Supermicro DataCenter SONiC

Configuration Guide

Revision 1.5

The information in this USER’S GUIDE has been carefully reviewed and is believed to be accurate. The vendor assumes no responsibility for any inaccuracies that may be contained in this document, makes no commitment to update or to keep current the information in this manual, or to notify any person organization of the updates. Please Note: For the most up-to-date version of this manual, please see our web site at www.supermicro.com.
Super Micro Computer, Inc. ("Supermicro") reserves the right to make changes to the product described in this manual at any time and without notice. This product, including software, if any, and documentation may not, in whole or in part, be copied, photocopied, reproduced, translated or reduced to any medium or machine without prior written consent.

DISCLAIMER OF WARRANTY ON SOFTWARE AND MATERIALS. You expressly acknowledge and agree that use of the Software and Materials is at your sole risk. FURTHERMORE, SUPER MICRO COMPUTER INC. DOES NOT WARRANT OR MAKE ANY REPRESENTATIONS REGARDING THE USE OR THE RESULTS OF THE USE OF THE SOFTWARE OR MATERIALS IN TERMS OF THEIR CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE. NO ORAL OR WRITTEN INFORMATION OR ADVICE GIVEN BY SUPER MICRO COMPUTER INC. OR SUPER MICRO COMPUTER INC. AUTHORIZED REPRESENTATIVE SHALL CREATE A WARRANTY OR IN ANY WAY INCREASE THE SCOPE OF THIS WARRANTY. SHOULD THE SOFTWARE AND/OR MATERIALS PROVE DEFECTIVE, YOU (AND NOT SUPER MICRO COMPUTER INC. OR A SUPER MICRO COMPUTER INC. AUTHORIZED REPRESENTATIVE) ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICE, REPAIR, OR CORRECTION.

LIMITATION OF LIABILITY. UNDER NO CIRCUMSTANCES INCLUDING NEGLIGENCE, SHALL SUPER MICRO COMPUTER INC. BE LIABLE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES THAT RESULT FROM THE USE OR INABILITY TO USE THE SOFTWARE OR MATERIALS, EVEN IF SUPER MICRO COMPUTER INC. OR A SUPER MICRO COMPUTER INC. AUTHORIZED REPRESENTATIVE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any disputes arising between manufacturer and customer shall be governed by the laws of Santa Clara County in the State of California, USA. The State of California, County of Santa Clara shall be the exclusive venue for the resolution of any such disputes. Super Micro’s total liability for all claims will not exceed the price paid for the hardware product.

Manual Revision 1.5

Release Date: 09-Nov-2023

Unless you request and receive written permission from Super Micro Computer, Inc., you may not copy any part of this document.

Information in this document is subject to change without notice. Other products and companies referred to herein are trademarks or registered trademarks of their respective companies or mark holders.

Copyright © by Super Micro Computer, Inc.
All rights reserved

Printed in the United States of America
### Document Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>09/30/2022</td>
<td>1.0</td>
<td>First release.</td>
</tr>
<tr>
<td>01/09/2023</td>
<td>1.1</td>
<td>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.</td>
</tr>
<tr>
<td>03/23/2023</td>
<td>1.2</td>
<td>Added route reflector, route reflector configuration and save the BGP configuration sections. Updated NTP section with IPv6 configurations. Added ACL and MCLAG section.</td>
</tr>
<tr>
<td>04/07/2023</td>
<td>1.3</td>
<td>Added IPv6 ACL section. Updated default Auto-negotiation and MCLAG show commands sections.</td>
</tr>
<tr>
<td>06/15/2023</td>
<td>1.4</td>
<td>Add Annex 1 for T7132</td>
</tr>
<tr>
<td>11/06/2023</td>
<td>1.5</td>
<td>Annex 1 is removed and it’s contents are merged with appropriate sections. Updated MCLAG with L2 and IPv6 configurations. Added port mirroring section.</td>
</tr>
</tbody>
</table>
Contents

1 Introduction ............................................................................................................................ 10

1.1 Switch initial configuration .............................................................................................. 10

1.1.1 Console Port .................................................................................................................. 10

1.1.2 Initial Switch Access ..................................................................................................... 11

1.1.3 Simple L2 Switch Configuration .................................................................................. 11

1.2 Definitions and Acronyms ............................................................................................... 13

1.3 Introduction about Switch Models ................................................................................... 15

1.3.1 SSE-G3748 .................................................................................................................. 15

1.3.2 SSE-T7132 .................................................................................................................. 15

2 System Configuration ........................................................................................................... 15

2.1 Management IP ................................................................................................................ 16

2.1.1 Interface IP Address Configuration ............................................................................. 16

2.1.2 Gateway Configuration ............................................................................................... 17

2.2 Management Access ....................................................................................................... 17

2.2.1 Defaults ...................................................................................................................... 17

2.2.2 Configure User ........................................................................................................... 17

2.2.3 Modify User ................................................................................................................ 19

2.2.4 Remove User .............................................................................................................. 19

2.3 Interface Properties ....................................................................................................... 20

2.3.1 Defaults ...................................................................................................................... 20

2.3.2 Description ................................................................................................................ 21

2.3.3 Auto-negotiation ......................................................................................................... 24

2.3.4 Forward Error Correction (FEC) Mode ....................................................................... 27

2.3.5 Speed .......................................................................................................................... 29

2.3.6 Shutdown / Startup ..................................................................................................... 30

2.3.7 MTU ............................................................................................................................ 31

2.3.8 Advertised-speed ....................................................................................................... 32

2.3.9 Advertised-type .......................................................................................................... 33

2.3.10 Configure IPv4 address ............................................................................................ 35

2.3.11 Remove IPv4 address ............................................................................................... 36

2.3.12 Configure IPv6 address ............................................................................................ 36
2.4.13 Remove IPv6 address ........................................................................................................... 38
2.4.14 Configure IPv6 address for Management Interface ........................................................... 39
2.4.15 Remove Management Interface IPv6 .................................................................................. 41
2.4.16 Enable IPv6 Link Local ..................................................................................................... 42
2.4.17 Disable IPv6 Link Local ..................................................................................................... 43
2.4.18 MAC .................................................................................................................................. 45
2.4.19 Type .................................................................................................................................. 46
2.4.20 Alias ................................................................................................................................. 47
2.4.21 Configure Interface Naming Mode ..................................................................................... 48
2.4.22 Counters ......................................................................................................................... 48
2.4.23 Configure loopback .......................................................................................................... 52
2.4.24 Remove loopback ............................................................................................................. 53
2.4.25 Storm Control .................................................................................................................. 54
2.4.26 Port splitting/HWSKU in SSE-T7132 ................................................................................ 55

