
        0 error bytes
admin@sonic:~$
```

2.3.23 Configure loopback

Follow the steps below to configure loopback.

Step	Command	Description
Step 1	**config loopback add <loopback_name>**	Creates loopback interface. Loopback name – A valid string with prefix “loopback” & suffix range of <0-999>
Step 2	**config interface ip add <interface_name> <ip_addr> <default gateway IP address>**	Configures interface ip Interface name – may be any of the following: Ethernet0 - Ethernet53 Ip addr – A Valid IPv4 address Gateway Ip addr – A Valid IPv4 address

Step 3	**show ip interfaces**	Displays the interfaces configuration.
Step 4	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to create loopback interface.

```
admin@sonic:~$ sudo config loopback add Loopback11
admin@sonic:~$ sudo config interface ip add Loopback11 10.1.0.2/32
admin@sonic:~$ show ip interfaces
Interface    Master    IPv4 address/mask    Admin/Oper    BGP Neighbor    Neighbor IP
-----------  --------  -------------------  ------------  --------------  -------------
Ethernet52             10.0.0.104/31        up/up         ARISTA26T0      10.0.0.105
Ethernet53             10.0.0.106/31        up/up         ARISTA27T0      10.0.0.107
Loopback0              10.1.0.1/32          up/up         N/A             N/A
Loopback11             10.1.0.2/32          up/up         N/A             N/A
docker0                240.127.1.1/24       up/down       N/A             N/A
eth0                   192.168.86.34/24     up/up         N/A             N/A
lo                     127.0.0.1/16         up/up         N/A             N/A
admin@sonic:~$
```

2.3.24 Remove loopback

Follow the steps below to remove loopback.

Step	Command	Description
Step 1	**config loopback del <loopback_name>**	Removes loopback interface. Loopback name – A valid string with prefix “loopback” & suffix range of <0-999>
Step 2	**show ip interfaces**	Displays the interfaces configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to remove loopback interface.

```
admin@sonic:~$ sudo config loopback del Loopback11
admin@sonic:~$ show ip interfaces
```

```
Interface    Master    IPv4 address/mask    Admin/Oper    BGP Neighbor    Neighbor IP
-----------  --------  -------------------  ------------  --------------  -------------
Ethernet52             10.0.0.104/31        up/up         ARISTA26T0      10.0.0.105
Ethernet53             10.0.0.106/31        up/up         ARISTA27T0      10.0.0.107
Loopback0              10.1.0.1/32          up/up         N/A             N/A
docker0                240.127.1.1/24       up/down       N/A             N/A
eth0                   192.168.86.34/24     up/up         N/A             N/A
lo                     127.0.0.1/16         up/up         N/A             N/A
admin@sonic:~$
```

2.3.25Storm Control

Storm control prevents traffic on a LAN from being disrupted by a broadcast, multicast, or unicast storm on one of the physical interfaces. A LAN storm occurs when packets flood the LAN due to errors or mistakes in network configurations, etc. LAN storms degrade network performance.

Storm Control monitors packets passing from an interface to the switching bus and determines if the packet is unicast, multicast, or broadcast. The switch counts the number of packets of a specified type received within the 1-second time interval and compares the measurement with a predefined suppression-level threshold. The port blocks traffic when the rising threshold is reached and remains blocked until the traffic rate drops below the falling threshold before resuming normal forwarding.

Follow the steps below to configure Storm control.

Step	Command	Description
Step 1	config interface storm-control add [OPTIONS] <port_name> <storm_type> <kbps_value>	Configure Storm control for broadcast or unknown-multicast or unknown-unicast packets.
Step 2	config interface storm-control del [OPTIONS] <port_name> <storm_type>	Delete Storm control for broadcast or unknown-multicast or unknown-unicast packets.
Step 3	show storm-control	Display the storm control configuration.
Step 4	sudo config save –y	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure Storm Control.

```
admin@sonic:~$ sudo config interface storm-control add Ethernet10 broadcast 200000
admin@sonic:~$ sudo config interface storm-control add Ethernet20 unknown-multicast 100000
admin@sonic:~$ sudo config interface storm-control add Ethernet20 unknown-unicast  10000
```

```
admin@sonic:~$ show storm-control
+------------------+------------------+--------------+
| Interface Name   | Storm Type       |   Rate (kbps) |
+==================+==================+==============+
| Ethernet10       | broadcast        |       200000 |
+------------------+------------------+--------------+
| Ethernet20       | unknown-multicast |       100000 |
+------------------+------------------+--------------+
| Ethernet20       | unknown-unicast   |       10000 |
+------------------+------------------+--------------+
```

The example below shows the commands used to delete Storm Control configuration.

```
admin@sonic:~$ sudo sudo config interface storm-control del Ethernet10 broadcast
admin@sonic:~$ show storm-control
+------------------+------------------+--------------+
| Interface Name   | Storm Type       |   Rate (kbps) |
+==================+==================+==============+
| Ethernet20       | unknown-multicast |       100000 |
+------------------+------------------+--------------+
| Ethernet20       | unknown-unicast   |       10000 |
+------------------+------------------+--------------+
```

2.3.26 Port splitting/HWSKU in SSE-T7132

Each QSFP-DD connector has 8 SerDes lanes and can have 8 logical ports at maximum. The maximum logical port count is 168 including the 2 SFP+ ports per the switch. Each SerDes lane can support 56Gbps PAM4 or 28Gbps NRZ. With combinations of different speed and lane numbers, the switch can have many physical interface configurations. The port configurations are hard coded in the profiles and only loaded at the boot. So, it is necessary to reboot the switch for a new interface configuration after any change. The switch does not support dynamic port breakout and warm boot features due to chipset limitation.

There are few predefined profiles for HWSKUs in the switch image as shown in the following table.

SKU Name	**Interfaces Speed/Type**	**Comment**
Supermicro_sse_t7132s	32 x 400G Ethernet interfaces	This is the default HWSKU
Supermicro_sse_t7132s_128x100	128 x 100G(PAM4) Ethernet interfaces	

Supermicro_sse_t7132s_32x100	32 100G(NRZ) Ethernet interfaces	One port per QSFP-DD connector. QSFP28 DACs or transceivers can be used.
Supermicro_sse_t7132s_64x100	64 x 100G(NRZ) Ethernet interfaces	Two ports per QSFP-DD connector.
Supermicro_sse_t7132s_64x200	64 x 200G Ethernet interfaces	Two ports per QSFP-DD connector.
Supermicro_sse_t7132s_16x400_64x100	64 x 100G(PAM4) and 16 x 400G Ethernet interfaces	First 16 QSFP-DD connectors will be split into 4 ports and operate at 100G (PAM4) and the last 16 QSFP-DD will operate at 400G speed.

Users can change the switch HWSKU by the sonic-cfggen tool. For example, to configure the switch to boot with Supermicro_sse_t7132s_32x100 HWKSU, follow the steps given below.

Step 1: Remove the current configuration.

```
admin@sonic:~$ sudo rm /etc/sonic/config_db.json
```

Step 2: Change the default SKU.

```
admin@sonic:~$ sudo -i
root@sonic:~# sudo echo “Supermicro_sse_t7132s_32x100 t1” >
/usr/share/sonic/device/x86_64-supermicro_sse_t7132s-r0/default_sku
```

Step 3: Reboot the switch to initialize it with new SKU profiles.

```
admin@sonic:~$sudo reboot
```

If there is no suitable predefined HWSKU for your applications, please contact Supermicro support to get the suitable configuration.

There are configuration files under each HWSKU folder to set the interface properties, the following files are for the default 400G SKU.

```
admin@sonic:/usr/share/sonic/device/x86_64-supermicro_sse_t7132s-r0/Supermicro_sse_t7132s$ ls
buffers_defaults_def_lossy.j2   ivm.sai.datapath.config.yaml
buffers_defaults_t1.j2          pg_profile_lookup.ini
buffers.json.j2                 port_config.ini
config_32x400G_sse_t7132s.yaml  qos_defaults_def_lossy.j2
innovium.77700_A                qos_defaults_t1.j2
innovium.77700_B                qos.json.j2
ivm.sai.config.yaml             sai.profile
```

The interface properties such as speed, FEC, and auto negotiation are configured in the config_xxx_sse_t7132s.yaml file, where xxx means the interface number or HWSKU.

Devport id is used to identify the switch physical SerDes lane in the configuration file. The first devport with eth type in the file maps to SONiC Ethernet0, the second devport with eth type in the file maps to SONiC EthernetX, the number of SerDes lanes used by Ethernet0 determines the value of X. For example, Ethernet0 is a 400G interface, then the next interface in SONiC is Ethernet8. The corresponding lane numbers are shown in “show interface status”.

The following is a portion from the default 400G SKU configuration file, config_32x400G_sse_t7132s.yaml, regarding the interface properties.

```
devports:
-   id: “0”   ← Devport ID
    sysport: “1000” ←System-port associated with this devport
  type: “cpu”    ←Devport type: to CPU
-   fec: “KPFEC”  ←FEC type for devport 241
    id: “241”    ←Devport ID
    lanes: “0:8”   ←SerDes lanes associated with this devport
    serdes_group: “30”  ← Innovium Serdes Group associated with this devport
    speed: “400G”    ←Speed
    sysport: “241”    ←system-port associated with devport
    type: “eth”    ←Devport type
-   fec: “KPFEC”
    id: “249”
    lanes: “0:8”
    serdes_group: “31”
    speed: “400G”
    sysport: “249”
    type: “eth”
-   fec: “KPFEC”
    id: “225”
    lanes: “0:8”
```

```
    serdes_group: “28”
    speed: “400G”
    sysport: “225”
    type: “eth”
```

To check the interface status use the command “show interface status”.

```
admin@sonic:~$ show interface status

  Interface                Lanes    Speed    MTU    FEC    Alias    Vlan    Oper    Admin    Type    Asym PFC
-----------  -----------------------------  -------  -----  -----  --------  ------  ------  --------  ------  ----------
  Ethernet0  241,242,243,244,245,246,247,248    400G   9100     rs     Eth1  routed    down      up     N/A
N/A
  Ethernet8  249,250,251,252,253,254,255,256    400G   9100     rs     Eth2  routed    down      up     N/A
N/A
 Ethernet16  225,226,227,228,229,230,231,232    400G   9100     rs     Eth3  routed    down      up     N/A
N/A
 Ethernet24  233,234,235,236,237,238,239,240    400G   9100     rs     Eth4  routed    down      up     N/A
N/A
```

Ethernet0 includes SerDes lanes from 241 to 248, which maps to devport id 241. Ethernet8 includes SerDes lanes from 249 to 256, which maps to devport id 249.

Ethernet16 includes SerDes lanes from 225 to 232, which maps to devport id 225.

Port_config.ini is a configuration file including interface name, SerDes lanes, alias, speed, index, MTU and FEC. Its content should be consistent with the SKU configuration file. The example below is a port_config.ini for 400G sku.

```
admin@sonic:~$ cat /usr/share/sonic/device/x86_64-supermicro_sse_t7132s-r0/Supermicro_sse_t7132s/port_config.ini
# name          lanes                                 alias      speed      index   mtu     fec
Ethernet0       241,242,243,244,245,246,247,248       Eth1       400000     0       9126    rs
Ethernet8       249,250,251,252,253,254,255,256       Eth2       400000     1       9126    rs
Ethernet16      225,226,227,228,229,230,231,232       Eth3       400000     2       9126    rs
Ethernet24      233,234,235,236,237,238,239,240       Eth4       400000     3       9126    rs
Ethernet32      217,218,219,220,221,222,223,224       Eth5       400000     4       9126    rs
Ethernet40      209,210,211,212,213,214,215,216       Eth6       400000     5       9126    rs
Ethernet48      201,202,203,204,205,206,207,208       Eth7       400000     6       9126    rs
Ethernet56      193,194,195,196,197,198,199,200       Eth8       400000     7       9126    rs
Ethernet64      185,186,187,188,189,190,191,192       Eth9       400000     8       9126    rs
Ethernet72      177,178,179,180,181,182,183,184       Eth10      400000     9       9126    rs
```

```
Ethernet80    169,170,171,172,173,174,175,176    Eth11    400000    10    9126    rs
Ethernet88    161,162,163,164,165,166,167,168    Eth12    400000    11    9126    rs
Ethernet96    153,154,155,156,157,158,159,160    Eth13    400000    12    9126    rs
Ethernet104   145,146,147,148,149,150,151,152    Eth14    400000    13    9126    rs
Ethernet112   137,138,139,140,141,142,143,144    Eth15    400000    14    9126    rs
Ethernet120   129,130,131,132,133,134,135,136    Eth16    400000    15    9126    rs
Ethernet128   121,122,123,124,125,126,127,128    Eth17    400000    16    9126    rs
Ethernet136   113,114,115,116,117,118,119,120    Eth18    400000    17    9126    rs
Ethernet144   105,106,107,108,109,110,111,112    Eth19    400000    18    9126    rs
Ethernet152   97,98,99,100,101,102,103,104       Eth20    400000    19    9126    rs
Ethernet160   89,90,91,92,93,94,95,96           Eth21    400000    20    9126    rs
Ethernet168   81,82,83,84,85,86,87,88           Eth22    400000    21    9126    rs
Ethernet176   73,74,75,76,77,78,79,80           Eth23    400000    22    9126    rs
Ethernet184   65,66,67,68,69,70,71,72           Eth24    400000    23    9126    rs
Ethernet192   57,58,59,60,61,62,63,64           Eth25    400000    24    9126    rs
Ethernet200   49,50,51,52,53,54,55,56           Eth26    400000    25    9126    rs
Ethernet208   41,42,43,44,45,46,47,48           Eth27    400000    26    9126    rs
Ethernet216   33,34,35,36,37,38,39,40           Eth28    400000    27    9126    rs
Ethernet224   25,26,27,28,29,30,31,32           Eth29    400000    28    9126    rs
Ethernet232   17,18,19,20,21,22,23,24           Eth30    400000    29    9126    rs
Ethernet240   9,10,11,12,13,14,15,16            Eth31    400000    30    9126    rs
Ethernet248   1,2,3,4,5,6,7,8                  Eth32    400000    31    9126    rs
Ethernet256   257                              Eth33    10000     32    9126    none
Ethernet257   258                              Eth34    10000     33    9126    none
```

To modify SONiC interface properties, the corresponding devport settings have to be changed and saved, then reboot the switch to apply those settings during switch initialization process.

2.4 System Management

SONiC switches can be administered by configuring or checking following operations.

2.4.1 System clock

Follow the steps below to display the system clock.

Step	Command	Description
Step 1	**show clock**	Displays date & time

The example below shows the command used to display system clock.

```
admin@sonic: ~$ show clock
Wed 21 Jul 2021 11:06:14 PM UTC
admin@sonic: ~$
```

2.4.2 Host Name

SONiC switches can be assigned a name for identification purposes. The default switch name issonic. The switch name is also used as a prompt.

Follow the steps below to configure the Host Name.

Step	Command	Description
Step 1	**config hostname <new_hostname>**	Configure Host Name. New hostname – Host name specified as alphanumeric characters
Step 2	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the command used to configure Host Name.

```
admin@sonic: ~$ sudo config hostname SONiC202106
Running command: service hostname-config restart
```

2.4.3 Display version

Follow the steps below to display the system version.

S. No	Command	Description
1	**show version**	Displays version

Example:

```
admin@sonic: ~$ show version

SONiC Software Version: SONiC.SSE-G3748_3.2.0-0003
Distribution: Debian 10.12
```

```
Kernel: 4.19.0-12-2-arm64
Build commit: 40a4b6649
Build date: Tue Jul 26 16:01:19 UTC 2022
Built by: selva@selva-Standard-PC-Q35-ICH9-2009

Platform: arm64-supermicro_sse_g3748-r0
HwSKU: sse_g3748
ASIC: marvell
ASIC Count: 1
Serial Number: SSG37AN02500016
Model Number: SSE-G3748
Hardware Revision: 2
Uptime: 04:33:37 up  9:39,  1 user,  load average: 2.37, 1.88, 1.91

Docker images:
REPOSITORY            TAG                 IMAGE ID         SIZE
docker-dhcp-relay     latest              a774c6cec02a     563MB
docker-syncd-mrvl     SSE-G3748_3.2.0-0003   d532a0cc0152     715MB
docker-syncd-mrvl     latest              d532a0cc0152     715MB
docker-teamd          SSE-G3748_3.2.0-0003   da29cf176d70     567MB
docker-teamd          latest              da29cf176d70     567MB
docker-nat            SSE-G3748_3.2.0-0003   65a4f316dcf2     569MB
docker-nat            latest              65a4f316dcf2     569MB
docker-platform-monitor  SSE-G3748_3.2.0-0003   98041aa6629b     736MB
docker-platform-monitor  latest              98041aa6629b     736MB
docker-lldp           SSE-G3748_3.2.0-0003   7e3a57bebf21     562MB
docker-lldp           latest              7e3a57bebf21     562MB
docker-database       SSE-G3748_3.2.0-0003   a959f82f26a4     556MB
docker-database       latest              a959f82f26a4     556MB
docker-router-advertiser  SSE-G3748_3.2.0-0003   7993c934ad77     556MB
docker-router-advertiser  latest              7993c934ad77     556MB
docker-orchagent      SSE-G3748_3.2.0-0003   db8160a889d0     666MB
docker-orchagent      latest              db8160a889d0     666MB
docker-snmp           SSE-G3748_3.2.0-0003   36490a24eb14     599MB
docker-snmp           latest              36490a24eb14     599MB
docker-sonic-telemetry  SSE-G3748_3.2.0-0003   d73a517f1aad     640MB
docker-sonic-telemetry  latest              d73a517f1aad     640MB
docker-fpm-frr        SSE-G3748_3.2.0-0003   bfbbc9034cd9     585MB
docker-fpm-frr        latest              bfbbc9034cd9     585MB
docker-sflow          SSE-G3748_3.2.0-0003   154be017ee0f     568MB
docker-sflow          latest              154be017ee0f     568MB
docker-macsec         SSE-G3748_3.2.0-0003   e64b26f32286     569MB
docker-macsec         latest              e64b26f32286     569MB
```

2.4.4 Display environment

Follow the steps below to display the system environment.

S. No	Command	Description
1	**show environment**	Displays Platform environmental

Example:

```
admin@sonic: ~$ show environment

lm75-i2c-0-49
Adapter: mv64xxx_i2c adapter
temp1:       +31.5 C  (high = +80.0 C, hyst = +75.0 C)

7f022004mdiomii00-mdio-0
Adapter: MDIO adapter
temp1:       +34.0 C  (crit = +100.0 C)

lm75-i2c-0-48
Adapter: mv64xxx_i2c adapter
temp1:       +37.0 C  (high = +80.0 C, hyst = +75.0 C)
```

2.4.5 Display reboot-cause

Follow the steps below to display the system reboot-cause.

S. No	Command	Description
1	**show reboot-cause**	Displays Cause of the previous reboot

Example:

```
admin@sonic: ~$ show reboot-cause history
Name                 Cause       Time                              User    Comment
-------------------  -------     --------------------------------  ------  ---------
2021_07_21_18_54_34  Unknown     N/A                                N/A    N/A
2021_07_21_18_54_33  Unknown     N/A                                N/A    N/A
2021_07_21_18_54_32  Unknown     N/A                                N/A    N/A
2021_07_21_18_54_31  reboot      Wed 21 Jul 2021 08:55:53 PM UTC    admin  N/A
admin@sonic: ~$
```

2.4.6 Display uptime

Follow the steps below to display the system uptime.

S. No	Command	Description
1	**show uptime**	Displays System uptime

Example:

```
admin@sonic: ~$ show uptime

up 10 hours, 10 minutes
```

2.4.7 Display logging

Follow the steps below to display the system logging.

S. No	Command	Description
1	**show logging [OPTIONS] [PROCESS]**	Displays Currently stored log message Process – Process name, If wanted specific process logging details Options: -l – shows the lines text -f - follow

Example:

```
admin@sonic: ~$ show logging (Note: Truncated output is added here)

Jul 21 19:00:02.100340 sonic INFO rsyslogd:  [origin software="rsyslogd" swVersion="8.1901.0" x-pid="1552" x-info="https://www.rsyslog.com"] rsyslogd was HUPed
Jul 21 19:00:16.496013 sonic WARNING pmon#thermalctld: fan get_speed speed is 25
Jul 21 19:00:16.497483 sonic WARNING pmon#thermalctld: fan get_target_speed speed is 22
Jul 21 19:00:16.504195 sonic WARNING pmon#thermalctld: fan get_speed speed is 24
Jul 21 19:00:16.505630 sonic WARNING pmon#thermalctld: fan get_target_speed speed is 22
```

```
Jul 21 19:01:00.508204 sonic INFO systemd[1]: run-docker-runtime\x2drunc-moby-a895e40a34721c72b7e3758449752d21341c8232a5d8bca92cedea0eea03d9f8-runc.7PwH7G.mount: Succeeded.
Jul 21 19:03:54.087508 sonic INFO syncd#/supervisord: syncd 19:03:54 SAI: WARNING PORT xpSaiPortCfgManager.c:1696 : Failed to apply admin state for port #22. Err=328
```

2.4.8 Display platform summary

Follow the steps below to display the system platform summary.

S. No	Command	Description
1	**show platform summary**	Displays Summary of the device's hardware platform

Example:

```
admin@sonic: ~$ show platform summary
Platform: arm64-supermicro_sse_g3748-r0
HwSKU: sse_g3748
ASIC: marvell
ASIC Count: 1
Serial Number: SSG37AN02500016
Model Number: SSE-G3748
Hardware Revision: 2
```

2.4.9 Display system EEPROM

Follow the steps below to display the system EEPROM.

S. No	Command	Description
1	**show platform syseeprom**	Displays Information stored on the system EEPROM

Example:

```
admin@sonic: ~$ show platform syseeprom
TlvInfo Header:
  Id String:    TlvInfo
  Version:      1
```

```
   Total Length: 192
TLV Name          Code       Len          Value
------------------    ------       -----        ----------------------------
Product Name       0x21      9            SSE-G3748
Part Number        0x22      9            SSE-G3748
Serial Number      0x23      15           SSG37AN02500016
Base MAC Address   0x24      6            0C:C4:7A:2E:16:35
Manufacture Date   0x25      19           06/06/2022 12:00:00
Device Version     0x26      12
Label Revision     0x27      11
Platform Name      0x28      29       arm64-supermicro_sse_g3748-r0
ONIE Version       0x29      24           2022.01.00.01_supermicro
MAC Addresses      0x2A      256
Manufacturer       0x2B      10           supermicro
Manufacture Country  0x2C    2        US
Vendor Name        0x2D      10       supermicro
Vendor Extension   0xFD      21
CRC-32             0xFE      4            0x882AC81B

(checksum valid)
```

2.4.10 Display power supply units

Follow the steps below to display the system power supply units.

S. No	Command	Description
1	**show platform psustatus**	Displays Status of the device's power supply units

Example:

```
admin@sonic: ~$ show platform psustatus
PSU     Model    Serial              HW Rev     Voltage (V)    Current (A)    Power (W)   Status    LED
-----    -------    --------------      --------    -------------    -------------    -----------    --------    -----
PSU 1    NA       K370150H5D0032    N/A          0.00           0.00           0           NOT OK    off
PSU 2    NA       K370150H5D0024    N/A          12.00          12.50          150         OK        green
admin@sonic: ~$
```

2.4.11 Display device's fans

Follow the steps below to display the system platform fan.

S. No	Command	Description
1	**show platform fan**	Displays Status of the device's fans

Example:

```
admin@sonic: ~$ show platform fan
 Drawer     LED    FAN    Speed   Direction  Presence    Status      Timestamp
 --------   -----  -----  -------  ----------  ----------  --------  -----------------
 drawer1    green  Fan1    25%     exhaust    Present      OK       20210722 04:49:16
 drawer1    green  Fan2    24%     exhaust    Present      OK       20210722 04:49:16
```

2.4.12 Display device's thermal sensors

Follow the steps below to display the system thermal sensors.

S. No	Command	Description
1	**show platform temperature**	Displays Status of the device's thermal sensors

Example:

```
admin@sonic: ~$ show platform temperature
 Sensor   Temperature   High TH   Low TH   Crit High TH   Crit Low TH   Warning   Timestamp
--------  -------------  ---------  --------  --------------  -------------  ---------  -----------------
  FRONT       37          80        N/A        N/A            N/A          False     20210722 04:51:16
  REAR        31.5        80        N/A        N/A            N/A          False     20210722 04:51:16
```

2.4.13 System State

2.4.13.1 Display CPU usage

Follow the steps below to display the system cpu usage.

Step	Command	Description

Step 1	**show processes cpu**	Displays Current CPU usage by process

The example below shows the command used to display current CPU usage by process.

```
admin@sonic: ~$ show processes cpu (Note: Truncated output is added here)
top - 04:55:06 up 10:01,  1 user,  load average: 2.12, 2.18, 2.13
Tasks: 195 total,   3 running, 187 sleeping,   0 stopped,   5 zombie
%Cpu(s): 52.6 us, 15.8 sy,  0.0 ni, 31.6 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   4014.3 total,   1963.1 free,   1184.9 used,    866.3 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.   2738.5 avail Mem

  PID     USER     PR  NI   VIRT     RES     SHR     S  %CPU  %MEM    TIME+    COMMAND
 3770     root     20   0   128268   40680   12280   R  88.2   1.0    91:36.40  python3
 2931     root     20   0   110368   19736   5624    R  11.8   0.5    67:10.02  python3
168235    admin    20   0   10264    3136    2768    R  11.8   0.1    0:00.05   top
168201    root     20   0    0         0       0      Z  0.0    0.0    0:00.27   python3
168215    admin    20   0   52308    37596   13084   S  0.0    0.9    0:01.81   show
```

2.4.13.2 Display Memory usage

Follow the steps below to display the system memory usage.

Step	Command	Description
Step 1	**show processes memory**	Displays Current memory usage by process

The example below shows the command used to display current memory usage by process.

```
admin@Sonic: ~$  show processes memory (Note: Truncated output is added here)
top - 04:56:31 up 10:02,  1 user,  load average: 2.49, 2.25, 2.16
Tasks: 192 total,   3 running, 187 sleeping,   0 stopped,   2 zombie
%Cpu(s): 59.5 us, 10.8 sy,  0.0 ni, 29.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   4014.3 total,   1962.9 free,   1184.8 used,    866.6 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.   2738.4 avail Mem

  PID     USER     PR  NI   VIRT      RES     SHR     S  %CPU  %MEM   TIME+    COMMAND
  567     root     20   0   1207656   85480   33084   S  0.0    2.1    4:18.27   dockerd
 2338     root     20   0   2221944   77580   32600   S  11.8   1.9    47:01.92  syncd
168325    root     20   0    0         0       0      Z  0.0    0.0    0:01.82   python3
168326    root     20   0    0         0       0      Z  0.0    0.0    0:00.28   python3
```

2.4.13.3 Display Summary usage

Follow the steps below to display the system summary usage.

Step	Command	Description
Step 1	**show processes summary**	Displays Current summary usage by process

The example below shows the command used to display current summary usage by process.

```
admin@Sonic: ~$ show processes summary (Note: Truncated output is added here)
  PID     PPID    CMD                              %MEM    %CPU
    1        0    /sbin/init                        0.2     0.2
    2        0    [kthreadd]                        0.0     0.0
    3        2    [rcu_gp]                          0.0     0.0
168629   489      [python3] <defunct>               0.0     0.4
168970   161906   /usr/bin/python3 /usr/local       0.9     86.5
168989   168970    /bin/sh -c ps -eo pid,ppid,      0.0     0.0
168990   168989    ps -eo pid,ppid,cmd,%mem,%c      0.0     0.0
```

2.4.14Troubleshooting

SONiC has the troubleshooting options. For troubleshooting and debugging purposes, *show techsupport* command gathers pertinent information about the state of the device; information is as diverse as syslog entries, database state, routing-stack state, etc., It then compresses it into an archive file. This archive file can be used for examination. Resulting archive file is saved as /var/dump/<DEVICE_HOST_NAME>_YYYYMMDD_HHMMSS.tar.gz

If the SONiC system was running for quite some time show techsupport will produce a large dump file. To reduce the amount of syslog and core files gathered during system dump use --since option:

Step	Command	Description
Step 1	**show techsupport**	Displays tech support options

The example below shows the command used to show the tech support options.

```
admin@sonic: ~$ show techsupport –since yesterday  # Will collect syslog and core files for the last 24 hours

main
mkdir: created directory '/var/dump/sonic_dump_SONiC202106_20210722_001118'
```

'/var/dump/sonic_dump_SONiC202106_20210722_001118/generate_dump' ->
'/usr/local/bin/generate_dump'
sonic_dump_SONiC202106_20210722_001118/
sonic_dump_SONiC202106_20210722_001118/generate_dump
mkdir: created directory '/var/dump/sonic_dump_SONiC202106_20210722_001118/proc'
'/proc/buddyinfo' -> '/var/dump/sonic_dump_SONiC202106_20210722_001118/proc/buddyinfo'
'/proc/cmdline' -> '/var/dump/sonic_dump_SONiC202106_20210722_001118/proc/cmdline'
'/proc/consoles' -> '/var/dump/sonic_dump_SONiC202106_20210722_001118/proc/consoles'
'/proc/cpuinfo' -> '/var/dump/sonic_dump_SONiC202106_20210722_001118/proc/cpuinfo'
'/proc/devices' -> '/var/dump/sonic_dump_SONiC202106_20210722_001118/proc/devices'
'/proc/diskstats' -> '/var/dump/sonic_dump_SONiC202106_20210722_001118/proc/diskstats'
'/proc/interrupts' -> '/var/dump/sonic_dump_SONiC202106_20210722_001118/proc/interrupts'
'/proc/iomem' -> '/var/dump/sonic_dump_SONiC202106_20210722_001118/proc/iomem'
'/proc/ioports' -> '/var/dump/sonic_dump_SONiC202106_20210722_001118/proc/ioports'
^Z
[2]+ Stopped show techsupport --since=yesterday
admin@sonic: ~$ show techsupport --since='hour ago' # Will collect syslog and core files for the last one hour

main
mkdir: created directory '/var/dump/sonic_dump_SONiC202106_20210722_001208'
'/var/dump/sonic_dump_SONiC202106_20210722_001208/generate_dump' ->
'/usr/local/bin/generate_dump'
sonic_dump_SONiC202106_20210722_001208/
sonic_dump_SONiC202106_20210722_001208/generate_dump
mkdir: created directory '/var/dump/sonic_dump_SONiC202106_20210722_001208/proc'
'/proc/buddyinfo' -> '/var/dump/sonic_dump_SONiC202106_20210722_001208/proc/buddyinfo'
'/proc/cmdline' -> '/var/dump/sonic_dump_SONiC202106_20210722_001208/proc/cmdline'
'/proc/consoles' -> '/var/dump/sonic_dump_SONiC202106_20210722_001208/proc/consoles'
'/proc/cpuinfo' -> '/var/dump/sonic_dump_SONiC202106_20210722_001208/proc/cpuinfo'
'/proc/devices' -> '/var/dump/sonic_dump_SONiC202106_20210722_001208/proc/devices'
'/proc/diskstats' -> '/var/dump/sonic_dump_SONiC202106_20210722_001208/proc/diskstats'
'/proc/interrupts' -> '/var/dump/sonic_dump_SONiC202106_20210722_001208/proc/interrupts'
'/proc/iomem' -> '/var/dump/sonic_dump_SONiC202106_20210722_001208/proc/iomem'
'/proc/ioports' -> '/var/dump/sonic_dump_SONiC202106_20210722_001208/proc/ioports'
'/proc/kallsyms' -> '/var/dump/sonic_dump_SONiC202106_20210722_001208/proc/kallsyms'
'/proc/loadavg' -> '/var/dump/sonic_dump_SONiC202106_20210722_001208/proc/loadavg'
[3]+ Stopped show techsupport --since='hour ago'
admin@SONiC202106:~$
admin@sonic: ~$ show techsupport
main

```
mkdir: created directory '/var/dump/sonic_dump_SONiC202106_20210722_050003'
'/var/dump/sonic_dump_SONiC202106_20210722_050003/generate_dump' ->
'/usr/local/bin/generate_dump'
sonic_dump_SONiC202106_20210722_050003/
sonic_dump_SONiC202106_20210722_050003/generate_dump
mkdir: created directory '/var/dump/sonic_dump_SONiC202106_20210722_050003/proc'
'/proc/buddyinfo' -> '/var/dump/sonic_dump_SONiC202106_20210722_050003/proc/buddyinfo'
'/proc/cmdline' -> '/var/dump/sonic_dump_SONiC202106_20210722_050003/proc/cmdline'
'/proc/consoles' -> '/var/dump/sonic_dump_SONiC202106_20210722_050003/proc/consoles'
'/proc/cpuinfo' -> '/var/dump/sonic_dump_SONiC202106_20210722_050003/proc/cpuinfo'
'/proc/devices' -> '/var/dump/sonic_dump_SONiC202106_20210722_050003/proc/devices'
'/proc/diskstats' -> '/var/dump/sonic_dump_SONiC202106_20210722_050003/proc/diskstats'
'/proc/interrupts' -> '/var/dump/sonic_dump_SONiC202106_20210722_050003/proc/interrupts'
'/proc/iomem' -> '/var/dump/sonic_dump_SONiC202106_20210722_050003/proc/iomem'
'/proc/ioports' -> '/var/dump/sonic_dump_SONiC202106_20210722_050003/proc/ioports'
'/proc/kallsyms' -> '/var/dump/sonic_dump_SONiC202106_20210722_050003/proc/kallsyms'
'/proc/loadavg' -> '/var/dump/sonic_dump_SONiC202106_20210722_050003/proc/loadavg'
'/proc/locks' -> '/var/dump/sonic_dump_SONiC202106_20210722_050003/proc/locks'
'/proc/meminfo' -> '/var/dump/sonic_dump_SONiC202106_20210722_050003/proc/meminfo'
'/proc/misc' -> '/var/dump/sonic_dump_SONiC202106_20210722_050003/proc/misc'
'/proc/modules' -> '/var/dump/sonic_dump_SONiC202106_20210722_050003/proc/modules'
'/proc/self/mounts' -> '/var/dump/sonic_dump_SONiC202106_20210722_050003/proc/mounts'
'/proc/self/net' -> '/var/dump/sonic_dump_SONiC202106_20210722_050003/proc/net'
'/proc/self/net/stat' -> '/var/dump/sonic_dump_SONiC202106_20210722_050003/proc/net/stat'
'/proc/self/net/stat/arp_cache' ->
'/var/dump/sonic_dump_SONiC202106_20210722_050003/proc/net/stat/arp_cache'
'/proc/self/net/stat/rt_cache' ->
'/var/dump/sonic_dump_SONiC202106_20210722_050003/proc/net/stat/rt_cache'
```

2.4.15 Display Services

S. No	Command	Description
1	**show services**	Displays Status of the system services

Example:

```
admin@sonic: ~$ show services (Note: Truncated output is added here)
snmp    docker
----------------------------
USER         PID    %CPU    %MEM     VSZ       RSS     TTY     STAT    START    TIME  COMMAND
```

```
root          1    0.0    0.5   30976   22928   pts/0   Ss+   18:57   0:17   /usr/bin/python3
/usr/local/ bin/supervisord
root          10   0.0    0.4   26288   18268   pts/0   S     18:57   0:03   python3/usr/bin
/supervisor-proc-exit-listener --container-name snmp
root          17   0.0    0.1   219868  5172    pts/0   Sl    18:57   0:00   /usr/sbin
/rsyslo gd -n -iNONE
database      docker
---------------------------
USER      PID   %CPU   %MEM   VSZ     RSS     TTY     STAT   START   TIME   COMMAND
root      1     0.0    0.5    27584   22640   pts/0   Ss+    18:54   0:15   /usr/bin/python3
/usr/local /bin/supervisord
root      48    0.0    0.4    22764   17856   pts/0   S      18:54   0:03   python3
/usr/bin/supervisor -proc-exit-listener --container-name database
root      49    0.0    0.0    219868  3152    pts/0   Sl   18:54   0:00   /usr/sbin/rsyslogd -n -
iNONE
root      50    16.3   0.8    87708   33260   pts/0   Sl   18:54   49:11    /usr/bin/redisserver127
.0.0.1:6379
admin@sonic: ~$
```

2.4.16 Display System-health

2.4.16.1 Display system-health detail

Step	Command	Description
Step 1	**sudo show system-health detail**	Displays Current system-health detail

The example below shows the command used to display current system-health detail.