2.4 System Management ........................................................................................................... 59
   2.4.1 System clock .................................................................................................................... 59
   2.4.2 Host Name ...................................................................................................................... 60
   2.4.3 Display version ............................................................................................................... 60
   2.4.4 Display environment ....................................................................................................... 61
   2.4.5 Display reboot-cause ...................................................................................................... 62
   2.4.6 Display uptime ................................................................................................................ 63
   2.4.7 Display logging ............................................................................................................... 63
   2.4.8 Display platform summary ............................................................................................ 64
   2.4.9 Display system EEPROM ............................................................................................... 64
   2.4.10 Display power supply units .......................................................................................... 65
   2.4.11 Display device’s fans .................................................................................................... 66
   2.4.12 Display device’s thermal sensors .................................................................................. 66
   2.4.13 System State ................................................................................................................ 66
   2.4.14 Troubleshooting ........................................................................................................... 68
   2.4.15 Display Services ........................................................................................................... 70
   2.4.16 Display System-health .................................................................................................. 71
   2.4.17 Display System-memory .............................................................................................. 75
2.5 Security Features ............................................................................................................. 75
  2.5.1 AAA......................................................................................................................... 75
  2.5.2 RADIUS ................................................................................................................... 77
  2.5.3 TACACS .................................................................................................................... 89
2.6 Configuration Management .............................................................................................. 97
  2.6.1 Save Startup-Config .................................................................................................. 97
  2.6.2 Save Running Configuration to File ........................................................................... 97
  2.6.3 Erase Startup-Config ................................................................................................. 98
  2.6.4 Reset-to-factory Defaults ......................................................................................... 98
  2.6.5 Boot-up options ....................................................................................................... 98
  2.6.6 Warm Reboot ........................................................................................................... 99
2.7 Switch features ............................................................................................................... 99
  2.7.1 Defaults .................................................................................................................... 99
  2.7.2 Configure state ........................................................................................................ 100
  2.7.3 Configure auto-restart ............................................................................................ 101
2.8 Reload ............................................................................................................................ 102
  2.8.1 Reload configuration ............................................................................................... 102
  2.8.2 Configure load ......................................................................................................... 103
2.9 SNMP ............................................................................................................................. 104
  2.9.1 Defaults .................................................................................................................... 104
  2.9.2 Configure SNMP Agent Address ............................................................................. 104
  2.9.3 Configure SNMP Trap ............................................................................................. 105
  2.9.4 Configure SNMP location ....................................................................................... 106
  2.9.5 Modify SNMP location ............................................................................................ 106
  2.9.6 Remove SNMP location .......................................................................................... 107
  2.9.7 Configure SNMP contact ......................................................................................... 108
  2.9.8 Modify SNMP contact ............................................................................................. 108
  2.9.9 Remove SNMP contact ............................................................................................ 109
  2.9.10 Configure SNMP community ............................................................................... 110
  2.9.11 Modify SNMP community ..................................................................................... 110
  2.9.12 Remove SNMP community .................................................................................... 111
  2.9.13 Configure SNMP users .......................................................................................... 112
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.9.14</td>
<td>Remove SNMP users</td>
<td>113</td>
</tr>
<tr>
<td>2.10</td>
<td>NTP</td>
<td>114</td>
</tr>
<tr>
<td>2.10.1</td>
<td>Configure NTP server Address</td>
<td>114</td>
</tr>
<tr>
<td>2.10.2</td>
<td>Delete NTP server</td>
<td>115</td>
</tr>
<tr>
<td>2.10.3</td>
<td>Configure Time Zone</td>
<td>116</td>
</tr>
<tr>
<td>2.11</td>
<td>System Logging (Syslog)</td>
<td>117</td>
</tr>
<tr>
<td>2.11.1</td>
<td>Configure syslog</td>
<td>117</td>
</tr>
<tr>
<td>2.11.2</td>
<td>Delete syslog</td>
<td>118</td>
</tr>
<tr>
<td>2.12</td>
<td>Zero Touch Provisioning (ZTP)</td>
<td>119</td>
</tr>
<tr>
<td>2.12.1</td>
<td>DHCP Scope Options to add in DHCP Server</td>
<td>119</td>
</tr>
<tr>
<td>2.12.2</td>
<td>Add Files to TFTP/HTTP Server</td>
<td>120</td>
</tr>
<tr>
<td>2.13</td>
<td>Firmware Upgrade</td>
<td>122</td>
</tr>
<tr>
<td>2.13.1</td>
<td>Upgrading from SONiC CLI</td>
<td>122</td>
</tr>
<tr>
<td>2.13.2</td>
<td>Upgrading from ONIE</td>
<td>122</td>
</tr>
<tr>
<td>3</td>
<td>Layer2 Configuration</td>
<td>124</td>
</tr>
<tr>
<td>3.1</td>
<td>VLAN</td>
<td>124</td>
</tr>
<tr>
<td>3.1.1</td>
<td>VLAN Numbers</td>
<td>124</td>
</tr>
<tr>
<td>3.1.2</td>
<td>VLAN Defaults</td>
<td>124</td>
</tr>
<tr>
<td>3.1.3</td>
<td>Creating VLANs</td>
<td>125</td>
</tr>
<tr>
<td>3.1.4</td>
<td>Removing VLANs</td>
<td>125</td>
</tr>
<tr>
<td>3.1.5</td>
<td>Port Based VLANs</td>
<td>126</td>
</tr>
<tr>
<td>3.1.6</td>
<td>VLAN Configuration Example</td>
<td>131</td>
</tr>
<tr>
<td>3.2</td>
<td>Link Aggregation</td>
<td>135</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Creating Port channels</td>
<td>136</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Remove Member Ports from a port channel</td>
<td>138</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Removing Port channels</td>
<td>138</td>
</tr>
<tr>
<td>3.2.4</td>
<td>Link Aggregation Configuration Example</td>
<td>140</td>
</tr>
<tr>
<td>3.3</td>
<td>LLDP</td>
<td>142</td>
</tr>
<tr>
<td>3.3.1</td>
<td>LLDP Overview</td>
<td>142</td>
</tr>
<tr>
<td>3.3.2</td>
<td>LLDP Configuration</td>
<td>142</td>
</tr>
<tr>
<td>3.3.3</td>
<td>LLDP Configuration Example</td>
<td>146</td>
</tr>
<tr>
<td>4</td>
<td>Layer3 Configuration</td>
<td>148</td>
</tr>
</tbody>
</table>
4.1 DHCP Relay ................................................................. 148
  4.1.1 IPv4 DHCP Relay .................................................. 149
  4.1.2 IPv6 DHCP Relay .................................................. 151
4.2 Layer3 VLAN Interface .................................................... 152
  4.2.1 Add an IP address for a VLAN interface ......................... 153
  4.2.2 Remove an IP address from a VLAN interface .................. 154
  4.2.3 Inter-VLAN Routing .............................................. 155
4.3 Static route .................................................................. 157
4.4 ARP ................................................................................. 158
4.5 BGP .............................................................................. 160
  4.5.1 EBGP ...................................................................... 160
  4.5.2 IBGP ...................................................................... 160
  4.5.3 Router ID ................................................................. 160
  4.5.4 Speaker and Peer ...................................................... 160
  4.5.5 Autonomous System (AS) ........................................... 160
  4.5.6 Attributes ................................................................ 161
  4.5.7 Filters ..................................................................... 161
  4.5.8 Synchronization ....................................................... 161
  4.5.9 BGP Path selection ................................................... 161
  4.5.10 Timers ..................................................................... 162
  4.5.11 BGP Route Reflector ................................................ 162
  4.5.12 BGP Configuration ................................................... 162
  4.5.13 BGP Configuration Example ...................................... 167
  4.5.14 Route Reflector Configuration .................................... 173
  4.5.15 BGP IPv6 Configuration ............................................ 174
4.6 Route Map ................................................................. 174
  4.6.1 Configure route-map .................................................. 174
5 Access Control Lists ................................................... 179
  5.1 IP Access Control List .................................................. 179
  5.1.1 IPv4 Access Control List ............................................ 179
  5.1.2 IPv6 Access Control List ............................................ 196
  5.1.3 Show Commands for ACL ........................................... 213
Port Mirroring ................................................................. 215
6.1.1 SPAN ........................................................................... 215

MCLAG .............................................................................. 221
7.1 MCLAG Layer-3-Ipv4 ...................................................... 221
7.1.1 MCLAG Layer-3 Configuration-Ipv4 ......................... 221
7.2 MCLAG Layer-3-Ipv6 ...................................................... 228
7.2.1 MCLAG Layer-3 Configuration-Ipv6 ......................... 228
7.3 MCLAG Layer-2 ................................................................ 232
7.3.1 MCLAG Configuration Combination of Layer-2 & Layer-3- Ipv4 ........................................ 232
7.4 MCLAG Combination of Layer-2 & Layer-3 .................... 236
7.4.1 MCLAG Configuration Combination of Layer-2 & Layer-3- Ipv4 ........................................ 236
7.4.2 MCLAG Configuration Combination of Layer-2 & Layer-3- Ipv6 ........................................ 240
1 Introduction

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

Software for Open Networking in the Cloud (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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management IP</td>
<td>DHCP</td>
</tr>
<tr>
<td>Login username</td>
<td>admin</td>
</tr>
<tr>
<td>Password</td>
<td>YourPaSSWoRd</td>
</tr>
<tr>
<td>Serial Baud rate</td>
<td>115200</td>
</tr>
</tbody>
</table>

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
```
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

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>DHCP</td>
<td>Dynamic Host Configuration Protocol</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>MTU</td>
<td>Maximum Transmission Unit</td>
</tr>
<tr>
<td>NTP</td>
<td>Network Time Protocol</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>TTL</td>
<td>Time to live</td>
</tr>
<tr>
<td>DSCP</td>
<td>Differentiated Services Code Point</td>
</tr>
<tr>
<td>TLV</td>
<td>Type Length Value</td>
</tr>
<tr>
<td>TACACS</td>
<td>Terminal Access Controller Access Control System</td>
</tr>
<tr>
<td>SNMP</td>
<td>Simple Network Management Protocol</td>
</tr>
<tr>
<td>VLAN</td>
<td>Virtual LAN</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>PVID</td>
<td>Port VLAN ID</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Interface Name</th>
<th>Interface Numbers</th>
<th>Speed</th>
<th>MTU</th>
<th>Autoneg</th>
<th>FEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>400 Gigabit ethernet</td>
<td>Ethernet0 – Ethernet248</td>
<td>400G</td>
<td>9100</td>
<td>Disabled</td>
<td>RS</td>
</tr>
<tr>
<td>10 Gigabit ethernet</td>
<td>Ethernet256 – Ethernet257</td>
<td>10G only</td>
<td>9100</td>
<td>Disabled</td>
<td>None</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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
```
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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name/Password</td>
<td>admin/YourPaSsWoRd</td>
</tr>
</tbody>
</table>

#### 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 1

**useradd [options] LOGIN**

- **useradd -D [options]**

  - **Create new user.**
  - **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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>usermod [options] [LOGIN]</code></td>
<td>Usermod - Modify user&lt;br&gt;LOGIN - new value of the login name&lt;br&gt;NOTE: Refer Linux manual for options related to modify user.</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>sudo users</code></td>
<td>Display the users currently logged in to the switch.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>show users</code></td>
<td>List of users currently logged in to the switch along with the IP address of the login session.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>userdel [options] LOGIN</code></td>
<td>Userdel - Delete user</td>
</tr>
</tbody>
</table>
### LOGIN

The login name of the user to be removed.

NOTE: Refer Linux manual for options related to delete user.

<table>
<thead>
<tr>
<th>Step 2</th>
<th>sudo users</th>
<th>Display current user</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step3</td>
<td>show users</td>
<td>List of users currently logged in to the device</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Interface Name</th>
<th>Interface Numbers</th>
<th>Speed</th>
<th>MTU</th>
<th>Autoneg</th>
<th>FEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gigabit ethernet</td>
<td>Ethernet0, Ethernet47</td>
<td>1G default, Can operate in 100Mb</td>
<td>9100</td>
<td>Enabled</td>
<td>N/A</td>
</tr>
<tr>
<td>Fx-ethernet</td>
<td>Ethernet48, Ethernet53</td>
<td>25G default, Can operate in 10G/1G</td>
<td>9100</td>
<td>Enabled</td>
<td>None</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>show interfaces description [interface_name]</code></td>
<td>Displays the interface description configuration. Interface name - may be any of the following: Ethernet0, Ethernet53</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Interface</th>
<th>Oper</th>
<th>Admin</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet33</td>
<td>down</td>
<td>up</td>
<td>Gi0/34</td>
<td></td>
</tr>
</tbody>
</table>

**Output from SSE-T7132:**

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

<table>
<thead>
<tr>
<th>Interface</th>
<th>Oper</th>
<th>Admin</th>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet0</td>
<td>down</td>
<td>up</td>
<td>Eth1</td>
<td></td>
</tr>
<tr>
<td>Ethernet8</td>
<td>down</td>
<td>up</td>
<td>Eth2</td>
<td></td>
</tr>
<tr>
<td>Ethernet16</td>
<td>down</td>
<td>up</td>
<td>Eth3</td>
<td></td>
</tr>
<tr>
<td>Ethernet24</td>
<td>down</td>
<td>up</td>
<td>Eth4</td>
<td></td>
</tr>
<tr>
<td>Ethernet32</td>
<td>down</td>
<td>up</td>
<td>Eth5</td>
<td></td>
</tr>
<tr>
<td>Ethernet40</td>
<td>down</td>
<td>up</td>
<td>Eth6</td>
<td></td>
</tr>
<tr>
<td>Ethernet48</td>
<td>down</td>
<td>up</td>
<td>Eth7</td>
<td></td>
</tr>
<tr>
<td>Ethernet56</td>
<td>down</td>
<td>up</td>
<td>Eth8</td>
<td></td>
</tr>
<tr>
<td>Ethernet64</td>
<td>down</td>
<td>up</td>
<td>Eth9</td>
<td></td>
</tr>
<tr>
<td>Ethernet72</td>
<td>down</td>
<td>up</td>
<td>Eth10</td>
<td></td>
</tr>
<tr>
<td>Ethernet80</td>
<td>down</td>
<td>up</td>
<td>Eth11</td>
<td></td>
</tr>
<tr>
<td>Ethernet88</td>
<td>down</td>
<td>up</td>
<td>Eth12</td>
<td></td>
</tr>
<tr>
<td>Ethernet96</td>
<td>up</td>
<td>up</td>
<td>Eth13</td>
<td></td>
</tr>
<tr>
<td>Ethernet104</td>
<td>down</td>
<td>up</td>
<td>Eth14</td>
<td></td>
</tr>
<tr>
<td>Ethernet112</td>
<td>down</td>
<td>up</td>
<td>Eth15</td>
<td></td>
</tr>
<tr>
<td>Ethernet120</td>
<td>down</td>
<td>up</td>
<td>Eth16</td>
<td></td>
</tr>
<tr>
<td>Ethernet128</td>
<td>down</td>
<td>up</td>
<td>Eth17</td>
<td></td>
</tr>
<tr>
<td>Ethernet136</td>
<td>down</td>
<td>up</td>
<td>Eth18</td>
<td></td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>config interface autoneg &lt;interface_name&gt; &lt;mode&gt;</code></td>
<td>Turn on/off the auto-negotiation. Interface name - may be any of the following interfaces: Ethernet0 – Ethernet53. <code>&lt;mode&gt;</code> - Enabled/disabled.</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>show interface autoneg status</code></td>
<td>Displays the auto-negotiation status for all interfaces.</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>admin@sonic:~$ sudo config interface autoneg Ethernet0 enabled</code></td>
</tr>
<tr>
<td>`admin@sonic:~$ show interface autoneg status</td>
</tr>
<tr>
<td><code>Ethernet0 enabled 1G N/A N/A N/A up up</code></td>
</tr>
<tr>
<td><code>admin@sonic:~$ sudo config interface autoneg Ethernet50 disabled</code></td>
</tr>
<tr>
<td>`admin@sonic:~$ show interface autoneg status</td>
</tr>
<tr>
<td><code>Ethernet50 disabled 25G N/A N/A N/A down up</code></td>
</tr>
</tbody>
</table>