```
admin@sonic: ~$ sudo show system-health detail (Note: Truncated output is added here)
System status summary

  System status LED  amber
  Services:
    Status: OK
  Hardware:
    Status: Not OK
    Reasons: Invalid voltage data for PSU 2, voltage=12.0, range=[N/A,N/A]
          PSU 1 is out of power
          routeCheck is not Status ok

System services and devices monitor list
```

```
Name                        Status      Type
--------------------------  --------    ----------
routeCheck                  Not OK      Program
sonic                       OK          System
rsyslog                     OK          Process
root-overlay                OK          Filesystem
var-log                     OK          Filesystem
diskCheck                   OK          Program
container_checker           OK          Program
vnetRouteCheck              OK          Program
container_memory_telemetry  OK          Program
snmp:snmpd                  OK          Process
snmp:snmp-subagent          OK          Process
telemetry:telemetry         OK          Process
telemetry:dialout           OK          Process
lldp:lldpd                  OK          Process
lldp:lldp-syncd             OK          Process
lldp:lldpmgrd               OK          Process
syncd:syncd                 OK          Process
teamd:teammgrd              OK          Process
teamd:teamsyncd             OK          Process
teamd:tlm_teamd             OK          Process
swss:orchagent              OK          Process
swss:portsyncd              OK          Process
swss:neighsyncd             OK          Process
swss:fdbsyncd               OK          Process
swss:vlanmgrd               OK          Process
swss:intfmgrd               OK          Process
swss:portmgrd               OK          Process
swss:buffermgrd             OK          Process
swss:vrfmgrd                OK          Process
swss:nbrmgrd                OK          Process
swss:vxlanmgrd              OK          Process
swss:coppmgrd               OK          Process
swss:tunnelmgrd             OK          Process
bgp:zebra                   OK          Process
bgp:staticd                 OK          Process
bgp:bgpd                    OK          Process
bgp:fpmsyncd                OK          Process
bgp:bgpcfgd                 OK          Process
database:redis              OK          Process
PSU 1                       Not OK      PSU
PSU 2                       Not OK      PSU
Fan1                        OK          Fan
Fan2                        OK          Fan

System services and devices ignore list
```

```
Name                 Status     Type
---------------      --------   ------
psu.temperature      Ignored    Device
asic                 Ignored    Device
admin@sonic: ~$
```

2.4.16.2 Display system-health monitor-list

Step	Command	Description
Step 1	**show system-health monitor-list**	Displays Current system-health monitor-list

The example below shows the command used to display current system-health monitor-list.

```
admin@sonic: ~$ sudo show system-health monitor-list (Note: Truncated output is added here)

System services and devices monitor list

Name                        Status          Type
-------------------------   --------        ----------
routeCheck                  Not OK          Program
sonic                       OK              System
rsyslog                     OK              Process
root-overlay                OK              Filesystem
var-log                     OK              Filesystem
diskCheck                   OK              Program
container_checker           OK              Program
vnetRouteCheck              OK              Program
container_memory_telemetry  OK              Program
snmp:snmpd                  OK              Process
snmp:snmp-subagent          OK              Process
telemetry:telemetry         OK              Process
telemetry:dialout           OK              Process
lldp:lldpd                  OK              Process
lldp:lldp-syncd             OK              Process
lldp:lldpmgrd               OK              Process
syncd:syncd                 OK              Process
teamd:teammgrd              OK              Process
teamd:teamsyncd             OK              Process
teamd:tlm_teamd             OK              Process
swss:orchagent              OK              Process
swss:portsyncd              OK              Process
swss:neighsyncd             OK              Process
```

```
swss:fdbsyncd                OK              Process
swss:vlanmgrd                OK              Process
swss:intfmgrd                OK              Process
swss:portmgrd                OK              Process
swss:buffermgrd              OK              Process
swss:vrfmgrd                 OK              Process
swss:nbrmgrd                 OK              Process
swss:vxlanmgrd               OK              Process
swss:coppmgrd                OK              Process
swss:tunnelmgrd              OK              Process
bgp:zebra                    OK              Process
bgp:staticd                  OK              Process
bgp:bgpd                     OK              Process
bgp:fpmsyncd                 OK              Process
bgp:bgpcfgd                  OK              Process
database:redis               OK              Process
PSU 1                        Not OK          PSU
PSU 2                        Not OK          PSU
Fan1                         OK              Fan
Fan2                         OK              Fan
admin@sonic: ~$
```

2.4.16.3 Display system-health summary

Step	Command	Description
Step 1	**show system-health summary**	Displays Current system-health summary

The example below shows the command used to display current system-health summary.

```
admin@sonic: ~$ sudo show system-health summary
System status summary

  System status LED  amber
  Services:
    Status: OK
  Hardware:
    Status: Not OK
    Reasons: Invalid voltage data for PSU 2, voltage=12.0, range=[N/A,N/A]
          PSU 1 is out of power
          routeCheck is not Status ok
admin@sonic: ~$
```

2.4.17 Display System-memory

S. No	Command	Description
1	**Show system-memory**	Displays Status of the system memory

Example:

```
admin@sonic: ~$ show system-memory
               total        used        free      shared  buff/cache   available
Mem:            4014        1161        1993          26         859        2749
Swap:              0           0           0
admin@sonic: ~$
```

2.5 Security Features

SONiC switches support two methods of user authentication: Local and remote. The remote authentication is supported using RADIUS and TACACS.

- RADIUS – Remote Authentication Dial-In User Service (RADIUS) uses AAA service for ID verification, granting access and tracking the actions of remote users.
- TACACS – *Terminal Access Controller Access Control System (TACACS)* provides accounting information and administrative control for authentication. RADIUS encrypts only passwords, whereas TACACS encrypts usernames as well, making it more secure.

2.5.1 AAA

2.5.1.1 Defaults

Parameter	Default Value
AAA login	Local
AAA failthrough	False
AAA fallback	N/A

2.5.1.2 Configure AAA authentication login

Follow the steps below to configure AAA authentication login.

Step	Command	Description
Step 1	**config aaa authentication login (tacacs+ | local | default)**	Configure AAA authentication login.
Step 2	**show aaa**	Displays the AAA configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure AAA authentication login.

```
admin@sonic:~$ sudo config aaa authentication login tacacs+
admin@sonic:~$ show aaa
AAA authentication login tacacs+
AAA authentication failthrough False (default)
AAA authorization login local (default)
AAA accounting login disable (default)
```

2.5.1.3 Configure AAA authentication failthrough

Follow the steps below to configure AAA authentication failthrough.

Step	Command	Description
Step 1	**config aaa authentication failthrough (enable | disable | default)**	Configure AAA authentication failthrough.
Step 2	**show aaa**	AAA configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure AAA authentication failthrough.

```
admin@sonic:~$ sudo config aaa authentication failthrough disable
admin@sonic:~$ show aaa
AAA authentication login tacacs+
AAA authentication failthrough False
AAA authorization login local (default)
AAA accounting login disable (default)
```

2.5.1.4 Configure AAA authentication fallback

Follow the steps below to configure AAA authentication fallback.

Step	Command	Description
Step 1	**config aaa authentication fallback (enable | disable | default)**	Configure AAA authentication fallback.
Step 2	**show aaa**	AAA configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure AAA authentication fallback.

```
admin@sonic:~$ sudo config aaa authentication fallback disable
admin@sonic:~$ show aaa
AAA authentication login tacacs+
AAA authentication failthrough False
AAA authentication fallback False
AAA authorization login local (default)
AAA accounting login disable (default)
```

2.5.2 RADIUS

A sequence of events occurs during RADIUS client-server communication at the time of user login.

- The username and password are encrypted by the client and sent to RADIUS server.
- The client receives a response from the RADIUS server:
  - ACCEPT—User authentication is successful.
  - REJECT—User authentication failed. User is prompted to re-enter username/password, or access is denied.
  - CHALLENGE—Additional data is requested from the user.
  - CHALLENGE PASSWORD—User is prompted to select a new password.

Along with ACCEPT or REJECT packets, service options (Telnet, SSH, rlogin, or privileged EXEC services) and connection parameters like user timeouts are sent by RADIUS server.

This section explains the Radius commands which are supported in SONiC switches

2.5.2.1 Defaults

Parameter	Default Value
RADIUS global auth_type	Pap
RADIUS global retransmit	3
RADIUS global timeout	5
RADIUS global passkey	<EMPTY_STRING>

2.5.2.2 Configure RADIUS Server

Sonic switches function as a RADIUS client. The RADIUS server that is to be contacted for authentication can be configured in the switch.

Follow the below steps to configure RADIUS server.

Step	Command	Description
Step 1	**config radius add [OPTIONS] <ip_address_or_domain_name>**	Configure the RADIUS server. Ip address – A valid IPv4 Address. Domain name – A valid domain name Options: -r, --retransmit - INTEGER RANGE Retransmit attempts, default 3 -t, --timeout - INTEGER RANGE Transmission timeout interval, default 5 -k, --key TEXT - Shared secret -a, --auth_type [chap|pap|mschapv2] - Authentication type, default pap -o, --auth-port - INTEGER RANGE UDP port range is 1 to 65535, default 1812

		-p, --pri INTEGER RANGE - Priority, default 1 -m, --use-mgmt-vrf - Management vrf, default is no vrf -s, --source-interface TEXT - Source Interface
Step 2	**show radius**	Displays the RADIUS configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure RADIUS server.

```
admin@sonic:~$ sudo config radius add 192.168.100.22
admin@sonic:~$ show radius
RADIUS global auth_type pap (default)
RADIUS global retransmit 3 (default)
RADIUS global timeout 5 (default)
RADIUS global passkey <EMPTY_STRING> (default)
RADIUS_SERVER address 192.168.100.22
            auth_port 1812
            priority 1
admin@sonic:~$
```

2.5.2.3 Configure RADIUS Server authtype

Follow the below steps to configure RADIUS server authtype parameters.

Step	Command	Description
Step 1	**config radius authtype [chap | pap | mschapv2]**	Configure the RADIUS server authtype. Chap – Configure chap authtype. pap – Configure pap authtype. mschapv2 – Configure mschapv2 authtype.
Step 2	**show radius**	Displays the RADIUS configuration.

Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure RADIUS server authtype.

```
admin@sonic:~$ sudo config radius authtype chap
admin@sonic:~$ show radius
RADIUS global auth_type chap
RADIUS global retransmit 3 (default)
RADIUS global timeout 5 (default)
RADIUS global passkey <EMPTY_STRING> (default)
RADIUS_SERVER address 192.168.100.22
        auth_port 1812
        priority 1
admin@sonic:~$
admin@sonic:~$ sudo config radius authtype pap
admin@sonic:~$ show radius
RADIUS global auth_type pap
RADIUS global retransmit 3 (default)
RADIUS global timeout 5 (default)
RADIUS global passkey <EMPTY_STRING> (default)
RADIUS_SERVER address 192.168.100.22
        auth_port 1812
        priority 1
admin@sonic:~$ sudo config radius authtype mschapv2
admin@sonic:~$ show radius
RADIUS global auth_type mschapv2
RADIUS global retransmit 3 (default)
RADIUS global timeout 5 (default)
RADIUS global passkey <EMPTY_STRING> (default)
RADIUS_SERVER address 192.168.100.22
        auth_port 1812
        priority 1
admin@sonic:~$
```

2.5.2.4 Configure RADIUS Server default parameters

Follow the below steps to configure RADIUS server default parameters.

Step	Command	Description

Step 1	**config radius default [OPTIONS]**	Configure the RADIUS server default parameters. Options: Authtype – Configure default authtype. Nasip – Configure default nas IP. Passkey – Configure default passkey. Retransmit – Configure default retransmit. Sourceip – Configure default source IP. Timeout – Configure default timeout.
Step 2	**show radius**	Displays the RADIUS configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure RADIUS server default parameters.

```
admin@sonic:~$ sudo config radius default authtype pap
admin@sonic:~$ show radius
RADIUS global auth_type pap (default)
RADIUS global retransmit 3 (default)
RADIUS global timeout 5 (default)
RADIUS global passkey <EMPTY_STRING> (default)
RADIUS_SERVER address 192.168.100.22
        auth_port 1812
        priority 1
admin@sonic:~$
```

2.5.2.5 Remove RADIUS Server

Follow the below steps to remove RADIUS server.

Step	Command	Description
Step 1	**config radius delete <ip_address_or_domain_name>**	Remove the RADIUS server.

		Ip address – A valid Ipv4 Address. Domain name – A valid domain name
Step 2	**show radius**	Displays the RADIUS configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to remove RADIUS server.

```
admin@sonic:~$ sudo config radius delete 192.168.100.22
admin@sonic:~$ show radius
RADIUS global auth_type pap (default)
RADIUS global retransmit 3 (default)
RADIUS global timeout 5 (default)
RADIUS global passkey <EMPTY_STRING> (default)
admin@sonic:~$
```

2.5.2.6 Configure RADIUS passkey

Follow the below steps to configure RADIUS server parameters.

Step	Command	Description
Step 1	**config radius passkey <secret_string>**	Configure the RADIUS passkey. Secret string – Secret string can be specified as alphanumeric characters.
Step 2	**show radius**	Displays the RADIUS configuration.
Step 3	**sudo config save –y**	Optional step – saves this configuration to be part of startup configuration.

The example below shows the commands used to configure RADIUS passkey.

```
admin@sonic:~$ sudo config radius passkey key1
admin@sonic:~$ show radius
RADIUS global auth_type pap (default)
RADIUS global retransmit 5
RADIUS global timeout 50
```

```
RADIUS global passkey key1
RADIUS global statistics True
RADIUS_SERVER address 192.168.100.11
        auth_port 1812
        priority 1
admin@sonic:~$
```

2.5.2.7 Configure RADIUS retransmit

Follow the below steps to configure RADIUS retransmit parameters.

Step	Command	Description
Step 1	**config radius retransmit <retry_attempts>**	Configure the RADIUS retransmit. Retry attempts – Retry attempt can be specified in the range of < 0 – 10 >.
Step 2	**show radius**	Displays the RADIUS configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure RADIUS retransmit.

```
admin@sonic:~$ sudo config radius retransmit 8
admin@sonic:~$ show radius
RADIUS global auth_type pap (default)
RADIUS global retransmit 8
RADIUS global timeout 50
RADIUS global passkey key1
RADIUS global statistics True
RADIUS_SERVER address 192.168.100.11
        auth_port 1812
        priority 1
admin@sonic:~$
```

2.5.2.8 Configure RADIUS statistics

Follow the below steps to configure RADIUS statistics parameters.

Step	Command	Description

Step 1	**config radius statistics [enable| disable | default]**	Configure the RADIUS statistics. Enable – enables radius statistics. Disable – disables radius statistics. default – default value of radius statistics.
Step 2	**show radius**	Displays the RADIUS configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure RADIUS statistics.

```
admin@sonic:~$ sudo config radius statistics enable
admin@sonic:~$ show radius
RADIUS global auth_type pap (default)
RADIUS global retransmit 8
RADIUS global timeout 50
RADIUS global passkey key1
RADIUS global statistics True
RADIUS_SERVER address 192.168.100.11
        auth_port 1812
        priority 1
admin@sonic:~$
admin@sonic:~$ sudo config radius statistics disable
admin@sonic:~$ show radius
RADIUS global auth_type pap (default)
RADIUS global retransmit 8
RADIUS global timeout 50
RADIUS global passkey key1
RADIUS global statistics False
RADIUS_SERVER address 192.168.100.11
        auth_port 1812
        priority 1
admin@sonic:~$ sudo config radius statistics default
admin@sonic:~$ show radius
RADIUS global auth_type pap (default)
RADIUS global retransmit 8
RADIUS global timeout 50
```

```
RADIUS global passkey key1

RADIUS_SERVER address 192.168.100.11
        auth_port 1812
        priority 1
admin@sonic:~$
```

2.5.2.9 Configure RADIUS timeout

Follow the below steps to configure RADIUS timeout parameters.

Step	Command	Description
Step 1	**config radius timeout <time_second>**	Configure the RADIUS timeout. Time second – Time seconds can be specified in the range of < 1 – 60 >.
Step 2	**show radius**	Displays the RADIUS configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure RADIUS timeout.

```
admin@sonic:~$ sudo config radius timeout 33
admin@sonic:~$ show radius
RADIUS global auth_type pap (default)
RADIUS global retransmit 8
RADIUS global timeout 33
RADIUS global passkey key1

RADIUS_SERVER address 192.168.100.11
        auth_port 1812
        priority 1

admin@sonic:~$
```

2.5.2.10 RADIUS Configuration Example

This section explains the configurations of RADIUS.

Step	Command	Description

Step 1	**useradd [options] LOGIN** **useradd -D [options]**	Useradd - Add user -D – Default (print or change default useradd configuration) LOGIN - new value of the login name NOTE: Refer Linux manual for options related to create user.
Step 2	**config radius add [OPTIONS] <ip_address_or_domain_name>**	Configure the RADIUS server. Ip address – A valid Ipv4 Address. Domain name – A valid domain name Options: -r, --retransmit - INTEGER RANGE Retransmit attempts, default 3 -t, --timeout - INTEGER RANGE Transmission timeout interval, default 5 -k, --key TEXT - Shared secret -a, --auth_type [chap|pap|mschapv2] - Authentication type, default pap -o, --auth-port - INTEGER RANGE UDP port range is 1 to 65535, default 1812 -p, --pri INTEGER RANGE - Priority, default 1 -m,--use-mgmt-vrf - Management vrf, default is no vrf -s, --source-interface TEXT - Source Interface
Step 3	**config radius passkey <secret_string>**	Configure the RADIUS passkey.

		Secret string – Secret string can be specified as alphanumeric characters.
Step 4	**config aaa authentication login (tacacs+ | local | default)**	Configure AAA authentication login.
Step 5	**config aaa authentication failthrough (enable | disable | default)**	Configure AAA authentication failthrough.
Step 6	**show radius**	Displays the RADIUS configuration.
Step 7	**show aaa**	AAA configuration.

The following example shows commands used to configure RADIUS.

```
admin@sonic:~$ sudo useradd -m -u 8787 -g admin -s /bin/bash SWtestradius
admin@sonic:~$ sudo config radius add 192.168.86.35
admin@sonic:~$ sudo config radius passkey testing123
admin@sonic:~$ sudo config aaa authentication login local radius
admin@sonic:~$ sudo config aaa authentication failthrough enable
admin@sonic:~$ show radius
RADIUS global auth_type pap (default)
RADIUS global retransmit 3 (default)
RADIUS global timeout 5 (default)
RADIUS global passkey testing123
RADIUS_SERVER address 192.168.86.35
        auth_port 1812
        priority 1
admin@sonic:~$ show aaa
AAA authentication login local,radius
AAA authentication failthrough True
AAA authorization login local (default)
AAA accounting login disable (default)
admin@sonic:~$
```

2.5.2.11 RADIUS Server Configuration

The RADIUS server has to be configured properly for the authentication to work. Below is the reference configuration for freeradius server running in ubuntu linux. The configuration may vary for different RADIUS servers.

Configure the switch details in the file /etc/freeradius/clients.conf in the RADIUS server. The IP address is the switch's IP address, which will be used in the communication with RADIUS server. The 'secret' is the passkey configured in the switch (refer to section Configure RADIUS passkey).

```
client 172.10.10.10/32 {
    secret        = key1
    shortname     = SupermicroSwitch
}
```

Create or add the users to the file /etc/freeradius/users in the RADIUS server. Each user has to be added to the file. Below is the configuration to create a user named "TestUser" with password "radius".

```
TestUser        Cleartext-Password := "radius"
            Management-Privilege-Level = 15
```

TEST:

login as: TestUser
TestUser@192.168.86.28's password:radius
Linux sonic 4.19.0-12-2-arm64 #1 SMP Debian 4.19.152-1 (2020-10-18) aarch64
You are on

```
  ____   ___  _   _ _  ____
 / ___| / _ \| \ | (_)/ ___|
 \___ \| | | |  \| | | |
  ___) | |_| | |\  | | |___
 |____/ \___/|_| \_|_|\____|
```

-- Software for Open Networking in the Cloud --

Unauthorized access and/or use are prohibited.
All access and/or use are subject to monitoring.

Help: http://azure.github.io/SONiC/

Last login: Wed Jul 21 21:47:17 2021
TestUser@sonic:~$

2.5.3 TACACS

TACACS provides access control to switch through a client-server model, similar to RADIUS except that it provides enhanced security by encryption of all messages and reliability via TCP.

2.5.3.1 Defaults

Parameter	Default Value
TACACS auth_type	Pap
TACACS timeout	5
TACACS passkey	<EMPTY_STRING>
TACPLUS_SERVER priority 1	1
TACPLUS_SERVER TCP port	49

2.5.3.2 Configure TACACS Server

Follow the steps below to configure TACACS server.

Step	Command	Description
Step 1	**config tacacs add <ip_address> [-t|--timeout <seconds>] [-k|--key <secret>] [-a|--type <type>] [-o|--port <port>] [-p|--pri <priority>] [-m|--use-mgmt-vrf]**	Configure TACACS server to be used. Ip address: TACACS server's IPv4/IPv6 address. timeout: Transmission timeout interval in seconds, range 1 to 60, default 5 key: Shared secret type: Authentication type, "chap" or "pap" or "mschap" or "login", default is "pap". port: TCP port range is 1 to 65535, default 49 pri: Priority, priority range 1 to 64, default 1.

		use-mgmt-vrf: This means that the server is part of Management vrf, default is "no vrf" Options: -t, --timeout - INTEGER Transmission timeout interval, default 5 -k, --key TEXT - Shared secret -a, --auth_type [chap |pap |mschap |login] - Authentication type, default pap -o, --port INTEGER RANGE - TCP port range is 1 to 65535, default 49 -p, --pri INTEGER RANGE - Priority, default 1 -m, --use-mgmt-vrf - Management vrf, default is no vrf
Step 2	**show tacacs**	Displays the TACACS configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure TACACS server with IPv4 address.

```
admin@sonic:~$ sudo config tacacs add 192.168.100.34 -t 10 -k testing789 -a mschap -o 50 -p 9
admin@sonic:~$ show tacacs
TACPLUS global auth_type pap (default)
TACPLUS global timeout 5 (default)
TACPLUS global passkey <EMPTY_STRING> (default)
TACPLUS_SERVER address 192.168.100.34
        auth_type mschap
        passkey testing789
        priority 9
        tcp_port 50
        timeout 10
admin@sonic:~$
```

The example below shows the commands used to configure TACACS server with IPv6 address.

```
admin@sonic:~$ sudo config tacacs add 2002::2222 -t 10 -k testing789 -a mschap -o 50 -p 9
admin@sonic:~$ show tacacs
TACPLUS global auth_type pap (default)
TACPLUS global timeout 5 (default)
TACPLUS global passkey <EMPTY_STRING> (default)

TACPLUS_SERVER address 2002::2222
        auth_type mschap
        passkey testing789
        priority 9
        tcp_port 50
        timeout 10
admin@sonic:~$
```

2.5.3.3 Delete TACACS Server

Follow the steps below to delete TACACS server.

Step	Command	Description
Step 1	**config tacacs delete <ip_address>**	Remove TACACS server. Ip address: TACACS server IP address.
Step 2	**show tacacs**	Displays the TACACS configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to delete TACACS server.

```
admin@sonic:~$ sudo config tacacs delete 192.168.100.34
admin@sonic:~$ show tacacs
TACPLUS global auth_type pap (default)
TACPLUS global timeout 5 (default)
TACPLUS global passkey <EMPTY_STRING> (default)
```

2.5.3.4 Configure TACACS authtype

Follow the steps below to configure a TACACS authtype.

Step	Command	Description
Step 1	**config tacacs authtype (chap | pap | mschap | login)**	Configure TACACS authtype.
Step 2	**show tacacs**	Displays the TACACS configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure TACACS authtype.

```
admin@sonic:~$ sudo config tacacs authtype mschap
admin@sonic:~$ show tacacs
TACPLUS global auth_type mschap
TACPLUS global timeout 5 (default)
TACPLUS global passkey <EMPTY_STRING> (default)
```

2.5.3.5 Configure TACACS default

Follow the steps below to configure TACACS default.

Step	Command	Description
Step 1	**config tacacs default (authtype | passkey | timeout)**	Configure TACACS default. Ip address: TACACS server IP address.
Step 2	**show tacacs**	Displays the TACACS configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure TACACS defaults.

```
admin@sonic:~$ sudo config tacacs default authtype
admin@sonic:~$ show tacacs
TACPLUS global auth_type pap (default)
TACPLUS global timeout 5 (default)
TACPLUS global passkey <EMPTY_STRING> (default)
```

2.5.3.6 Configure TACACS passkey

Follow the steps below to configure TACACS passkey.

Step	Command	Description
Step 1	**config tacacs passkey <pass_key>**	Configure TACACS passkey. pass_key - TACACS server passkey.
Step 2	**show tacacs**	Displays the TACACS configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure TACACS passkey.

```
admin@sonic:~$ sudo config tacacs passkey testing123
admin@sonic:~$ show tacacs
TACPLUS global auth_type pap (default)
TACPLUS global timeout 5 (default)
TACPLUS global passkey testing123
```

2.5.3.7 Configure TACACS timeout

Follow the steps below to configure TACACS timeout.

Step	Command	Description
Step 1	**config tacacs [default] timeout [<timeout_value_in_seconds>]**	Configure TACACS timeout. Timeout value in seconds: Timeout value TACACS server in seconds.
Step 2	**show tacacs**	Displays the TACACS configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure TACACS timeout.

```
admin@sonic:~$ sudo config tacacs timeout 60
```

```
admin@sonic:~$ show tacacs
TACPLUS global auth_type pap (default)
TACPLUS global timeout 60
TACPLUS global passkey testing123
admin@sonic:~$
```

2.5.3.8 TACACS Configuration Example

This section explains the configurations of TACACS.

Step	Command	Description
Step 1	**config tacacs add <ip_address> [-t|--timeout <seconds>] [-k|--key <secret>] [-a|--type <type>] [-o|--port <port>] [-p|--pri <priority>] [-m|--use-mgmt-vrf]**	Configure TACACS server to be used. Ip address: TACACS server IP address. timeout: Transmission timeout interval in seconds, range 1 to 60, default 5 key: Shared secret type: Authentication type, "chap" or "pap" or "mschap" or "login", default is "pap". port: TCP port range is 1 to 65535, default 49 pri: Priority, priority range 1 to 64, default 1. use-mgmt-vrf: This means that the server is part of Management vrf, default is "no vrf" Options: -t, --timeout - INTEGER Transmission timeout interval, default 5 -k, --key TEXT - Shared secret -a, --auth_type [chap |pap |mschap |login] - Authentication type, default pap

		-o, --port INTEGER RANGE - TCP port range is 1 to 65535, default 49 -p, --pri INTEGER RANGE - Priority, default 1 -m, --use-mgmt-vrf - Management vrf, default is no vrf
Step 2	**config tacacs passkey <pass_key>**	Configure TACACS passkey. pass_key - TACACS server passkey.
Step 3	**config aaa authentication login (tacacs+ | local | default)**	Configure AAA authentication login.
Step 4	**config aaa authentication failthrough (enable | disable | default)**	Configure AAA authentication failthrough.
Step 5	**show tacacs**	Displays the TACACS configuration.
Step 6	**show aaa**	Displays the AAA configuration.

The following example shows commands used to configure TACACS.

```
admin@sonic:~$ sudo config tacacs add 192.168.86.35
admin@sonic:~$ sudo config tacacs passkey testing123
admin@sonic:~$ sudo config aaa authentication login local tacacs+
admin@sonic:~$ sudo config aaa authentication failthrough enable
admin@sonic:~$ show tacacs
TACPLUS global auth_type pap (default)
TACPLUS global timeout 5 (default)
TACPLUS global passkey testing123
TACPLUS_SERVER address 192.168.86.35
        priority 1
        tcp_port 49

admin@sonic:~$ show aaa
AAA authentication login local,tacacs+
AAA authentication failthrough True
AAA authorization login local (default)
AAA accounting login disable (default)
admin@sonic:~$
```

2.5.3.9 TACACS Server Configuration

The TACACS server has to be configured properly for the authentication to work. Below is the reference configuration for TACACS+ server running in ubuntu linux. The configuration may vary for different TACACS servers.

Create or add the users to the file /etc/tacacs+/tac_plus.conf users in the TACACS server. Each user has to be added to the file. The 'key' is the passkey configured in the switch (refer to section Configure TACACS passkey). Below is the configuration to create a user named "TestUser" with password "tacacs".

```
key = testing123

user = TestUser {
  default service = permit
  name = "TACACS User"
  pap = cleartext "tacacs"
  service = exec {
  priv-lvl = 15
  }
}
```

TEST:

```
login as: TestUser
TestUser@192.168.86.28's password:tacacs
Linux sonic 4.19.0-12-2-arm64 #1 SMP Debian 4.19.152-1 (2020-10-18) aarch64
You are on

 ____   ___  _   _ _  ____
/ ___| / _ \| \ | (_)/ ___|
\___ \| | | |  \| | | |
 ___) | |_| | |\  | | |___
|____/ \___/|_| \_|_|\____|

-- Software for Open Networking in the Cloud --

Unauthorized access and/or use are prohibited.
All access and/or use are subject to monitoring.

Help:    http://azure.github.io/SONiC/
TestUser@sonic:~$
```

2.6 Configuration Management

This section describes the steps to save and manage the configuration files on the SONiC switch.

2.6.1 Save Startup-Config

Follow the steps below to save the config DB configuration into the default /etc/sonic/config_db.json.

Step	Command	Description
Step 1	**config save [-y|--yes] [<filename>]** **sudo config save –y**	Saves the running configuration to be part of startup configuration.

The example below shows the commands used to save startup config in default file.

```
admin@sonic: ~$ sudo config save -y
Running command: /usr/local/bin/sonic-cfggen -d --print-data > /etc/sonic/config_db.json
admin@sonic: ~$ sudo reboot (Note: Truncated output is added here)
requested COLD shutdown
/var/log: 205.9 MiB (215916544 bytes) trimmed on /dev/loop1
Tue 30 Nov 2021 02:16:00 PM UTC Issuing OS-level reboot ...
```

2.6.2 Save Running Configuration to File

Follow the steps below to save the config DB configuration into the user-specified filename.

Step	Command	Description
Step 1	**config save [-y|--yes] [<filename>]** **sudo config save -y /etc/sonic/config2.json**	Saves the running configuration to the filename mentioned. filename – filename in which configuration should be saved

The example below shows the commands used to write existing switch configuration to a file.

```
admin@sonic: ~$ sudo config save -y /etc/sonic/config2.json
Running command: /usr/local/bin/sonic-cfggen -d --print-data > /etc/sonic/config2.json
```

2.6.3 Erase Startup-Config

Follow the steps below to Erase the existing config DB configuration and store default configuration into /etc/sonic/config_db.json.

Step	Command	Description
Step 1	**sudo rm /etc/sonic/config_db.json**	Remove the /etc/sonic/config_db.json
Step 2	**sudo config-setup factory**	Generate factory default configuration
Step 3	**sudo reboot**	Restore default configuration file on /etc/sonic/config_db.json

The example below shows the commands used to Erase the Startup config.

```
admin@sonic:~$ sudo rm /etc/sonic/config_db.json
admin@sonic:~$ sudo config-setup factory
admin@sonic:~$ sudo reboot
```

2.6.4 Reset-to-factory Defaults

Follow the steps below to reset the switch to factory-default-configuration.

Step	Command	Description
Step 1	**sudo config system reset-to-factory**	Resets the switch to factory-default-configuration.

The example below shows the command used to reset the switch to factory-default-configuration.

```
admin@sonic:~$ sudo config system reset-to-factory
This command will reset settings to factory defaults. After resetting to factory defaults, all configs will be lost and switch will be reloaded immediately. Do you really want to execute this command and reload the switch? [y/N]: y
```

2.6.5 Boot-up options

Follow the steps below to display the images installed on the device.

Step	Command	Description
Step 1	**show boot**	Displays the images installed on the device

The example below shows the commands used to display current boot options.

```
admin@sonic:~$ show boot
Current: SONiC-OS-Supermicro_sse-g3748_3.2.0-0011
Next: SONiC-OS-Supermicro_sse-g3748_3.2.0-0011
Available:
SONiC-OS-Supermicro_sse-g3748_3.2.0-0011

admin@sonic:~$
```

2.6.6 Warm Reboot

Follow the steps below to warm reboot of the device.

Step	Command	Description
Step 1	**sudo warm-reboot**	Initiates a warm reboot of the device

The example below shows the commands used to Initiates a warm reboot of the device.

```
admin@sonic:~$ sudo warm-reboot -v
Wed 21 Jul 2021 07:13:34 PM UTC Saving counters folder before warmboot...
cat: /host/grub/grub.cfg: No such file or directory
Wed 21 Jul 2021 07:13:35 PM UTC warm-reboot failure (1) cleanup ...
Wed 21 Jul 2021 07:13:37 PM UTC Cancel warm-reboot: code (0)
admin@sonic:~$
```

2.7 Switch features

2.7.1 Defaults

Feature	State	AutoRestart
BGP	Enabled	Enabled

Database	Always_enabled	Always_enabled
Dhcp_relay	Disabled	Enabled
LLDP	Enabled	Enabled
Macsec	Disabled	Enabled
mgmt-framework	Enabled	Enabled
NAT	disabled	Enabled
Pmon	Enabled	Enabled
Radv	Enabled	Enabled
Sflow	disabled	Enabled
SNMP	Enabled	Enabled
Swss	Enabled	Enabled
Syncd	Enabled	Enabled
Teamd	Enabled	Enabled
Telemetry	enabled	Enabled

2.7.2 Configure state

Follow the steps below to configure state for a specific feature.

Step	Command	Description
Step 1	**config feature state <feature_name> (enabled | disabled)**	Configure state for a specific feature. Feature name – Feature name (e.g bgp, lldp, pmon)
Step 2	**show feature status [<feature_name>]**	Status of feature state.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure state for a specific feature.

```
admin@sonic: ~$ sudo config feature state bgp disabled
admin@sonic: ~$ show feature status
Feature              State           AutoRestart          SetOwner
----------           --------------  --------------        ----------
bgp                  disabled        enabled
database             always_enabled  always_enabled
dhcp_relay           disabled        enabled                 local
lldp                 enabled         enabled
macsec               disabled        enabled
nat                  disabled        enabled
```

```
pmon            enabled         enabled
radv            enabled         enabled
sflow           disabled        enabled
snmp            enabled         enabled
swss            enabled         enabled
syncd           enabled         enabled
teamd           enabled         enabled
telemetry       enabled         enabled
admin@sonic: ~$
```

2.7.3 Configure auto-restart

Follow the steps below to configure auto-restart for feature.

Step	Command	Description
Step 1	**config feature autorestart <feature_name> (enabled | disabled)**	Configure auto-restart for a particular feature. Feature name – Feature name (e.g bgp, lldp, pmon)
Step 2	**show feature autorestart [<feature_name>]**	Status of auto-restart for feature.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure auto-restart for feature.

```
admin@sonic: ~$ sudo config feature autorestart bgp disabled
admin@sonic: ~$ show feature autorestart
Feature              AutoRestart
----------           --------------
bgp                  disabled
database             always_enabled
dhcp_relay           enabled
lldp                 enabled
macsec               enabled
nat                  enabled
pmon                 enabled
radv                 enabled
sflow                enabled
```

```
snmp            enabled
swss            enabled
syncd           enabled
teamd           enabled
telemetry       enabled
admin@sonic: ~$
```

2.8 Reload

This section explains the reload configuration support in SONiC switches.

2.8.1 Reload configuration

This command is used to clear current configuration and import new configuration from the input file or from /etc/sonic/config_db.json. This command shall stop all services before clearing the configuration and it then restarts those services.

Follow the steps below to reload configuration.

Step	Command	Description
Step 1	**config reload [OPTIONS] [FILENAME]**	Configure reload options. Filename – Names of configuration file(s) to load, separated by comma with no spaces in between Options: "-y" or "--yes", - Forces the loading without prompting the user for confirmation. If the argument is not specified, it prompts the user to confirm whether user really wants to load this configuration file. "-n" or "--no-service-restart", - clear and loads the configuration without restarting dependent services running on the device. One use case for this option is during boot time when config-setup service loads existing old configuration and there is no services running on the device. "-f" or "--force" - ignores the system sanity checks. By default a list of

		sanity checks are performed and if one of the checks fail, the command will not execute. The sanity checks include ensuring the system status is not starting, all the essential services are up and swss is in ready state.

The example below shows the command used to configure reload options.

```
admin@sonic: ~$ sudo config reload
running
admin@sonic: ~$ sudo config reload -y
running
admin@sonic: ~$
```

2.8.2 Configure load

This command reads the specified JSON file and writes it to the config database for addition and replacement as running settings. This command loads the configuration from the input file (if user specifies this optional filename, it will use that input file. Otherwise, it will use the default /etc/sonic/config_db.json file as the input file) into CONFIG_DB.

The configuration present in the input file is applied on top of the already running configuration. This command does not flush the config DB before loading the new configuration (i.e., If the configuration present in the input file is same as the current running configuration, nothing happens) If the config present in the input file is not present in running configuration, it will be added. If the config present in the input file differs (when key matches) from that of the running configuration, it will be modified as per the new values for those keys.

Follow the steps below to load the configuration.

Step	Command	Description
Step 1	**config load [OPTIONS] [FILENAME]**	Configure load. Filename – Names of configuration file(s) to load, separated by comma with no spaces in between. OPTIONS - -y indicates yes to reload the current configuration file

The example below shows the command used to load configuration.

```
admin@sonic: ~$ sudo config load -y
Running command: /usr/local/bin/sonic-cfggen -j /etc/sonic/config_db.json –write-to-db
admin@sonic: ~$ sudo config load
Load config from the default config file(s) ? [y/N]: y
Running command: /usr/local/bin/sonic-cfggen -j /etc/sonic/config_db.json –write-to-db
admin@sonic: ~$
```

2.9 SNMP

The SNMP agent also resides on the switch. It processes the SNMP requests received from the SNMP manager. SNMP agents also send voluntary traps to SNMP managers. Traps are sent to alert the SNMP managers on events happening on the switch.

2.9.1 Defaults

Parameter	Default Value
location	public
Community_string	public
Community_type	RO

2.9.2 Configure SNMP Agent Address

Follow the steps below to configure the snmp agent address.

Step	Command	Description
Step 1	**config snmpagentaddress add <SNMP AGENT LISTENING IP Address>**	Add the snmp agent address SNMP AGENT LISTENING IP Address – valid ipv4 address
Step 2	**config snmpagentaddress del <SNMP AGENT LISTENING IP Address>**	Delete the snmp agent address SNMP AGENT LISTENING IP Address – valid ipv4 address
Step 3	**show snmpagentaddress**	snmp agent address

Step 4	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure the snmp agent address.

```
admin@sonic: ~$ sudo config snmpagentaddress add 192.168.100.11
admin@sonic: ~$ show snmpagentaddress
ListenIP              ListenPort    ListenVrf
--------------        ------------      -----------
192.168.100.11
admin@sonic: ~$ sudo config snmpagentaddress del 192.168.100.11
admin@sonic: ~$ show snmpagentaddress
ListenIP    ListenPort    ListenVrf
----------  ------------  -----------
admin@sonic: ~$
```

2.9.3 Configure SNMP Trap

Follow the steps below to configure the snmp trap.

Step	Command	Description
Step 1	**config snmptrap modify <SNMP Version> <SNMP TRAP SERVER IP Address>** **config snmptrap del <SNMP Version>**	Configure the snmp trap SNMP AGENT LISTENING IP Address – valid ipv4 address
Step 2	**show runningconfiguration snmp**	Running configuration of the snmp module
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure the snmp trap.

```
admin@sonic: ~$ sudo config snmptrap modify 1 192.168.100.11
admin@sonic: ~$ show snmptrap
  Version    TrapReceiverIP     Port      VRF     Community
  ---------     ----------------       ------      -----    -----------
      1        192.168.100.34     162       None     public
admin@sonic: ~$
```

```
admin@sonic: ~$ sudo config snmptrap del 1
admin@sonic: ~$ show snmptrap
Version    TrapReceiverIP    Port    VRF    Community
---------      ----------------        ------  -----  -----------
admin@sonic: ~$
```

2.9.4 Configure SNMP location

Follow the steps below to configure the SNMP location.

Step	Command	Description
Step 1	**config snmp location add <location>**	Configures SNMP location. location - A valid location string.
Step 2	**show runningconfiguration snmp location**	Displays the SNMP location configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the command used to configure snmp location.

```
admin@sonic: ~$ sudo config snmp location add LA
SNMP Location LA city has been added to configuration
Restarting SNMP service...
admin@sonic: ~$ show runningconfiguration snmp location
Location
----------
LA
admin@sonic: ~$
```

2.9.5 Modify SNMP location

Follow the steps below to modify the SNMP location.

Step	Command	Description
Step 1	**config snmp location mod <location>**	Modifies SNMP location.

		location - A valid location string.
Step 2	**show runningconfiguration snmp location**	Displays the SNMP location configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the command used to modify the snmp location.

```
admin@sonic: ~$ sudo config snmp location mod New York
SNMP location New York modified in configuration
Restarting SNMP service...
admin@sonic: ~$ show runningconfiguration snmp location
Location
----------
New York
 admin@sonic: ~$
```

2.9.6 Remove SNMP location

Follow the steps below to remove the SNMP location.

Step	Command	Description
Step 1	**config snmp location del <location>**	Deletes SNMP location. location - A valid location string.
Step 2	**show runningconfiguration snmp location**	Displays the SNMP location configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the command used to remove snmp location.

```
admin@sonic: ~$ sudo config snmp location del New York
SNMP Location New York removed from configuration
Restarting SNMP service...
admin@sonic: ~$ show runningconfiguration snmp location
Location
```

```
----------
admin@sonic: ~$
```

2.9.7 Configure SNMP contact

Follow the steps below to configure the SNMP contact.

Step	Command	Description
Step 1	**config snmp contact add <contact_name> <contact_email>**	Configures SNMP contact. Contact name - A valid string. Contact email - A valid contact email string.
Step 2	**show runningconfiguration snmp contact**	Displays the SNMP contact configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the command used to configure snmp contact.

```
admin@sonic: ~$ sudo config snmp contact add user user@email.com
Contact name sonic and contact email user@email.com have been added to configuration
Restarting SNMP service...
admin@sonic: ~$ show runningconfiguration snmp contact
Contact    Contact Email
---------  ---------------
User       user@email.com
admin@sonic: ~$
```

2.9.8 Modify SNMP contact

Follow the steps below to modify the SNMP contact.

Step	Command	Description
Step 1	**config snmp contact mod <contact> <contact email>**	Modify SNMP contact. Contact name - A valid string.

		Contact email - A valid contact email string.
Step 2	**show runningconfiguration snmp contact**	Displays the SNMP contact configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the command used to modify the snmp contact.

```
admin@sonic: ~$ sudo config snmp contact mod user user1@email.com
SNMP contact user and contact email user1@email.com updated
Restarting SNMP service...
admin@sonic: ~$ show runningconfiguration snmp contact
Contact    Contact Email
---------  ------------------
User       user1@email.com
admin@sonic: ~$
```

2.9.9 Remove SNMP contact

Follow the steps below to remove the SNMP contact.

Step	**Command**	**Description**
Step 1	**config snmp contact del <contact>**	Delete SNMP contact. Contact name - A valid string.
Step 2	**show runningconfiguration snmp contact**	Displays the SNMP contact configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the command used to remove snmp contact.

```
admin@sonic: ~$ sudo config snmp contact del user
SNMP contact user removed from configuration
Restarting SNMP service...
admin@sonic: ~$ show runningconfiguration snmp contact
```

```
Contact    Contact Email
---------      ---------------
admin@sonic: ~$
```

2.9.10 Configure SNMP community

Follow the steps below to configure the SNMP community.

Step	Command	Description
Step 1	**config snmp community add <snmp_community> <RO|RW>**	Configures SNMP community. Snmp Community - A valid string.
Step 2	**show runningconfiguration snmp community**	Displays the SNMP community configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the command used to configure the snmp community.

```
admin@sonic: ~$ sudo config snmp community add user ro
SNMP community user added to configuration
Restarting SNMP service...
admin@sonic: ~$ show runningconfiguration snmp community
Community String    Community Type
------------------          ----------------
public                      RO
user                        RO
admin@sonic: ~$
```

2.9.11 Modify SNMP community

Follow the steps below to modify the SNMP community.

Step	Command	Description
Step 1	**config snmp community replace <current_community_string> <new_community_string>**	Modify SNMP community. Snmp Community - A valid string. New Snmp Community - A valid string.

Step 2	**show runningconfiguration snmp community**	Displays the SNMP community configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the command used to modify snmp community.

```
admin@sonic: ~$ sudo config snmp community replace sonic user1
SNMP community user1 added to configuration
SNMP community user1 replace community sonic
Restarting SNMP service...
admin@sonic: ~$ show runningconfiguration snmp community
Community String    Community Type
------------------  ----------------
public               RO
user1                RO
admin@sonic: ~$
```

2.9.12 Remove SNMP community

Follow the steps below to remove the SNMP community.

Step	Command	Description
Step 1	**config snmp community del <snmp_community>**	Remove SNMP community. Snmp Community - A valid string.
Step 2	**show runningconfiguration snmp community**	Displays the SNMP community configuration.
Step 3	**sudo config save –y**	Optional step - Saves this current configuration to be part of startup configuration.

The example below shows the command used to remove snmp community.

```
admin@sonic: ~$ sudo config snmp community del user1
SNMP community user1 removed from configuration
Restarting SNMP service...
admin@sonic: ~$ show runningconfiguration snmp community
```

```
Community String    Community Type
------------------  ----------------
public               RO
admin@sonic: ~$
```

2.9.13 Configure SNMP users

Follow the steps below to configure the SNMP users.

Step	Command	Description
Step 1	**config snmp user add <snmp_user> <noAuthNoPriv | AuthNoPriv |Priv> <RO|RW> <MD5|SHA|HMAC-SHA-2> <auth_password> <DES|AES> <encrypt_password>**	Configures SNMP users. Snmp user - A valid string. Auth password - A valid string.
Step 2	**show runningconfiguration snmp user**	Displays the SNMP users configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the command used to configure snmp users.

```
admin@sonic: ~$ sudo config snmp user add testuser1 noauthnopriv ro
SNMP user testuser1 added to configuration
Restarting SNMP service...
admin@sonic: ~$ show runningconfiguration snmp user
User        Permission Type    Type        Auth Type    Auth Password    Encryption Type    Encryption
Password
---------   -----------------  ----------  ----------   ---------------  -----------------  --------------------
testuser1    RO             noAuthNoPriv
admin@sonic: ~$
admin@sonic: ~$ sudo config snmp user add testuser2 authnopriv ro sha testuser2_auth_pass
SNMP user testuser2 added to configuration
Restarting SNMP service...
admin@sonic: ~$ show runningconfiguration snmp user
User        Permission Type    Type        Auth Type    Auth Password      Encryption Type    Encryption
Password
---------   -----------------  ----------  ----------   -----------------  -----------------  -------------
---
testuser1    RO             noAuthNoPriv
testuser2    RO             AuthNoPriv        SHA       testuser2_auth_pass
```

```
admin@sonic: ~$
admin@sonic: ~$ sudo config snmp user add testuser3 priv rw md5 testuser3_auth_pass aes testuser3_encrypt_pass
SNMP user testuser3 added to configuration
Restarting SNMP service...
admin@sonic: ~$ show runningconfiguration snmp user
User      Permission Type   Type         Auth Type   Auth Password      Encryption Type   Encryption Password
--------- ----------------- ------------ ----------- ------------------ ----------------- ------------------
testuser1 RO                noAuthNoPriv
testuser2 RO                AuthNoPriv   SHA         testuser2_auth_pass
testuser3 RW                Priv         MD5         testuser3_auth_pass AES               testuser3_encrypt_pass
admin@sonic: ~$
```

2.9.14 Remove SNMP users

Follow the steps below to remove the SNMP users.

Step	Command	Description
Step 1	**config snmp user del <snmp_user>**	Remove SNMP users. Snmp user - A valid string.
Step 2	**show runningconfiguration snmp users**	Displays the SNMP users configuration.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the command used to remove SNMP users.

```
admin@sonic: ~$ sudo config snmp user del testuser1
SNMP user testuser1 removed from configuration
Restarting SNMP service...
admin@sonic: ~$ show runningconfiguration snmp user
User      Permission Type   Type         Auth Type   Auth Password      Encryption Type   Encryption Password
--------- ----------------- ------------ ----------- ------------------ ----------------- ------------------
testuser2 RO                AuthNoPriv   SHA         testuser2_auth_pass
testuser3 RW                Priv         MD5         testuser3_auth_pass AES               testuser3_encrypt_pass
```

```
admin@sonic: ~$
```

2.10NTP

The Network Time Protocol (NTP) helps to keep the switch's clock synchronized with a network time server. Maintaining the synchronized time across all devices would help in troubleshooting the network events that spans multiple devices.

2.10.1Configure NTP server Address

Follow the steps below to configure the NTP server address.

Step	Command	Description
Step 1	**config ntp add [OPTIONS] <ntp_ip_address>**	Add NTP server address. ntp_ip_address – valid IPv4 or IPv6 address.
Step 2	**config ntp del [OPTIONS] <ntp_ip_address>**	Delete NTP server address. ntp_ip_address – valid IPv4 or IPv6 address.
Step 3	**show runningconfiguration ntp**	Displays the current NTP configuration.
Step 4	**show ntp**	Show NTP information.
Step 5	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure IPv4 NTP server addresses.

```
admin@sonic: ~$ sudo config ntp add 171.66.97.126
admin@sonic: ~$ sudo config ntp add 216.40.34.37
admin@sonic: ~$ show ntp
MGMT_VRF_CONFIG is not present.
synchronised to NTP server (171.66.97.126) at stratum 3
  time correct to within 89 ms
  polling server every 64 s
    remote         refid      st t when poll reach   delay   offset  jitter
==============================================================================
*171.66.97.126   171.64.7.73     2 u   36   64   37    5.631    0.425   1.073
```

```
 216.40.34.37    .INIT.          16 u    -  64    0    0.000    0.000   0.000

admin@sonic: ~$ show runningconfiguration ntp
NTP Servers
-------------
171.66.97.126
216.40.34.37
admin@sonic:~$
```

The example below shows the commands used to configure IPv6 NTP server address.

```
admin@sonic: ~$ sudo config ntp add 2001:0:0:1::1
NTP server 2001:0:0:1::1 added to configuration
Restarting ntp-config service...
admin@sonic: ~$ show ntp
MGMT_VRF_CONFIG is not present.
unsynchronised
   polling server every 8 s
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 2001:0:0:1::1   .INIT.          16 u    -  64    0    0.000    0.000   0.000

admin@sonic: ~$ show runningconfiguration ntp
NTP Servers
-------------
2001:0:0:1::1
admin@sonic:~$
```

2.10.2 Delete NTP server

Follow the below steps to delete NTP server.

Step	Command	Description
Step 1	**config ntp del <ip_address>**	Removes NTP server. Ip address - A valid IPv4 or IPv6 address.
Step 2	**show ntp**	Displays the NTP configuration.

Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to delete the IPv4 NTP server address.

```
admin@sonic: ~$ sudo config ntp del 216.40.34.37

NTP server 216.40.34.37 removed from configuration

Restarting ntp-config service...
```

The example below shows the commands used to delete the IPv6 NTP server address.

```
admin@sonic: ~$ sudo config ntp del 2001:0:0:1::1

NTP server 2001:0:0:1::1 removed from configuration

Restarting ntp-config service...
```

2.10.3 Configure Time Zone

Follow the steps below to configure the time zone for the switch.

Step	Command	Description
Step 1	**timedatectl set-timezone <zone>**	Configures the timezone of the switch. zone – valid timezone.
Step 2	**timedatectl list-timezones**	Lists the valid time zones.
Step 3	**timedatectl status**	Displays the current time zone.

The example below shows the commands used to add the NTP server addresses.

```
admin@sonic: ~$ sudo timedatectl set-timezone America/Los_Angeles
admin@sonic:~$ timedatectl status
               Local time: Thu 2022-09-29 18:03:53 PDT
           Universal time: Fri 2022-09-30 01:03:53 UTC
                 RTC time: n/a
                Time zone: America/Los_Angeles (PDT, -0700)
System clock synchronized: yes
              NTP service: active
```

```
RTC in local TZ: no
```

2.11System Logging (Syslog)

SONiC switches send system message outputs to a logging process and this is called System Message Logging (Syslog). This displays all the currently stored log messages. All the latest processes and corresponding transactions are stored in the "syslog" file. This file is saved in the path /var/log and can be viewed by giving the command *sudo cat syslog* as this requires root login.

2.11.1 Configure syslog

Follow the below steps to configure syslog server parameters.

Step	Command	Description
Step 1	**config syslog add <ip_address>**	Configures SYSLOG server. Ip address - A valid IPv4/IPv6 Address.
Step 2	**show logging [OPTIONS] [PROCESS]**	Displays Currently stored log message Process – Process name, if wanted specific process logging details Options: -l – shows the lines text -f - follow
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to configure IPv4 SYSLOG server.

```
admin@sonic:~$ sudo config syslog add 192.168.86.24
Syslog server 192.168.86.24 added to configuration
Restarting rsyslog-config service...
admin@sonic:~$ show logging –follow (Note: Truncated output is added here)
Jul 21 19:04:59.164028 sonic INFO systemd[1]: rsyslog.service: Succeeded.
Jul 21 19:04:59.164386 sonic INFO systemd[1]: Stopped System Logging Service.
Jul 21 19:04:59.178349 sonic INFO systemd[1]: Starting System Logging Service...
Jul 21 19:04:59.221866 sonic INFO rsyslogd: imuxsock: Acquired UNIX socket '/run
/systemd/journal/syslog' (fd 3) from systemd.  [v8.1901.0]
```

The example below shows the commands used to configure IPv4 SYSLOG server.

```
admin@sonic:~$ sudo config syslog add fddd:0:0:1::1
Syslog server fddd:0:0:1::1 added to configuration
Restarting rsyslog-config service…
root@sonic:/var/log#
```

2.11.2 Delete syslog

Follow the below steps to delete configured SYSLOG server parameters.

Step	Command	Description
Step 1	**config syslog del <ip_address>**	Removes SYSLOG server. IP address - A valid IPv4 Address.
Step 2	**show logging [OPTIONS] [PROCESS]**	Displays Currently stored log message Process – Process name, if wanted specific process logging details Options: -l – shows the lines text -f - follow
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

The example below shows the commands used to delete SYSLOG server.

```
admin@sonic:~$ sudo config syslog del 192.168.86.24
Syslog server 192.168.86.24 removed from configuration
Restarting rsyslog-config service...
admin@sonic:~$ show logging –follow  (Note: Truncated output is added here)
Jul 21 19:06:51.627399 sonic INFO dhclient[1633]: XMT: Solicit on eth0, interval 114510ms.
Jul 21 19:06:54.788887 sonic INFO rsyslogd:  [origin software="rsyslogd" swVersion="8.1901.0" x-pid="5744" x-info="https://www.rsyslog.com"] exiting on signal 15.
Jul 21 19:06:54.794217 sonic INFO systemd[1]: Stopping System Logging Service...
Jul 21 19:06:54.797112 sonic INFO systemd[1]: rsyslog.service: Succeeded.
```

```
Jul 21 19:06:54.799299 sonic INFO systemd[1]: Stopped System Logging Service.
Jul 21 19:06:54.813666 sonic INFO systemd[1]: Starting System Logging Service...
```

2.12Zero Touch Provisioning (ZTP)

Zero Touch Provisioning (ZTP) helps to auto provision Supermicro switches without manual intervention. ZTP also helps to upgrade the switch firmware automatically.

Supermicro SONiC switches come with the default management IP address set to DHCP mode. When switches boot up, the management IP address is received from the DHCP server. The DHCP server can also be configured to supply the switch configurations and firmware image when assigning IP addresses to Supermicro switches.

ZTP is enabled by default in Supermicro SONiC switches. For the proper ZTP operation ensure the management port is connected to correct network, DHCP server has necessary configurations and TFTP/HTTP servers has necessary files.

2.12.1DHCP Scope Options to add in DHCP Server

The switch expects two information, TFTP server IP address and ZTP config file, from the DHCP server. The TFTP server IP address is supplied via the DHCP option tftp-server-name and the ZTP configuration file name is supplied via the DHCP option bootfile-name. Add the tftp-server-name and bootfile-name options to your DHCP scope as shown below. The DHCP service may need to be restarted for the change to take effect.

```
subnet 10.5.5.0 netmask 255.255.255.0 {
   range 10.5.5.11 10.5.5.250;
   option routers 10.5.5.1;
   default-lease-time 6000;
   max-lease-time 6000;
   option tftp-server-name "10.5.5.5";
   option bootfile-name "G3748-sonic-ZTP-config.json";
 }
```

If a different config_db.json configuration needs to be applied on a specific switch, then create a host specific option as shown below within the DHCP scope. The filename mentioned in the bootfilename option should point to different filename based on the management mac address of the switch.

```
subnet 10.5.5.0 netmask 255.255.255.0 {
   range 10.5.5.11 10.5.5.250;
   option routers 10.5.5.1;
   default-lease-time 6000;
   max-lease-time 6000;
   option tftp-server-name "10.5.5.5";
   option bootfile-name "G3748-sonic-ZTP-config.json";
```

```
        host special-sonicg3748 {
          hardware ethernet 88:5a:85:fa:2d:a9;
          option bootfile-name "G3748-special-sonic_config_db.json";
        }
}
```

2.12.2 Add Files to TFTP/HTTP Server

The switch downloads the configuration file and firmware file from the TFTP server in 2 steps. First the ZTP configuration file that is mentioned in the bootfile-name option of the DHCP scope is downloaded and parsed. The actual switch configuration file mentioned in the ZTP configuration file is downloaded and copied to the /etc/sonic/config_db.json. If this ZTP configuration has the upgrade section, then the firmware file is downloaded and installed. So, based on the user need, the following files need to be uploaded to the TFTP/HTTP server.

1. ZTP configuration file.
2. Switch Configuration file (config_db.json).
3. Firmware image file.

1. Sample ZTP configuration file is shown below. Make sure the filename mentioned in DHCP server configuration matches with actual name of the configuration file in TFTP server.

ZTP Configuration file: G3748-sonic-ZTP-config.json

```
root@TFTP-server:/home/tftp# cat G3748-sonic-ZTP-config.json
{
  "ztp": {
    "01-configdb-json": {
       "url": {
          "source": "tftp://10.5.5.5/G3748-config_db.json",
          "destination": "/etc/sonic/config_db.json",
          "secure": false
       }
     },

    "02-firmware": {
       "install": {
          "url": "http://10.5.5.5/SONiC_SSE-G3748_3.2.0-0009.bin",
          "skip-reboot": true
       }
     }
  }
}
```

2. Upload a valid SONiC configuration file to the TFTP server. The name of the configuration file should match with the name of the file mentioned as source in the ZTP configuration file. See the filename “G3748-config_db.json” in source line in the above example. Note that the switch doesn’t download the configuration file every time when it reboots. The switch just keeps a copy of the configuration file as /etc/sonic/config_db.json.

3. Upload the switch firmware image file to the HTTP server. The name of the firmware image file must match with the filename mentioned in the firmware section in ZTP configuration.

2.13 Firmware Upgrade

The SONiC firmware can be upgraded either from SONiC CLI or from the ONIE shell. The firmware image shall be obtained from the website supermicro.com. Make sure that the firmware image is the right file for the switch model. The management port has to be connected to the correct network and the switch must get an IP address for the firmware upgrade to work.

2.13.1 Upgrading from SONiC CLI

The sonic-installer command is used to upgrade the SONiC firmware from the SONiC CLI. Some firmware may mandate the upgrade to be done from the ONIE. Please refer to the release notes for any such instructions. Upgrade from SONiC CLI can be done from a SSH terminal.

```
sonic-installer install http://<ip-address>/<path-to-image>
```

Reboot the switch when the installation is finished.

2.13.2 Upgrading from ONIE

All the configurations/settings will be lost if the switch is upgraded from ONIE. So, please backup your configurations before upgrading. Installation from ONIE has to be done from switch console.

Steps:

1) Reboot the switch and interrupt the boot during countdown by pressing any key. The switch will boot fast and one has to be very quick to interrupt at this boot point.

```
Hit any key to stop autoboot:  3
Marvell>>
```

2) Start ONIE as shown below.

```
Marvell>> run onie_bootcmd
```

3) From ONIE the firmware can be installed either using the remote HTTP/TFTP server or using an USB drive.

   a. Upgrade using HTTP/TFTP:

      Stop the auto-discovery process. This just gives a clean console and is optional.

      ```
      ONIE:/# onie-stop
      ```

      Start the SONiC upgrade from ONIE as shown below.

      ```
      ONIE:/# onie-nos-install http://<ip-address>/<path-to-image>
                                      or
      ONIE:/# onie-nos-install tftp://<ip-address>/<path-to-image>
      ```

b. Upgrade using USB drive:

   Copy the firmware image file to a FAT formatted USB drive with the filename 'onie-installer.bin'. Insert the USB drive in to the USB slot that is located between the console port and the management port. The ONIE auto-discovery process will automatically start the installation. The switch will reboot after the installation completes.

3 Layer2 Configuration

This Section describes the Layer2 features supported in SONiC switch.

3.1 VLAN

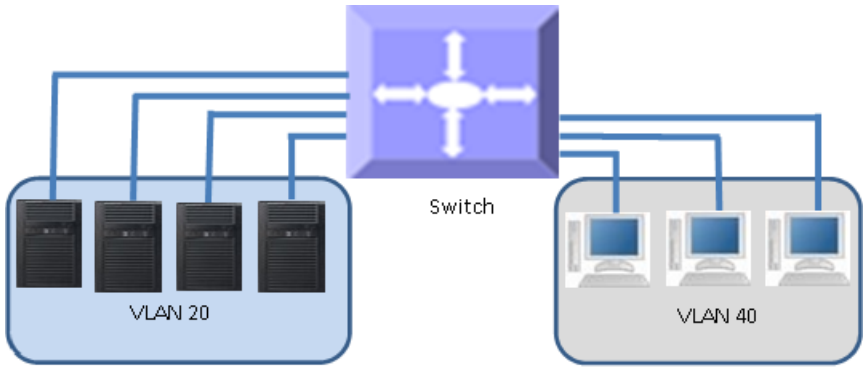
A Virtual LAN (VLAN) is a logical switched LAN formed by segmenting physical Local Area Networks (LANs).

Segmenting a switched LAN as one or more VLANs provides the following advantages:

- Multicast and broadcast floods are limited only to the required segments of the LAN to save LAN bandwidth
- It provides a secured LAN access by limiting traffic to specific LAN segments
- Eases management by logically grouping ports across multiple switches

VLANs work in same way as physical LANs. The packets from the end stations of a VLAN are switched only to other end stations or network devices inside that VLAN.

Figure VLAN-1: VLANs on a Switched LAN



To reach devices in another VLAN, the packets have to be routed from one VLAN to another.

SONiC switch supports such InterVLAN Routing to route packets across different VLANs.

InterVLAN Routing is done by creating “Layer 3 VLAN Interface”.

3.1.1 VLAN Numbers

SONiC support VLAN identifiers from 1 to 4094 for user created VLANs.

3.1.2 VLAN Defaults

There is no default VLAN configuration in SONiC switch.

3.1.3 Creating VLANs

Follow the steps below to create VLANs in SONiC.

Step	Command	Description
Step 1	**config vlan add <vid>**	Create a VLAN. vid - May be any vlan number, Range 1 to 4094.
Step 2	**show vlan config**	Displays the configured VLANs
Step 3	**show vlan brief**	Displays all bridge information
Step 4	**sudo config save -y**	Optional step - saves this configuration to be part of startup configuration.

The following example shows how to create a VLAN:

```
admin@sonic: ~$ sudo config vlan add 100

admin@sonic: ~$ show vlan config

Name        VID

----------  ------

Vlan100     100

admin@sonic: ~$ show vlan brief

+-------------+---------------+----------+------------------+---------------+------------------------------+

|  VLAN ID | IP Address  | Ports  | Port Tagging  | Proxy ARP  | DHCP Helper Address  |

+========+==========+=======+==========+==========+====================+

|     100  |              |          |                | disabled    |                            |

+-------------+---------------+----------+------------------+---------------+------------------------------+
```

3.1.4 Removing VLANs

Follow the steps below to remove VLANs from Sonic.

Step	Command	Description
Step 1	**config vlan del <vid>**	Remove VLAN. vid - May be any vlan number, Range 1 to 4094.
Step 2	**show vlan config**	Displays the configured VLANs
Step 3	**show vlan brief**	Displays all bridge information
Step 4	**sudo config save -y**	Optional step - saves this configuration to be part of startup configuration.

The following example shows how to remove a VLAN:

```
admin@sonic: ~$ sudo config vlan del 100
admin@sonic: ~$ show vlan config
Name    VID    Member    Mode
--------    -----    ------------    ---------
admin@sonic: ~$ show vlan brief
+-------------+---------------+----------+-------------------+---------------+-----------------------------+
|  VLAN ID | IP Address   | Ports   | Port Tagging   | Proxy ARP   | DHCP Helper Address   |
+========+===========+======+===========+==========+==================+
+-------------+---------------+----------+-------------------+---------------+-----------------------------+
```

3.1.5 Port Based VLANs

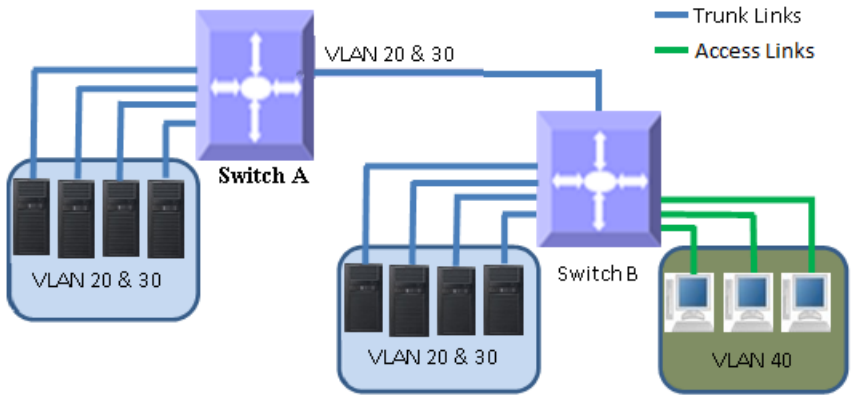
Port based VLANs are the simplest and most useful type of VLAN.

In port based VLAN deployment, switch ports are associated with one or more VLANs as member ports.

The VLAN traffic sent on the ports is decided by the VLAN membership modes of the ports. Mostly ports are associated with VLANs as either “untagged (access)” port members or “tagged (trunk)” port members.

Port Channel interfaces also can be configured as VLAN member ports.

Figure VLAN-2: Port Based VLANs



3.1.5.1 Untagged (Access) Ports

Access ports carry traffic of only one VLAN. Any switch ports can be configured as access ports. Mostly switch ports connected to end stations (computers / servers) that have only one type of traffic are configured as access ports.

When a switch port is configured as an access port to any VLAN, that port is added as an untagged member port of the given VLAN. Also, the Port based VLAN identifier (PVID) of that port is configured as the given VLAN. Each port can be configured as untagged member of only one VLAN.

Switch strips the VLAN tag header from all packets sent out on an access port. Hence, access ports are also called untagged ports.

When a packet is received on an access port, the switch identifies the VLAN for the received packet fromthe packet's VLAN tag header. If the received packet did not have a VLAN identifier the port PVID is used as VLAN for all the received untagged.

Follow the steps below to add a member port as untagged port in a VLAN.

Step	Command	Description
Step 1	**config vlan member add [-u|--untagged] <vlan_id> <member_portname>**	Add an untagged member port in the already created VLAN by using the option -u or --untagged vlan_id - may be any vlan number

		member_portname - any interface name which is not a router interface
Step 2	**show vlan config**	Displays the configured VLANs
Step 3	**show vlan brief**	Displays all bridge information
Step 4	**sudo config save -y**	Optional step - saves this configuration to be part of startup configuration.

The following example shows how to add an untagged port in a VLAN

```
admin@sonic: ~$ sudo config vlan member add -u 100 Ethernet48
admin@sonic: ~$ show vlan config
Name      VID   Member       Mode
----------  -----  -------------  ------------
Vlan100   100   Ethernet48   untagged
Vlan200   200
admin@sonic: ~$ show vlan brief
+-------------+---------------+---------------+-------------------+---------------+-----------------------------+
|  VLAN ID | IP Address  | Ports         | Port Tagging  | Proxy ARP  | DHCP Helper Address  |
+========+==========+==========+============+==========+====================+
|     100  |               | Ethernet48 | untagged      | disabled     |                            |
+------------+----------------+---------------+-------------------+---------------+-----------------------------+
|     200  |               |               |                | disabled     |                            |
+------------+----------------+---------------+-------------------+---------------+-----------------------------+
```

3.1.5.2 Tagged (Trunk) Port

Tagged (Trunk) ports carry the traffic of one or more VLANs. Any switch ports can be configured as trunk ports. Mostly switch ports connected between switches are configured as trunk ports to carry multiple VLAN traffic across switches. Switch ports connected to end stations (computers / servers) that have multiple VLANs are also configured as trunk ports.

Switch adds the VLAN tag header to all packets sent out on the trunk port. When a packet is received on a trunk port, the switch identifies the VLAN for the received packet from the packet's VLAN tag header. If the received packet did not have a VLAN the port PVID is used to determine the VLAN for all untagged and priority tagged packets that are received.

Follow the steps below to add a member port as tagged port in a VLAN.

Step	Command	Description

Step 1	**config vlan member add <vlan_id> <member_portname>**	Add a tagged member port in the already created VLAN vlan_id - may be any vlan number member_portname - any interface name which is not a router interface
Step 2	**show vlan config**	Displays the configured VLANs
Step 3	**show vlan brief**	Displays all bridge information
Step 4	**sudo config save -y**	Optional step - saves this configuration to be part of startup configuration.

The following example shows how to add a tagged port in a VLAN