### 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/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
  - id: "241"  # Devport ID
    fec: "KPFEC"  # FEC type for devport 241
    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"
  - id: "241"
    fec: "KPFEC"  # Not mandatory when auto-nego is true
    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_neg is true
    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-solomon (RS), and Fire-code (FC) FECs. By default on all Fx-ethernet ports the FEC is set RS.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>config interface fec &lt;interface_name&gt;</code></td>
<td>Configure FEC on the interface.</td>
</tr>
<tr>
<td></td>
<td><code>&lt;interface_fec&gt;</code></td>
<td><code>&lt;interface_name&gt;</code> - may be any of the following interfaces:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet48 – Ethernet53</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>show interface status</code></td>
<td>Displays the interface status for all interfaces with the current FEC.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>sudo config save -y</code></td>
<td>Optional step - Saves this current configuration to be part of startup</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>MAC Speed</th>
<th>PCS-FEC</th>
<th>Lanes</th>
<th>Start Lane</th>
<th>Restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>400GbE</td>
<td>CL119 PCS with KP-FEC</td>
<td>8</td>
<td>0</td>
<td>None</td>
</tr>
<tr>
<td>Speed</td>
<td>Type</td>
<td>Devports</td>
<td>FEC Configuration</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------------------</td>
<td>----------</td>
<td>-------------------</td>
<td></td>
</tr>
<tr>
<td>200GbE</td>
<td>CL119 PCS with KP-FEC</td>
<td>8, 0</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>200GbE</td>
<td>CL119 PCS with KP-FEC</td>
<td>4, 0, 4</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>100GbE</td>
<td>CL91 PCS with KP-FEC/KR-FEC</td>
<td>2, 0, 2, 4, 6</td>
<td>0, 2: no KR-FEC mixed with KP-FEC 4, 6: no KR-FEC mixed with KP-FEC</td>
<td></td>
</tr>
<tr>
<td>100GbE</td>
<td>CL91 PCS with KP-FEC/KR-FEC</td>
<td>4, 0, 4</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>100GbE</td>
<td>CL82 PCS with no FEC</td>
<td>4, 0, 4</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>50GbE</td>
<td>CL133 PCS with KP-FEC/KR-FEC</td>
<td>1, 0, 1, 2, 3, 7</td>
<td>0, 1, 2, 3: no KR-FEC mixed with KP-FEC 4, 5, 6, 7: no KR-FEC mixed with KP-FEC</td>
<td></td>
</tr>
<tr>
<td>50GbE</td>
<td>CL133PCS with KP-FEC/KR-FEC/no FEC</td>
<td>2, 0, 2, 4, 6</td>
<td>0, 2: no KR-FEC mixed with KP-FEC 4, 6: no KR-FEC mixed with KP-FEC</td>
<td></td>
</tr>
<tr>
<td>25GbE</td>
<td>CL107/25GEC PCS with KR-FEC/FC-FEC/no FEC</td>
<td>1, 0, 1, 2, 3, 7</td>
<td>0, 1, 2, 3: no KR-FEC mixed with KP-FEC 4, 5, 6, 7: no KR-FEC mixed with KP-FEC</td>
<td></td>
</tr>
<tr>
<td>40GbE</td>
<td>CL82 PCS with no FEC</td>
<td>4, 0, 4</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>10GbE</td>
<td>CL49 PCS with no FEC</td>
<td>1, 0, 1, 2, 3, 7</td>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

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.

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

Supermicro SONiC Configuration Guide v1.4
- 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
```

<table>
<thead>
<tr>
<th>Interface</th>
<th>Lanes</th>
<th>Speed</th>
<th>MTU</th>
<th>FEC</th>
<th>Alias</th>
<th>Vlan</th>
<th>Oper</th>
<th>Admin</th>
<th>Type</th>
<th>Asym</th>
<th>PFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet0</td>
<td>241,242,243,244,245,246,247,248</td>
<td>400G</td>
<td>9100</td>
<td>none</td>
<td>Eth1</td>
<td>down</td>
<td>up</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>

Supermicro SONiC Configuration Guide v1.4 30
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.

```bash
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.

```bash
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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>config interface mtu &lt;interface_name&gt; &lt;mtu_value&gt;</code></td>
<td>Configures interface mtu. Interface name – may be any of the following: Ethernet0, Ethernet53. Mtu value – maximum transmission unit.</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>show interface status</code></td>
<td>Displays the interface configuration.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>sudo config save –y</code></td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>config interface autoneg [OPTIONS] &lt;interface_name&gt; &lt;mode&gt;</code></td>
<td>Configures interface advertised-speeds. Interface name – may be any of the following: Ethernet0, Ethernet53.</td>
</tr>
<tr>
<td></td>
<td><code>config interface advertised-speeds &lt;interface_name&gt; &lt;speed_list&gt;</code></td>
<td></td>
</tr>
</tbody>
</table>

---

Supermicro SONiC Configuration Guide v1.4 32
### Mode
- **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
```

---

#### 2.3.9 Advertised-type
Follow the steps below to configure Interface advertised-type.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><strong>config interface autoneg [OPTIONS]</strong> &lt;interface_name&gt; &lt;mode&gt;</td>
<td>Configures interface advertised-type</td>
</tr>
</tbody>
</table>
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.

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

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             N/A
docker0           240.127.1.1/24       up/down       N/A             N/A             N/A
eth0              172.18.0.154/24      up/up         N/A             N/A             N/A
lo                127.0.0.1/16         up/up         N/A             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             N/A
Ethernet3         192.168.80.13/24     up/up         N/A             N/A             N/A
docker0           240.127.1.1/24       up/down       N/A             N/A             N/A
eth0              172.18.0.154/24      up/up         N/A             N/A             N/A
```
2.3.11 Remove IPv4 address
Follow the steps below to remove IPv4 address from an interface.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>config interface ip remove &lt;interface_name&gt; &lt;ip_addr&gt;</code></td>
<td>Configures interface ip Interface name – may be any of the following: Ethernet0 - Ethernet53 Ip addr – A Valid IPv4 address</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>show interface status</code></td>
<td>Displays the interface configuration.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>sudo config save --y</code></td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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
```

<table>
<thead>
<tr>
<th>Interface</th>
<th>Master</th>
<th>IPv4 address/mask</th>
<th>Admin/Oper</th>
<th>BGP Neighbor</th>
<th>Neighbor IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet0</td>
<td></td>
<td>10.0.0.0/31</td>
<td>up/up</td>
<td>BGPNeigh01</td>
<td>10.0.0.1</td>
</tr>
<tr>
<td>Ethernet1</td>
<td></td>
<td>10.0.0.2/31</td>
<td>up/up</td>
<td>BGPNeigh02</td>
<td>10.0.0.3</td>
</tr>
<tr>
<td>Ethernet2</td>
<td></td>
<td>10.0.0.4/31</td>
<td>up/up</td>
<td>BGPNeigh03</td>
<td>10.0.0.5</td>
</tr>
<tr>
<td>Ethernet3</td>
<td></td>
<td>10.0.0.6/31</td>
<td>up/up</td>
<td>BGPNeigh04</td>
<td>10.0.0.7</td>
</tr>
<tr>
<td>192.168.80.13/24</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet4</td>
<td></td>
<td>10.0.0.8/31</td>
<td>up/up</td>
<td>BGPNeigh05</td>
<td>10.0.0.9</td>
</tr>
<tr>
<td>Ethernet5</td>
<td></td>
<td>10.0.0.10/31</td>
<td>up/up</td>
<td>BGPNeigh06</td>
<td>10.0.0.11</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
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)

<table>
<thead>
<tr>
<th>Interface</th>
<th>Master</th>
<th>IPv4 address/mask</th>
<th>Admin/Oper</th>
<th>BGP Neighbor</th>
<th>Neighbor IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bridge</td>
<td></td>
<td>fe80::886e:d3ff:fe7c:5551%Bridge/64</td>
<td>up/down</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Ethernet0</td>
<td></td>
<td>2::2/64</td>
<td>up/up</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Ethernet0</td>
<td></td>
<td>fe80::ec4:7aff:fe2e:17bd%Ethernet0/64</td>
<td>up/up</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Ethernet1</td>
<td></td>
<td>fe80::ec4:7aff:fe2e:17bd%Ethernet1/64</td>
<td>up/up</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Ethernet2</td>
<td></td>
<td>fe80::ec4:7aff:fe2e:17bd%Ethernet2/64</td>
<td>up/up</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Ethernet3</td>
<td></td>
<td>fe80::ec4:7aff:fe2e:17bd%Ethernet3/64</td>
<td>up/up</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Ethernet4</td>
<td></td>
<td>fe80::ec4:7aff:fe2e:17bd%Ethernet4/64</td>
<td>up/up</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Ethernet5</td>
<td></td>
<td>fe80::ec4:7aff:fe2e:17bd%Ethernet5/64</td>
<td>up/up</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>
```

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:CsAp9CFqVpli4Lz4Liexf1AzzXiUUs4HZZUlPfXqJzU.
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
```
### 2.3.13 Remove IPv6 address