```
admin@sonic: ~$ sudo config vlan member add 100 Ethernet52
admin@sonic: ~$ show vlan config
Name        VID  Member       Mode
----------  -----  ---------------  --------
Vlan100     100  Ethernet48   untagged
Vlan100     100  Ethernet52   tagged
Vlan200     200
admin@sonic: ~$ show vlan brief
+-------------+---------------+----------------+-------------------+---------------+-----------------------------+
|  VLAN ID | IP Address  | Ports          | Port Tagging  | Proxy ARP  | DHCP Helper Address  |
+========+==========+===========+============+==========+==================+
|     100  |             | Ethernet48  | untagged      | disabled     |                          |
|          |             | Ethernet52  | tagged        |              |                          |
+------------+----------------+----------------+-------------------+----------------+-----------------------------+
|     200  |             |                |                   | disabled     |                          |
+------------+----------------+----------------+-------------------+----------------+-----------------------------+
```

Ensure the port configuration of the VLAN member is not a router port. If the router port is configured as a VLAN member, the following error is displayed.

admin@sonic: ~$ sudo config vlan member add 100 Ethernet52

Usage: config vlan member add [OPTIONS] <vid> port

Try "config vlan member add -h" for help.

Error: Ethernet52 is a L3 interface!

3.1.5.3 *Remove Port from VLAN*

Follow the steps below to remove a member port from a VLAN.

Step	Command	Description
Step 1	**config vlan member del <vlan_id> <member_portname>**	Remove untagged/tagged member from a VLAN vlan_id - may be any vlan number member_portname - any interface name which is not a router interface
Step 2	**show vlan config**	Displays the configured VLANs
Step 3	**show vlan brief**	Displays all bridge information
Step 4	**sudo config save -y**	Optional step - saves this configuration to be part of startup configuration.

The following examples show how to delete an untagged or a tagged port from a VLAN

```
admin@sonic: ~$ sudo config vlan member del 100 Ethernet48
admin@sonic: ~$ show vlan config
Name        VID  Member       Mode
----------  -----  ---------------  --------
Vlan100     100  Ethernet52   tagged
Vlan200     200
admin@sonic: ~$ show vlan brief
+-------------+---------------+----------------+-------------------+---------------+-----------------------------+
|  VLAN ID | IP Address  | Ports          | Port Tagging  | Proxy ARP  | DHCP Helper Address  |
+========+==========+===========+============+==========+==================+
|            |               | Ethernet52  | tagged          |               |                           |
+------------+----------------+---------------+-----------------+----------------+----------------------------+
|     200   |               |               |                 | disabled      |                           |
+------------+----------------+---------------+-----------------+----------------+----------------------------+
Remove an interface Ethernet52 from a VLAN 100
admin@sonic: ~$ sudo config vlan member del 100 Ethernet52
admin@sonic: ~$ show vlan config
Name        VID  Member       Mode
----------  -----  ---------------  --------
Vlan100     100
Vlan200     200
admin@sonic: ~$ show vlan brief
+-------------+---------------+----------------+-------------------+---------------+-----------------------------+
```

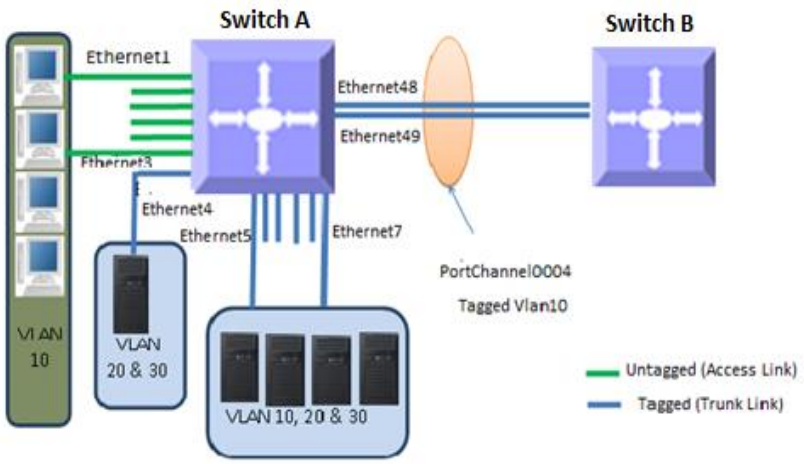
VLAN ID	IP Address	Ports	Port Tagging	Proxy ARP	DHCP Helper Address
100				disabled	
200				disabled	

3.1.6 VLAN Configuration Example

Configure the following requirements on SONiC, as shown below in Figure VLAN-3.

1. Ports Ethernet1 to Ethernet3 are untagged access ports for VLAN 10.
2. Port Ethernet4 is a trunk/tagged port connected to storage, which carries VLAN 20 and 30.
3. Ports Ethernet5 to Ethernet7 are tagged/trunk ports connected to servers that have VLANs 20, 30 and 10. Here, VLAN 10 is untagged.
4. Ports Ethernet48 and Ethernet52 are part of a tagged/trunk port channel that carries all the VLANs to other switches with untagged VLAN 10.

Figure VLAN – 3: VLAN Configuration Example



#Configure VLAN's 10,20 and 30

```
admin@sonic: ~$ sudo config vlan add 10
admin@sonic: ~$ sudo config vlan add 20
admin@sonic: ~$ sudo config vlan add 30
```

#Configure Ethernet1 to Ethernet3 as an untagged port in VLAN 10

```
admin@sonic: ~$ sudo config vlan member add 10 -u Ethernet1
admin@sonic: ~$ sudo config vlan member add 10 -u Ethernet2
admin@sonic: ~$ sudo config vlan member add 10 -u Ethernet3
```

#Configure Ethernet4 as a tagged port in VLAN 20,30

```
admin@sonic: ~$ sudo config vlan member add 20 Ethernet4
admin@sonic: ~$ sudo config vlan member add 30 Ethernet4
```

#Configure Ethernet5 to Ethernet7 as a tagged port in VLAN 10,20 and 30

```
admin@sonic: ~$ sudo config vlan member add 20 Ethernet5
admin@sonic: ~$ sudo config vlan member add 20 Ethernet6
admin@sonic: ~$ sudo config vlan member add 20 Ethernet7
admin@sonic: ~$ sudo config vlan member add 30 Ethernet7
admin@sonic: ~$ sudo config vlan member add 30 Ethernet6
admin@sonic: ~$ sudo config vlan member add 30 Ethernet5
admin@sonic: ~$ sudo config vlan member add 10 Ethernet7
admin@sonic: ~$ sudo config vlan member add 10 Ethernet6
admin@sonic: ~$ sudo config vlan member add 10 Ethernet5
```

#Configure Ethernet48 and Ethernet49 in port channel PortChannel0004 and configure port channel as a tagged port in 10

```
admin@sonic: ~$ sudo config portchannel add PortChannel0004
admin@sonic: ~$ sudo config portchannel member add PortChannel0004 Ethernet48
admin@sonic: ~$ sudo config portchannel member add PortChannel0004 Ethernet49
admin@sonic: ~$ sudo config vlan member add 10 Portchannel0004
admin@sonic: ~$ show vlan config
Name      VID  Member             Mode
---------  -----  --------------------  --------
Vlan10    10   Ethernet1          untagged
Vlan10    10   Ethernet2          untagged
Vlan10    10   Ethernet3          untagged
Vlan10    10   Ethernet5          tagged
Vlan10    10   Ethernet6          tagged
Vlan10    10   Ethernet7          tagged
Vlan10    10   PortChannel0004    tagged
```

```
Vlan20     20    Ethernet4            tagged
Vlan20     20    Ethernet5            tagged
Vlan20     20    Ethernet6            tagged
Vlan20     20    Ethernet7            tagged
Vlan30     30    Ethernet4            tagged
Vlan30     30    Ethernet5            tagged
Vlan30     30    Ethernet6            tagged
Vlan30     30    Ethernet7            tagged
Vlan100    100
Vlan200    200
admin@sonic: ~$ show vlan brief
+-------------+---------------+-----------------------+--------------------+---------------+-----------------------------+
|  VLAN ID | IP Address  | Ports                 | Port Tagging    | Proxy ARP  | DHCP Helper Address  |
+========+==========+================+=============+==========+====================+
|      10    |               | Ethernet1          | untagged         | disabled    |                          |
|            |               | Ethernet2          | untagged         |             |                          |
|            |               | Ethernet3          | untagged         |             |                          |
|            |               | Ethernet5          | tagged           |             |                          |
|            |               | Ethernet6          | tagged           |             |                          |
|            |               | Ethernet7          | tagged           |             |                          |
|            |               | PortChannel0004 | tagged           |             |                          |
+-----------+----------------+----------------------+---------------------+---------------+---------------------------
=+
|      20    |               | Ethernet4          | tagged           | disabled    |                          |
|            |               | Ethernet5          | tagged           |             |                          |
|            |               | Ethernet6          | tagged           |             |                          |
|            |               | Ethernet7          | tagged           |             |                          |
+-----------+----------------+----------------------+---------------------+---------------+---------------------------
+
|      30  |               | Ethernet4          | tagged           | disabled    |                          |
|           |               | Ethernet5          | tagged           |             |                          |
|           |               | Ethernet6          | tagged           |             |                          |
|           |               | Ethernet7          | tagged           |             |                          |
+-----------+----------------+----------------------+---------------------+---------------+---------------------------
+
|     100 |               |                       |                   | disabled     |                          |
+----------+----------------+----------------------+--------------------+---------------+---------------------------
+
|     200 |               |                       |                   | disabled     |                          |
+----------+----------------+----------------------+--------------------+---------------+---------------------------
+
admin@sonic: ~$ sudo bridge vlan
```

```
port              vlan ids
docker0            1    PVID Egress Untagged
Bridge             10
                   20
                   30
                   100
                   200
dummy              1    PVID Egress Untagged
Ethernet1          10 PVID Egress Untagged
Ethernet2          10 PVID Egress Untagged
Ethernet3          10 PVID Egress Untagged
Ethernet4          20
                   30
Ethernet5          10
                   20
                   30
Ethernet6          10
                    20
                    30
Ethernet7          10
                   20
                   30
PortChannel0004    10
```

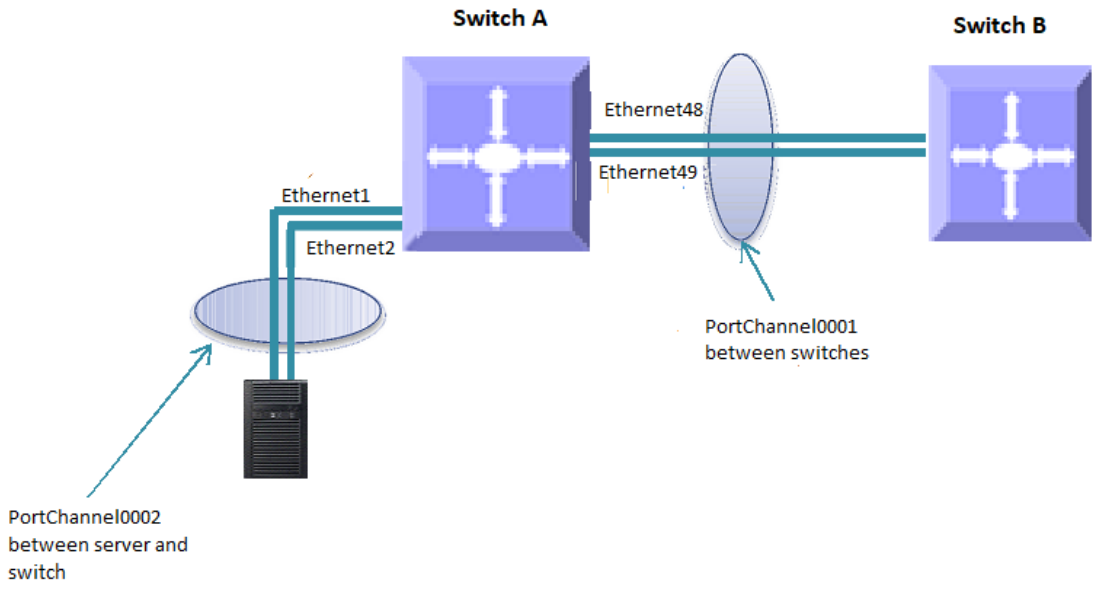
3.2 Link Aggregation

The Link Aggregation feature helps connecting two or more physical links between two network devices without forming loops. Link Aggregation can be used between switches, servers and routers.

Link Aggregation provides the following advantages:

- Increased bandwidth – User can connect more than one physical links between devices to increase the link bandwidth.
- Incremental bandwidth – Users can start aggregation with a fewer number of ports and then increase the number of ports in aggregation incrementally based on the bandwidth requirements.
- Redundancy - When one of the physical links fails, traffic will be distributed over the other remaining links in the aggregation.

Figure LA-1: Link Aggregation



The term “port channel” is used synonymously to refer to aggregated links.

3.2.1 Creating Port channels

Port channel creation involves two steps: the first step is creating the port channel interfaces and the second step is adding member ports to the port channel interfaces.

3.2.1.1 Creating Port Channel Interfaces

Follow the steps below to create port channel interfaces in SONiC:

Step	Command	Description
Step 1	**config portchannel add <portchannel_name>**	Create a port channel interface. portchannel_name - In the format "PortChannelxxxx", where "xxxx" is number of 1 to 4 digits. Ex: "PortChannel0001"
Step 2	**show interfaces portchannel**	Displays the configured port channel interfaces
Step 3	**sudo config save -y**	Optional step - saves this configuration to be part of startup configuration.

The following example shows how to create port channel interface

```
admin@sonic: ~$ sudo config portchannel add PortChannel0001
admin@sonic: ~$ show interfaces portchannel
Flags: A - active, I - inactive, Up - up, Dw - Down, N/A - not available,
       S - selected, D - deselected, * - not synced
No.    Team Dev            Protocol        Ports
-----  ------------------  ----------      -------
0001   PortChannel0001     LACP(A)(Dw)
```

It is recommended to use port channel names in the format "PortChannelxxxx", where "xxxx" is number of 1 to 4 digits. Ex: "**PortChannel0002**".

NOTE: If users specify any other name like "pc10", command will succeed, but such names are not supported as those names not printed properly in the "**show interface portchannel**" command. So, it is not recommended to use.

3.2.1.2 Adding member ports to port channels

Follow the steps below to add a member port to the already created port channel. Maximum 8

members shall be added to a portchannel.

Step	Command	Description
Step 1	**config portchannel member add <portchannel_name> <member_portname>**	Add a member port in the already created port channel portchannel_name - In the format "PortChannelxxxx", where "xxxx" is number of 1 to 4 digits. Ex: "PortChannel0001" member_portname - any interface name
Step 2	**show interfaces portchannel**	Displays the configured port channel information
Step 3	**sudo config save -y**	Optional step - saves this configuration to be part of startup configuration.

Only ports of same speed can be added to port channel interfaces.

The IP addresses must be removed before adding the port to the portchannel.

The following example shows how to add a member port in a port channel:

Add an interface Ethernet48 as a member port in PortChannel0001

```
admin@sonic: ~$ sudo config portchannel member add PortChannel0001 Ethernet48
```

When there is no LACP in peer

```
admin@sonic: ~$ show interface portchannel
Flags: A - active, I - inactive, Up - up, Dw - Down, N/A - not available,
       S - selected, D - deselected, * - not synced
No.    Team Dev            Protocol       Ports
-----  ------------------  -----------    -------
0001   PortChannel0001  LACP(A)(Dw)   Ethernet48(D)
admin@sonic: ~$ sudo config portchannel member add PortChannel0001 Ethernet52
```

When there is LACP in Peer

```
admin@sonic: ~$ show interface portchannel
Flags: A - active, I - inactive, Up - up, Dw - Down, N/A - not available,
       S - selected, D - deselected, * - not synced
No.    Team Dev            Protocol       Ports
-----  ------------------  -----------    -------
```

```
0001   PortChannel0001   LACP(A)(Dw)  Ethernet52(D) Ethernet48(D)
```

3.2.2 Remove Member Ports from a port channel

Follow the steps below to remove a member port from a port channel

Step	Command	Description
Step 1	**config portchannel member del <portchannel_name> <member_portname>**	Remove member port from a port channel portchannel_name - In the format "PortChannelxxxx", where "xxxx" is number of 1 to 4 digits. Ex: "PortChannel0001" member_portname – any interface name
Step 2	**show interfaces portchannel**	Displays the configured port channel information
Step 3	**sudo config save -y**	Optional step - saves this configuration to be part of startup configuration.

The following examples show how to delete a member port from a port channel:

Delete an interface Ethernet48 from a port channel PortChannel0001

```
admin@sonic: ~$ sudo config portchannel member del PortChannel0001 Ethernet48
admin@sonic: ~$ sudo config portchannel member del PortChannel0001 Ethernet52
admin@sonic: ~$ show interface portchannel
Flags: A - active, I - inactive, Up - up, Dw - Down, N/A - not available,
       S - selected, D - deselected, * - not synced
No.    Team Dev            Protocol        Ports
-----  ------------------  ------------    -------
0001   PortChannel0001   LACP(A)(Dw)
```

3.2.3 Removing Port channels

Follow the steps below to remove port channels from Sonic.

Step	Command	Description
Step 1	**config portchannel del <portchannel_name>**	Deletes a port channel interface

		portchannel_name - In the format "PortChannelxxxx", where "xxxx" is number of 1 to 4 digits. Ex: "PortChannel0001"
Step 2	**show interfaces portchannel**	Displays the configured port channel interfaces
Step 3	**sudo config save -y**	Optional step - saves this configuration to be part of startup configuration.

The following example shows how to remove a port channel:

Delete a port channel PortChannel0001

```
admin@sonic: ~$ sudo config portchannel del PortChannel0001
admin@sonic: ~$ show interface portchannel
Flags: A - active, I - inactive, Up - up, Dw - Down, N/A - not available,
       S - selected, D - deselected, * - not synced
No.    Team Dev            Protocol        Ports
-----  ------------------------  -----------  -------
```

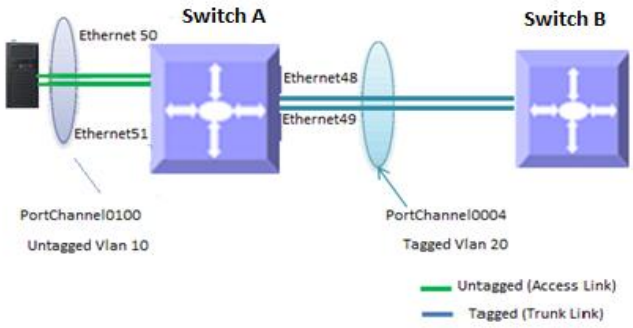
The port channel cannot be deleted when there is a member in it. Remove the member ports before deleting a port channel

3.2.4 Link Aggregation Configuration Example

Configure the SONiC switch as shown below in Figure LA-2.

1. Aggregate ports Ethernet48 and Ethernet49 in a port channel PortChannel0004. Also configure this aggregation as a tagged interface with VLAN 20.
2. Aggregate ports Ethernet50 and Ethernet51 in a port channel PortChannel0100. Configure this aggregation as an untagged port on VLAN 10.

Figure LA-2: Link Aggregation Configuration Example



#Create VLAN's 10 and 20 First

```
admin@sonic: ~$ sudo config vlan add 10
admin@sonic: ~$ sudo config vlan add 20
```

#Create Port channel PortChannel0004 and add Ethernet48, Ethernet49 as a member

```
admin@sonic: ~$ sudo config portchannel add PortChannel0004
admin@sonic: ~$ sudo config portchannel member add PortChannel0004 Ethernet48
admin@sonic: ~$ sudo config portchannel member add PortChannel0004 Ethernet49
```

#Add port channel PortChannel0100 as a tagged member for VLAN 20

```
admin@sonic: ~$ sudo config vlan member add 20 PortChannel0004
```

#Create Port channel PortChannel0100 and add Ethernet50, Ethernet51 as a member

```
admin@sonic: ~$ sudo config portchannel add PortChannel0100
```

```
admin@sonic: ~$ sudo config portchannel member add PortChannel0100 Ethernet50
admin@sonic: ~$ sudo config portchannel member add PortChannel0100 Ethernet51
```

#Add port channel PortChannel0004 as an untagged member for VLAN 10

```
admin@sonic: ~$ sudo config vlan member add -u 10 PortChannel0100
admin@sonic: ~$ show interface portchannel
Flags: A - active, I - inactive, Up - up, Dw - Down, N/A - not available,
       S - selected, D - deselected, * - not synced
No.    Team Dev            Protocol       Ports
-----  -------------------  -----------    -------
0001   PortChannel0001   LACP(A)(Dw)
0004   PortChannel0004   LACP(A)(Dw)   Ethernet48(D) Ethernet49(D)
0100   PortChannel0100   LACP(A)(Dw)   Ethernet50(D) Ethernet51(D)
admin@sonic: ~$ show vlan config
Name      VID   Member              Mode
-------   -----  ----------------------  ---------
Vlan10    10    Ethernet1           untagged
Vlan10    10    Ethernet2           untagged
Vlan10    10    Ethernet3           untagged
Vlan10    10    Ethernet5           tagged
Vlan10    10    Ethernet6           tagged
Vlan10    10    Ethernet7           tagged
Vlan10    10    PortChannel0100   untagged
Vlan20    20    Ethernet4            tagged
Vlan20    20    Ethernet5            tagged
Vlan20    20    Ethernet6            tagged
Vlan20    20    Ethernet7            tagged
Vlan20    20    PortChannel0004    tagged
Vlan30    30    Ethernet4            tagged
Vlan30    30    Ethernet5            tagged
Vlan30    30    Ethernet6            tagged
Vlan30    30    Ethernet7             tagged
Vlan100   100
Vlan200   200
admin@sonic: ~$ show vlan brief
+-------------+---------------+------------------------+--------------------+---------------+-----------------------------+
|  VLAN ID | IP Address  | Ports                 | Port Tagging    | Proxy ARP  | DHCP Helper Address  |
+========+==========+================+=============+==========+====================+
|      10   |               | Ethernet1          | untagged         | disabled    |                          |
|           |               | Ethernet2          | untagged         |               |                          |
|           |               | Ethernet3          | untagged         |               |                          |
```

```
|           |                | Ethernet5          | tagged             |               |                           |
|           |                | Ethernet6          | tagged             |               |                           |
|           |                | Ethernet7          | tagged             |               |                           |
|           |                | PortChannel0100    | untagged           |               |                           |
+-----------+----------------+--------------------+--------------------+---------------+---------------------------=+
|      20   |                | Ethernet4          | tagged             | disabled      |                           |
|           |                | Ethernet5          | tagged             |               |                           |
|           |                | Ethernet6          | tagged             |               |                           |
|           |                | Ethernet7          | tagged             |               |                           |
|           |                | PortChannel0004    | tagged             |               |                           |
+-----------+----------------+--------------------+--------------------+---------------+---------------------------+
|      30   |                | Ethernet4          | tagged             | disabled      |                           |
|           |                | Ethernet5          | tagged             |               |                           |
|           |                | Ethernet6          | tagged             |               |                           |
|           |                | Ethernet7          | tagged             |               |                           |
+-----------+----------------+--------------------+--------------------+---------------+---------------------------+
|     100   |                |                    |                    | disabled      |                           |
+-----------+----------------+--------------------+--------------------+---------------+---------------------------+
|     200   |                |                    |                    | disabled      |                           |
+-----------+----------------+--------------------+--------------------+---------------+---------------------------+
```

3.3 LLDP

3.3.1 LLDP Overview

LLDP is a neighbor discovery protocol that is used for network devices to advertise information about themselves to other devices on the network. This protocol runs over the data-link layer, which allows two systems running different network layer protocols to learn about each other.

Devices in a LAN maintain operations-related configuration information in management information bases (MIBs). LLDP helps avoid misconfiguration problems in LANs by enabling LAN devices to be aware of other devices' configuration information.

LLDP supports a set of attributes that it uses to discover neighbor devices. These attributes contain type, length, and value descriptions and are referred to as TLVs. LLDP supported devices can use TLVs to receive and send information to their neighbors. Details such as configuration information, device capabilities, and device identity can be advertised using LLDP.

3.3.2 LLDP Configuration

3.3.2.1 Default Configuration

Parameter	Default Value
LLDP Status	Enabled
LLDP PDU interval	30 secs

3.3.2.2 Disable LLDP

There is no SONiC command to disable LLDP. However, LLDP can be disabled using the below command.

Step	Command	Description
Step 1	**systemctl disable lldp**	Disables LLDP
Step 2	**show lldp table**	Displays the LLDP neighbors in tabular format
Step 3	**show lldp neighbors**	Displays the LLDP neighbors

Disable LLDP

```
admin@sonic: ~$ sudo systemctl disable lldp
Removed /etc/systemd/system/sonic.target.wants/lldp.service.
admin@sonic: ~$ show lldp table
Capability codes: (R) Router, (B) Bridge, (O) Other
LocalPort    RemoteDevice    RemotePortID    Capability    RemotePortDescr
-------------  --------------------  ------------------  -------------  ----------------------
------------------------------------------------------------------
Total entries displayed:  1
admin@sonic: ~$ show lldp neighbors
------------------------------------------------------------------------------
LLDP neighbors:
------------------------------------------------------------------------------
Interface:    Ethernet0, via: LLDP, RID: 3, Time: 0 day, 00:03:33
  Chassis:
     Chasid:  mac 0c:c4:7a:f7:d0:d5
  Port:
     PortID:    ifalias Gi0/41
     TTL:       120
------------------------------------------------------------------------------
```

3.3.2.3 Enable LLDP

LLDP is enabled by default in SONiC switch. There is no specific SONiC Command to enable it. However, it can be enabled by using the below command.

Step	Command	Description
Step 1	**systemctl enable lldp**	Enables LLDP
Step 2	**show lldp table**	Displays the LLDP neighbors in tabular format
Step 3	**show lldp neighbors**	Displays the LLDP neighbors

Enable LLDP

```
admin@sonic: ~$ sudo systemctl enable lldp
admin@sonic: ~$ show lldp table
Capability codes: (R) Router, (B) Bridge, (O) Other
LocalPort    RemoteDevice    RemotePortID    Capability    RemotePortDescr
-----------  --------------  --------------  ------------  ---------------
Ethernet0                    Gi0/41
------------------------------------------------------------
Total entries displayed:  1
admin@sonic: ~$ show lldp neighbors
------------------------------------------------------------------------------
LLDP neighbors:
------------------------------------------------------------------------------
Interface:    Ethernet0, via: LLDP, RID: 3, Time: 0 day, 00:05:03
  Chassis:
    ChassisID:    mac 0c:c4:7a:f7:d0:d5
  Port:
      PortID:     ifalias Gi0/41
      TTL:        120
------------------------------------------------------------------------------
```

3.3.2.4 Start LLDP service

LLDP Service can be started using the below command

Step	Command	Description

Step 1	**systemctl start lldp**	Starts LLDP service
Step 2	**show lldp table**	Displays the LLDP neighbors in tabular format
Step 3	**show lldp neighbors**	Displays the LLDP neighbors

Start LLDP Service

```
admin@sonic: ~$ sudo systemctl start lldp
admin@sonic: ~$ show lldp table
Capability codes: (R) Router, (B) Bridge, (O) Other
LocalPort    RemoteDevice    RemotePortID    Capability    RemotePortDescr
-----------  --------------  --------------  ------------  -----------------------
Ethernet0                    Gi0/41
------------------------------------------------------------
Total entries displayed:  1
admin@sonic: ~$ show lldp neighbors
--------------------------------------------------------------------------------
LLDP neighbors:
--------------------------------------------------------------------------------
Interface:    Ethernet0, via: LLDP, RID: 3, Time: 0 day, 00:08:32
  Chassis:
    ChassisID:    mac 0c:c4:7a:f7:d0:d5
  Port:
    PortID:       ifalias Gi0/41
    TTL:          120
--------------------------------------------------------------------------------
```

3.3.2.5 Stop LLDP service

LLDP Service can be stopped using the below command

Step	Command	Description
Step 1	**systemctl stop lldp**	Stops LLDP service
Step 2	**show lldp table**	Displays the LLDP neighbors in tabular format
Step 3	**show lldp neighbors**	Displays the LLDP neighbors

Stop LLDP Service

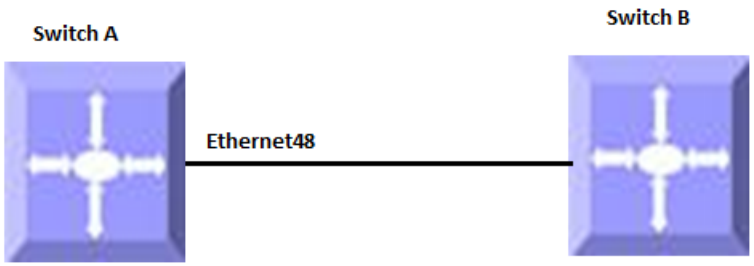
```
admin@sonic: ~$ sudo systemctl stop lldp
admin@sonic: ~$ show lldp table
Error response from daemon: Container 852441f50ba91588ab5d4e1803feb583c0c9bb6d31c11e73bc4c18a9b9578ede is not running

admin@sonic: ~$ show lldp neighbors
Error response from daemon: Container 852441f50ba91588ab5d4e1803feb583c0c9bb6d31c11e73bc4c18a9b9578ede is not running
```

3.3.3 LLDP Configuration Example

The example below shows the below configuration in the LLDP enabled Sonic switches connected by a port in between.

1. Stop LLDP service
2. Start LLDP service

Figure LLDP-1: LLDP Configuration Example



#Stop the lldp service

```
admin@sonic: ~$ sudo systemctl stop lldp
admin@sonic: ~$ show lldp table
Error response from daemon: Container 852441f50ba91588ab5d4e1803feb583c0c9bb6d31c11e73bc4c18a9b9578ede is not running
admin@sonic: ~$ show lldp neighbors
```

```
Error response from daemon: Container 852441f50ba91588ab5d4e1803feb583c0c9bb6d31c11e73bc4c18a9b9578ede is not running
```

#Start the lldp service

```
admin@sonic: ~$ sudo systemctl start lldp
admin@sonic: ~$ show lldp table
Capability codes: (R) Router, (B) Bridge, (O) Other
LocalPort    RemoteDevice    RemotePortID    Capability    RemotePortDescr
-----------  ----------------  ----------------  ----------  ----------------------
Ethernet0                      Gi0/41
------------------------------------------------------------------
Total entries displayed:  1
admin@sonic: ~$ show lldp neighbors
--------------------------------------------------------------------------------
LLDP neighbors:
--------------------------------------------------------------------------------
Interface:    Ethernet0, via: LLDP, RID: 1, Time: 0 day, 00:00:21
  Chassis:
    ChassisID:    mac 0c:c4:7a:f7:d0:d5
  Port:
    PortID:       ifalias Gi0/41
    TTL:          120
--------------------------------------------------------------------------------
```

After starting LLDP service, the switch would take few seconds to exchange LLDP packets with it's neighbor and show the neighbors in the LLDP neighbor table.

4 Layer3 Configuration

Internet Protocol (IP), the foundation of the IP protocol suite, is a packet-based protocol used for the exchange of data over computer networks. IP is a network layer that contains addressing and control information to allow routing of data packets. IP handles addressing, fragmentation, reassembly, and protocol de-multiplexing.
Supermicro switches support both TCP and UDP at the transport layer for maximum flexibility in services.

- Transmission Control Protocol (TCP) is a connection-oriented protocol built upon the IP layer. TCP specifies the format of data and acknowledgments used in the transfer of data and also the procedures used to ensure that the data arrives in correct order. With TCP, multiple applications on a system can communicate concurrently as it handles all de-multiplexing of the incoming traffic among the application programs.

- With UDP, applications can send messages (also called datagrams) to other hosts on an IP network without prior setup of transmission channels or data paths. UDP is suitable when error checking and correction is either not necessary or performed in the application, avoiding the overhead of such processing at the network interface level.

4.1 DHCP Relay

In small networks with only one IP subnet, DHCP clients can communicate directly with DHCP servers. In large networks, DHCP servers provide IP addresses for multiple subnets. In such cases, a DHCP client that has not yet obtained an IP address from the DHCP server cannot communicate with the DHCP server using IP routing.

A DHCP relay agent forwards DHCP packets between clients and servers when they are not on the same physical subnet. The relay agent receives the broadcast from the DHCP client and unicasts it to one or more DHCP servers.

DHCP VLAN Relay can be applied in a scenario where a DHCP server is deployed to offer IP addresses to clients in multiple VLANs. These VLANs do not have VLAN interfaces.

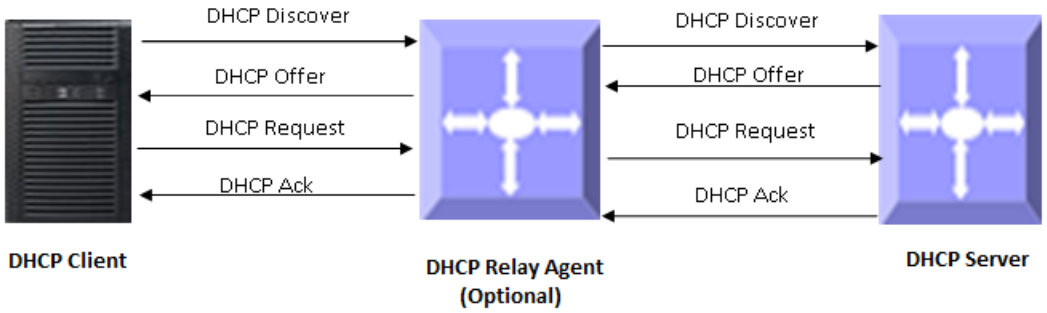


Figure IP-1: DHCP Relay Agent

DHCP VLAN Relay can manually designate an L3 interface for all the VLANs as the default relay agent interface. All the DHCP packets can be forwarded through this interface so that the clients can get IP addresses from the DHCP Server.

This document assumes that the DHCP client and DHCP Server, which are beyond the scope of this document, are configured and ready.

4.1.1 IPv4 DHCP Relay

4.1.1.1 Add DHCP Relay Destination IP address (es) for a VLAN interface

Follow the steps below to add the DHCP Relay Destination IP address (es) for a VLAN interface. Note that more than one DHCP Relay Destination IP address can be added on a VLAN interface.

Step	Command	Description
Step 1	**config feature state dhcp_relay enabled**	Enable the dhcp relay.
Step 2	**config vlan dhcp_relay add <vlan_id> <dhcp_relay_destination_ip>**	Add a DHCP Relay Destination IP address to the VLAN vlan_id - may be any vlan number dhcp_relay_destination_ip - IPv4 address
Step 3	**show vlan brief**	Displays the configured VLAN information.
Step 4	**sudo config save -y**	Optional step - saves this configuration to be part of startup configuration.

The following example shows how to add the DHCP Relay Destination IPv4 address for a VLAN

```
Add an IP 192.168.200.20 as a DHCP Relay address for VLAN 100
admin@sonic: ~$ sudo config feature state dhcp_relay enabled
admin@sonic: ~$ sudo config vlan dhcp_relay add 100 192.168.200.20
admin@sonic: ~$ show vlan brief
+-------------+----------------+----------+-----------------+----------------+----------------------------+
|  VLAN ID    | IP Address     | Ports    | Port Tagging    | Proxy ARP      | DHCP Helper Address        |
+=========+==========+======+============+==========+====================+
|     100     |192.168.100.1|Ethernet0|Untagged|   disabled    |     192.168.200.20       |
+-------------+----------------+----------+-----------------+----------------+----------------------------+

Add another IP 192.168.200.22 as a DHCP Relay address for VLAN 100
admin@sonic: ~$ sudo config vlan dhcp_relay add 100 192.168.200.22
admin@sonic: ~$ show vlan brief
+-------------+----------------+----------+-----------------+----------------+----------------------------+
|  VLAN ID    | IP Address     | Ports    | Port Tagging    | Proxy ARP      | DHCP Helper Address        |
+=========+==========+======+============+==========+====================+
|     100     |192.168.100.1| Ethernet0|Untagged |   disabled    |     192.168.200.20       |
|             |                |          |                 |                |     192.168.200.22       |
+-------------+----------------+----------+-----------------+----------------+----------------------------+
```

4.1.1.2 Remove DHCP Relay Destination IP address (es) from a VLAN interface

Follow the steps below to remove the DHCP Relay Destination IP address (es) from a VLAN interface.

Step	Command	Description
Step 1	**config vlan dhcp_relay del <vlan_id> <dhcp_relay_destination_ip>**	Delete a configured DHCP Relay Destination IP address from a VLAN interface vlan_id - may be any vlan number dhcp_relay_destination_ip - IPv4 address
Step 2	**show vlan brief**	Displays the configured VLAN information.
Step 3	**sudo config save -y**	Optional step - saves this configuration to be part of startup configuration.

The following example shows how to remove the DHCP Relay Destination IPv4 address for a VLAN

Remove DHCP Relay address 192.168.200.22 from VLAN 100

```
admin@sonic: ~$ sudo config vlan dhcp_relay del 100 192.168.200.22
admin@sonic: ~$ show vlan brief
+-----------+---------------+---------+----------------+---------------+----------------------------+
|  VLAN ID  | IP Address    | Ports   | Port Tagging   | Proxy ARP     | DHCP Helper Address        |
+=========+==========+======+============+==========+====================+
|     100   |192.168.100.1| Ethernet0|Untagged |   disabled    |     192.168.100.20         |
+-----------+---------------+---------+----------------+---------------+----------------------------+
```

4.1.2 IPv6 DHCP Relay

4.1.2.1 Add DHCP Relay Destination IP address (es) for a VLAN interface

Follow the steps below to add the DHCP Relay Destination IP address (es) for a VLAN interface. Note that more than one DHCP Relay Destination IP address can be added on a VLAN interface.

Step	Command	Description
Step 1	**config feature state dhcp_relay enabled**	Enable the dhcp relay.
Step 2	**config vlan dhcp_relay add <vlan_id> <dhcp_relay_destination_ip>**	Add a DHCP Relay Destination IP address to the VLAN vlan_id - may be any vlan number dhcp_relay_destination_ip – IPv6 address
Step 3	**show vlan brief**	Displays the configured VLAN information.
Step 4	**sudo config save -y**	Optional step - saves this configuration to be part of startup configuration.

The following example shows how to add the DHCP Relay Destination IPv6 address for a VLAN.

Add an IP 2001:192:168:20::120 as a DHCP Relay address for VLAN 10.