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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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 |
### 2.3.14 Configure IPv6 address for Management Interface

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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>config interface ip add &lt;interface_name&gt; &lt;ip_addr&gt; &lt;default gateway IP address&gt;</td>
<td>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</td>
</tr>
<tr>
<td>Step 2</td>
<td>show ipv6 interfaces</td>
<td>Displays IPv6 Address of all interfaces.</td>
</tr>
<tr>
<td>Step 3</td>
<td>show management_interface address</td>
<td>Displays the management interface IP configuration.</td>
</tr>
<tr>
<td>Step 4</td>
<td>sudo config save –y</td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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] mveneta 7f020000.ethernet eth0: Link is Down
[ 5734.212789] mveneta 7f020000.ethernet eth0: PHY [7f022004.mdio-mii:00] driver [Marvell 88E1510]
[ 5734.221825] mveneta 7f020000.ethernet eth0: configuring for phy/sgmii link mode
[ 5734.229268] mveneta 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
```
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

Test the connectivity between the switches over IPv6.

SWITCH B:

admin@sonic:~$ sudo config interface ip add eth0 2001::2/64

Management IP address = 2001::2/64

admin@sonic:~$ show management_interface address
Management IP address = 2001::2/64

admin@sonic:~$ show ipv6 interfaces (Note: Truncated output is added here)

<table>
<thead>
<tr>
<th>Interface</th>
<th>Master</th>
<th>IPv4 address/mask</th>
<th>Admin/Oper</th>
<th>BGP Neighbor</th>
<th>Neighbor IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth0</td>
<td></td>
<td>2001::2/64</td>
<td>up/up</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>fe80::ec4:7aff:fe2e:1635%eth0/64</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lo</td>
<td></td>
<td>::1/128</td>
<td>up/up</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

admin@sonic:~$
--- 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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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.eternet eth0: Link is Down
[ 7428.759705] mvneta 7f020000.eternet eth0: PHY [7f022004.mdio-mii:00] driver [Marvell 88E1510]
```
1.42

[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:~$

admin@sonic:~$ show management_interface address
admin@sonic:~$ show ipv6 interfaces

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:~$ show ipv6 interfaces

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| Step 1 | config interface ipv6 enable use-link-local-only <interface_name> | Enables interface IPv6

Interface name – may be any of the following:
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)
```

<table>
<thead>
<tr>
<th>Interface Name</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet5</td>
<td>Enabled</td>
</tr>
<tr>
<td>Ethernet50</td>
<td>Disabled</td>
</tr>
<tr>
<td>Ethernet51</td>
<td>Disabled</td>
</tr>
<tr>
<td>Ethernet52</td>
<td>Disabled</td>
</tr>
<tr>
<td>Ethernet53</td>
<td>Disabled</td>
</tr>
<tr>
<td>Ethernet6</td>
<td>Disabled</td>
</tr>
<tr>
<td>Ethernet7</td>
<td>Disabled</td>
</tr>
<tr>
<td>Ethernet8</td>
<td>Disabled</td>
</tr>
<tr>
<td>Ethernet9</td>
<td>Disabled</td>
</tr>
<tr>
<td>PortChannel0033</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

2.3.17 Disable IPv6 Link Local

Follow the steps below to disables Interface IPv6.
<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>config interface ipv6 disable use-link-local-only &lt;interface_name&gt;</code></td>
<td>Disables interface IPv6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Interface name – may be any of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet0 - Ethernet53</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>show ipv6 link-local-mode</code></td>
<td>Display IPv6 link-local-mode</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>sudo config save -y</code></td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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:~$
```
### 2.3.18 MAC

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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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
```
2.3.19 Type

Follow the steps below to configure Interface type.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| Step 1 | `config interface type <interface_name> <interface_type_value>` | Configures interface type  
Interface name – may be any of the following:  
Ethernet0 - Ethernet53 |

<table>
<thead>
<tr>
<th>Step 2</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>show interface status</td>
<td>Displays the interface configuration.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>sudo config save -y</td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Interface</th>
<th>Auto-Neg Mode</th>
<th>Speed</th>
<th>Adv Speeds</th>
<th>Type</th>
<th>Adv Types</th>
<th>Oper</th>
<th>Admin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet4</td>
<td>disabled</td>
<td>1G</td>
<td>N/A</td>
<td>CR4</td>
<td>N/A</td>
<td>down</td>
<td>up</td>
</tr>
</tbody>
</table>
```

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>show interfaces alias [interface_name]</td>
<td>Displays the interface alias configuration.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Interface name - may be any of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet0 - Ethernet53</td>
</tr>
</tbody>
</table>

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

```
admin@sonic:~$ show interfaces alias (Note: Truncated output is added here)
```
### 2.3.21 Configure Interface Naming Mode

Follow the steps below to configure interface naming mode.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>`config interface_naming_mode (default</td>
<td>alias)`</td>
</tr>
<tr>
<td></td>
<td></td>
<td>default – Default interface name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>alias – Alias interface name</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>show interfaces naming_mode</code></td>
<td>Displays the interface naming configuration.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>sudo config save -y</code></td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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 1: `show interfaces counters [options]`

**Description**
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.

```bash
admin@sonic:$ show interfaces counters
```

<table>
<thead>
<tr>
<th>Interface</th>
<th>State</th>
<th>RX_OK</th>
<th>RX_BPS</th>
<th>RX_UTIL</th>
<th>RX_ERR</th>
<th>RX_DRP</th>
<th>RX_OVR</th>
<th>TX_OK</th>
<th>TX_BPS</th>
<th>TX_UTIL</th>
<th>TX_ERR</th>
<th>TX_DRP</th>
<th>TX_OVR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet0</td>
<td>D</td>
<td>0</td>
<td>0.00 B/s</td>
<td>0.00%</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00 B/s</td>
<td>0.00%</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Ethernet1</td>
<td>D</td>
<td>0</td>
<td>0.00 B/s</td>
<td>0.00%</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00 B/s</td>
<td>0.00%</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Ethernet2</td>
<td>D</td>
<td>0</td>
<td>0.00 B/s</td>
<td>0.00%</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00 B/s</td>
<td>0.00%</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### 2.3.22.1 Counters Detailed

Follow the steps below to display interface counters detailed.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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:**
The example below shows the commands used to display interface counters in detail.

```
admin@sonic:~$ show interfaces counters detailed Ethernet3
               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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>show interfaces counters errors [OPTIONS]</code></td>
<td>Displays the interface counters errors</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Options:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-p, --period - TEXT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-d, --display [all] - Show internal interfaces [default: all]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-n, --namespace [] - Namespace name or all</td>
</tr>
</tbody>
</table>

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

```
admin@sonic:~$ show interfaces counters errors
               IFACE    STATE   RX_ERR  RX_DRP  RX_OVR  TX_ERR  TX_DRP  TX_OVR
              --------  ------   ------   ------   ------   ------   ------   ------
```

"
### 2.3.22.3 Counters Rates

Follow the steps below to display interface counters rates.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>show interfaces counters rates [OPTIONS]</code></td>
<td>Displays the interface counters rates</td>
</tr>
<tr>
<td></td>
<td>Options:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-p, --period - TEXT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-d, --display [all] - Show internal interfaces</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[default: all]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-n, --namespace [] - Namespace name or all</td>
<td></td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>IFACE</th>
<th>STATE</th>
<th>RX_OK</th>
<th>RX_BPS</th>
<th>RX_PPS</th>
<th>RX_UTIL</th>
<th>TX_OK</th>
<th>TX_BPS</th>
<th>TX_PPS</th>
<th>TX_UTIL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet0</td>
<td>D</td>
<td>0</td>
<td>0.00 B/s</td>
<td>0.00/s</td>
<td>0.00%</td>
<td>0</td>
<td>0.00 B/s</td>
<td>0.00/s</td>
<td>0.00%</td>
</tr>
<tr>
<td>Ethernet1</td>
<td>D</td>
<td>0</td>
<td>0.00 B/s</td>
<td>0.00/s</td>
<td>0.00%</td>
<td>0</td>
<td>0.00 B/s</td>
<td>0.00/s</td>
<td>0.00%</td>
</tr>
</tbody>
</table>

### 2.3.22.4 Counters Rif

Follow the steps below to display interface counters rif.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>show interfaces counters rif [OPTIONS]</code></td>
<td>Displays all the interface RIFs counters</td>
</tr>
<tr>
<td></td>
<td><code>&lt;interface_name&gt;</code></td>
<td>Interface name - may be any of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet0 - Ethernet53</td>
</tr>
<tr>
<td></td>
<td>Options:</td>
<td></td>
</tr>
</tbody>
</table>


-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
```

2.3.23 Configure loopback

Follow the steps below to configure loopback.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>config loopback add &lt;loopback_name&gt;</code></td>
<td>Creates loopback interface.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Loopback name – A valid string with prefix “loopback” &amp; suffix range of &lt;0-999&gt;</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>config interface ip add &lt;interface_name&gt; &lt;ip_addr&gt; &lt;default gateway IP address&gt;</code></td>
<td>Configures interface ip</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Interface name – may be any of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet0 - Ethernet53</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ip addr – A Valid IPv4 address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gateway Ip addr – A Valid IPv4 address</td>
</tr>
</tbody>
</table>
### Step 3

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show ip interfaces</code></td>
<td>Displays the interfaces configuration.</td>
</tr>
</tbody>
</table>

### Step 4

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>sudo config save</code></td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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
```

<table>
<thead>
<tr>
<th>Interface</th>
<th>Master</th>
<th>IPv4 address/mask</th>
<th>Admin/Oper</th>
<th>BGP Neighbor</th>
<th>Neighbor IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet52</td>
<td></td>
<td>10.0.0.104/31</td>
<td>up/up</td>
<td>ARISTA26T0</td>
<td>10.0.0.105</td>
</tr>
<tr>
<td>Ethernet53</td>
<td></td>
<td>10.0.0.106/31</td>
<td>up/up</td>
<td>ARISTA27T0</td>
<td>10.0.0.107</td>
</tr>
<tr>
<td>Loopback0</td>
<td></td>
<td>10.1.0.1/32</td>
<td>up/up</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Loopback11</td>
<td></td>
<td>10.1.0.2/32</td>
<td>up/up</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>docker0</td>
<td></td>
<td>240.127.1.1/24</td>
<td>up/down</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>eth0</td>
<td></td>
<td>192.168.86.34/24</td>
<td>up/up</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>lo</td>
<td></td>
<td>127.0.0.1/16</td>
<td>up/up</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

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

#### 2.3.24 Remove loopback

Follow the steps below to remove loopback.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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` | 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 Configuration

<table>
<thead>
<tr>
<th>Interface</th>
<th>Master</th>
<th>IPv4 address/mask</th>
<th>Admin/Oper</th>
<th>BGP Neighbor</th>
<th>Neighbor IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet52</td>
<td></td>
<td>10.0.0.104/31</td>
<td>up/up</td>
<td>ARISTA26T0</td>
<td>10.0.0.105</td>
</tr>
<tr>
<td>Ethernet53</td>
<td></td>
<td>10.0.0.106/31</td>
<td>up/up</td>
<td>ARISTA27T0</td>
<td>10.0.0.107</td>
</tr>
<tr>
<td>Loopback0</td>
<td></td>
<td>10.0.1.0/32</td>
<td>up/up</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>docker0</td>
<td></td>
<td>240.127.1.1/24</td>
<td>up/down</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>eth0</td>
<td></td>
<td>192.168.86.34/24</td>
<td>up/up</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>lo</td>
<td></td>
<td>127.0.0.1/16</td>
<td>up/up</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

### 2.3.25 Storm 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.

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

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

```
admin@sonic:~$ sudo config interface storm-control del Ethernet10 broadcast

admin@sonic:~$ show storm-control
```

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.

<table>
<thead>
<tr>
<th>SKU Name</th>
<th>Interfaces Speed/Type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supermicro_sse_t7132s</td>
<td>32 x 400G Ethernet interfaces</td>
<td>This is the default HWSKU</td>
</tr>
<tr>
<td>Supermicro_sse_t7132s_128x100</td>
<td>128 x 100G(PAM4) Ethernet interfaces</td>
<td></td>
</tr>
<tr>
<td>Supermicro_sse_t7132s_32x100</td>
<td>32 100G(NRZ) Ethernet interfaces</td>
<td>One port per QSFP-DD connector. QSFP28 DACs or transceivers can be used.</td>
</tr>
<tr>
<td>Supermicro_sse_t7132s_64x100</td>
<td>64 x 100G(NRZ) Ethernet interfaces</td>
<td>Two ports per QSFP-DD connector.</td>
</tr>
<tr>
<td>Supermicro_sse_t7132s_64x200</td>
<td>64 x 200G Ethernet interfaces</td>
<td>Two ports per QSFP-DD connector.</td>
</tr>
<tr>
<td>Supermicro_sse_t7132s_16x400_64x100</td>
<td>64 x 100G(PAM4) and 16 x 400G Ethernet interfaces</td>
<td>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.</td>
</tr>
</tbody>
</table>

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:~# 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.
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”.

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

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-superserver_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
```
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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>show clock</td>
<td>Displays date &amp; time</td>
</tr>
</tbody>
</table>
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 is sonic. The switch name is also used as a prompt.

Follow the steps below to configure the Host Name.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>config hostname &lt;new_hostname&gt;</td>
<td>Configure Host Name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>New hostname – Host name specified as alphanumeric characters</td>
</tr>
<tr>
<td>Step 2</td>
<td>sudo config save –y</td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>S. No</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>show version</td>
<td>Displays version</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>REPOSITORY</th>
<th>TAG</th>
<th>IMAGE ID</th>
<th>SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>docker-dhcp-relay</td>
<td>latest</td>
<td>a774c6cec02a</td>
<td>563MB</td>
</tr>
<tr>
<td>docker-syncd-mrvl</td>
<td>SSE-G3748_3.2.0-0003</td>
<td>d532a0cc0152</td>
<td>715MB</td>
</tr>
<tr>
<td>docker-syncd-mrvl</td>
<td>latest</td>
<td>d532a0cc0152</td>
<td>715MB</td>
</tr>
<tr>
<td>docker-teamd</td>
<td>SSE-G3748_3.2.0-0003</td>
<td>da29cf16d70</td>
<td>567MB</td>
</tr>
<tr>
<td>docker-teamd</td>
<td>latest</td>
<td>da29cf16d70</td>
<td>567MB</td>
</tr>
<tr>
<td>docker-nat</td>
<td>SSE-G3748_3.2.0-0003</td>
<td>65a4f316dcf2</td>
<td>569MB</td>
</tr>
<tr>
<td>docker-nat</td>
<td>latest</td>
<td>65a4f316dcf2</td>
<td>569MB</td>
</tr>
<tr>
<td>docker-platform-monitor</td>
<td>SSE-G3748_3.2.0-0003</td>
<td>98041aaa6629b</td>
<td>736MB</td>
</tr>
<tr>
<td>docker-platform-monitor</td>
<td>latest</td>
<td>98041aaa6629b</td>
<td>736MB</td>
</tr>
<tr>
<td>docker-lldp</td>
<td>SSE-G3748_3.2.0-0003</td>
<td>7e3a57bebfb1</td>
<td>562MB</td>
</tr>
<tr>
<td>docker-lldp</td>
<td>latest</td>
<td>7e3a57bebfb1</td>
<td>562MB</td>
</tr>
<tr>
<td>docker-database</td>
<td>SSE-G3748_3.2.0-0003</td>
<td>a959f82f26a4</td>
<td>556MB</td>
</tr>
<tr>
<td>docker-database</td>
<td>latest</td>
<td>a959f82f26a4</td>
<td>556MB</td>
</tr>
<tr>
<td>docker-router-advertiser</td>
<td>SSE-G3748_3.2.0-0003</td>
<td>7993c934ad77</td>
<td>556MB</td>
</tr>
<tr>
<td>docker-router-advertiser</td>
<td>latest</td>
<td>7993c934ad77</td>
<td>556MB</td>
</tr>
<tr>
<td>docker-orchagent</td>
<td>SSE-G3748_3.2.0-0003</td>
<td>db8160a889d0</td>
<td>666MB</td>
</tr>
<tr>
<td>docker-orchagent</td>
<td>latest</td>
<td>db8160a889d0</td>
<td>666MB</td>
</tr>
<tr>
<td>docker-snmp</td>
<td>SSE-G3748_3.2.0-0003</td>
<td>36490a24eb14</td>
<td>599MB</td>
</tr>
<tr>
<td>docker-snmp</td>
<td>latest</td>
<td>36490a24eb14</td>
<td>599MB</td>
</tr>
<tr>
<td>docker-sonic-telemetry</td>
<td>SSE-G3748_3.2.0-0003</td>
<td>d73a517f1aad</td>
<td>640MB</td>
</tr>
<tr>
<td>docker-sonic-telemetry</td>
<td>latest</td>
<td>d73a517f1aad</td>
<td>640MB</td>
</tr>
<tr>
<td>docker-fpm-frr</td>
<td>SSE-G3748_3.2.0-0003</td>
<td>bfbbc9034cd9</td>
<td>585MB</td>
</tr>
<tr>
<td>docker-fpm-frr</td>
<td>latest</td>
<td>bfbbc9034cd9</td>
<td>585MB</td>
</tr>
<tr>
<td>docker-sflow</td>
<td>SSE-G3748_3.2.0-0003</td>
<td>154be017ee0f</td>
<td>568MB</td>
</tr>
<tr>
<td>docker-sflow</td>
<td>latest</td>
<td>154be017ee0f</td>
<td>568MB</td>
</tr>
<tr>
<td>docker-macsec</td>
<td>SSE-G3748_3.2.0-0003</td>
<td>e64b26f32286</td>
<td>569MB</td>
</tr>
<tr>
<td>docker-macsec</td>
<td>latest</td>
<td>e64b26f32286</td>
<td>569MB</td>
</tr>
</tbody>
</table>

2.4.4 Display environment

Follow the steps below to display the system environment.
<table>
<thead>
<tr>
<th>S. No</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>show environment</td>
<td>Displays Platform environmental</td>
</tr>
</tbody>
</table>

Example:

```
admin@sonic: ~$ show environment

Im75-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)

Im75-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.

<table>
<thead>
<tr>
<th>S. No</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>show reboot-cause</td>
<td>Displays Cause of the previous reboot</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>S. No</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>show uptime</td>
<td>Displays System uptime</td>
</tr>
</tbody>
</table>

Example:

```
admin@sonic: ~$ show uptime
up 10 hours, 10 minutes
```

2.4.7 Display logging
Follow the steps below to display the system logging.

<table>
<thead>
<tr>
<th>S. No</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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
```
2.4.8 Display platform summary

Follow the steps below to display the system platform summary.

<table>
<thead>
<tr>
<th>S. No</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>show platform summary</td>
<td>Displays Summary of the device's hardware platform</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>S. No</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>show platform syseeprom</td>
<td>Displays Information stored on the system EEPROM</td>
</tr>
</tbody>
</table>

Example:

```
admin@sonic: ~$ show platform syseeprom
TlvInfo Header:
  Id String: TlvInfo
  Version: 1
```
Total Length: 192

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

(checksum valid)

2.4.10 Display power supply units

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

<table>
<thead>
<tr>
<th>S. No</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>show platform psustatus</td>
<td>Displays Status of the device's power supply units</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>S. No</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>show platform fan</td>
<td>Displays Status of the device's fans</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>S. No</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>show platform temperature</td>
<td>Displays Status of the device's thermal sensors</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>

Supermicro SONiC Configuration Guide v1.4
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
```

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

### 2.4.13.2 Display Memory usage

Follow the steps below to display the system memory usage.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>show processes memory</td>
<td>Displays Current memory usage by process</td>
</tr>
</tbody>
</table>

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
```

<table>
<thead>
<tr>
<th>PID</th>
<th>USER</th>
<th>PR</th>
<th>NI</th>
<th>VIRT</th>
<th>RES</th>
<th>SHR</th>
<th>S</th>
<th>%CPU</th>
<th>%MEM</th>
<th>TIME+</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>567</td>
<td>root</td>
<td>20</td>
<td>0</td>
<td>1207656</td>
<td>85480</td>
<td>33084</td>
<td>S</td>
<td>0.0</td>
<td>2.1</td>
<td>4:18.27</td>
<td>dockerd</td>
</tr>
<tr>
<td>2338</td>
<td>root</td>
<td>20</td>
<td>0</td>
<td>2221944</td>
<td>77580</td>
<td>32600</td>
<td>S</td>
<td>11.8</td>
<td>1.9</td>
<td>47:01.92</td>
<td>syncd</td>
</tr>
<tr>
<td>168325</td>
<td>root</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Z</td>
<td>0.0</td>
<td>0.0</td>
<td>0:01.82</td>
<td>python3</td>
</tr>
<tr>
<td>168326</td>
<td>root</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Z</td>
<td>0.0</td>
<td>0.0</td>
<td>0:00.28</td>
<td>python3</td>
</tr>
</tbody>
</table>
### 2.4.13.3 Display Summary usage