```
admin@sonic: ~$ sudo config feature state dhcp_relay enabled
admin@sonic: ~$ sudo config vlan dhcp_relay add 10 2001:192:168:20::120
admin@sonic: ~$ sudo config dhcp_relay ipv6 destination add 10 2001:192:168:20::120
admin@sonic: ~$ show dhcprelay_helper ipv6
```

```
+------------+-----------------------------+
|Interface | DHCP Relay Address   |
+========+==================+
|Vlan10    |2001:192:168:20::120|
+-------------+-----------------------------+

admin@sonic: ~$ show dhcp_relay ipv6 destination
---------  -----------------------------
Vlan10  2001:192:168:20::120
--------- -----------------------------
```

4.1.2.2 Remove DHCP Relay Destination IP address (es) from a VLAN interface

Follow the steps below to remove the DHCP Relay Destination IP address (es) from a VLAN interface.

Step	Command	Description
Step 1	**config vlan dhcp_relay del <vlan_id> <dhcp_relay_destination_ip>**	Delete a configured DHCP Relay Destination IP address from a VLAN interface vlan_id - may be any vlan number dhcp_relay_destination_ip – IPv6 address
Step 2	**show vlan brief**	Displays the configured VLAN information.
Step 3	**sudo config save -y**	Optional step - saves this configuration to be part of startup configuration.

The following example shows how to remove the DHCP Relay Destination IPv6 address for a VLAN.

Remove DHCP Relay address 2001:192:168:20::120 from VLAN 10

```
admin@sonic: ~$ sudo config vlan dhcp_relay del 10 2001:192:168:20::120
```

4.2 Layer3 VLAN Interface

VLANs typically operate at Layer2. When a Layer2 VLAN is configured with an IP address, it behaves as a logical Layer3 VLAN interface.

A Layer3 VLAN interface provides logical routing interfaces to VLANs on Layer2 switches.

It is also called a Switch Virtual Interface (SVI) and handles processing for all the packets associated with that VLAN.

4.2.1 Add an IP address for a VLAN interface

Follow the steps below to add IP address for a VLAN interface.

Step	Command	Description
Step 1	**config interface ip add Vlan<vlan_id> <ip_addr>**	Add an IP address for a VLAN. vlan_id - may be any vlan number ip_addr - ip address
Step 2	**show vlan brief**	Displays the configured VLAN information.
Step 3	**show ip interface**	Displays IP Address of all interfaces.
Step 4	**sudo config save -y**	Optional step - saves this configuration to be part of startup configuration.

The following examples show how to add IP address for a VLAN:

Add an IP 192.168.100.10 for VLAN 100

```
admin@sonic: ~$ sudo config interface ip add Vlan100 192.168.100.10 /24
admin@sonic: ~$ show vlan brief
+-----------+-------------------------+---------+-----------------+----------------+-----------------------------+
|  VLAN ID  |       IP Address        | Ports   | Port Tagging    | Proxy ARP      | DHCP Helper Address         |
+===========+=========================+=========+=================+================+=============================+
|    100    | 192.168.100.10/24       |         |                 | disabled       |    192.168.100.20           |
+-----------+-------------------------+---------+-----------------+----------------+-----------------------------+
admin@sonic: ~$
admin@sonic: ~$ show ip interface (Note: Truncated output is added here)
Interface    Master    IPv4 address/mask    Admin/Oper    BGP Neighbor    Neighbor IP
-----------  --------  -------------------  ------------  --------------  -------------
Ethernet53             10.0.0.106/31        up/up         ARISTA27T0      10.0.0.107
Loopback0              10.1.0.1/32          up/up         N/A             N/A
Vlan100                192.168.100.10/24    up/down       N/A             N/A
docker0                240.127.1.1/24       up/down       N/A             N/A
lo                     127.0.0.1/16         up/up         N/A             N/A
```

```
admin@sonic: ~$
```

The command to add IP address to non-existing VLAN interface fails silently without explicit error message.

4.2.2 Remove an IP address from a VLAN interface

Follow the steps below to delete an IP address from a VLAN interface.

Step	Command	Description
Step 1	**config interface ip remove Vlan<vlan_id> <ip_addr>**	Remove an IP address for a VLAN. vlan_id - may be any vlan number ip_addr - ip address
Step 2	**show vlan brief**	Displays the configured VLAN information.
Step 3	**show ip interface**	Displays IP Address of all interfaces.
Step 4	**sudo config save -y**	Optional step - saves this configuration to be part of startup configuration.

The following examples show how to delete an IP address from a VLAN:

Remove an IP 192.168.100.10 from VLAN 100

```
admin@sonic: ~$ sudo config interface ip remove Vlan100 192.168.100.10/24
admin@sonic: ~$ show vlan brief
+-----------+-------------+---------+----------------+-------------+-----------------------+
| VLAN ID   | IP Address  | Ports   | Port Tagging   | Proxy ARP   | DHCP Helper Address   |
+===========+=============+=========+================+=============+=======================+
|    100    |             |         |                |  disabled   |    192.168.100.20     |
+-----------+-------------+---------+----------------+-------------+-----------------------+
admin@sonic: ~$
admin@sonic: ~$ show ip interface (Note: Truncated output is added here)
Interface    Master   IPv4 address/mask    Admin/Oper    BGP Neighbor    Neighbor IP
-----------  -------  -------------------  ------------  --------------  -------------
Ethernet53            10.0.0.106/31        up/up         ARISTA27T0      10.0.0.107
Loopback0             10.1.0.1/32          up/up         N/A             N/A
```

docker0	240.127.1.1/24	up/down	N/A	N/A
lo	127.0.0.1/16	up/up	N/A	N/A
admin@sonic: ~$				

The attempt to remove a wrong or non-existing IP address from an interface fails silently without explicit error message.

4.2.3 Inter-VLAN Routing

VLANs enable splitting traffic across several manageable broadcast domains. Devices within a VLAN can communicate with one another without requiring routing. Whenever hosts in one VLAN need to communicate with hosts in another VLAN, the traffic must be routed between them. This is known as Inter-VLAN Routing.

By default, all interfaces are assigned IPv4 address. Only L2 interface can be added as VLAN member.

Application of Inter-VLAN routing:

The network can be divided based on the group or function of its devices. For example, an engineering department VLAN would only have devices associated with the engineering department, while an HR VLAN would only have HR related devices. With Inter-VLAN routing, the devices in each VLAN can talk to one another without all the devices being in the same broadcast domain.

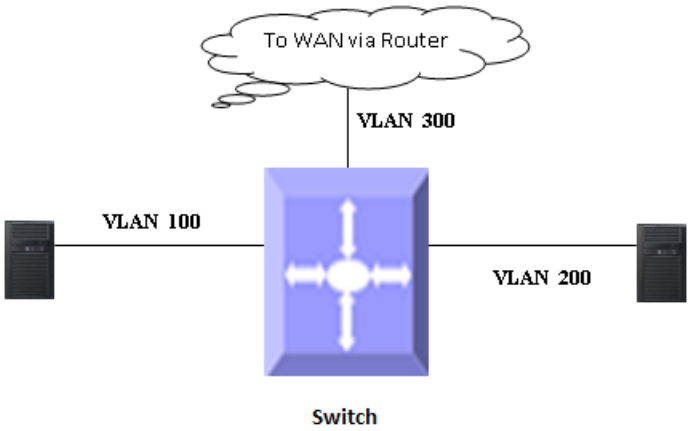


Figure IP-2: Inter – VLAN Routing

Follow the steps below to configure Inter-VLAN routing.

1. Create two VLANs and added an Ethernet48 in VLAN 100 and Ethernet52 in VLAN 200.
2. Configure an IP address for both VLANs.
3. Execute show ip route to check if the VLAN routes specified by VLAN IP address are displayed as connected routes.

Create VLAN's and add the member ports

```
admin@sonic: ~$ sudo config vlan add 100
admin@sonic: ~$ sudo config vlan add 200
admin@sonic: ~$ sudo config interface ip remove Ethernet50 10.0.0.100/31
admin@sonic: ~$ sudo config interface ip remove Ethernet52 10.0.0.104/31
admin@sonic: ~$ sudo config vlan member add 100 Ethernet50
admin@sonic: ~$ sudo config vlan member add 200 Ethernet52
```

Configure IP address for both the VLAN's

```
admin@sonic: ~$ sudo config interface ip add Vlan100 192.168.100.30
admin@sonic: ~$ sudo config interface ip add Vlan200 192.168.100.40
admin@sonic: ~$ show ip interface (Note: Truncated output is added here)
Interface    Master    IPv4 address/mask    Admin/Oper    BGP Neighbor    Neighbor IP
-----------  --------  -------------------  ------------  --------------  -------------
Ethernet53             10.0.0.106/31        up/up         ARISTA27T0      10.0.0.107
Loopback0              10.1.0.1/32          up/up         N/A             N/A
Vlan100                192.168.100.30/32    up/up         N/A             N/A
Vlan200                192.168.100.40/32    up/up         N/A             N/A
docker0                240.127.1.1/24       up/down       N/A             N/A
lo                     127.0.0.1/16         up/up         N/A             N/A
admin@sonic: ~$
admin@sonic: ~$ show vlan brief
+-----------+-------------------+-----------+----------------+-------------+----------------------+
| VLAN ID   | IP Address        | Ports     | Port Tagging   | Proxy ARP   | DHCP Helper Address  |
+===========+===================+===========+================+=============+======================+
| 100       | 192.168.100.30/32 | Ethernet50 | tagged        | disabled    |                      |
| 200       | 192.168.100.40/32 | Ethernet52 | tagged        | disabled    |                      |
+-----------+-------------------+-----------+----------------+-------------+----------------------+
admin@sonic: ~$ show ip route (Note: Truncated output is added here)
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
```

```
      F - PBR, f - OpenFabric,
      > - selected route, * - FIB route, q - queued, r - rejected, b - backup

C>* 10.0.0.0/31 is directly connected, Ethernet0, 01:16:59
C>* 10.0.0.106/31 is directly connected, Ethernet53, 01:16:47
C>* 10.1.0.1/32 is directly connected, Loopback0, 01:17:11
C>* 192.168.86.0/24 is directly connected, Ethernet0, 01:16:59
C>* 192.168.100.0/24 is directly connected, Ethernet9, 00:57:03
C>* 192.168.100.30/32 is directly connected, Vlan100, 00:04:24
C>* 192.168.100.40/32 is directly connected, Vlan200, 00:04:12
admin@sonic: ~$
```

4.3 Static route

A Static route defines an explicit path between two routers. Manual reconfiguration of static routes is required whenever network changes occur. Static routes use less bandwidth than dynamic routes. No CPU cycles are used to calculate and analyze routing updates.

Routers forward packets using either route information from manually configured route table entries or by using the route information calculated with dynamic routing algorithms.

Use of Static Routes:

- Static routes can be used in environments where network traffic is predictable and the network design is simple.
- Static routes are also useful for specifying a gateway of last resort (a default router to which all non-routable packets are sent).

Follow the steps below to configure a static route

Step	Command	Description
Step 1	**config route add prefix <A.B.C.D/M> nexthop <dev <dev_name>>**	Add a static route A.B.C.D/M - ip address with subnet mask dev_name - any interface name
Step 2	**show ip route**	Displays the configured route information
Step 3	**sudo config save -y**	Optional step - saves this configuration to be part of startup configuration.

sudo config route del prefix <A.B.C.D/M> nexthop <dev <dev_name>> command deletes the static route

The following example shows the commands used to configure a static route.

```
admin@sonic: ~$ sudo config vlan add 100
admin@sonic: ~$ sudo config vlan member add 100 Ethernet48
admin@sonic: ~$ sudo config interface ip add Vlan100 192.168.100.30
admin@sonic: ~$ sudo config route add prefix 192.168.200.1/24 nexthop dev Vlan100
admin@sonic: ~$ show ip route (Note: Truncated output is added here)
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
C>* 10.0.0.0/31 is directly connected, Ethernet0, 00:07:12
C>* 10.0.0.106/31 is directly connected, Ethernet53, 00:06:54
C>* 10.1.0.1/32 is directly connected, Loopback0, 00:07:22
C>* 192.168.86.0/24 is directly connected, Ethernet0, 00:07:12
S>* 192.168.200.0/24 [1/0] is directly connected, Vlan100, weight 1, 00:05:41
C>* 192.168.100.30/32 is directly connected, Vlan100, 00:06:55
```

Configuring static routes via both SONiC CLI and FRRouting must be avoided. Configuring static routes in both SONiC CLI and FRRouting will cause conflicts and some routes may not be installed.

4.4 ARP

The Address Resolution Protocol (ARP) feature finds the hardware address, also known as the Media Access Control (MAC) address, of a host from its known IP address. This mapping of MAC addresses to IP addresses is stored in a table called the *ARP cache*.

Follow the steps below to display arp table

Step	Command	Description
Step 1	**show arp [OPTIONS] [IPADDRESS]**	Displays the arp table Options:

		-if, --iface - Interface name IP ADDRESS - IPv4 address

The following example shows the ARP entries in ARP table.

```
admin@sonic: ~$ show arp
Address            MacAddress           Iface         Vlan
-----------------  -------------------  ------------  -------
192.168.86.1       28:bd:89:25:3e:0a        eth0        -
Total number of entries 1
admin@sonic: ~$ show arp -if eth0
Address            MacAddress           Iface         Vlan
-----------------  -------------------  ------------  -------
192.168.86.1       28:bd:89:25:3e:0a        eth0        -
Total number of entries 1
admin@sonic: ~$ show arp -iface eth0
Address            MacAddress           Iface         Vlan
-----------------  -------------------  ------------  -------
192.168.86.1       28:bd:89:25:3e:0a        eth0        -
Total number of entries 1
```

The ARP entries are listed only for Layer3 interfaces; no ARP entries will be displayed for the Layer2 switch port. For example the port Ethernet1 is a layer2 port part of VLAN 11 and zero ARP entries listed for that port.

```
root@test:~# show arp
Address        MacAddress        Iface     Vlan
-------------  -----------------  ---------  ------
7.7.1.2        0c:c4:7a:14:fd:4e  Ethernet0  10
7.7.2.2        0c:c4:7a:15:0f:ae  Ethernet1  11
7.7.3.3        0c:c4:7a:15:0f:ae  Ethernet1  11
172.30.0.1     00:25:90:01:d4:44  eth0       -
172.30.0.253   3c:ec:ef:48:86:6b  eth0       -
Total number of entries 10
root@test:~# show arp -if Ethernet1
Address    MacAddress    Iface    Vlan
---------  ------------  -------  ------
Total number of entries 0
root@test:~#
```

4.5 BGP

Border Gateway Protocol (BGP) is an inter-domain routing protocol designed to provide loop-free routing links between organizations. BGP is designed to run over a reliable transport protocol using Port 179. BGP is used to connect a local network to an external network in order to access the Internet or to connect to other organizations.

4.5.1 EBGP

EBGP stands for External Border Gateway Protocol. It runs between two BGP routers in different autonomous system. EBGP routes received from an EBGP peer can be advertised to EBGP and IBGP peers. It is used between organization or between organization and Internet Service provider. When connecting to an external organization, external BGP peering sessions are created. In EBGP peers, attributes like local preference are not sent. When route is advertised to EBGP peer, next hop is changed to local router.

4.5.2 IBGP

IBGP stands for Internal Border Gateway Protocol. It runs between two BGP routers in the same autonomous system. IBGP routes received from an IBGP peer cannot be advertised to another IBGP peer but can be advertised to an EBGP peer. It is used within the same organization. BGP peers within the same organization exchange routing information through internal BGP peering sessions. In IBGP peers, attributes like local preference are sent. When route is advertised to IBGP peer, next hop remains unchanged.

4.5.3 Router ID

BGP uses router ID to identify BGP-speaking peers. The BGP router ID is represented by an IPv4 address. The BGP router ID must be unique to the BGP peers in a network.

4.5.4 Speaker and Peer

A peer device is a BGP-speaking router that has an active TCP connection to another BGP-speaking device. BGP devices need not be necessarily directly connected. A BGP speaker is the local router and a peer is any other BGP speaking network device.

When a TCP connection is established between peers, each BGP peer initially exchanges all its routes—the complete BGP routing table with the other peer. After this only incremental updates are sent after a change in network topology or routing policy. Peers exchange special messages called keep alive messages.

4.5.5 Autonomous System (AS)

An autonomous system is a network controlled by a single technical administration entity. In BGP autonomous systems are used in individual routing domains with local routing policies. Each routing domain can support multiple routing protocols. However, each routing protocol is administrated separately. Other routing protocols can dynamically exchange routing information with BGP through redistribution.

4.5.6 Attributes

BGP has a number of complex attributes used to determine a path to a remote network. These attributes allow greater flexibility and enable a complex routing decision to ensure that the path to a remote network is the best possible path. BGP always propagates the best path to any peers. BGP attributes are carried in update packets.

4.5.6.1 Local preference Attribute

If there are multiple exit points from the AS, the local preference attribute is used to select the exit point for a specific route. A higher local preference is always preferred.

4.5.6.2 Next-Hop Attribute

The EBGP next-hop attribute is the IP address that is used to reach the advertising router. For EBGP peers, the next-hop address is the IP address of the connection between the peers. For IBGP, the EBGP next-hop address is carried into the local AS.

4.5.7 Filters

A number of different filter methods control the send and receive of BGP updates. BGP updates can be filtered with route information as a basis, or with communities as a basis. Packets that do not match the configured filters are dropped.

4.5.8 Synchronization

When a BGP router receives information about a network from an IBGP neighbor, it does not use that information until a matching route is learned via an IGP or static route. This is called Synchronization. It also does not advertise that route to an EBGP neighbor unless a matching route is in the routing table. It is recommended to turn off synchronization when all routers in the autonomous system run BGP.

4.5.9 BGP Path selection

When a BGP speaker receives updates from multiple autonomous systems that describe different paths to the same destination, it must choose the single best path for reaching that destination. When chosen, the selected path is entered into the BGP routing table and propagated to its neighbors. The decision is based on the value of attributes that the update contains and other BGP-configurable factors.

1. If the next hop address is reachable, consider it.
2. Prefer the largest local preference attribute.
3. If the local preference is the same, prefer the route this local router originated.
4. Prefer the route with the shortest AS path.
5. If this is equal, prefer the route with the origin set to originated (through BGP); IGP is preferred to EGP followed by incomplete.
6. If the origin codes are the same, prefer the route with the lowest MED.
7. If the MED is the same, prefer EBGP over IBGP.
8. Prefer the closest path.
9. Finally, if all paths are equal, prefer the path with lowest BGP router ID.

4.5.10 Timers

BGP implementation maintains different timers for Peers and Route updates.

- The keep alive interval is the time within which keep alive messages are sent to peers.
- The hold time is the interval after which a peer is declared inactive after not receiving a keep alive message from it.
- Route advertisement interval is the interval between sending BGP routing updates.
- Connection Retry timer is the amount of time to wait before re-opening a TCP connection.
- AS Originate Interval is the interval between two subsequent update messages for internal peers.

4.5.11 BGP Route Reflector

To avoid loops, an IBGP router doesn't advertise the prefix it learnt from one IBGP neighbor to another IBGP neighbor. So, all the IBGP neighbors has to be fully meshed with each other to learn the complete network. But this is not practical in a large IBGP network. If there are X number of IBGP routers, then there will be X * [X-1]/2 IBGP sessions has to be established, which would be a huge administrative overhead. In this case, route reflectors are used.

Route reflector is a way to avoid full mesh between IBGP neighbors, but still get the benefits of full mesh. In route reflector method, a IBGP router is selected to act as route reflector. Other IBGP routers in the network act as route reflector clients. When a route reflector learns a prefix from one of its IBGP neighbor, route reflector advertises the prefix to all it's route reflector clients. For redundancy purposes, more than one router can be configured to act as route reflector.

Route reflector has to adhere to the rules below in advertising the routes while advertising the prefixes.

1. Route reflector can re-advertise the prefixes it learnt from non-RR IBGP neighbors, RR IBGP neighbors and EBGP neighbors to its RR client.
2. Route reflector should not re-advertise the prefixes it learnt from a non-RR IBGP neighbor to other non-RR IBGP neighbors.
3. Route reflector can re-advertise the prefixes it learnt from RR IBGP neighbors to its non-RR IBGP neighbors.

4.5.12 BGP Configuration

This section explains basic BGP configuration commands. For more details, please refer FRRouting document.

4.5.12.1 BGP Default Configuration

Parameter	Default Value
BGP Status	Active
Synchronization	Disabled
Preference	None
Peer	None
Connection retry time	120 seconds

Hold time	180 seconds
Keep alive	60 seconds
Route Advertisement Interval	30 seconds
EBGP Multihop	Disable
AS Number	65100
Router ID	None

Figure IP-3: BGP topology

4.5.12.2 Save the BGP Configuration

This section explains the steps to save the BGP configuration. The BGP routing are handled via FRR module and the configuration has to be saved in two steps.

1) Edit the /etc/config_db.json and in the DEVICE_METADATA section, delete the bgp_asn line and add the docker_routing_config_mode line as shown below. This step is needed only once.

```
        "DEVICE_METADATA": {
            "localhost": {
                "buffer_model": "traditional",
                "default_bgp_status": "up",
                "default_pfcwd_status": "disable",
                "hostname": "sonic",
                "hwsku": "sse_g3748",
                "mac": "0C:C4:7A:2E:1D:6D",
                "platform": "arm64-supermicro_sse_g3748-r0",
                "bgp_asn": "65100",
                "docker_routing_config_mode": "split",
                "type": "not-provisioned"
            }
        },
```

2) Reboot the switch for the above change to take effect.
3) The vtysh command invokes the FRRouting mode. After configure the BGP, the configuration has to be saved within the FRRouting mode and again in the SONiC mode.

Example Config:

```
admin@sonic:~$ sudo -i
root@sonic:~# vtysh
Hello, this is FRRouting (version 7.5.1-sonic).
Copyright 1996-2005 Kunihiro Ishiguro, et al.
sonic# configure terminal
sonic(config)# router bgp 65100
sonic(config-router)# no bgp ebgp-requires-policy
sonic(config-router)# neighbor 10.0.0.2 remote-as 65100
sonic(config-router)# end
```

```
sonic# write
sonic# exit
root@sonic:~# config save -y
```

If the command “no bgp ebgp-requires-policy” is not used, then the routes may be Exchanged with BGP peer without proper policies.

4.5.12.3 Enable BGP

BGP is disabled by default. Follow the steps below to enable BGP.

Step	Command	Description
Step 1	**sudo vtysh**	Enter FRRouting
Step 2	**configure terminal**	Enters the configuration mode
Step 3	**router bgp <AS no (1-65535)>**	Enable BGP and configure the AS number of the BGP Speaker
Step 4	**End**	Exits Configuration mode.

“no router bgp” command disables BGP in the switch.

4.5.12.4 BGP Peer

Follow the steps below to configure BGP Peer.

Step	Command	Description
Step 1	**sudo vtysh**	Enter FRRouting
Step 2	**configure terminal**	Enters the configuration mode
Step 3	**router bgp <AS no (1-65535)>**	Enable BGP and configure the AS number of the BGP Speaker
Step 4	**bgp router-id <bgp router id (ip-address)>**	Configures the BGP Identifier of the BGP Speaker.
Step 5	**neighbor <ip-address> remote-as <AS no (1-65535)>**	Creates a Peer and initiates the connection to the peer.
Step 6	**neighbor <ip-address> {advertisement-interval <0-600 seconds>}**	(Optional) Configures neighbor interval.
Step 7	**neighbor <ip-address> timers {keepalive <seconds> | holdtime <seconds>}**	(Optional) Configures neighbor KeepAlive Time and Hold Time Intervals

Step 8	**Exit**	Exits BGP Router Mode
Step 9	**End**	Exits Configuration mode.

no neighbor <ip-address>
no neighbor <ip-address> {advertisement-interval}
no neighbor <ip-address> timers {keepalive | holdtime}

4.5.12.5 Attributes

Follow the steps below to configure BGP Attributes.

Step	Command	Description
Step 1	**sudo vtysh**	Enter FRRouting
Step 2	**configure terminal**	Enters the configuration mode
Step 3	**router bgp <AS no (1-4294967295)>**	Enable BGP and configure the AS number of the BGP Speaker
Step 4	bgp router-id <bgp router id (ip-address)>	Configures the BGP Identifier of the BGP Speaker.
Step 5	neighbor <ip-address> remote-as <AS no (1-4294967295)>	Creates a Peer and initiates the (Optional) connection to the peer.
Step 6	bgp default local-preference <local Pref Value 0-4294967295>	(Optional) Configures the Default Local Preference value.
Step 7	**neighbor <ip-address> ebgp-multihop**	(Optional) Enables BGP to establish connection with external peers that are not directly connected
Step 8	**Exit**	Exits BGP Router Mode
Step 9	**End**	Exits Configuration mode.

no bgp default local-preference
no neighbor <ip-address> ebgp-multihop

4.5.12.6 Network

Follow the steps below to configure Network through BGP.

Step	Command	Description
Step 1	**sudo vtysh**	Enter FRRouting
Step 2	**configure terminal**	Enters the configuration mode
Step 3	**router bgp <AS no (1-65535)>**	Enable BGP and configure the AS number of the BGP Speaker
Step 4	**no bgp ebgp-requires-policy**	Disable Require policy on EBGP

Step 5	**address-family ipv4 unicast**	Declare neighbors with whom need to exchange normal "IPv4 unicast" routes
Step 6	**network <A.B.C.D> <A.B.C.D/M>**	Configure network A.B.C.D - Network number A.B.C.D/M - IPv4 prefix
Step 7	**exit-address-family**	Exits address-family mode
Step 8	**Exit**	Exits BGP Router Mode
Step 9	**End**	Exits Configuration mode.

The command “no network <A.B.C.D> <A.B.C.D/M>” removes the configured network.

If the command “no bgp ebgp-requires-policy” is not used, then the routes may be Exchanged with BGP peer without proper policies.

4.5.12.7 Redistribute connected

Follow the steps below to redistribute connect through BGP.

Step	Command	Description
Step 1	**sudo vtysh**	Enter FRRouting
Step 2	**configure terminal**	Enters the configuration mode
Step 3	**router bgp <AS no (1-65535)>**	Enable BGP and configure the AS number of the BGP Speaker
Step 4	**no bgp ebgp-requires-policy**	Disable Require policy on EBGP
Step 5	**address-family ipv4 unicast**	Declare neighbors with whom need to exchange normal "IPv4 unicast" routes
Step 6	**redistribute connected**	Redistributes connected routes to internal and external BGP peers
Step 7	**exit-address-family**	Exits address-family mode
Step 8	**Exit**	Exits BGP Router Mode
Step 9	**End**	Exits Configuration mode.

no redistribute connected stops the connected routes to internal and external BGP peers

4.5.12.8 Redistribute static

Follow the steps below to redistribute connect through BGP.

Step	Command	Description

Step 1	**sudo vtysh**	Enter FRRouting
Step 2	**configure terminal**	Enters the configuration mode
Step 3	**router bgp <AS no (1-65535)>**	Enable BGP and configure the AS number of the BGP Speaker
Step 4	**no bgp ebgp-requires-policy**	Disable Require policy on EBGP
Step 5	**address-family ipv4 unicast**	Declare neighbors with whom need to exchange normal "IPv4 unicast" routes
Step 6	**redistribute static**	Redistributes static routes to internal and external BGP peers
Step 7	**exit-address-family**	Exits address-family mode
Step 8	**Exit**	Exits BGP Router Mode
Step 9	**End**	Exits Configuration mode.

no redistribute static stops the static routes to internal and external BGP peers

4.5.13 BGP Configuration Example

This section shows a sample BGP configuration.

Step	Command	Description
Step 1	**sudo vtysh**	Enter FRRouting
Step 2	**configure**	Enter configuration mode
Step 3	**no route bgp < (1-4294967295) AS number>**	Remove default router (65100)
Step 4	**router bgp < (1-4294967295) AS number>**	Add new router
Step 5	**bgp router-id <A.B.C.D>**	Manually configure router identifier Router-id – Ipv4/Ipv6 address
Step 6	**neighbor <A.B.C.D Neighbor address> remote-as < (1-4294967295) AS number>**	Manually configure neighbor address and remote-as
Step 7	**bgp default local-preference (0-4294967295)**	Configure default local preference value
Step 8	**neighbor <A.B.C.D Neighbor address> ebgp-multihop (1-255)**	Configure ebgp-multihop ebgp-multihop - maximum hop count

Step 9	**neighbor <A.B.C.D Neighbor address> timers (0-65535) connect**	Configure timers timers - Keepalive interval connect - BGP connect timer
Step 10	**neighbor <A.B.C.D Neighbor address> advertisement-interval (0-600)**	Configure advertisement interval advertisement-interval - time in seconds
Step 11	**end**	Exit configure mode in FRRouting.
Step 12	**show ip bgp neighbors**	Displays configured BGP neighbor
Step 13	**show ip bgp summary**	Displays configured BGP details
Step 14	**exit**	Exit FRRouting
Step 15	**sudo config save -y**	Optional step - saves this configuration to be part of startup configuration.

The following example shows commands used to configure and display BGP.

SWITCH A (Sonic):

```
admin@sonic: ~$ sudo vtysh
Hello, this is FRRouting (version 7.5.1-sonic).
Copyright 1996-2005 Kunihiro Ishiguro, et al.
sonic# configure terminal
sonic(config)# no router bgp 65100
sonic(config)# router bgp 200
sonic(config-router) # bgp router-id 192.168.100.1
sonic(config-router) # no bgp ebgp-requires-policy
sonic(config-router) # neighbor 192.168.100.2 remote-as 300
sonic(config-router) # bgp default local-preference 50
sonic(config-router) # neighbor 192.168.100.2 ebgp-multihop
sonic(config-router) # neighbor 192.168.100.2 timers 10 10
sonic(config-router) # neighbor 192.168.100.2 advertisement-interval 5
sonic(config-router) # end
sonic# exit
admin@sonic: ~$
```

SWITCH B:

```
admin@sonic: ~$ sudo vtysh
Hello, this is FRRouting (version 7.5.1-sonic).
```

```
Copyright 1996-2005 Kunihiro Ishiguro, et al.
sonic# configure terminal
sonic(config)# no router bgp 65100
sonic(config)# router bgp 300
sonic(config-router) # bgp router-id 192.168.100.2
sonic(config-router) # no bgp ebgp-requires-policy
sonic(config-router) # neighbor 192.168.100.1 remote-as 200
sonic(config-router) # bgp default local-preference 50
sonic(config-router) # neighbor 192.168.100.1 ebgp-multihop
sonic(config-router) # neighbor 192.168.100.1 timers 10 10
sonic(config-router) # neighbor 192.168.100.1 advertisement-interval 5
sonic(config-router) # end
sonic# exit
admin@sonic: ~$
```

OUTPUT:

SWITCH A:

```
admin@sonic: ~$ show ip bgp neighbors
BGP neighbor is 192.168.100.2, remote AS 300, local AS 200, external link
  BGP version 4, remote router ID 192.168.100.2, local router ID 192.168.100.1
  BGP state = Established, up for 00:02:51
  Last read 00:00:05, Last write 00:00:21
  Hold time is 10, keepalive interval is 3 seconds
  Configured hold time is 10, keepalive interval is 3 seconds
  Neighbor capabilities:
    4 Byte AS: advertised and received
    AddPath:
      IPv4 Unicast: RX advertised IPv4 Unicast
    Route refresh: advertised and received(new)
    Address Family IPv4 Unicast: advertised and received
    Hostname Capability: advertised (name: sonic, domain name: n/a) not received
    Graceful Restart Capability: advertised and received
      Remote Restart timer is 120 seconds
      Address families by peer:
        IPv4 Unicast(preserved)
  Graceful restart information:
    End-of-RIB send: IPv4 Unicast
    End-of-RIB received: IPv4 Unicast
    Local GR Mode: Helper*
    Remote GR Mode: Restart
    R bit: False
    Timers:
      Configured Restart Time(sec): 120
      Received Restart Time(sec): 120
    IPv4 Unicast:
      F bit: True
```

```
    End-of-RIB sent: Yes
    End-of-RIB sent after update: Yes
    End-of-RIB received: Yes
    Timers:
      Configured Stale Path Time(sec): 360
 Message statistics:
  Inq depth is 0
  Outq depth is 0
                     Sent       Rcvd
  Opens:                3          3
  Notifications:        6          0
  Updates:              2          2
  Keepalives:           7         17
  Route Refresh:        1          0
  Capability:           0          0
  Total:               19         22
 Minimum time between advertisement runs is 5 seconds
For address family: IPv4 Unicast
 Update group 3, subgroup 2
 Packet Queue length 0
 Community attribute sent to this neighbor(all)
 Inbound updates discarded due to missing policy
 Outbound updates discarded due to missing policy
 0 accepted prefixes

 Connections established 2; dropped 1
 Last reset 00:02:55, No AFI/SAFI activated for peer
 Message received that caused BGP to send a NOTIFICATION:
  FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF
  00350104  012C005A  C0A86402  18021601
  04000100  01020040  06005A00  01018041
  04000001  2C
 External BGP neighbor may be up to 255 hops away.
Local host: 192.168.100.1, Local port: 179
Foreign host: 192.168.100.2, Foreign port: 38373
Nexthop: 192.168.100.1
Nexthop global: fe80::ec4:7aff:fe2e:1635
Nexthop local: fe80::ec4:7aff:fe2e:1635
BGP connection: shared network
BGP Connect Retry Timer in Seconds: 120
Estimated round trip time: 2 ms
Read thread: on Write thread: on FD used: 25
admin@sonic: ~$
admin@sonic: ~$ sudo vtysh
Hello, this is FRRouting (version 7.5.1-sonic).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

sonic# show bgp summary
```

```
IPv4 Unicast Summary:
BGP router identifier 192.168.100.1, local AS number 200 vrf-id 0
BGP table version 0
RIB entries 0, using 0 bytes of memory
Peers 1, using 21 KiB of memory

Neighbor        V     AS   MsgRcvd   MsgSent   TblVer  InQ  OutQ  Up/Down      State/PfxRcd  PfxSnt
192.168.100.2   4     300        69        36        0    0     0  00:11:24            (Policy) (Policy)

Total number of neighbors 1
sonic#
```

SWITCH B:

```
admin@sonic: ~$ show ip bgp neighbors
BGP neighbor is 192.168.100.1, remote AS 200, local AS 300, external link
 BGP version 4, remote router ID 192.168.100.1, local router ID 192.168.100.2
 BGP state = Established, up for 00:03:51
 Last read 00:00:05, Last write 00:00:21
 Hold time is 10, keepalive interval is 3 seconds
 Configured hold time is 10, keepalive interval is 3 seconds
 Neighbor capabilities:
  4 Byte AS: advertised and received
  AddPath:
   IPv4 Unicast: RX advertised IPv4 Unicast
  Route refresh: advertised and received(new)
  Address Family IPv4 Unicast: advertised and received
  Hostname Capability: advertised (name: sonic, domain name: n/a) not received
  Graceful Restart Capability: advertised and received
   Remote Restart timer is 120 seconds
   Address families by peer:
    IPv4 Unicast(preserved)
 Graceful restart information:
  End-of-RIB send: IPv4 Unicast
  End-of-RIB received: IPv4 Unicast
  Local GR Mode: Helper*
  Remote GR Mode: Restart
  R bit: False
  Timers:
   Configured Restart Time(sec): 120
   Received Restart Time(sec): 120
  IPv4 Unicast:
   F bit: True
   End-of-RIB sent: Yes
   End-of-RIB sent after update: Yes
   End-of-RIB received: Yes
```

```
    Timers:
      Configured Stale Path Time(sec): 360
  Message statistics:
    Inq depth is 0
    Outq depth is 0
                         Sent       Rcvd
    Opens:               5          5
    Notifications:       5          6
    Updates:             2          2
    Keepalives:          17         15
    Route Refresh:       1          0
    Capability:          0          0
    Total:               30         28
  Minimum time between advertisement runs is 5 seconds
 For address family: IPv4 Unicast
  Update group 3, subgroup 2
  Packet Queue length 0
  Community attribute sent to this neighbor(all)
  Inbound updates discarded due to missing policy
  Outbound updates discarded due to missing policy
  0 accepted prefixes

  Connections established 2; dropped 1
  Last reset 00:02:55, No AFI/SAFI activated for peer
  Message received that caused BGP to send a NOTIFICATION:
    FFFFFFFF    FFFFFFFF    FFFFFFFF    FFFFFFFF
    00350104    012C005A    C0A86402    18021601
    04000100    01020040    06005A00    01018041
    04000001    2C
  External BGP neighbor may be up to 255 hops away.
Local host: 192.168.100.2, Local port: 179
Foreign host: 192.168.100.1, Foreign port: 38373
Nexthop: 192.168.100.2
Nexthop global: fe80::ec4:7ade:fe2f:8675
Nexthop local: fe80::ec4:7ade:fe2f:8675
BGP connection: shared network
BGP Connect Retry Timer in Seconds: 120
Estimated round trip time: 2 ms
Read thread: on Write thread: on FD used: 25
admin@sonic: ~$
admin@sonic: ~$ sudo vtysh
Hello, this is FRRouting (version 7.5.1-sonic).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

sonic# show bgp summary
IPv4 Unicast Summary:
BGP router identifier 192.168.100.3, local AS number 300 vrf-id 0
BGP table version 0
```

```
RIB entries 0, using 0 bytes of memory
Peers 1, using 21 KiB of memory

Neighbor        V     AS   MsgRcvd   MsgSent   TblVer  InQ  OutQ  Up/Down      State/PfxRcd  PfxSnt
192.168.100.1   4     200       36        69        0    0     0  00:15:24            (Policy) (Policy)

Total number of neighbors 1
sonic#
```

4.5.14 Route Reflector Configuration

This section shows a sample BGP configuration for route reflector.

Step	Command	Description
Step 1	**sudo vtysh**	Enter FRRouting
Step 2	**configure**	Enter configuration mode
Step 3	**router bgp < (1-4294967295) AS number>**	Add new router
Step 4	**neighbor <A.B.C.D Neighbor address> remote-as < (1-4294967295) AS number>**	Configure neighbor address and remote-as
Step 5	**neighbor <A.B.C.D Neighbor address> route-reflector-client**	Configure neighbor as route-reflector-client.
Step 6	**end**	Exit configure mode in FRRouting.
Step 7	**write**	Optional step - saves this configuration to be part of bgpd.conf.
Step 8	**exit**	Exit FRRouting
Step 9	**sudo config save -y**	Optional step - saves this configuration to be part of startup configuration.

The following example shows commands used to configure route-reflector-client.

```
admin@sonic:~$ sudo -i
root@sonic:~# vtysh
Hello, this is FRRouting (version 7.5.1-sonic).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

sonic# configure terminal
sonic(config)# router bgp 65100
sonic(config-router)# neighbor 10.0.0.2 remote-as 65100
```

```
sonic(config-router)# neighbor 10.0.0.2 route-reflector-client
sonic(config-router)# end
sonic# write
sonic# exit
root@sonic:~# config save -y
```

4.5.15 BGP IPv6 Configuration

This section shows a sample IPv6 BGP configuration.

Step	Command	Description
Step 1	**sudo vtysh**	Enter FRRouting
Step 2	**configure**	Enter configuration mode
Step 3	**router bgp < (1-4294967295) AS number>**	Add new router
Step 4	**neighbor <Neighbor's IPv6 address> remote-as < (1-4294967295) AS number>**	Configure neighbor's IPv6 address and remote-as
Step 5	**neighbor <Neighbor's IPv6 address> route-reflector-client**	Configure neighbor as route-reflector-client.
Step 6	**end**	Exit configure mode in FRRouting.
Step 7	**write**	Optional step - saves this configuration to be part of bgpd.conf.
Step 8	**exit**	Exit FRRouting
Step 9	**sudo config save -y**	Optional step - saves this configuration to be part of startup configuration.

4.6 Route Map

This section explains the routing policy that takes precedence over the other route processes that are configured.

By default, any packet or route that does not match any particular entry in the route map will be dropped.

4.6.1 Configure route-map

Follow the steps below to configure Route-Map parameters.

Step	Command	Description
Step 1	**sudo vtysh**	Enter FRRouting
Step 2	**configure**	Enter configuration mode
Step 3	**route-map map-name {permit | deny} [sequence-number (1-65535)]**	Configure Route-map map-name - A valid route-map name permit – To permit the route deny – To deny the route sequence number – A valid number in range (1-65535)
Step 4	**call <WORD>**	Call to another route-map WORD - Target route-map name
Step 5	**description < description_string>**	Describing this route-map rule description_string - A valid string
Step 6	**match as-path <WORD>**	WORD - AS path access-list name
Step 7	**match community COMMUNITY_LIST**	Matches the specified community_list COMMUNITY_LIST - It can be Community-list number or WORD Community-list number - standard (1-99) or expanded (100-500) WORD - Community-list name
Step 8	**match evpn [default-route | rd | route-type[macip|multicast|prefix] | vni [VNI ID]]**	BGP EVPN specific match default-route - default EVPN type-5 route rd - Route Distinguisher route-type - Match route-type

		macip - mac-ip route multicast - IMET route prefix - prefix route vni - Match VNI, VNI ID - (1-16777215)
Step 9	**match extcommunity COMMUNITY_LIST**	Matches the specified extcommunity COMMUNITY_LIST - It can be Community-list number or WORD Community-list number - standard (1-99) or expanded (100-500) WORD - Community-list name
Step 10	**match interface IFNAME**	Matches the specified interface IFNAME - Interface name
Step 11	**match ip [address [IP access-list number|WORD|prefix-len (0-32) |prefix-list<WORD>] |next-hop|route-source]**	address - Match address of route IP access-list number – standard (1-199) or expanded (1300-2699) WORD - IP Access-list name prefix-len - Match prefix length of IP address, range 0-32 prefix-list - Match entries of prefix-lists next-hop - Match next-hop address of route route-source - Match advertising source address of route
Step 12	**set ip next-hop <A.B.C.D>**	A.B.C.D - IP address of next hop
Step 13	**set local-preference (0-4294967295)**	Configure local preference value
Step 14	**set community <none|COMMUNITY> additive**	Sets the community value

Step 15	**Exit**	Exits Route-map Mode
Step 16	**End**	Exits Configuration mode
Step 17	**show route-map [ROUTE_MAP_NAME]**	Displays the Route-map. ROUTE_MAP_NAME – Name of the route-map.

"no route-map" command deletes configured Route-map

The following example shows the command used to display the Route-map.

```
admin@sonic: ~$ sudo vtysh
Hello, this is FRRouting (version 7.5.1-sonic).
Copyright 1996-2005 Kunihiro Ishiguro, et al.
sonic# configure
sonic(config)# route-map rm-supermicro permit 10
sonic(config-route-map) # description supermicro
sonic(config-route-map) # set ip next-hop 192.168.100.1
sonic(config-route-map) # set local-preference 10
sonic(config-route-map) # set community additive no-export
sonic(config-route-map) # exit
sonic(config)# exit
sonic# show route-map rm-supermicro
ZEBRA:
route-map: rm-supermicro Invoked: 0 Optimization: enabled Processed Change: false
 permit, sequence 10 Invoked 0
  Description:
    supermicro
  Match clauses:
  Set clauses:
  Call clause:
  Action:
    Exit routemap
BGP:
route-map: rm-supermicro Invoked: 0 Optimization: enabled Processed Change: false
 permit, sequence 10 Invoked 0
  Description:
```

```
    supermicro
  Match clauses:
  Set clauses:
    ip next-hop 192.168.100.1
    local-preference 10
    community no-export additive
  Call clause:
  Action:
    Exit routemap
sonic#
```

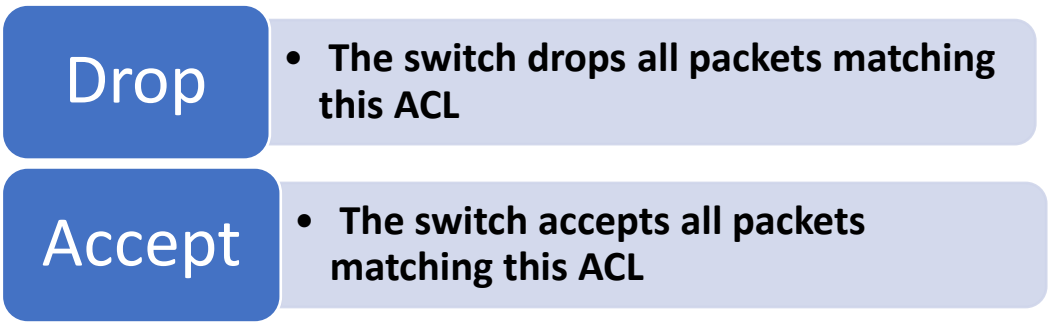
5 Access Control Lists

ACL is used to filter any particular traffic flow on the switch.

ACLs can be configured to match packets based on Layer3 or Layer 4 TCP/UDP Parameters.

Every packet entering/exiting the switch is checked for the configured ACLs. If any packet contents match any of the configured ACLs, that packet will be handled according to the matched ACL configured action.

The ACL configuration provides the following actions that can be applied on matched traffic flow.



ACL is implemented in hardware ASIC (Application Specific Integrated Circuit) to provide line rate processing for all incoming traffic.

ASIC analyzes the first 128 bytes of every received packet and extracts the packet contents for key fields in the Layer 2, Layer 3 and Layer 4 headers. ASIC then looks up the ACL tables to find a matching ACL rule for the extracted content of the packet. ASIC compares the values of the configured fields only and treats all other fields as “do not care”. Once a matching ACL rule is found, ASIC stops looking in that ACL table.

ASIC applies the configured action of the matching ACL rule to the matched packet. This could result in dropping that packet or allowing the packet to be forwarded through the switch.

5.1 IP Access Control List

An IP ACL allows users to control traffic based on fields in an IP header, ICMP header, TCP header and UDP header. Users can configure the traffic flow based on source IP address, destination IP address, TCP port number or UDP port number.

Users can deny or permit the packet flow using an ACL rule for ingress/egress traffic.

5.1.1 IPv4 Access Control List

ACL configuration for IPv4 packets is explained below.

5.1.1.1 Configure ACL table and ACL Rules

ACL configuration has two steps. First the ACL table has to be created and then the ACL rules need to be added in JSON format.

Follow the below steps to create ACL table.

Step	Command	Description
Step 1	config acl add table [OPTIONS] <table_name> <table_type> [-d <description>] [-p <ports>] [-s (ingress | egress)]	Create ACL table. add table – Creates table. table_type – L3 -d – Description of the table. -p – Ports to bind the ACL table. -s – ingress/egress direction.
Step 2	**show acl table**	Displays the ACL table.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

Example command to create the ACL table is given below.

```
root@sonic:~# config acl add table ACL_RULES_1 L3 -s egress -
pEthernet15,Ethernet16,Ethernet17,Ethernet18,Ethernet19,Ethernet20 -d"External access rules."
root@sonic:~# config acl add table ACL_RULES_2 L3 -s ingress -
pEthernet26,Ethernet27,Ethernet28,Ethernet29,Ethernet30 -d"Finance/accounting dept."

root@sonic:~# show acl table
Name          Type   Binding      Description                Stage
------------       ------  ----------       ------------------------       -------
ACL_RULES_1  L3    Ethernet15  External access rules.    egress
                   Ethernet16
                   Ethernet17
                   Ethernet18
                   Ethernet19
                   Ethernet20
ACL_RULES_2  L3    Ethernet26  Finance/accounting dept.  ingress
                   Ethernet27
                   Ethernet28
                   Ethernet29
                   Ethernet30
root@sonic:~#
```

Then the ACL rule has to be added/updated using the below command. Use the sample JSON given in the next subsections for the ACL rule configurations.

Follow the below steps to add/update the ACL rules.

Step	Command	Description
Step 1	config acl update <full/incremental> <JSON-filename>	Configure the ACL rules. full – Full update of ACL rules configuration. incremental – Incremental update of ACL rule configuration. JSON-filename – Name of the file containing the ACL rule in JSON format.
Step 2	**show acl rule**	Displays the ACL rules.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

Example command to update the ACL rule is given below.

```
root@sonic:~# config acl update full /tmp/ACL-Rules.json

root@sonic:~# show acl rule
Table       Rule    Priority   Action    Match
----------- ------ ---------- -------- ----------------------
ACL_RULES_1  RULE_1  9999       DROP     ETHER_TYPE: 2048
                            SRC_IP: 172.31.0.19/32
ACL_RULES_1  RULE_2  9998       DROP     ETHER_TYPE: 2048
                            L4_SRC_PORT: 179
root@sonic:~#
```

5.1.1.2 Sample ACL Configuration Based on Source IPv4 Address

Below is the ACL configuration to accept the packets with source IPv4 address 172.31.0.19.

```
{
   "acl": {
      "acl-sets": {
         "acl-set": {
            "ACL_Rules_1":{
               "acl-entries": {
                  "acl-entry": {
                     "1": {
                        "actions":{
                           "config":{
```

```
                        "forwarding-action":"ACCEPT"
                      }
                    },
                    "config":{
                      "sequence-id":1
                      },
                    "ip":{
                      "config":{
                        "source-ip-address":"172.31.0.19/32",
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
```

Below is the ACL configuration to drop the packets with source IP address 172.31.0.19.

```
{
  "acl": {
    "acl-sets": {
      "acl-set": {
        "ACL_Rules_1":{
          "acl-entries": {
            "acl-entry": {
              "1": {
                "actions":{
                  "config":{
                    "forwarding-action":"DROP"
                  }
                },
                "config":{
                  "sequence-id":1
                  },
                "ip":{
                  "config":{
                    "source-ip-address":"172.31.0.19/32"
                  }
                }
              }
            }
          }
        }
      }
```

```
      }
    }
}
```

5.1.1.3 Sample ACL Configuration Based on Destination IPv4 Address

Below is the ACL configuration to accept the packets with destination IPv4 address 172.31.0.22.

```
{
   "acl": {
      "acl-sets": {
         "acl-set": {
            "ACL_Rules_1":{
               "acl-entries": {
                  "acl-entry": {
                     "1": {
                        "actions":{
                           "config":{
                              "forwarding-action":"ACCEPT"
                           }
                        },
                        "config":{
                           "sequence-id":1
                           },
                        "ip":{
                           "config":{
                              "destination-ip-address":"172.31.0.22/32"
                           }
                        }
                     }
                  }
               }
            }
         }
      }
   }
}
```

Below is the ACL configuration to drop the packets with destination IP address 172.31.0.22.

```
{
   "acl": {
      "acl-sets": {
         "acl-set": {
            "ACL_Rules_1":{
               "acl-entries": {
                  "acl-entry": {
                     "1": {
                        "actions":{
```

```
                    "config":{
                      "forwarding-action":"DROP"
                    }
                  },
                  "config":{
                    "sequence-id":1
                    },
                  "ip":{
                    "config":{
                      "destination-ip-address":"172.31.0.22/32"
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
}
```

5.1.1.4 Sample ACL Configuration Based on Source Port

Below is the ACL configuration to accept the packets with source port 179.

```
{
  "acl": {
    "acl-sets": {
      "acl-set": {
        "ACL_Rules_1":{
          "acl-entries": {
            "acl-entry": {
              "1": {
                "actions":{
                  "config":{
                    "forwarding-action":"ACCEPT"
                  }
                },
                "config":{
                  "sequence-id":1
                  },
                "transport":{
                  "config":{
                    "source-port":179
                  }
                }
              }
            }
```

```
            }
          }
        }
      }
    }
}
```

Below is the ACL configuration to drop the packets with source port 179.

```
{
   "acl": {
      "acl-sets": {
         "acl-set": {
            "ACL_Rules_1":{
               "acl-entries": {
                  "acl-entry": {
                     "1": {
                        "actions":{
                           "config":{
                              "forwarding-action":"DROP"
                           }
                        },
                        "config":{
                           "sequence-id":1
                           },
                        "transport":{
                           "config":{
                              "source-port":179
                           }
                        }
                     }
                  }
               }
            }
         }
      }
   }
}
```

Below is the ACL configuration to accept the packets with source port range from 179 to 182.

```
{
   "acl": {
      "acl-sets": {
         "acl-set": {
            "ACL_Rules_1":{
               "acl-entries": {
                  "acl-entry": {
                     "1": {
```

```
                    "actions":{
                      "config":{
                        "forwarding-action":"ACCEPT"
                      }
                    },
                    "config":{
                      "sequence-id":1
                      },
                    "transport":{
                      "config":{
                        "source-port":"179..182"
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
}
```

Below is the ACL configuration to drop the packets with source port range from 179 to 182.

```
{
  "acl": {
    "acl-sets": {
      "acl-set": {
        "ACL_Rules_1":{
          "acl-entries": {
            "acl-entry": {
              "1": {
                "actions":{
                  "config":{
                    "forwarding-action":"DROP"
                  }
                },
                "config":{
                  "sequence-id":1
                  },
                "transport":{
                  "config":{
                    "source-port":"179..182"
                  }
                }
              }
            }
          }
```

```
          }
        }
      }
    }
}
```

5.1.1.5 Sample ACL Configuration Based on Destination Port

Below is the ACL configuration to accept the packets with destination port 179.

```
{
   "acl": {
      "acl-sets": {
         "acl-set": {
            "ACL_Rules_1":{
               "acl-entries": {
                  "acl-entry": {
                     "1": {
                        "actions":{
                           "config":{
                              "forwarding-action":"ACCEPT"
                           }
                        },
                        "config":{
                           "sequence-id":1
                           },
                        "transport":{
                           "config":{
                              "destination-port":179
                           }
                        }
                     }
                  }
               }
            }
         }
      }
   }
}
```

Below is the ACL configuration to drop the packets with destination port 179.

```
{
   "acl": {
      "acl-sets": {
         "acl-set": {
            "ACL_Rules_1":{
               "acl-entries": {
                  "acl-entry": {
```

```
                "1": {
                  "actions":{
                    "config":{
                      "forwarding-action":"DROP"
                    }
                  },
                  "config":{
                    "sequence-id":1
                    },
                  "transport":{
                    "config":{
                      "destination-port":179
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
}
```

Below is the ACL configuration to accept the packets with destination port range from 179 to 182.

```
{
   "acl": {
     "acl-sets": {
       "acl-set": {
         "ACL_Rules_1":{
           "acl-entries": {
             "acl-entry": {
               "1": {
                 "actions":{
                   "config":{
                     "forwarding-action":"ACCEPT"
                   }
                 },
                 "config":{
                   "sequence-id":1
                   },
                 "transport":{
                   "config":{
                     "destination-port":"179..182"
                   }
                 }
               }
             }
```

```
            }
          }
        }
      }
    }
}
```

Below is the ACL configuration to drop the packets with destination port range from 179 to 182.

```
{
   "acl": {
      "acl-sets": {
         "acl-set": {
            "ACL_Rules_1":{
               "acl-entries": {
                  "acl-entry": {
                     "1": {
                        "actions":{
                           "config":{
                              "forwarding-action":"DROP"
                           }
                        },
                        "config":{
                           "sequence-id":1
                           },
                        "transport":{
                           "config":{
                              "destination-port":"179..182"
                           }
                        }
                     }
                  }
               }
            }
         }
      }
   }
}
```

5.1.1.6 Sample ACL Configuration Based on Protocols

Below is the ACL configuration to accept the TCP packets.

```
{
   "acl": {
      "acl-sets": {
         "acl-set": {
            "ACL_Rules_1":{
               "acl-entries": {
```

```
                "acl-entry": {
                  "1": {
                    "actions":{
                      "config":{
                        "forwarding-action":"ACCEPT"
                      }
                    },
                    "config":{
                      "sequence-id":1
                      },
                    "ip":{
                      "config":{
                        "protocol":"IP_TCP"
                      }
                    }
                  }
                }
              }
            }
          }
        }
    }
}
```

Below is the ACL configuration to drop the TCP packets.

```
{
   "acl": {
      "acl-sets": {
         "acl-set": {
            "ACL_Rules_1":{
               "acl-entries": {
                  "acl-entry": {
                    "1": {
                      "actions":{
                        "config":{
                          "forwarding-action":"DROP"
                        }
                      },
                      "config":{
                        "sequence-id":1
                        },
                      "ip":{
                        "config":{
                          "protocol":"IP_TCP"
                        }
                      }
                    }
```

```
            }
          }
        }
      }
    }
  }
}
```

Below is the ACL configuration to accept the UDP packets.

```
{
  "acl": {
    "acl-sets": {
      "acl-set": {
        "ACL_Rules_1":{
          "acl-entries": {
            "acl-entry": {
              "1": {
                "actions":{
                  "config":{
                    "forwarding-action":"ACCEPT"
                  }
                },
                "config":{
                  "sequence-id":1
                  },
                "ip":{
                  "config":{
                    "protocol":"IP_UDP"
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

Below is the ACL configuration to drop the UDP packets.

```
{
  "acl": {
    "acl-sets": {
      "acl-set": {
        "ACL_Rules_1":{
          "acl-entries": {
            "acl-entry": {
```

```
                "1": {
                  "actions":{
                    "config":{
                      "forwarding-action":"DROP"
                    }
                  },
                  "config":{
                    "sequence-id":1
                    },
                  "ip":{
                    "config":{
                      "protocol":"IP_UDP"
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
}
```

Below is the ACL configuration to accept the TCP packets with acknowledgement flag set.

```
{
   "acl": {
     "acl-sets": {
       "acl-set": {
         " ACL_Rules_1": {
           "acl-entries": {
             "acl-entry": {
               "5": {
                 "actions":{
                   "config": {
                     "forwarding-action": "ACCEPT"
                   }
                 },
                 "config": {
                   "sequence-id": 5
                 },
                 "transport":{
                   "config":{
                     "tcp-flags":"TCP_ACK"
                   }
                 }
               }
             }
```

```
            }
          }
        }
      }
    }
}
```

Below is the ACL configuration to accept the ICMP packets with source IP address 172.31.0.19.

```
{
   "acl": {
      "acl-sets": {
         "acl-set": {
            "ACL_Rules_1":{
               "acl-entries": {
                  "acl-entry": {
                     "1": {
                        "actions":{
                           "config":{
                              "forwarding-action":"ACCEPT"
                           }
                        },
                        "config":{
                           "sequence-id":1
                           },
                        "ip":{
                           "config":{
                              "protocol":"IP_ICMP",
                              "source-ip-address":"172.31.0.19/32"
                           }
                        }
                     }
                  }
               }
            }
         }
      }
   }
}
```

Below is the ACL configuration to accept the UDP packets with source IP address 172.31.0.19.

```
{
   "acl": {
      "acl-sets": {
         "acl-set": {
            "ACL_Rules_1":{
               "acl-entries": {
                  "acl-entry": {
```

```
                "1": {
                  "actions":{
                    "config":{
                      "forwarding-action":"ACCEPT"
                    }
                  },
                  "config":{
                    "sequence-id":1
                    },
                  "ip":{
                    "config":{
                      "protocol":"IP_UDP",
                      "source-ip-address":"172.31.0.19/32"
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
}
```

5.1.1.7 Sample ACL Configuration With Multiple Rules/Parameters

Below is the ACL configuration with two rules. First rule to drop the packets with source IPv4 address 172.31.0.19 and second rule to drop the packets with source port 179.

```
{
   "acl": {
      "acl-sets": {
         "acl-set": {
            "ACL_Rules_1":{
               "acl-entries": {
                  "acl-entry": {
                     "1": {
                        "actions":{
                           "config":{
                              "forwarding-action":"DROP"
                           }
                        },
                        "config":{
                           "sequence-id":1
                           },
                        "ip":{
                           "config":{
                              "source-ip-address":"172.31.0.19/32"
```

```
                    }
                  }
                },
                "2": {
                  "actions":{
                    "config":{
                      "forwarding-action":"DROP"
                    }
                  },
                  "config":{
                    "sequence-id":2
                    },
                  "transport":{
                    "config":{
                      "source-port":179
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
}
```

Below is the ACL configuration with four parameters. This ACL accepts the packets with source IPv4 address 1.1.1.1, destination IPv4 address 2.2.2.2, source port 179 and destination port 182.

```
{
   "acl": {
      "acl-sets": {
        "acl-set": {
          " ACL_Rules_1":{
            "acl-entries": {
              "acl-entry": {
                "9": {
                  "actions":{
                    "config":{
                      "forwarding-action":"ACCEPT"
                    }
                  },
                  "config":{
                    "sequence-id":9
                    },
                  "ip":{
                    "config":{
                      "source-ip-address":"1.1.1.1/32",
```

```
                    "destination-ip-address":"2.2.2.2/32",
                    "protocol":"IP_TCP"
                  }
                },
                "transport":{
                  "config":{
                    "source-port":179,
                    "destination-port":182
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

5.1.2 IPv6 Access Control List

ACL configuration for IPv6 packets is explained below.

5.1.2.1 Configure ACL table and ACL Rules

ACL configuration has two steps. First the ACL table has to be created and then the ACL rules need to be added in JSON format.

Follow the below steps to create ACL table.

Step	Command	Description
Step 1	config acl add table [OPTIONS] <table_name> <table_type> [-d <description>] [-p <ports>] [-s (ingress | egress)]	Create ACL table. add table – Creates table. table_type – L3V6 -d – Description of the table. -p – Ports to bind the ACL table. -s – ingress/egress direction.
Step 2	**show acl table**	Displays the ACL table.

Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

Example command to create the ACL table is given below.

```
root@sonic:~# config acl add table ACL_RULES_1 L3V6 -s egress -
pEthernet15,Ethernet16,Ethernet17,Ethernet18,Ethernet19,Ethernet20 -d"External access rules."
root@sonic:~# config acl add table ACL_RULES_2 L3V6 -s ingress -
pEthernet26,Ethernet27,Ethernet28,Ethernet29,Ethernet30 -d"Finance/accounting dept."

root@sonic:~# show acl table
Name          Type   Binding      Description                Stage
------------       ------  ----------      ------------------------       -------
ACL_RULES_1  L3V6    Ethernet15  External access rules.    egress
                      Ethernet16
                      Ethernet17
                      Ethernet18
                      Ethernet19
                      Ethernet20
ACL_RULES_2  L3V6    Ethernet26  Finance/accounting dept.  ingress
                      Ethernet27
                      Ethernet28
                      Ethernet29
                      Ethernet30
root@sonic:~#
```

Then the ACL rule has to be added/updated using the below command. Use the sample JSON given in the next subsections for the ACL rule configurations.

Follow the below steps to add/update the ACL rules.

Step	**Command**	**Description**
Step 1	config acl update <full/incremental> <JSON-filename>	Configure the ACL rules. full – Full update of ACL rules configuration. incremental – Incremental update of ACL rule configuration. JSON-filename – Name of the file containing the ACL rule in JSON format.

Step 2	**show acl rule**	Displays the ACL rules.
Step 3	**sudo config save –y**	Optional step - saves this configuration to be part of startup configuration.

Example command to update the ACL rule is given below.

```
root@sonic:~# config acl update full /tmp/ACL-Rules.json

root@sonic:~# show acl rule
Table       Rule    Priority    Action    Match
----------- ------ ---------- -------- ----------------------
ACL_RULES_1  RULE_1  9999       DROP     ETHER_TYPE: 2048
                              SRC_IPV6: fe80::ec4:7aff:fe2e:1000/124
ACL_RULES_1  RULE_2  9998       DROP     ETHER_TYPE: 2048
                              L4_SRC_PORT: 179
root@sonic:~#
```

5.1.2.2 Sample ACL Configuration Based on Source IPv6 Address

Below is the ACL configuration to accept the packets with source IPv6 address fe80::ec4:7aff:fe2e:1000/124.

```
{
   "acl": {
      "acl-sets": {
         "acl-set": {
            "ACL_Rules_1":{
               "acl-entries": {
                  "acl-entry": {
                     "1": {
                        "actions":{
                           "config":{
                              "forwarding-action":"ACCEPT"
                           }
                        },
                        "config":{
                           "sequence-id":1
                           },
                        "ip":{
                           "config":{
                              "source-ip-address":"fe80::ec4:7aff:fe2e:1000/124",
                           }
                        }
                     }
                  }
               }
            }
         }
```

```
        }
      }
    }
}
```

Below is the ACL configuration to drop the packets with source IP address fe80::ec4:7aff:fe2e:2000/124.

```
{
  "acl": {
    "acl-sets": {
      "acl-set": {
        "ACL_Rules_1":{
          "acl-entries": {
            "acl-entry": {
              "1": {
                "actions":{
                  "config":{
                    "forwarding-action":"DROP"
                  }
                },
                "config":{
                  "sequence-id":1
                  },
                "ip":{
                  "config":{
                    "source-ip-address":"fe80::ec4:7aff:fe2e:2000/124"
                  }
                }
              }
            }
          }
        }
      }
    }
}
```

5.1.2.3 Sample ACL Configuration Based on Destination IPv6 Address

Below is the ACL configuration to accept the packets with destination IPv6 address fe80::ec4:7aff:fe2e:1000/124.

```
{
  "acl": {
    "acl-sets": {
      "acl-set": {
        "ACL_Rules_1":{
          "acl-entries": {
```

```
                "acl-entry": {
                  "1": {
                    "actions":{
                      "config":{
                        "forwarding-action":"ACCEPT"
                      }
                    },
                    "config":{
                      "sequence-id":1
                      },
                    "ip":{
                      "config":{
                        "destination-ip-address":"fe80::ec4:7aff:fe2e:1000/124"
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
```

Below is the ACL configuration to drop the packets with destination IP address fe80::ec4:7aff:fe2e:2000/124.

```
{
   "acl": {
      "acl-sets": {
         "acl-set": {
            "ACL_Rules_1":{
               "acl-entries": {
                  "acl-entry": {
                     "1": {
                        "actions":{
                           "config":{
                              "forwarding-action":"DROP"
                           }
                        },
                        "config":{
                           "sequence-id":1
                           },
                        "ip":{
                           "config":{
                              "destination-ip-address":"fe80::ec4:7aff:fe2e:2000/124"
                           }
                        }
```

```
                }
              }
            }
          }
        }
      }
    }
}
```

5.1.2.4 Sample ACL Configuration Based on Source Port

Below is the ACL configuration to accept the packets with source port 179.

```
{
  "acl": {
    "acl-sets": {
      "acl-set": {
        "ACL_Rules_1":{
          "acl-entries": {
            "acl-entry": {
              "1": {
                "actions":{
                  "config":{
                    "forwarding-action":"ACCEPT"
                  }
                },
                "config":{
                  "sequence-id":1
                  },
                "transport":{
                  "config":{
                    "source-port":179
                  }
                }
              }
            }
          }
        }
      }
    }
}
```

Below is the ACL configuration to drop the packets with source port 179.

```
{
  "acl": {
    "acl-sets": {
      "acl-set": {
```

```
            "ACL_Rules_1":{
              "acl-entries": {
                "acl-entry": {
                  "1": {
                    "actions":{
                      "config":{
                        "forwarding-action":"DROP"
                      }
                    },
                    "config":{
                      "sequence-id":1
                      },
                    "transport":{
                      "config":{
                        "source-port":179
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
```

Below is the ACL configuration to accept the packets with source port range from 179 to 182.

```
{
  "acl": {
    "acl-sets": {
      "acl-set": {
        "ACL_Rules_1":{
          "acl-entries": {
            "acl-entry": {
              "1": {
                "actions":{
                  "config":{
                    "forwarding-action":"ACCEPT"
                  }
                },
                "config":{
                  "sequence-id":1
                  },
                "transport":{
                  "config":{
                    "source-port":"179..182"
                  }
```

```
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
}
```

Below is the ACL configuration to drop the packets with source port range from 179 to 182.

```
{
   "acl": {
      "acl-sets": {
         "acl-set": {
            "ACL_Rules_1":{
               "acl-entries": {
                  "acl-entry": {
                     "1": {
                        "actions":{
                           "config":{
                              "forwarding-action":"DROP"
                           }
                        },
                        "config":{
                           "sequence-id":1
                           },
                        "transport":{
                           "config":{
                              "source-port":"179..182"
                           }
                        }
                     }
                  }
               }
            }
         }
      }
   }
}
```

5.1.2.5 Sample ACL Configuration Based on Destination Port

Below is the ACL configuration to accept the packets with destination port 179.

```
{
   "acl": {
      "acl-sets": {
```

```
        "acl-set": {
          "ACL_Rules_1":{
            "acl-entries": {
              "acl-entry": {
                "1": {
                  "actions":{
                    "config":{
                      "forwarding-action":"ACCEPT"
                    }
                  },
                  "config":{
                    "sequence-id":1
                    },
                  "transport":{
                    "config":{
                      "destination-port":179
                    }
                  }
                }
              }
            }
          }
        }
    }
}
```

Below is the ACL configuration to drop the packets with destination port 179.