Follow the steps below to display the system summary usage.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>show processes summary</code></td>
<td>Displays current summary usage by process</td>
</tr>
</tbody>
</table>

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)
<table>
<thead>
<tr>
<th>PID</th>
<th>PPID</th>
<th>CMD</th>
<th>%MEM</th>
<th>%CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>/sbin/init</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>[kthreadd]</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>[rcu_gp]</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>168629</td>
<td>489</td>
<td>[python3] &lt;defunct&gt;</td>
<td>0.0</td>
<td>0.4</td>
</tr>
<tr>
<td>168970</td>
<td>161906</td>
<td>/usr/bin/python3 /usr/local</td>
<td>0.9</td>
<td>86.5</td>
</tr>
<tr>
<td>168989</td>
<td>168970</td>
<td>/bin/sh -c ps -eo pid,ppid,</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>168990</td>
<td>168989</td>
<td>ps -eo pid,ppid,cmd,%mem,%c</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
```

### 2.4.14 Troubleshooting

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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>show techsupport</code></td>
<td>Displays tech support options</td>
</tr>
</tbody>
</table>

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
```

```
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'
^Z
[3]+ Stopped show techsupport --since='hour ago'
admin@SONiC202106:~$
admin@sonic: ~$ show techsupport
main
2.4.15 Display Services

<table>
<thead>
<tr>
<th>S. No</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>show services</td>
<td>Displays Status of the system services</td>
</tr>
</tbody>
</table>

Example:

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

```
	nsnmp
tsnmp

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```

```
2.4.16 Display System-health

2.4.16.1 Display system-health detail

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>sudo show system-health detail</td>
<td>Displays Current system-health detail</td>
</tr>
</tbody>
</table>

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

System services and devices ignore list
<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>psu.temperature</td>
<td>Ignored</td>
<td>Device</td>
</tr>
<tr>
<td>asic</td>
<td>Ignored</td>
<td>Device</td>
</tr>
<tr>
<td>admin@sonic: ~$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.4.16.2 Display system-health monitor-list

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><strong>show system-health monitor-list</strong></td>
<td>Displays Current system-health monitor-list</td>
</tr>
</tbody>
</table>

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

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

admin@sonic: ~$ 2.4.16.3 Display system-health summary

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>show system-health summary</td>
<td>Displays Current system-health summary</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>S. No</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Show system-memory</td>
<td>Displays Status of the system memory</td>
</tr>
</tbody>
</table>

Example:

```
admin@sonic: ~$ show system-memory
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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAA login</td>
<td>Local</td>
</tr>
<tr>
<td>AAA failthrough</td>
<td>False</td>
</tr>
<tr>
<td>AAA fallback</td>
<td>N/A</td>
</tr>
</tbody>
</table>

2.5.1.2 Configure AAA authentication login

Follow the steps below to configure AAA authentication login.
### Configure AAA Authentication Login

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>`config aaa authentication login (tacacs+</td>
<td>local</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>show aaa</code></td>
<td>Displays the AAA configuration.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>sudo config save --y</code></td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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)
```

### Configure AAA Authentication Failthrough

Follow the steps below to configure AAA authentication failthrough.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>`config aaa authentication failthrough (enable</td>
<td>disable</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>show aaa</code></td>
<td>AAA configuration.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>sudo config save --y</code></td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>config aaa authentication fallback (enable</td>
<td>disable</td>
</tr>
<tr>
<td>Step 2</td>
<td>show aaa</td>
<td>AAA configuration.</td>
</tr>
<tr>
<td>Step 3</td>
<td>sudo config save –y</td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RADIUS global auth_type</td>
<td>Pap</td>
</tr>
<tr>
<td>RADIUS global retransmit</td>
<td>3</td>
</tr>
<tr>
<td>RADIUS global timeout</td>
<td>5</td>
</tr>
</tbody>
</table>
| 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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>config radius add [OPTIONS] &lt;ip_address_or_domain_name&gt;</code></td>
<td>Configure the RADIUS server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ip address – A valid IPv4 Address.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Domain name – A valid domain name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Options:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-r, --retransmit - INTEGER RANGE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Retransmit attempts, default 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-t, --timeout - INTEGER RANGE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transmission timeout interval, default 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-k, --key TEXT - Shared secret</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-a, --auth_type [chap</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-o, --auth-port - INTEGER RANGE UDP port range is 1 to 65535, default 1812</td>
</tr>
</tbody>
</table>
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.

```bash
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 authype**

Follow the below steps to configure RADIUS server authype parameters.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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. |
The example below shows the commands used to configure RADIUS server auth type.

```
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:~$
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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td><code>sudo config save -y</code></td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>config radius delete &lt;ip_address_or_domain_name&gt;</td>
<td>Remove the RADIUS server.</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>config radius passkey &lt;secret_string&gt;</td>
<td>Configure the RADIUS passkey. Secret string – Secret string can be specified as alphanumeric characters.</td>
</tr>
<tr>
<td>Step 2</td>
<td>show radius</td>
<td>Displays the RADIUS configuration.</td>
</tr>
<tr>
<td>Step 3</td>
<td>sudo config save –y</td>
<td>Optional step – saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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
2.5.2.7 Configure RADIUS retransmit
Follow the below steps to configure RADIUS retransmit parameters.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>config radius retransmit &lt;retry_attempts&gt;</td>
<td>Configure the RADIUS retransmit. Retry attempts – Retry attempt can be specified in the range of &lt; 0 – 10 &gt;.</td>
</tr>
<tr>
<td>Step 2</td>
<td>show radius</td>
<td>Displays the RADIUS configuration.</td>
</tr>
<tr>
<td>Step 3</td>
<td>sudo config save –y</td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>`config radius statistics [enable</td>
<td>disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enable – enables radius statistics.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Disable – disables radius statistics.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>default – default value of radius statistics.</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>show radius</code></td>
<td>Displays the RADIUS configuration.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>sudo config save -y</code></td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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

```bash
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:$
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
```
**2.5.2.9 Configure RADIUS timeout**

Follow the below steps to configure RADIUS timeout parameters.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>config radius timeout &lt;time_second&gt;</code></td>
<td>Configure the RADIUS timeout. Time second – Time seconds can be specified in the range of &lt; 1 – 60 &gt;.</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>show radius</code></td>
<td>Displays the RADIUS configuration.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>sudo config save -y</code></td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>`config aaa authentication login (tacacs+</td>
<td>local</td>
</tr>
<tr>
<td>5</td>
<td>`config aaa authentication failthrough (enable</td>
<td>disable</td>
</tr>
<tr>
<td>6</td>
<td><code>show radius</code></td>
<td>Displays the RADIUS configuration.</td>
</tr>
<tr>
<td>7</td>
<td><code>show aaa</code></td>
<td>AAA configuration.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TACACS auth_type</td>
<td>Pap</td>
</tr>
<tr>
<td>TACACS timeout</td>
<td>5</td>
</tr>
<tr>
<td>TACACS passkey</td>
<td>&lt;EMPTY_STRING&gt;</td>
</tr>
<tr>
<td>TACPLUS_SERVER priority 1</td>
<td>1</td>
</tr>
<tr>
<td>TACPLUS_SERVER TCP port</td>
<td>49</td>
</tr>
</tbody>
</table>

2.5.3.2 Configure TACACS Server

Follow the steps below to configure TACACS server.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>`config tacacs add &lt;ip_address&gt; [-t</td>
<td>--timeout &lt;seconds&gt;] [-k</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ip address: TACACS server’s IPv4/IPv6 address.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>timeout: Transmission timeout interval in seconds, range 1 to 60, default 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>key: Shared secret</td>
</tr>
<tr>
<td></td>
<td></td>
<td>type: Authentication type, &quot;chap&quot; or &quot;pap&quot; or &quot;mschap&quot; or &quot;login&quot;, default is &quot;pap&quot;.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>port: TCP port range is 1 to 65535, default 49</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pri: Priority, priority range 1 to 64, default 1.</td>
</tr>
</tbody>
</table>
**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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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.
### 2.5.3.5 Configure TACACS authtype

Follow the steps below to configure TACACS authtype.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>`config tacacs authtype (chap</td>
<td>pap</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>show tacacs</code></td>
<td>Displays the TACACS configuration.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>sudo config save -y</code></td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>`config tacacs default (authtype</td>
<td>passkey</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>show tacacs</code></td>
<td>Displays the TACACS configuration.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>sudo config save -y</code></td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>config tacacs passkey &lt;pass_key&gt;</code></td>
<td>Configure TACACS passkey. <em>pass_key</em> - TACACS server passkey.</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>show tacacs</code></td>
<td>Displays the TACACS configuration.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>sudo config save</code></td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td><code>show tacacs</code></td>
<td>Displays the TACACS configuration.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>sudo config save</code></td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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

```
admin@sonic:~$ sudo config tacacs timeout 60
```
2.5.3.8 **TACACS Configuration Example**

This section explains the configurations of TACACS.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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
### Step 1

- **-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.

```bash
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”.

```plaintext
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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>`config save [ -y</td>
<td>--yes ] [&lt;filename&gt;]`</td>
</tr>
<tr>
<td></td>
<td><code>sudo config save -y</code></td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>`config save [-y</td>
<td>--yes] [&lt;filename&gt;]`</td>
</tr>
<tr>
<td></td>
<td><code>sudo config save -y /etc/sonic/config2.json</code></td>
<td>filename – filename in which configuration should be saved</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>sudo rm /etc/sonic/config_db.json</td>
<td>Remove the /etc/sonic/config_db.json</td>
</tr>
<tr>
<td>Step 2</td>
<td>sudo config-setup factory</td>
<td>Generate factory default configuration</td>
</tr>
<tr>
<td>Step 3</td>
<td>sudo reboot</td>
<td>Restore default configuration file on /etc/sonic/config_db.json</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>sudo config system reset-to-factory</td>
<td>Resets the switch to factory-default-configuration.</td>
</tr>
</tbody>
</table>

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.
<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>show boot</td>
<td>Displays the images installed on the device</td>
</tr>
</tbody>
</table>

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
```

2.6.6 Warm Reboot

Follow the steps below to warm reboot of the device.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>sudo warm-reboot</td>
<td>Initiates a warm reboot of the device</td>
</tr>
</tbody>
</table>