```
{
   "acl": {
      "acl-sets": {
        "acl-set": {
          "ACL_Rules_1":{
            "acl-entries": {
              "acl-entry": {
                "1": {
                  "actions":{
                    "config":{
                      "forwarding-action":"DROP"
                    }
                  },
                  "config":{
                    "sequence-id":1
                    },
                  "transport":{
                    "config":{
                      "destination-port":179
```

```
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
}
```

Below is the ACL configuration to accept the packets with destination port range from 179 to 182.

```
{
  "acl": {
    "acl-sets": {
      "acl-set": {
        "ACL_Rules_1":{
          "acl-entries": {
            "acl-entry": {
              "1": {
                "actions":{
                  "config":{
                    "forwarding-action":"ACCEPT"
                  }
                },
                "config":{
                  "sequence-id":1
                  },
                "transport":{
                  "config":{
                    "destination-port":"179..182"
                  }
                }
              }
            }
          }
        }
      }
    }
}
```

Below is the ACL configuration to drop the packets with destination port range from 179 to 182.

```
{
  "acl": {
    "acl-sets": {
      "acl-set": {
```

```
            "ACL_Rules_1":{
              "acl-entries": {
                "acl-entry": {
                  "1": {
                    "actions":{
                      "config":{
                        "forwarding-action":"DROP"
                      }
                    },
                    "config":{
                      "sequence-id":1
                      },
                    "transport":{
                      "config":{
                        "destination-port":"179..182"
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
```

5.1.2.6 Sample ACL Configuration Based on Protocols

Below is the ACL configuration to accept the TCP packets.

```
{
  "acl": {
    "acl-sets": {
      "acl-set": {
        "ACL_Rules_1":{
          "acl-entries": {
            "acl-entry": {
              "1": {
                "actions":{
                  "config":{
                    "forwarding-action":"ACCEPT"
                  }
                },
                "config":{
                  "sequence-id":1
                  },
                "ip":{
                  "config":{
```

```
                        "protocol":"IP_TCP"
                    }
                }
            }
        }
    }
  }
}
```

Below is the ACL configuration to drop the TCP packets.

```
{
  "acl": {
    "acl-sets": {
      "acl-set": {
        "ACL_Rules_1":{
          "acl-entries": {
            "acl-entry": {
              "1": {
                "actions":{
                  "config":{
                    "forwarding-action":"DROP"
                  }
                },
                "config":{
                  "sequence-id":1
                  },
                "ip":{
                  "config":{
                    "protocol":"IP_TCP"
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

Below is the ACL configuration to accept the UDP packets.

```
{
  "acl": {
    "acl-sets": {
```

```
        "acl-set": {
          "ACL_Rules_1":{
            "acl-entries": {
              "acl-entry": {
                "1": {
                  "actions":{
                    "config":{
                      "forwarding-action":"ACCEPT"
                    }
                  },
                  "config":{
                    "sequence-id":1
                    },
                  "ip":{
                    "config":{
                      "protocol":"IP_UDP"
                    }
                  }
                }
              }
            }
          }
        }
    }
}
```

Below is the ACL configuration to drop the UDP packets.

```
{
   "acl": {
      "acl-sets": {
        "acl-set": {
          "ACL_Rules_1":{
            "acl-entries": {
              "acl-entry": {
                "1": {
                  "actions":{
                    "config":{
                      "forwarding-action":"DROP"
                    }
                  },
                  "config":{
                    "sequence-id":1
                    },
                  "ip":{
                    "config":{
                      "protocol":"IP_UDP"
```

```
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
}
```

Below is the ACL configuration to accept the TCP packets with acknowledgement flag set.

```
{
  "acl": {
    "acl-sets": {
      "acl-set": {
        " ACL_Rules_1": {
          "acl-entries": {
            "acl-entry": {
              "5": {
                "actions":{
                  "config": {
                    "forwarding-action": "ACCEPT"
                  }
                },
                "config": {
                  "sequence-id": 5
                },
                "transport":{
                  "config":{
                    "tcp-flags":"TCP_ACK"
                  }
                }
              }
            }
          }
        }
      }
    }
}
```

Below is the ACL configuration to accept the ICMP packets with source IP address fe80::ec4:7aff:fe2e:1000/124.

```
{
  "acl": {
    "acl-sets": {
```

```
        "acl-set": {
          "ACL_Rules_1":{
            "acl-entries": {
              "acl-entry": {
                "1": {
                  "actions":{
                    "config":{
                      "forwarding-action":"ACCEPT"
                    }
                  },
                  "config":{
                    "sequence-id":1
                    },
                  "ip":{
                    "config":{
                      "protocol":"IP_ICMP",
                      "source-ip-address":"fe80::ec4:7aff:fe2e:1000/124"
                    }
                  }
                }
              }
            }
          }
        }
    }
}
```

Below is the ACL configuration to accept the UDP packets with source IP address fe80::ec4:7aff:fe2e:1000/124.

```
{
   "acl": {
      "acl-sets": {
        "acl-set": {
          "ACL_Rules_1":{
            "acl-entries": {
              "acl-entry": {
                "1": {
                  "actions":{
                    "config":{
                      "forwarding-action":"ACCEPT"
                    }
                  },
                  "config":{
                    "sequence-id":1
                    },
                  "ip":{
```

```
                    "config":{
                      "protocol":"IP_UDP",
                      "source-ip-address":"fe80::ec4:7aff:fe2e:1000/124"
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
}
```

5.1.2.7 Sample ACL Configuration With Multiple Rules/Parameters

Below is the ACL configuration with two rules. First rule to drop the packets with source IPv6 address fe80::ec4:7aff:fe2e:1000/124 and second rule to drop the packets with source port 179.

```
{
  "acl": {
    "acl-sets": {
      "acl-set": {
        "ACL_Rules_1":{
          "acl-entries": {
            "acl-entry": {
              "1": {
                "actions":{
                  "config":{
                    "forwarding-action":"DROP"
                  }
                },
                "config":{
                  "sequence-id":1
                  },
                "ip":{
                  "config":{
                    "source-ip-address":"fe80::ec4:7aff:fe2e:1000/124"
                  }
                }
              },
              "2": {
                "actions":{
                  "config":{
                    "forwarding-action":"DROP"
                  }
                },
                "config":{
```

```
                    "sequence-id":2
                    },
                  "transport":{
                    "config":{
                      "source-port":179
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
}
```

Below is the ACL configuration with four parameters. This ACL accepts the packets with source IPv6 address fe80::ec4:7aff:fe2e:1000/124, destination IPv6 address fe80::ec4:7aff:fe2e:2000/124, source port 179 and destination port 182.

```
{
   "acl": {
      "acl-sets": {
        "acl-set": {
          " ACL_Rules_1":{
            "acl-entries": {
              "acl-entry": {
                "9": {
                  "actions":{
                    "config":{
                      "forwarding-action":"ACCEPT"
                    }
                  },
                  "config":{
                    "sequence-id":9
                    },
                  "ip":{
                    "config":{
                      "source-ip-address":"fe80::ec4:7aff:fe2e:1000/124",
                      "destination-ip-address":"fe80::ec4:7aff:fe2e:2000/124",
                      "protocol":"IP_TCP"
                    }
                  },
                  "transport":{
                    "config":{
                      "source-port":179,
                      "destination-port":182
                    }
```

```
                }
              }
            }
          }
        }
      }
    }
  }
}
```

5.1.3 Show Commands for ACL

The show commands to check the ACL status and troubleshoot the ACL are given below.

Use the below command to display the ACL tables configured and the interface bindings of the tables.

```
show acl table [<table_name>]

Example-1:
root@sonic:~# show acl table
Name          Type   Binding      Description                Stage
------------  ------ ----------   ------------------------   -------
ACL_RULES_1   L3     Ethernet15   External access rules.     egress
                     Ethernet16
                     Ethernet17
                     Ethernet18
                     Ethernet19
                     Ethernet20
ACL_RULES_2   L3     Ethernet26   Finance/accounting dept.   ingress
                     Ethernet27
                     Ethernet28
                     Ethernet29
                     Ethernet30
root@sonic:~#

Example-2:
root@sonic:~# show acl table
Name          Type   Binding      Description                Stage
------------  ------ ----------   ------------------------   -------
ACL_RULES_1   L3V6   Ethernet15   External access rules.     egress
                     Ethernet16
                     Ethernet17
                     Ethernet18
                     Ethernet19
                     Ethernet20
ACL_RULES_2   L3V6   Ethernet26   Finance/accounting dept.   ingress
                     Ethernet27
                     Ethernet28
```

```
                    Ethernet29
                    Ethernet30
root@sonic:~#
```

Use the below command to display the ACL tables configured and the interface bindings of the tables.

```
Example-1:
root@sonic:~# show acl rule
Table       Rule    Priority    Action    Match
----------- ------ ---------- -------- ----------------------
ACL_RULES_1 RULE_1 9999       DROP     ETHER_TYPE: 2048
                           SRC_IP: 172.31.0.19/32
ACL_RULES_1 RULE_2 9998       DROP     ETHER_TYPE: 2048
                           L4_SRC_PORT: 179
root@sonic:~#

Example-2:
root@sonic:~# show acl rule
Table       Rule    Priority    Action    Match
----------- ------ ---------- -------- ----------------------
ACL_RULES_1 RULE_1 9999       DROP     ETHER_TYPE: 2048
                           SRC_IPV6: fe80::ec4:7aff:fe2e:1000/124
ACL_RULES_1 RULE_2 9998       DROP     ETHER_TYPE: 2048
                           L4_SRC_PORT: 179
root@sonic:~#
```

Use the below command to display the ACL counters.

```
root@sonic:~#  aclshow -a -vv
Reading ACL info...
Total number of ACL Tables: 1
Total number of ACL Rules: 2

RULE NAME     TABLE NAME      PRIO  PACKETS COUNT    BYTES COUNT
------------ ------------ ------ --------------- -------------
RULE_1       ACL_RULES_1     9999  116          12118
DEFAULT_RULE  ACL_RULES_1       1  156            6800
```

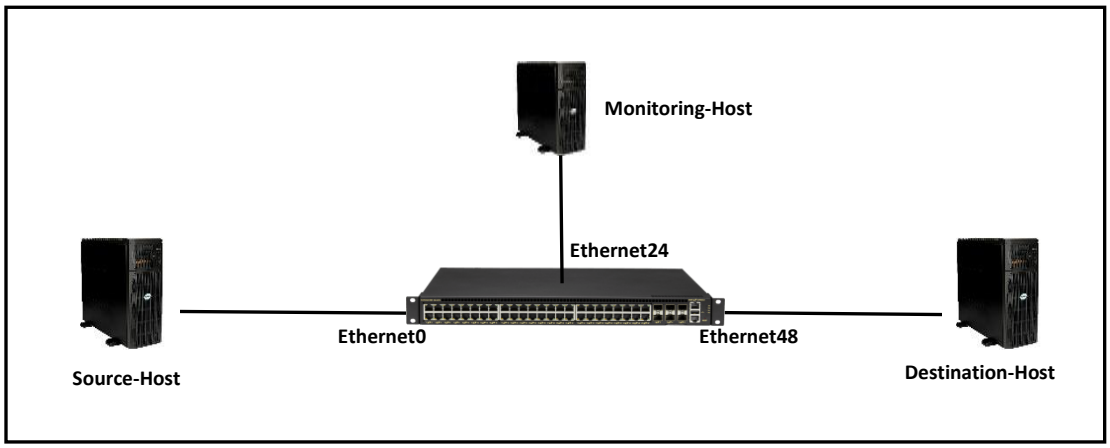
6 Port Mirroring

The port mirroring feature is a handy tool to use while debugging any complex issue in a network. When the port is mirrored, the switch sends a copy of the packets received and/or packets transmitted from the monitored port to the destination port. This helps to check whether the packet under study was actually received/transmitted by the port. In the networking world, the port mirroring has helped to identify the root cause in several long debugging sessions.

The port mirroring feature is for debugging. Enabling port mirroring may slow down the switch in high traffic conditions. So, use this feature with caution in production environments.

6.1.1 SPAN

The steps to create a SPAN mirror session is explained below.



In the above topology, say there is an issue in the traffic flow between the source host and the destination host and we suspect that the source host did not send the packet. In this scenario, the port Ethernet0 can be mirrored to Ethernet24 and all the traffic received/transmitted via Ethernet0 can be monitored.

Step	Command	Description
Step 1	**config mirror_session span add [OPTIONS] <session_name> <dst_port> [src_port]**	Creates a SPAN session. session_name – Name of the span session to be created. Dst_port – Destination port where the monitoring host is connected.

		Src_port – The source port that need to monitored.
Step 2	**config mirror_session remove [OPTIONS] <session_name>**	Deletes a mirror session
Step 3	**show mirror_session**	Displays the configured VLAN information.

The example below shows the command used to create SPAN session.

```
root@sonic:~# config mirror_session span add test-session Ethernet24 Ethernet0
```

The example below shows the command used to display the mirror session.

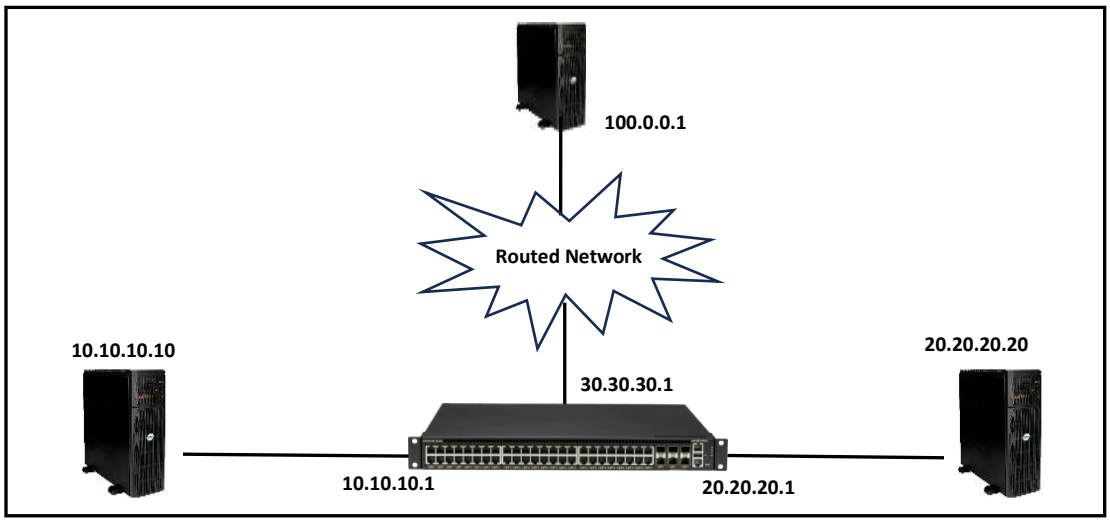
```
root@sonic:~# show mirror_session
ERSPAN Sessions
Name    Status    SRC IP    DST IP    GRE    DSCP    TTL    Queue    Policer    Monitor Port    SRC Port    Direction
------  --------  --------  --------  -----  ------  -----  -------  ---------  --------------  ----------  -----------

SPAN Sessions
Name         Status    DST Port    SRC Port    Direction    Queue    Policer
-----------  --------  ----------  ----------  -----------  -------  ---------
test-session  active    Ethernet24  Ethernet0   both
```

The example below shows the command used to delete a mirror session.

```
root@sonic:~# config mirror_session remove test-session
```

6.1.2 Everflow

The everflow can be called as next generation port mirroring. In the modern datacenter world, it may not be easy to get physical access to the switch and the switch ports. In these kind of remote work culture, everflow helps to study the ingress packets to a given switch port from a remote computer.



In the above topology, say there is an issue in the traffic flow between the source host 10.10.10.10 and the destination host 20.20.20.20 and we suspect that the source host 10.10.10.10 did not send the packet. In this scenario, the port Ethernet0 can be mirrored to Ethernet24 and all the traffic received by Ethernet0 can be monitored.

Step	Command	Description
Step 1	**config mirror_session add [OPTIONS] <session_name> <src_ip> <dst_ip> <dscp> <ttl> [gre_type] [queue]**	Creates a SPAN session. session_name – Name of the everflow session to be created. dst_ip – IP address of the monitoring host. src_ip – IP address of the switch to use as source IP in the forwarded packets. dscp – The DSCP value to be set for the forwarded packets. ttl – The ttl value to be set for the forwarded packets. gre_type –GRE type.
Step 2	**config mirror_session remove [OPTIONS] <session_name>**	Deletes a mirror session
Step 3	**show mirror_session**	Displays the configured VLAN information.

Following three steps are required to create an everflow monitoring session.

Step-1: Create ACL table

Step-2: Create ACL rule

Step-3: Create the everflow monitoring session

Step-1: Create ACL table

The steps for creating ACL table are explained detailly in Access Control Lists section. Below shown command is a simple example for quick reference.

The ACL table type must be set to MIRROR.

```
root@sonic:~# config acl add table Everflow-ACL-Table MIRROR --description 'ACL for Everflow mirror session' --stage ingress --ports Ethernet0
```

Step-2: Create ACL rule

To create ACL rule, first the JSON file has to be created with the rules and then the JSON file has to be loaded. The steps for creating ACL rule are explained detailly in Access Control Lists section. Below shown command is a simple example for quick reference.

In the below example, the TCP packets with priority '0', source IP 10.10.10.10 and destination IP 20.20.20.20 will be monitored. The value of the IP_PROTOCOL in the JSON file is the protocol number of the monitored protocol.

```
root@sonic:~# cat /tmp/ACL-for-everflow.json
{
   "ACL_RULE": {
      "Everflow-ACL-Table|Everflow_Rule": {
         "DST_IP": "20.20.20.20/24",
         "IP_PROTOCOL": "6",
         "MIRROR_ACTION": "Everflow_session",
         "PRIORITY": "0",
         "SRC_IP": "10.10.10.10/24"
      }
   }
}

root@sonic:~# config load /tmp/ACL-for-everflow.json
Load config from the file(s) /tmp/ACL-for-everflow.json ? [y/N]: y
Running command: /usr/local/bin/sonic-cfggen -j /tmp/ACL-for-everflow.json --write-to-db
root@sonic:~#
```

Step-3: Create the everflow monitoring session

In the example below, the everflow session is created with the destination IP 100.0.0.1. The packets selected by the ACL rule created in step-2 will be forwarded to the destination IP with ttl 255 and priority '0'. The name of the mirror session should be the same as the MIRROR_ACTION defined in the JSON file used in step-2.

```
root@sonic:~# config mirror_session add Everflow_session 30.30.30.1 100.0.0.1 0 255
```

The below commands can be used to check the status of the mirror session. Note the everflow session was created in three steps, so the ACL table, ACL rule and the mirror session should be checked if there is any problem.

Example command to check the ACL table.

```
root@sonic:~# show acl table
Name             Type    Binding    Description                  Stage    Status
---------------  ------  ---------  ---------------------------  -------  --------
Everflow-ACL-Table  MIRROR  Ethernet0  ACL for Everflow mirror session  ingress  Active
```

Example command to check the ACL rule.

```
root@sonic:~# show acl rule
Table       Rule         Priority   Action                    Match                Status
----------  -----------  ----------  --------------------------------  ---------------------  --------
Everflow-ACL-Table  Everflow_Rule  0       MIRROR INGRESS: Everflow_session  DST_IP: 20.20.20.20/24
N/A
                                                              IP_PROTOCOL: 6
                                                              SRC_IP: 10.10.10.10/24
```

Example command to check the mirror session.

```
root@sonic:~# show mirror_session
ERSPAN Sessions
Name           Status    SRC IP     DST IP    GRE     DSCP    TTL  Queue    Policer    Monitor Port    SRC Port
Direction
----------------  --------  ----------  ---------  -----  ------  -----  -------  ---------  --------------  ----------  -----------
Everflow_session  active  30.30.30.1  100.0.0.1          0    255                Ethernet24

SPAN Sessions
Name    Status    DST Port    SRC Port    Direction    Queue    Policer
------  --------  ----------  ----------  -----------  -------  ---------
```

If the next-hop ARP is not resolved for the destination IP, then the status will be displayed as inactive and the monitored port will be blank as shown below.

```
root@sonic:~# show mirror_session
ERSPAN Sessions
Name           Status    SRC IP     DST IP    GRE     DSCP    TTL  Queue    Policer    Monitor Port    SRC Port
Direction
----------------  --------  ----------  ---------  -----  ------  -----  -------  ---------  --------------  ----------  -----------
Everflow_session  inactive  30.30.30.1  100.0.0.1          0    255

SPAN Sessions
```

```
Name    Status    DST Port    SRC Port    Direction    Queue    Policer
------  --------  ----------  ----------  -----------  -------  ---------
```

7 MCLAG

Conventionally, all the members of a port-channel have to be terminated in the same switch. The Multi Chassis Link Aggregation Group (MCLAG) helps to terminate the members of a port-channel at two different switches. This provides redundancy for the port-channel if one of the switch fails.

MCLAG shall be configured to work at layer-3 or layer-2.

7.1 MCLAG Layer-3-IPv4

MCLAG Layer-3 configuration is given below. The layer-3 routing is a pre-requisite; so configure layer-3 routing either using static routes or by using a dynamic routing protocol. All the IP addresses should be reachable. The routing is out of scope of this section; for routing refer to Layer-3 configuration section in this document.

7.1.1 MCLAG Layer-3 Configuration-IPv4

Configuring MCLAG has four main steps and an optional step. These steps has to be configured on both MCLAG peer switches.

Step-1: Create port-channels and add member ports.

Step-2: Configure IP address to the port-channel interfaces.

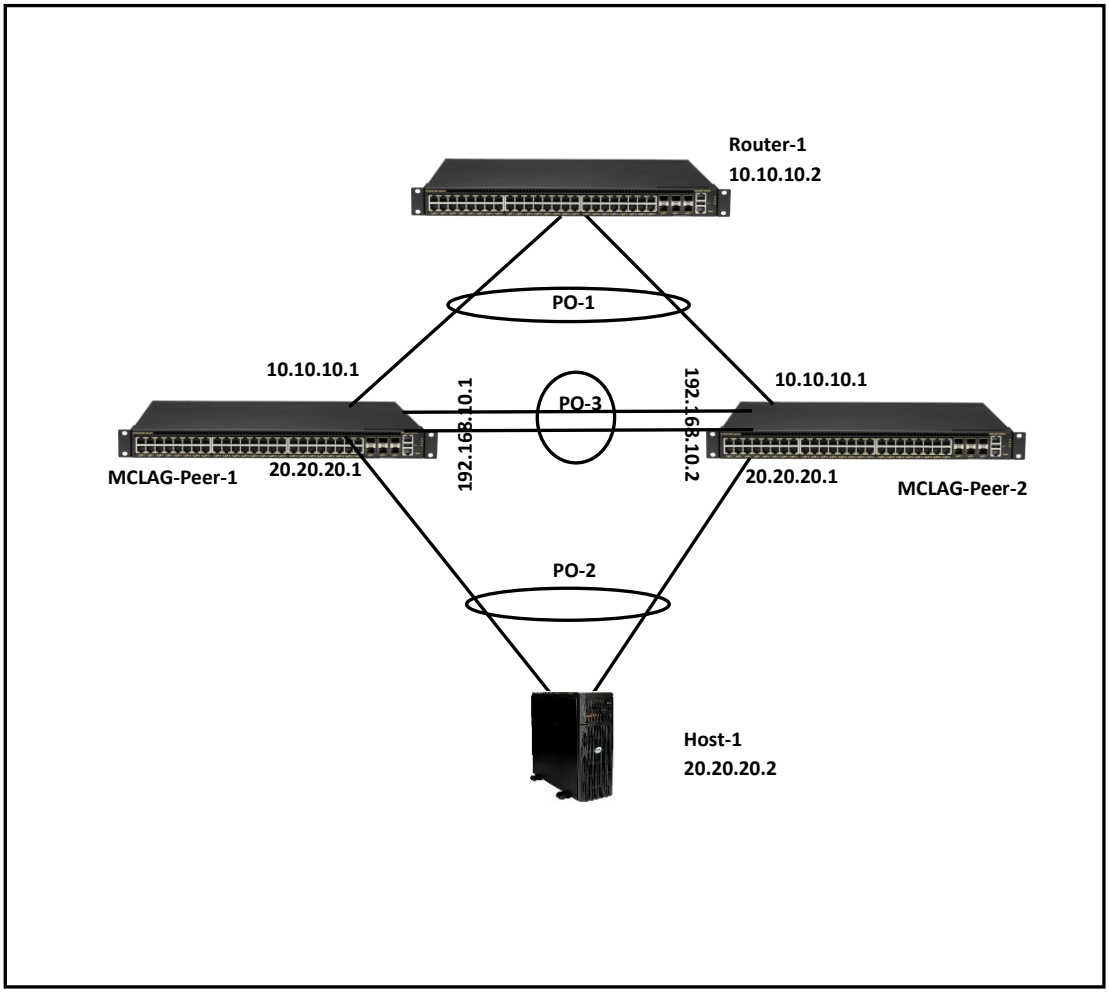
Step-3: Create MCLAG domain.

Step-4: Add MCLAG member port-channels to the MCLAG domain.

Step-5: Add static routes.

7.1.1.1 Sample Layer-3 IPv4 MCLAG Topology

The picture below shows a sample Layer-3 MCLAG topology.



7.1.1.2 Step-1: Create port-channels and add member ports

Creating the port-channels and adding member is explained in the link aggregation section of this document. The example given below is for quick reference. This step has to be completed on both MCLAG peer switches.

Example command to create port-channel and add member ports is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config portchannel add PortChannel01 config portchannel add PortChannel02 config portchannel add PortChannel03 config portchannel member add PortChannel01 Ethernet48 config portchannel member add PortChannel02 Ethernet49 config portchannel member add PortChannel03 Ethernet53	config portchannel add PortChannel01 config portchannel add PortChannel02 config portchannel add PortChannel03 config portchannel member add PortChannel01 Ethernet48 config portchannel member add PortChannel02 Ethernet49 config portchannel member add PortChannel03 Ethernet53

7.1.1.3 Step-2: Configure IP address to the port-channel interfaces

Configuring IP address to an interface is explained in the interface IP address configuration section of this document. The example given below is for quick reference. This step has to be completed on both MCLAG peer switches.

Example command to configure IP address for the port-channel is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config int ip add PortChannel01 10.10.10.1/24 config int ip add PortChannel02 20.20.20.1/24 config int ip add PortChannel03 192.168.10.1/24	config int ip add PortChannel01 10.10.10.1/24 config int ip add PortChannel02 20.20.20.1/24 config int ip add PortChannel03 192.168.10.2/24

7.1.1.4 Step-3: Create MCLAG domain

The MCLAG domain has to be created; the MCLAG domain will be identified with the domain-id. The IP address of the port-channel, which will serve as the peer-link will be used as source IP address. The IP address from the MCLAG peer switch on the other end of the peer-link port-channel will be used as the destination IP address. This step has to be completed on both MCLAG peer switches.

Follow the below steps to create MCLAG domain.

Step	Command	Description
Step 1	config mclag add [OPTIONS] <domain_id> <source_ip_addr> <peer_ip_addr> <peer_ifname>	Add MCLAG domain. domain_id – The MCLAG domain id. source_ip_addr – IP address of the port-channel.

		peer_ip_addr – Ports to bind the ACL table. peer_ifname – ingress/egress direction.
Step 2	sudo config save –y	Optional step - saves this configuration to be part of startup configuration.

Example command to create MCLAG domain is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config mclag add 1 192.168.10.1 192.168.10.2	config mclag add 1 192.168.10.2 192.168.10.1

7.1.1.5 Step-4: Add MCLAG member port-channels to the MCLAG domain

After the MCLAG domain is created, the MCLAG port-channels has to be added to the MCLAG domain. This step has to be completed on both MCLAG peer switches.

Follow the below steps to add member port-channels to MCLAG domain.

Step	**Command**	**Description**
Step 1	config mclag member add [OPTIONS] <domain_id> <portchannel_names>	Add member MCLAG interfaces. domain_id – The MCLAG domain id. portchannel_names – Name of the port-channel.
Step 2	sudo config save –y	Optional step - saves this configuration to be part of startup configuration.

Example command to add MCLAG port-channels to the domain is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config mclag member add 1 PortChannel01 config mclag member add 1 PortChannel02	config mclag member add 1 PortChannel01 config mclag member add 1 PortChannel02

7.1.1.6 Step-5: Add static routes

This step is not needed if there is a dynamic routing configured and running in the setup. If dynamic routing is not used, then the static routes are essential to forward the packets in failure cases. Please refer to the static route section for more details about static routes configuration.

Example command to add MCLAG port-channels to the domain is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config route add prefix 10.10.10.0/24 nexthop 192.168.10.2 config route add prefix 20.20.20.0/24 nexthop 192.168.10.2	config route add prefix 10.10.10.0/24 nexthop 192.168.10.1 config route add prefix 20.20.20.0/24 nexthop 192.168.10.1

7.1.1.7 MCLAG Show Commands

The commands to check the MCLAG status and to debug the MCLAG issues are given below.

The command to display the current state of the MCLAG.

```
mclagdctl dump state

Example:
root@sonic:~# mclagdctl dump state
The MCLAG's keepalive is: OK
MCLAG info sync is: completed
Domain id: 1
Local Ip: 192.168.10.2
Peer Ip: 192.168.10.1
Peer Link Interface: Unknown
Keepalive time: 1
sesssion Timeout : 15
Peer Link Mac: 00:00:00:00:00:00
Role: Standby
MCLAG Interface: PortChannel02,PortChannel01
Loglevel: NOTICE
root@sonic:~#
```

The command to display the ARP entries of the MCLAG.

```
mclagdctl dump arp -i <mclag-domain-id>

Example:
root@sonic:~# mclagdctl dump arp -i 1
No.  IP             MAC               DEV            Flag
1    192.168.10.1   88:5a:85:fa:2a:d1  PortChannel03  L
2    20.20.20.10    ac:1f:6b:38:73:52  PortChannel02  R
```

```
3    10.10.10.10       ac:1f:6b:38:6e:f9   PortChannel01          R
4    20.20.20.11       ac:1f:6b:1b:7d:e3   PortChannel02          R
5    10.10.10.11       ac:1f:6b:59:39:6c   PortChannel01          R
root@sonic:~#
```

The command to display the MAC addresses learnt by the MCLAG.

```
mclagdctl dump mac -i <mclag-domain-id>

Example:
root@sonic:~# mclagdctl dump mac -i 1
TYPE: S-STATIC, D-DYNAMIC; AGE: L-Local age, P-Peer age
No.  TYPE MAC               VID  DEV              ORIGIN-DEV        AGE
1    D   ac:1f:6b:38:6e:f9   10   PortChannel01      PortChannel01      L
2    D   ac:1f:6b:38:6e:fa   10   PortChannel01      PortChannel01      P
3    D   ac:1f:6b:59:39:6c   10   PortChannel01      PortChannel01      L
4    D   ac:1f:6b:1b:7d:e3   20   PortChannel02      PortChannel02      L
5    D   ac:1f:6b:38:73:52   20   PortChannel02      PortChannel02      L
6    D   ac:1f:6b:38:73:53   20   PortChannel02      PortChannel02      P
7    S   88:5a:85:fa:2a:d1   100  PortChannel03      PortChannel03      L
root@sonic:~#
```

The command to list the local MCLAG ports.

```
mclagdctl dump portlist local -i <mclag-domain-id>

Example:
root@sonic:~# mclagdctl dump portlist local -i 1
------------------------------------------------------------
Ifindex: 13
Type: Ethernet
PortName: Ethernet1
State: Up
VlanList:
------------------------------------------------------------

------------------------------------------------------------
Ifindex: 12
Type: Ethernet
PortName: Ethernet0
State: Up
VlanList:
------------------------------------------------------------
```

```
------------------------------------------------------------
Ifindex: 5
Type: PortChannel
PortName: PortChannel02
MAC: 88:5a:85:fa:2a:d1
IPv4Address: 0.0.0.0
Prefixlen: 32
State: Up
IsL3Interface: No
MemberPorts: Ethernet1
PortchannelIsUp: 1
IsIsolateWithPeerlink: Yes
IsTrafficDisable: No
VlanList: 20
------------------------------------------------------------

------------------------------------------------------------
Ifindex: 4
Type: PortChannel
PortName: PortChannel01
MAC: 88:5a:85:fa:2a:d1
IPv4Address: 0.0.0.0
Prefixlen: 32
State: Up
IsL3Interface: No
MemberPorts: Ethernet0
PortchannelIsUp: 1
IsIsolateWithPeerlink: Yes
IsTrafficDisable: No
VlanList: 10
------------------------------------------------------------

------------------------------------------------------------
Ifindex: 6
Type: PortChannel
PortName: PortChannel03
MAC: 0c:c4:7a:2e:16:6d
IPv4Address: 0.0.0.0
Prefixlen: 32
State: Up
IsL3Interface: No
MemberPorts: Ethernet53
PortchannelIsUp: 1
IsIsolateWithPeerlink: No
IsTrafficDisable: No
VlanList: 10 20 100
------------------------------------------------------------
```

```
root@sonic:~#
```

The command to list the remote MCLAG ports.

```
mclagdctl dump portlist peer -i <mclag-domain-id>

Example:
root@sonic:~# mclagdctl dump portlist peer -i 1
------------------------------------------------------------
Ifindex: 1
Type: PortChannel
PortName: PortChannel01
MAC: 88:5a:85:fa:2a:d1
State: Up
------------------------------------------------------------


------------------------------------------------------------
Ifindex: 2
Type: PortChannel
PortName: PortChannel02
MAC: 88:5a:85:fa:2a:d1
State: Up
------------------------------------------------------------
root@sonic:~#
```

7.2 MCLAG Layer-3-IPv6

Steps to configure layer-3 MCLAG Layer-3 is given below. The layer-3 routing is a pre-requisite; so configure layer-3 routing either using static routes or by using a dynamic routing protocol. All the IP addresses should be reachable. The routing is out of scope of this section; for routing refer to Layer-3 configuration section in this document.

7.2.1 MCLAG Layer-3 Configuration-IPv6

NOTE: THE IP(s) on the peer-switches for Control-Plane traffic should still be IPv4 in addition to IPv6

Configuring MCLAG has four main steps and an optional step. These steps has to be configured on both MCLAG peer switches.

Step-1: Create port-channels and add member ports.

Step-2: Configure IP address to the port-channel interfaces.

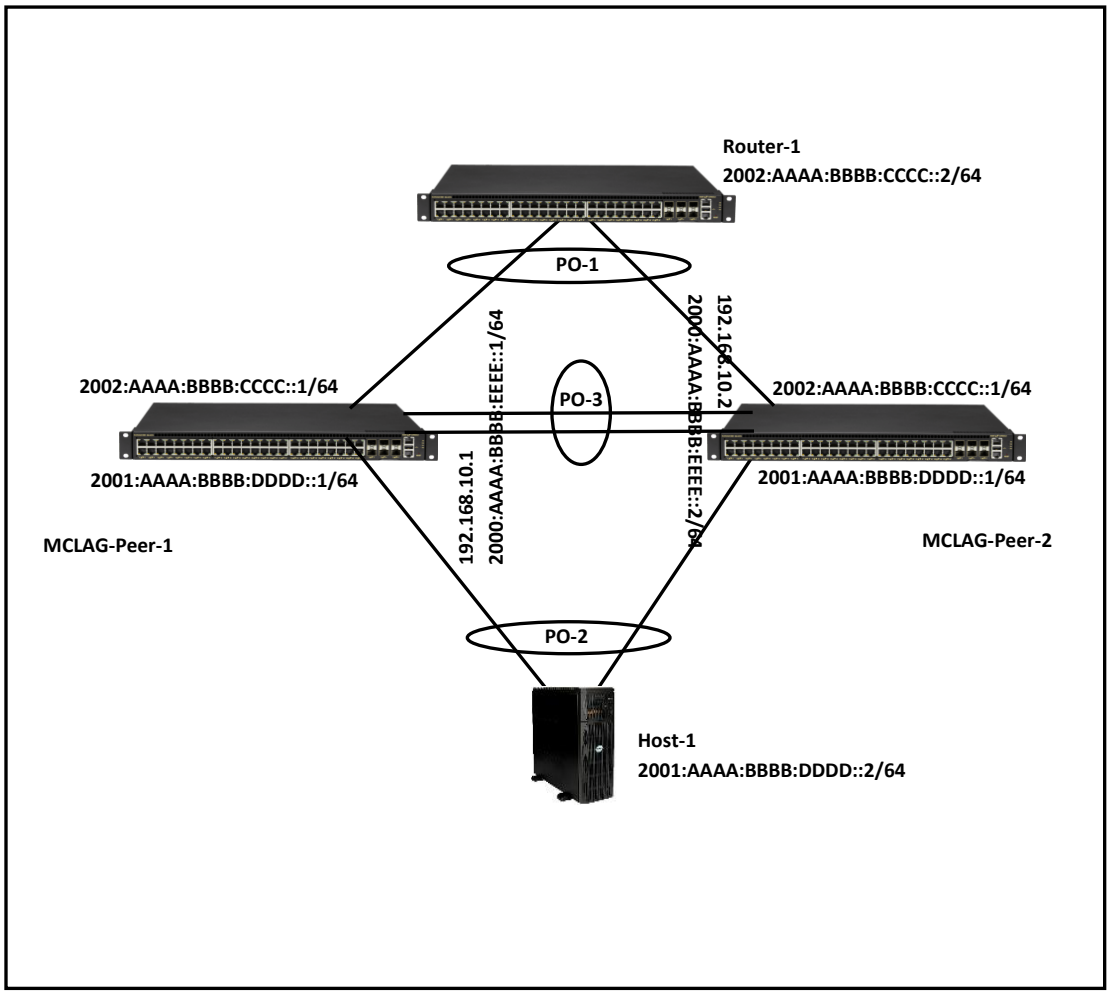
Step-3: Create MCLAG domain.

Step-4: Add MCLAG member port-channels to the MCLAG domain.

Step-5: Add static routes.

7.2.1.1 Sample Layer-3 IPv6 MCLAG Topology

The picture below shows a sample Layer-3 MCLAG topology.



7.2.1.2 Step-1: Create port-channels and add member ports

Creating the port-channels and adding member is explained in the link aggregation section of this document. The example given below is for quick reference. This step has to be completed on both MCLAG peer switches.

Example command to create port-channel and add member ports is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config portchannel add PortChannel01 config portchannel add PortChannel02 config portchannel add PortChannel03 config portchannel member add PortChannel01 Ethernet48 config portchannel member add PortChannel02 Ethernet49 config portchannel member add PortChannel03 Ethernet53	config portchannel add PortChannel01 config portchannel add PortChannel02 config portchannel add PortChannel03 config portchannel member add PortChannel01 Ethernet48 config portchannel member add PortChannel02 Ethernet49 config portchannel member add PortChannel03 Ethernet53

7.2.1.3 Step-2: Configure IP address to the port-channel interfaces

Configuring IP address to an interface is explained in the interface IP address configuration section of this document. The example given below is for quick reference. This step has to be completed on both MCLAG peer switches.

Example command to configure IP address for the port-channel is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config int ip add PortChannel01 2002:aaaa:bbbb:cccc::1/64 config int ip add PortChannel02 2001:aaaa:bbbb:dddd::1/64 config int ip add PortChannel03 192.168.10.1/24 config int ip add PortChannel03 2000:aaaa:bbbb:eeee::1/64	config int ip add PortChannel01 2002:aaaa:bbbb:cccc::1/64 config int ip add PortChannel02 2001:aaaa:bbbb:dddd::1/64 config int ip add PortChannel03 192.168.10.2/24 config int ip add PortChannel03 2000:aaaa:bbbb:eeee::2/64

7.2.1.4 Step-3: Create MCLAG domain

The MCLAG domain has to be created; the MCLAG domain will be identified with the domain-id. The IP address of the port-channel, which will serve as the peer-link will be used as source IP address. The IP address from the MCLAG peer switch on the other end of the peer-link port-channel will be used as the destination IP address. This step has to be completed on both MCLAG peer switches.

Follow the below steps to create MCLAG domain.

Step	Command	Description
Step 1	config mclag add [OPTIONS] <domain_id> <source_ip_addr> <peer_ip_addr> <peer_ifname>	Add MCLAG domain. domain_id – The MCLAG domain id. source_ip_addr – IP address of the port-channel. peer_ip_addr – Ports to bind the ACL table. peer_ifname – ingress/egress direction.
Step 2	sudo config save –y	Optional step - saves this configuration to be part of startup configuration.

Example command to create MCLAG domain is given below.

MCLAG Switch - 1	MCLAG Switch - 2

config mclag add 1 192.168.10.1 192.168.10.2	config mclag add 1 192.168.10.2 192.168.10.1

7.2.1.5 Step-4: Add MCLAG member port-channels to the MCLAG domain

After the MCLAG domain is created, the MCLAG port-channels has to be added to the MCLAG domain. This step has to be completed on both MCLAG peer switches.

Follow the below steps to add member port-channels to MCLAG domain.

Step	Command	Description
Step 1	config mclag member add [OPTIONS] <domain_id> <portchannel_names>	Add member MCLAG interfaces. domain_id – The MCLAG domain id. portchannel_names – Name of the port-channel.
Step 2	sudo config save –y	Optional step - saves this configuration to be part of startup configuration.

Example command to add MCLAG port-channels to the domain is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config mclag member add 1 PortChannel01 config mclag member add 1 PortChannel02	config mclag member add 1 PortChannel01 config mclag member add 1 PortChannel02

7.2.1.6 Step-5: Add static routes

This step is not needed if there is a dynamic routing configured and running in the setup. If dynamic routing is not used, then the static routes are essential to forward the packets in failure cases. Please refer to the static route section for more details about static routes configuration.

Example command to add MCLAG port-channels to the domain is given below.

MCLAG Switch - 1	MCLAG Switch - 2
vtysh configure terminal ipv6 route 2002:AAAA:BBBB:CCCC::/64 2000:AAAA:BBBB:EEEE::2 ipv6 route 2001:AAAA:BBBB:DDDD::1/64 2000:AAAA:BBBB:EEEE::2	vtysh configure terminal ipv6 route 2002:AAAA:BBBB:CCCC::/64 2000:AAAA:BBBB:EEEE::1 ipv6 route 2001:AAAA:BBBB:DDDD::1/64 2000:AAAA:BBBB:EEEE::1

7.2.1.7 MCLAG Show Commands

The commands to check the MCLAG status and to debug the MCLAG issues are given below.

The command to display the current state of the MCLAG.

```
mclagdctl dump state

Example:
root@sonic:~# mclagdctl dump state
The MCLAG's keepalive is: OK
MCLAG info sync is: completed
Domain id: 1
Local Ip: 192.168.10.2
Peer Ip: 192.168.10.1
Peer Link Interface: Unknown
Keepalive time: 1
sesssion Timeout : 15
Peer Link Mac: 00:00:00:00:00:00
Role: Standby
MCLAG Interface: PortChannel02,PortChannel01
Loglevel: NOTICE
root@sonic:~#
```

7.3 MCLAG Layer-2

MCLAG Layer-2 configuration is given below. The layer-3 routing is a pre-requisite; so configure layer-3 routing either using static routes or by using a dynamic routing protocol. All the IP addresses should be reachable. The routing is out of scope of this section; for routing refer to Layer-3 configuration section in this document.

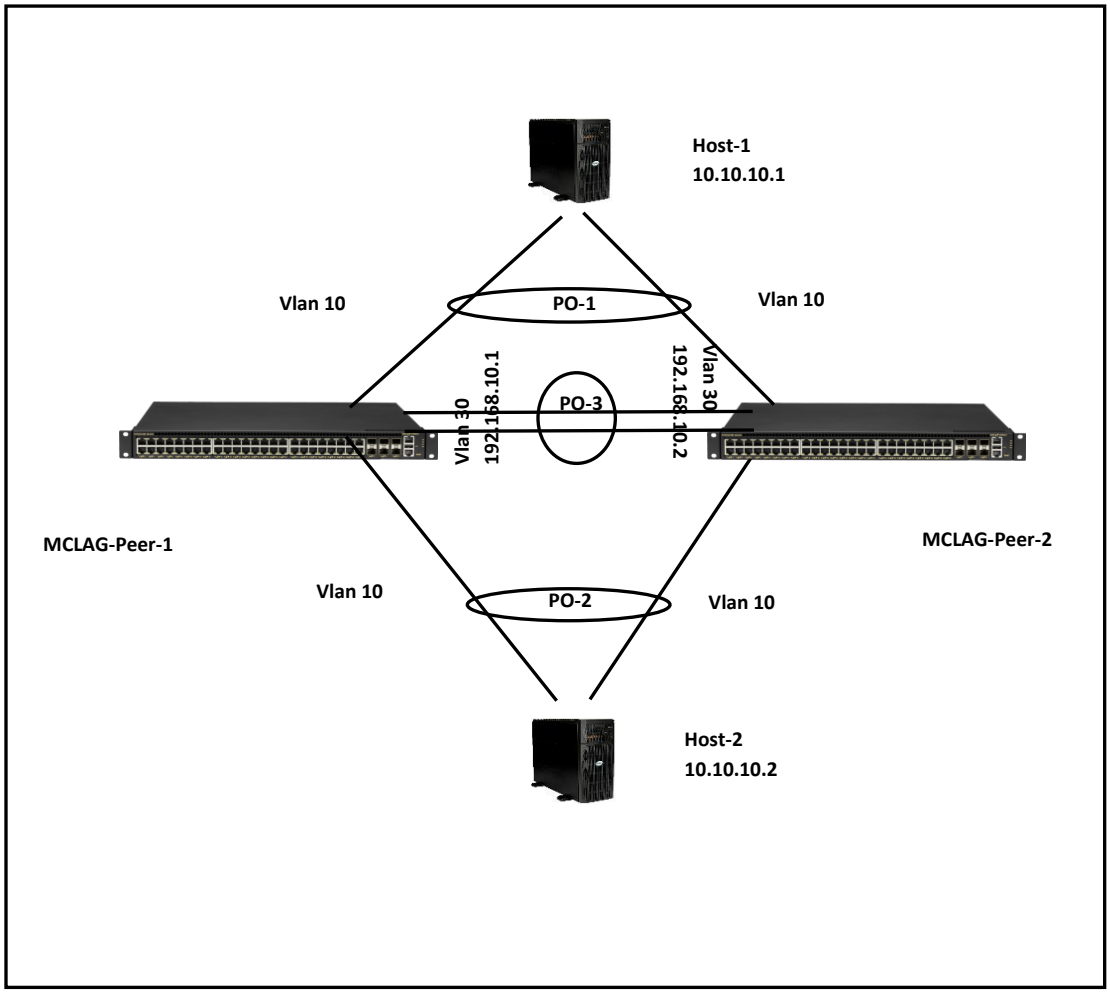
7.3.1 MCLAG Configuration Combination of Layer-2 & Layer-3- IPv4

Configuring MCLAG has six steps. These steps have to be followed on both the MCLAG peer switches.

Step-1: Create port-channels
Step-2: Create VLANS
Step-3: Remove IP addresses associated with the relevant interfaces
Step-4: Add port-channel members
Step-5: Add VLAN members and assign IP
Step-6: Create MCLAG domain and assign unique-IP
Step-7: Add member port-channels to the MCLAG domain

7.3.1.1 Sample Layer-2 MCLAG Topology

The picture below shows a sample L2 MCLAG topology



7.3.1.2 Step-1: Create port-channels

Creating the port-channels is explained in the link aggregation section of this document. The example given below is for quick reference. This step has to be completed on both MCLAG peer switches.

Example command to create port-channel is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config portchannel add PortChannel01 config portchannel add PortChannel02 config portchannel add PortChannel03	config portchannel add PortChannel01 config portchannel add PortChannel02 config portchannel add PortChannel03

7.3.1.3 Step-2: Create VLANS

Creating VLANS is explained in the VLAN configuration section of this document. The example given below is for quick reference. This step has to be completed on both MCLAG peer switches.

Example command to Create VLANS is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config vlan add 10 config vlan add 30	config vlan add 10 config vlan add 30

7.3.1.4 Step-3: Remove IP addresses associated with the relevant interfaces

The interfaces on SONiC, by default, are configured as routed ports. The interfaces have default IP address that need to be removed to make them function as L2 ports. The following command is used to remove the associated IP addresses. This step has to be completed on both MCLAG peer switches.

Example command to remove IP address is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config int ip rem Ethernet48 10.0.0.96/31 config int ip rem Ethernet49 10.0.0.98/31 config int ip rem Ethernet53 10.0.0.106/31	config int ip rem Ethernet48 10.0.0.96/31 config int ip rem Ethernet49 10.0.0.98/31 config int ip rem Ethernet53 10.0.0.106/31

7.3.1.5 Step-4: Add Port-Channel members

Adding port-channel member is explained in the link aggregation section of this document. The example given below is for quick reference. This step has to be completed on both MCLAG peer switches.

Example command to add port-channel member ports is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config portchannel member add PortChannel01 Ethernet48 config portchannel member add PortChannel02 Ethernet49 config portchannel member add PortChannel03 Ethernet53	config portchannel member add PortChannel01 Ethernet48 config portchannel member add PortChannel02 Ethernet49 config portchannel member add PortChannel03 Ethernet53

7.3.1.6 Step-5: Add VLAN members and assign IP

Adding member ports to a VLAN is explained in the VLAN configuration section. Configuring IP is explained in the Interface Properties section. The example given below is for quick reference. This step has to be completed on both MCLAG peer switches.

Example command to add VLAN members and configuring VLAN IP is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config vlan member add 30 PortChannel03 config vlan member add -u 10 PortChannel01 config vlan member add -u 10 PortChannel02 config vlan member add 10 PortChannel03	config vlan member add 30 PortChannel03 config vlan member add -u 10 PortChannel01 config vlan member add -u 10 PortChannel02 config vlan member add 10 PortChannel03

config int ip add Vlan30 192.168.10.1/24	config int ip add Vlan30 192.168.10.2/24

7.3.1.7 Step-6: Create MCLAG domain and assign unique-IP

The MCLAG domain has to be created; the MCLAG domain will be identified with the domain-id. The IP address of the port-channel, which will serve as the peer-link will be used as source IP address. The IP address from the MCLAG peer switch on the other end of the peer-link port-channel will be used as the destination IP address. The unique-IP will be used to forward MCLAG control-traffic to the peer switch. This step has to be completed on both MCLAG peer switches.

Follow the below steps to create MCLAG domain and assign unique-ip.

Step	Command	Description
Step 1	config mclag add [OPTIONS] <domain_id> <source_ip_addr> <peer_ip_addr> <peer_ifname>	Add MCLAG domain. domain_id – The MCLAG domain id. source_ip_addr – IP address of the port-channel. peer_ip_addr – Ports to bind the ACL table. peer_ifname – ingress/egress direction.
Step 2	sudo config save –y	Optional step - saves this configuration to be part of startup configuration.

Example command to create MCLAG domain and configuring a unique-ip is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config mclag add 1 192.168.10.1 192.168.10.2 config mclag unique-ip add Vlan3	config mclag add 1 192.168.10.2 192.168.10.1 config mclag unique-ip add Vlan3

7.3.1.8 Step-7: Add MCLAG member port-channels to the MCLAG domain

After the MCLAG domain is created, the MCLAG port-channels have to be added to the MCLAG domain. This step has to be completed on both MCLAG peer switches.

Follow the below steps to add member port-channels to MCLAG domain.

Step	Command	Description
Step 1	config mclag member add [OPTIONS] <domain_id> <portchannel_names>	Add member MCLAG interfaces.

		domain_id – The MCLAG domain id. portchannel_names – Name of the port-channel.
Step 2	sudo config save –y	Optional step - saves this configuration to be part of startup configuration.

Example command to add MCLAG port-channels to the domain is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config mclag member add 1 PortChannel01 config mclag member add 1 PortChannel02	config mclag member add 1 PortChannel01 config mclag member add 1 PortChannel02

7.4 MCLAG Combination of Layer-2 & Layer-3

In some deployments, both layer-2 and layer-3 MCLAG are deployed. In most setups inter VLAN routing will be required. If your setup requires inter VLAN routing, then the layer-3 routing is a pre-requisite; so configure layer-3 routing either using static routes or by using a dynamic routing protocols. All the IP addresses should be reachable. The routing is out of scope of this section; for routing refer to Layer-3 configuration section in this document.

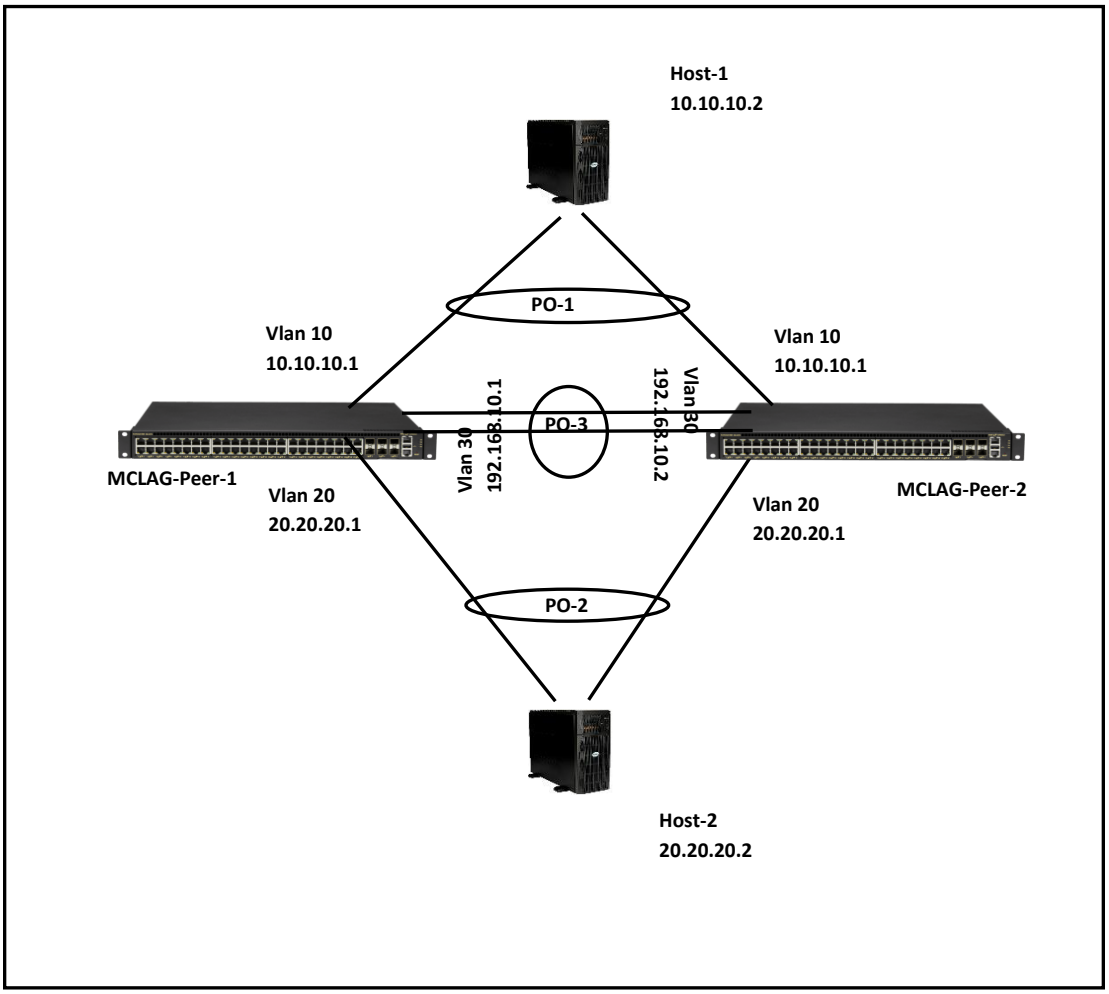
7.4.1 MCLAG Configuration Combination of Layer-2 & Layer-3- IPv4

Configuring MCLAG has six steps. These steps have to be followed on both the MCLAG peer switches.

Step-1: Create port-channels
Step-2: Create VLANS
Step-3: Remove IP addresses associated with the relevant interfaces
Step-4: Add port-channel members
Step-5: Add VLAN members and assign IP addresses
Step-6: Create MCLAG domain and assign unique-IP
Step-7: Add member port-channels to the MCLAG domain

7.4.1.1 Combination of Layer-2 and Layer-3 MCLAG Topology-IPv4

The picture below shows a sample MCLAG topology with Layer-2 and Layer-3.



7.4.1.2 Step-1: Create port-channels

Creating the port-channels is explained in the link aggregation section of this document. The example given below is for quick reference. This step has to be completed on both MCLAG peer switches.

Example command to create port-channel is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config portchannel add PortChannel01 config portchannel add PortChannel02 config portchannel add PortChannel03	config portchannel add PortChannel01 config portchannel add PortChannel02 config portchannel add PortChannel03

7.4.1.3 Step-2: Create VLANS

Creating VLANS has been explained in the VLAN configuration section of this document. The example given below is for quick reference. This step has to be completed on both MCLAG peer switches.

Example command to Create VLANS is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config vlan add 10 config vlan add 20 config vlan add 30	config vlan add 10 config vlan add 20 config vlan add 30

7.4.1.4 Step-3: Remove IP addresses associated with the relevant interfaces

The interfaces on SONiC, by default, are configured as routed ports. The interfaces have default IP address that need to be removed to make them function as L2 ports. The following commands are used to remove the associated IP addresses. This step has to be completed on both MCLAG peer switches.

Example command to remove IP address is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config int ip rem Ethernet48 10.0.0.96/31 config int ip rem Ethernet49 10.0.0.98/31 config int ip rem Ethernet53 10.0.0.106/31	config int ip rem Ethernet48 10.0.0.96/31 config int ip rem Ethernet49 10.0.0.98/31 config int ip rem Ethernet53 10.0.0.106/31

7.4.1.5 Step-4: Add Port-Channel members

Adding port-channel member is explained in the link aggregation section of this document. The example given below is for quick reference. This step has to be completed on both MCLAG peer switches.

Example command to add port-channel member ports is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config portchannel member add PortChannel01 Ethernet48 config portchannel member add PortChannel02 Ethernet49 config portchannel member add PortChannel03 Ethernet53	config portchannel member add PortChannel01 Ethernet48 config portchannel member add PortChannel02 Ethernet49 config portchannel member add PortChannel03 Ethernet53

7.4.1.6 Step-5: Add VLAN members and assign IP addresses

Adding member(s) to a VLAN has been explained in the VLAN configuration section. Configuring IP has been explained in the Interface Properties section. The example given below is for quick reference. This step has to be completed on both MCLAG peer switches.

Example command to add VLAN members and configuring VLAN IP is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config vlan member add 30 PortChannel03 config vlan member add -u 10 PortChannel01 config vlan member add 10 PortChannel03	config vlan member add 30 PortChannel03 config vlan member add -u 10 PortChannel01 config vlan member add 10 PortChannel03

config vlan member add -u 20 PortChannel02 config vlan member add 20 PortChannel03 config int ip add Vlan10 10.10.10.1/24 config int ip add Vlan20 20.20.20.1/24 config int ip add Vlan30 192.168.10.1/24	config vlan member add -u 20 PortChannel02 config vlan member add 20 PortChannel03 config int ip add Vlan10 10.10.10.1/24 config int ip add Vlan20 20.20.20.1/24 config int ip add Vlan30 192.168.10.2/24

7.4.1.7 Step-6: Create MCLAG domain and assign unique-IP

The MCLAG domain has to be created; the MCLAG domain will be identified with the domain-id. The IP address of the port-channel, which will serve as the peer-link will be used as source IP address. The IP address from the MCLAG peer switch on the other end of the peer-link port-channel will be used as the destination IP address. The unique-ip will be used to forward control-traffic to the peer switch. This step has to be completed on both MCLAG peer switches.

Follow the below steps to create MCLAG domain and assign unique-IP.

Step	Command	Description
Step 1	config mclag add [OPTIONS] <domain_id> <source_ip_addr> <peer_ip_addr> <peer_ifname>	Add MCLAG domain. domain_id – The MCLAG domain id. source_ip_addr – IP address of the port-channel. peer_ip_addr – Ports to bind the ACL table. peer_ifname – ingress/egress direction.
Step 2	sudo config save –y	Optional step - saves this configuration to be part of startup configuration.

Example command to create MCLAG domain and configuring a unique-ip is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config mclag add 1 192.168.10.1 192.168.10.2 config mclag unique-ip add Vlan3	config mclag add 1 192.168.10.2 192.168.10.1 config mclag unique-ip add Vlan3

7.4.1.8 Step-7: Add MCLAG member port-channels to the MCLAG domain

After the MCLAG domain is created, the MCLAG port-channels have to be added to the MCLAG domain. This step has to be completed on both MCLAG peer switches.

Follow the below steps to add member port-channels to MCLAG domain.

Step	Command	Description
Step 1	config mclag member add [OPTIONS] <domain_id> <portchannel_names>	Add member MCLAG interfaces. domain_id – The MCLAG domain id. portchannel_names – Name of the port-channel.
Step 2	sudo config save –y	Optional step - saves this configuration to be part of startup configuration.

Example command to add MCLAG port-channels to the domain is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config mclag member add 1 PortChannel01 config mclag member add 1 PortChannel02	config mclag member add 1 PortChannel01 config mclag member add 1 PortChannel02

7.4.2 MCLAG Configuration Combination of Layer-2 & Layer-3- IPv6

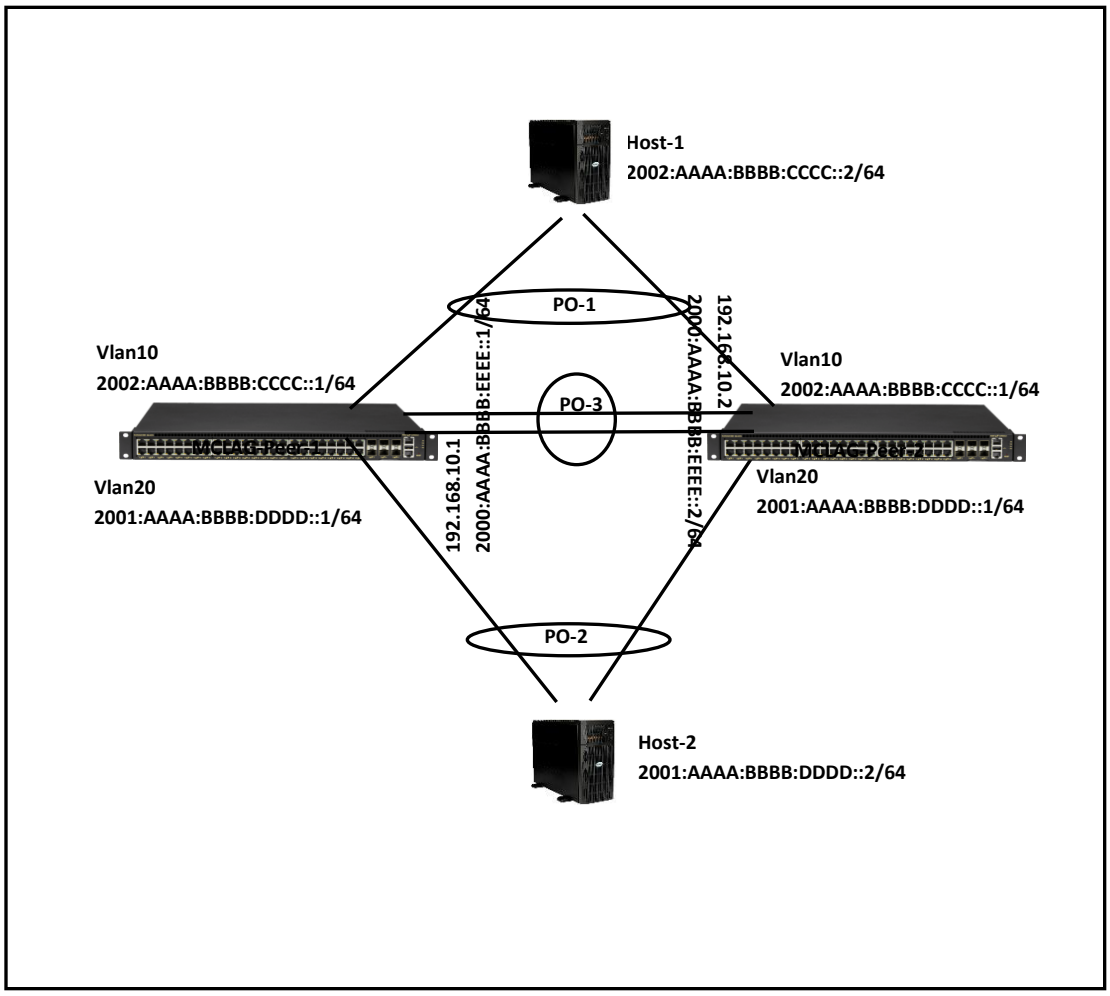
NOTE: THE IP(s) on the peer-switches for Control-Plane traffic should still be IPv4

Configuring MCLAG has six steps. These steps have to be followed on both the MCLAG peer switches.

Step-1: Create port-channels
Step-2: Create VLANS
Step-3: Remove IP addresses associated with the relevant interfaces
Step-4: Add port-channel members
Step-5: Add VLAN members and assign IP addresses
Step-6: Create MCLAG domain and assign unique-IP
Step-7: Add member port-channels to the MCLAG domain

7.4.2.1 Combination of Layer-2 and Layer-3 MCLAG Topology-IPv6

The picture below shows a sample MCLAG topology with Layer-2 and Layer-3.



7.4.2.2 Step-1: Create port-channels

Creating the port-channels is explained in the link aggregation section of this document. The example given below is for quick reference. This step has to be completed on both MCLAG peer switches.

Example command to create port-channel is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config portchannel add PortChannel01 config portchannel add PortChannel02 config portchannel add PortChannel03	config portchannel add PortChannel01 config portchannel add PortChannel02 config portchannel add PortChannel03

7.4.2.3 Step-2: Create VLANS

Creating VLANS has been explained in the VLAN configuration section of this document. The example given below is for quick reference. This step has to be completed on both MCLAG peer switches.

Example command to Create VLANS is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config vlan add 10 config vlan add 20 config vlan add 30	config vlan add 10 config vlan add 20 config vlan add 30

7.4.2.4 Step-3: Remove IP addresses associated with the relevant interfaces

The interfaces on SONiC, by default, are configured as routed ports. The interfaces have default IP address that need to be removed to make them function as L2 ports. The following commands are used to remove the associated IP addresses. This step has to be completed on both MCLAG peer switches..

Example command to remove IP address is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config int ip rem Ethernet48 10.0.0.96/31 config int ip rem Ethernet49 10.0.0.98/31 config int ip rem Ethernet53 10.0.0.106/31	config int ip rem Ethernet48 10.0.0.96/31 config int ip rem Ethernet49 10.0.0.98/31 config int ip rem Ethernet53 10.0.0.106/31

7.4.2.5 Step-4: Add Port-Channel members

Adding port-channel member is explained in the link aggregation section of this document. The example given below is for quick reference. This step has to be completed on both MCLAG peer switches.

Example command to add port-channel member ports is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config portchannel member add PortChannel01 Ethernet48 config portchannel member add PortChannel02 Ethernet49 config portchannel member add PortChannel03 Ethernet53	config portchannel member add PortChannel01 Ethernet48 config portchannel member add PortChannel02 Ethernet49 config portchannel member add PortChannel03 Ethernet53

7.4.2.6 Step-5: Add VLAN members and assigning IP(s)

Adding member(s) to a VLAN is explained in the VLAN configuration section. Configuring IP is explained in the Interface Properties section. The example given below is for quick reference. This step has to be completed on both MCLAG peer switches.

Example command to add VLAN members and configuring VLAN IP is given below.

MCLAG Switch - 1	MCLAG Switch - 2

config vlan member add 30 PortChannel03 config vlan member add -u 10 PortChannel01 config vlan member add 10 PortChannel03 config vlan member add -u 20 PortChannel02 config vlan member add 20 PortChannel03 config int ip add Vlan10 2002:aaaa:bbbb:cccc::1/64 config int ip add Vlan20 2001:aaaa:bbbb:dddd::1/64 config int ip add Vlan30 192.168.10.1/24 config int ip add Vlan30 2000:aaaa:bbbb:eeee::1/64	config vlan member add 30 PortChannel03 config vlan member add -u 10 PortChannel01 config vlan member add 10 PortChannel03 config vlan member add -u 20 PortChannel02 config vlan member add 20 PortChannel03 config int ip add Vlan10 2002:aaaa:bbbb:cccc::1/64 config int ip add Vlan20 2001:aaaa:bbbb:dddd::1/64 config int ip add Vlan30 192.168.10.2/24 config int ip add Vlan30 2000:aaaa:bbbb:eeee::2/64

7.4.2.7 Step-6: Create MCLAG domain and assign unique-IP

The MCLAG domain has to be created; the MCLAG domain will be identified with the domain-id. The IP address of the port-channel, which will serve as the peer-link will be used as source IP address. The IP address from the MCLAG peer switch on the other end of the peer-link port-channel will be used as the destination IP address. The unique-ip will be used to forward control-traffic to the peer switch. This step has to be completed on both MCLAG peer switches.

Follow the below steps to create MCLAG domain and assign unique-ip.

Step	Command	Description
Step 1	config mclag add [OPTIONS] <domain_id> <source_ip_addr> <peer_ip_addr> <peer_ifname>	Add MCLAG domain. domain_id – The MCLAG domain id. source_ip_addr – IP address of the port-channel. peer_ip_addr – Ports to bind the ACL table. peer_ifname – ingress/egress direction.
Step 2	sudo config save –y	Optional step - saves this configuration to be part of startup configuration.

Example command to create MCLAG domain and configuring a unique-IP is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config mclag add 1 192.168.10.1 192.168.10.2	config mclag add 1 192.168.10.2 192.168.10.1

config mclag unique-ip add Vlan3	config mclag unique-ip add Vlan3

7.4.2.8 Step-7: Add MCLAG member port-channels to the MCLAG domain

After the MCLAG domain is created, the MCLAG port-channels have to be added to the MCLAG domain. This step has to be completed on both MCLAG peer switches.

Follow the below steps to add member port-channels to MCLAG domain.

Step	Command	Description
Step 1	config mclag member add [OPTIONS] <domain_id> <portchannel_names>	Add member MCLAG interfaces. domain_id – The MCLAG domain id. portchannel_names – Name of the port-channel.
Step 2	sudo config save –y	Optional step - saves this configuration to be part of startup configuration.

Example command to add MCLAG port-channels to the domain is given below.

MCLAG Switch - 1	MCLAG Switch - 2
config mclag member add 1 PortChannel01 config mclag member add 1 PortChannel02	config mclag member add 1 PortChannel01 config mclag member add 1 PortChannel02

7.4.2.9 MCLAG Show Commands

The commands to check the MCLAG status and to debug the MCLAG issues are given below.

The command to display the current state of the MCLAG.

```
mclagdctl dump state

Example:
The MCLAG's keepalive is: OK
MCLAG info sync is: completed
Domain id: 1
Local Ip: 192.168.10.1
Peer Ip: 192.168.10.2
Peer Link Interface: PortChannel03
Keepalive time: 1
sesssion Timeout : 15
Peer Link Mac: 0c:c4:7a:3e:18:2d
Role: Active
MCLAG Interface: PortChannel02,PortChannel01
Loglevel: NOTICE
root@sonic:~#
```