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)
```

2.7 Switch features

2.7.1 Defaults

<table>
<thead>
<tr>
<th>Feature</th>
<th>State</th>
<th>AutoRestart</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP</td>
<td>Enabled</td>
<td>Enabled</td>
</tr>
</tbody>
</table>
### 2.7.2 Configure state

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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><strong>config feature state</strong> <code>&lt;feature_name&gt;</code> (enabled</td>
<td>Configure state for a specific feature.</td>
</tr>
<tr>
<td></td>
<td>disabled)</td>
<td>Feature name – Feature name (e.g. bgp, lldp, pmon)</td>
</tr>
<tr>
<td>Step 2</td>
<td><strong>show feature status</strong> <code>&lt;feature_name&gt;</code> ]</td>
<td>Status of feature state.</td>
</tr>
<tr>
<td>Step 3</td>
<td><strong>sudo config save –y</strong></td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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
<table>
<thead>
<tr>
<th>Feature</th>
<th>State</th>
<th>AutoRestart</th>
<th>SetOwner</th>
</tr>
</thead>
<tbody>
<tr>
<td>bgp</td>
<td>disabled</td>
<td>enabled</td>
<td></td>
</tr>
<tr>
<td>database</td>
<td>always_enabled</td>
<td>always_enabled</td>
<td></td>
</tr>
<tr>
<td>dhcp relay</td>
<td>disabled</td>
<td>enabled</td>
<td>local</td>
</tr>
<tr>
<td>lldp</td>
<td>enabled</td>
<td>enabled</td>
<td></td>
</tr>
<tr>
<td>macsec</td>
<td>disabled</td>
<td>enabled</td>
<td></td>
</tr>
<tr>
<td>nat</td>
<td>disabled</td>
<td>enabled</td>
<td></td>
</tr>
</tbody>
</table>
```
2.7.3 Configure auto-restart

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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>config feature autorestart &lt;feature_name&gt; (enabled</td>
<td>disabled)</td>
</tr>
<tr>
<td>Step 2</td>
<td>show feature autorestart [&lt;&gt;feature_name&gt;]</td>
<td>Status of auto-restart for feature.</td>
</tr>
<tr>
<td>Step 3</td>
<td>sudo config save -y</td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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
pmmon          enabled
radv           enabled
sflow          enabled
```

### 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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>config reload [OPTIONS] [FILENAME]</code></td>
<td>Configure reload options.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Filename – Names of configuration file(s) to load, separated by comma with no spaces in between</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Options:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;-y&quot; or &quot;--yes&quot;, - 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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;-n&quot; or &quot;--no-service-restart&quot;, - 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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;-f&quot; or &quot;--force&quot; - ignores the system sanity checks. By default a list of</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>config load [OPTIONS] [FILENAME]</code></td>
<td>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</td>
</tr>
</tbody>
</table>
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: ~$
```

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>location</td>
<td>public</td>
</tr>
<tr>
<td>Community_string</td>
<td>public</td>
</tr>
<tr>
<td>Community_type</td>
<td>RO</td>
</tr>
</tbody>
</table>

2.9.2 Configure SNMP Agent Address

Follow the steps below to configure the snmp agent address.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>config snmpagentaddress add &lt;SNMP AGENT LISTENING IP Address&gt;</code></td>
<td>Add the snmp agent address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNMP AGENT LISTENING IP Address – valid ipv4 address</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>config snmpagentaddress del &lt;SNMP AGENT LISTENING IP Address&gt;</code></td>
<td>Delete the snmp agent address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNMP AGENT LISTENING IP Address – valid ipv4 address</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>show snmpagentaddress</code></td>
<td>snmp agent address</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| Step 1 | `config snmptrap modify <SNMP Version> <SNMP TRAP SERVER IP Address>`  | Configure the snmp trap
|        | `config snmptrap del <SNMP Version>`                                    | 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: ~$
```
2.9.4 Configure SNMP location

Follow the steps below to configure the SNMP location.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>config snmp location add &lt;location&gt;</code></td>
<td>Configures SNMP location. location - A valid location string.</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>show runningconfiguration snmp location</code></td>
<td>Displays the SNMP location configuration.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>sudo config save –y</code></td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>config snmp location mod &lt;location&gt;</code></td>
<td>Modifies SNMP location.</td>
</tr>
</tbody>
</table>
### 2.9.6 Remove SNMP location

Follow the steps below to remove the SNMP location.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>config snmp location del &lt;location&gt;</code></td>
<td>Deletes SNMP location. Location - A valid location string.</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>show runningconfiguration snmp location</code></td>
<td>Displays the SNMP location configuration.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>sudo config save --y</code></td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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
-------------
New York
admin@sonic: ~$
```
2.9.7 Configure SNMP contact

Follow the steps below to configure the SNMP contact.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>config snmp contact add &lt;contact_name&gt; &lt;contact_email&gt;</code></td>
<td>Configures SNMP contact. Contact name - A valid string. Contact email - A valid contact email string.</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>show runningconfiguration snmp contact</code></td>
<td>Displays the SNMP contact configuration.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>sudo config save --y</code></td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>config snmp contact mod &lt;contact&gt; &lt;contact_email&gt;</code></td>
<td>Modify SNMP contact. Contact name - A valid string.</td>
</tr>
</tbody>
</table>
Contact email - A valid contact email string.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>show runningconfiguration snmp contact</td>
<td>Displays the SNMP contact configuration.</td>
</tr>
<tr>
<td>3</td>
<td>sudo config save –y</td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>config snmp contact del &lt;contact&gt;</td>
<td>Delete SNMP contact.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contact name - A valid string.</td>
</tr>
<tr>
<td>2</td>
<td>show runningconfiguration snmp contact</td>
<td>Displays the SNMP contact configuration.</td>
</tr>
<tr>
<td>3</td>
<td>sudo config save –y</td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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
```
2.9.10 Configure SNMP community
Follow the steps below to configure the SNMP community.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>`config snmp community add &lt;snmp_community&gt; &lt;RO</td>
<td>RW&gt;`</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>show runningconfiguration snmp community</code></td>
<td>Displays the SNMP community configuration.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>sudo config save -y</code></td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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
```

2.9.11 Modify SNMP community
Follow the steps below to modify the SNMP community.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>config snmp community replace &lt;current_community_string&gt; &lt;new_community_string&gt;</code></td>
<td>Modify SNMP community. Snmp Community - A valid string. New Snmp Community - A valid string.</td>
</tr>
</tbody>
</table>
### Step 2: Modify SNMP Community

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td><code>show runningconfiguration snmp community</code></td>
<td>Displays the SNMP community configuration.</td>
</tr>
<tr>
<td>3</td>
<td><code>sudo config save --y</code></td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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
```

<table>
<thead>
<tr>
<th>Community String</th>
<th>Community Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>public</td>
<td>RO</td>
</tr>
<tr>
<td>user1</td>
<td>RO</td>
</tr>
</tbody>
</table>

```
```

### 2.9.12 Remove SNMP Community

Follow the steps below to remove the SNMP community.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>config snmp community del &lt;snmp_community&gt;</code></td>
<td>Remove SNMP community. Snmp Community - A valid string.</td>
</tr>
<tr>
<td>2</td>
<td><code>show runningconfiguration snmp community</code></td>
<td>Displays the SNMP community configuration.</td>
</tr>
<tr>
<td>3</td>
<td><code>sudo config save –y</code></td>
<td>Optional step - Saves this current configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>config snmp user add &lt;snmp_user&gt; &lt;noAuthNoPriv</td>
<td>Configures SNMP users.</td>
</tr>
<tr>
<td></td>
<td>AuthNoPriv</td>
<td>Snmp user - A valid string.</td>
</tr>
<tr>
<td></td>
<td>Priv&gt; &lt;RO</td>
<td>RW&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;MD5</td>
<td>SHA</td>
</tr>
<tr>
<td></td>
<td>&lt;auth_password&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;DES</td>
<td>AES&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;encrypt_password&gt;</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>show runningconfiguration snmp user</td>
<td>Displays the SNMP users configuration.</td>
</tr>
<tr>
<td>Step 3</td>
<td>sudo config save –y</td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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: ~$ 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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>config snmp user del &lt;snmp_user&gt;</td>
<td>Remove SNMP users. Snmp user - A valid string.</td>
</tr>
<tr>
<td>Step 2</td>
<td>show runningconfiguration snmp users</td>
<td>Displays the SNMP users configuration.</td>
</tr>
<tr>
<td>Step 3</td>
<td>sudo config save –y</td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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
2.10 NTP

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.

2.10.1 Configure NTP server Address

Follow the steps below to configure the NTP server address.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>config ntp add [OPTIONS] &lt;ntp_ip_address&gt;</code></td>
<td>Add NTP server address. ntp_ip_address – valid IPv4 or IPv6 address.</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>config ntp del [OPTIONS] &lt;ntp_ip_address&gt;</code></td>
<td>Delete NTP server address. ntp_ip_address – valid IPv4 or IPv6 address.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>show runningconfiguration ntp</code></td>
<td>Displays the current NTP configuration.</td>
</tr>
<tr>
<td>Step 4</td>
<td><code>show ntp</code></td>
<td>Show NTP information.</td>
</tr>
<tr>
<td>Step 5</td>
<td><code>sudo config save</code></td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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
```

Supermicro SONiC Configuration Guide v1.4
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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>config ntp del &lt;ip_address&gt;</td>
<td>Removes NTP server. Ip address - A valid IPv4 or IPv6 address.</td>
</tr>
<tr>
<td>Step 2</td>
<td>show ntp</td>
<td>Displays the NTP configuration.</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>timedatectl set-timezone &lt;zone&gt;</code></td>
<td>Configures the timezone of the switch. <code>&lt;zone&gt;</code> - valid timezone.</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>timedatectl list-timezones</code></td>
<td>Lists the valid time zones.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>timedatectl status</code></td>
<td>Displays the current time zone.</td>
</tr>
</tbody>
</table>

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

```
admin@sonic: ~$ sudo timedatectl set-timezone America/Los_Angeles
admin@sonic:~$ sudo 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
```
2.11 System 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.

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

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 19:04:59.178349 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.

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

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 19:06:54.794217 sonic INFO systemd[1]: Stopping System Logging Service...
Jul 21 19:06:54.797112 sonic INFO systemd[1]: rsyslog.service: Succeeded.
```
2.12 Zero 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.

2.12.1 DHCP 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 bootfile-name 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";
}
```
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 the 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 supports 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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>config vlan add &lt;vid&gt;</td>
<td>Create a VLAN. vid - May be any vlan number, Range 1 to 4094.</td>
</tr>
<tr>
<td>Step 2</td>
<td>show vlan config</td>
<td>Displays the configured VLANs</td>
</tr>
<tr>
<td>Step 3</td>
<td>show vlan brief</td>
<td>Displays all bridge information</td>
</tr>
<tr>
<td>Step 4</td>
<td>sudo config save -y</td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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       |           |       |             |          |                     |
+-----------------+-----------+-------+-------------+-----------+---------------------+
```

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.
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 from the 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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>`config vlan member add [-u</td>
<td>--untagged] &lt;vlan_id&gt; &lt;member_portname&gt;`</td>
</tr>
</tbody>
</table>
### Supermicro SONiC Configuration Guide

#### 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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>sudo config vlan member add -u 100 Ethernet48</td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>
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  Ethernet48  untagged
Vlan200  200  Ethernet52  tagged
admin@sonic: ~$ show vlan brief
+------------------------+------------------------+------------------------+------------------------+------------------------+------------------------+
| VLAN ID | IP Address | Ports | Port Tagging | Proxy ARP | DHCP Helper Address |
+----------+-----------+-------+-------------+-----------+---------------------+
| 100      |           | Ethernet48 | untagged    | disabled  |                     |
| 100      |           | 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.

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

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
admin@sonic: ~$ show vlan brief
Vlan100 100
Vlan200 200
admin@sonic: ~$ sudo config vlan member del 100 Ethernet52
admin@sonic: ~$ show vlan config
Name       VID  Member        Mode
----------  ---  --------------- ------
Vlan100 100  Ethernet52    tagged
admin@sonic: ~$ show vlan brief
```

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
admin@sonic: ~$ show vlan brief
```
### 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**

```bash
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

<table>
<thead>
<tr>
<th>Name</th>
<th>VID</th>
<th>Member</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vlan10</td>
<td>10</td>
<td>Ethernet1</td>
<td>untagged</td>
</tr>
<tr>
<td>Vlan10</td>
<td>10</td>
<td>Ethernet2</td>
<td>untagged</td>
</tr>
<tr>
<td>Vlan10</td>
<td>10</td>
<td>Ethernet3</td>
<td>untagged</td>
</tr>
<tr>
<td>Vlan10</td>
<td>10</td>
<td>Ethernet5</td>
<td>tagged</td>
</tr>
<tr>
<td>Vlan10</td>
<td>10</td>
<td>Ethernet6</td>
<td>tagged</td>
</tr>
<tr>
<td>Vlan10</td>
<td>10</td>
<td>Ethernet7</td>
<td>tagged</td>
</tr>
<tr>
<td>Vlan10</td>
<td>10</td>
<td>PortChannel0004</td>
<td>tagged</td>
</tr>
</tbody>
</table>
```
<table>
<thead>
<tr>
<th>VLAN ID</th>
<th>IP Address</th>
<th>Ports</th>
<th>Port Tagging</th>
<th>Proxy ARP</th>
<th>DHCP Helper Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td></td>
<td>Ethernet1</td>
<td>untagged</td>
<td>disabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet2</td>
<td>untagged</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet3</td>
<td>untagged</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet5</td>
<td>tagged</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet6</td>
<td>tagged</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet7</td>
<td>tagged</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PortChannel0004</td>
<td>tagged</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>Ethernet4</td>
<td>tagged</td>
<td>disabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet5</td>
<td>tagged</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet6</td>
<td>tagged</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet7</td>
<td>tagged</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>Ethernet4</td>
<td>tagged</td>
<td>disabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet5</td>
<td>tagged</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet6</td>
<td>tagged</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet7</td>
<td>tagged</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td>disabled</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td></td>
<td></td>
<td></td>
<td>disabled</td>
<td></td>
</tr>
<tr>
<td>port</td>
<td>vlan ids</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>docker0</td>
<td>1  PVID Egress Untagged</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bridge</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>200</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dummy</td>
<td>1  PVID Egress Untagged</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet1</td>
<td>10 PVID Egress Untagged</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet2</td>
<td>10 PVID Egress Untagged</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet3</td>
<td>10 PVID Egress Untagged</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet4</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet5</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet6</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet7</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PortChannel0004</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>config portchannel add &lt;portchannel_name&gt;</code></td>
<td>Create a port channel interface. portchannel_name - In the format &quot;PortChannelxxxx&quot;, where &quot;xxxx&quot; is number of 1 to 4 digits. Ex: &quot;PortChannel0001&quot;</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>show interfaces portchannel</code></td>
<td>Displays the configured port channel interfaces</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>sudo config save</code> -y</td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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  
----- ----------------- -------- --------  

3.2.2 Remove Member Ports from a port channel

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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>config portchannel member del &lt;portchannel_name&gt; &lt;member_portname&gt;</td>
<td>Remove member port from a port channel</td>
</tr>
</tbody>
</table>
| | | 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

```bash
admin@sonic: ~$ sudo config portchannel member del PortChannel0001 Ethernet48
admin@sonic: ~$ sudo config portchannel member del PortChannel0001 Ethernet52
admin@sonic: ~$ sudo config portchannel member del PortChannel0001 Ethernet48

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:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>config portchannel del &lt;portchannel_name&gt;</td>
<td>Deletes a port channel interface</td>
</tr>
</tbody>
</table>
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
```
Add port channel PortChannel0004 as an untagged member for VLAN 10

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

Flags: A - active, I - inactive, Up - up, Dw - Down, N/A - not available,
S - selected, D - deselected, * - not synced

<table>
<thead>
<tr>
<th>No.</th>
<th>Team Dev</th>
<th>Protocol</th>
<th>Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>PortChannel0001</td>
<td>LACP(A)(Dw)</td>
<td>Ethernet48(D) Ethernet49(D)</td>
</tr>
<tr>
<td>0004</td>
<td>PortChannel0004</td>
<td>LACP(A)(Dw)</td>
<td>Ethernet50(D) Ethernet51(D)</td>
</tr>
</tbody>
</table>

```bash
admin@sonic: ~$ show vlan config
```

<table>
<thead>
<tr>
<th>Name</th>
<th>VID</th>
<th>Member</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vlan10</td>
<td>10</td>
<td>Ethernet1</td>
<td>untagged</td>
</tr>
<tr>
<td>Vlan10</td>
<td>10</td>
<td>Ethernet2</td>
<td>untagged</td>
</tr>
<tr>
<td>Vlan10</td>
<td>10</td>
<td>Ethernet3</td>
<td>untagged</td>
</tr>
<tr>
<td>Vlan10</td>
<td>10</td>
<td>Ethernet5</td>
<td>tagged</td>
</tr>
<tr>
<td>Vlan10</td>
<td>10</td>
<td>Ethernet6</td>
<td>tagged</td>
</tr>
<tr>
<td>Vlan10</td>
<td>10</td>
<td>Ethernet7</td>
<td>tagged</td>
</tr>
<tr>
<td>Vlan10</td>
<td>10</td>
<td>PortChannel0100</td>
<td>untagged</td>
</tr>
<tr>
<td>Vlan20</td>
<td>20</td>
<td>Ethernet4</td>
<td>tagged</td>
</tr>
<tr>
<td>Vlan20</td>
<td>20</td>
<td>Ethernet5</td>
<td>tagged</td>
</tr>
<tr>
<td>Vlan20</td>
<td>20</td>
<td>Ethernet6</td>
<td>tagged</td>
</tr>
<tr>
<td>Vlan20</td>
<td>20</td>
<td>Ethernet7</td>
<td>tagged</td>
</tr>
<tr>
<td>Vlan20</td>
<td>20</td>
<td>PortChannel0004</td>
<td>tagged</td>
</tr>
<tr>
<td>Vlan30</td>
<td>30</td>
<td>Ethernet4</td>
<td>tagged</td>
</tr>
<tr>
<td>Vlan30</td>
<td>30</td>
<td>Ethernet5</td>
<td>tagged</td>
</tr>
<tr>
<td>Vlan30</td>
<td>30</td>
<td>Ethernet6</td>
<td>tagged</td>
</tr>
<tr>
<td>Vlan30</td>
<td>30</td>
<td>Ethernet7</td>
<td>tagged</td>
</tr>
<tr>
<td>Vlan100</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vlan200</td>
<td>200</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```bash
admin@sonic: ~$ show vlan brief
```

```bash
<table>
<thead>
<tr>
<th>VLAN ID</th>
<th>IP Address</th>
<th>Ports</th>
<th>Port Tagging</th>
<th>Proxy ARP</th>
<th>DHCP Helper Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td></td>
<td>Ethernet1</td>
<td>untagged</td>
<td>disabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet2</td>
<td>untagged</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet3</td>
<td>untagged</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLDP Status</td>
<td>Enabled</td>
</tr>
<tr>
<td>LLDP PDU interval</td>
<td>30 secs</td>
</tr>
</tbody>
</table>

#### 3.3.2.2 Disable LLDP

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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>systemctl disable lldp</td>
<td>Disables LLDP</td>
</tr>
<tr>
<td>Step 2</td>
<td>show lldp table</td>
<td>Displays the LLDP neighbors in tabular format</td>
</tr>
<tr>
<td>Step 3</td>
<td>show lldp neighbors</td>
<td>Displays the LLDP neighbors</td>
</tr>
</tbody>
</table>

**Disable LLDP**

```bash
admin@sonic: ~$ sudo systemctl disable lldp
Removed /etc/systemd/system/sonic.target.wants/lldp.service.
admin@sonic: ~$ sudo systemctl disable lldp
Removed /etc/systemd/system/sonic.target.wants/lldp.service.
admin@sonic: ~$
```

Capability codes: (R) Router, (B) Bridge, (O) Other

```
LocalPort    RemoteDevice    RemotePortID    Capability    RemotePortDescr
-------------    -----------------    ----------------    ------------    -----------------
-------------------------------------------------------------------
Total entries displayed:  1
admin@sonic: ~$ sudo show lldp table
Capability codes: (R) Router, (B) Bridge, (O) Other
LocalPort    RemoteDevice    RemotePortID    Capability    RemotePortDescr
-------------    -----------------    ----------------    ------------    -----------------
-------------------------------------------------------------------
```

admin@sonic: ~$
```
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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>systemctl enable lldp</td>
<td>Enables LLDP</td>
</tr>
<tr>
<td>Step 2</td>
<td>show lldp table</td>
<td>Displays the LLDP neighbors in tabular format</td>
</tr>
<tr>
<td>Step 3</td>
<td>show lldp neighbors</td>
<td>Displays the LLDP neighbors</td>
</tr>
</tbody>
</table>

**Enable LLDP**

```bash
admin@sonic: ~$ sudo systemctl enable lldp
admin@sonic: ~$ sudo systemctl status lldp
```

**Capability codes: (R) Router, (B) Bridge, (O) Other**

<table>
<thead>
<tr>
<th>LocalPort</th>
<th>RemoteDevice</th>
<th>RemotePortID</th>
<th>Capability</th>
<th>RemotePortDescr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet0</td>
<td>Gi0/41</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total entries displayed: 1

```bash
admin@sonic: ~$ sudo systemctl status lldp
```

**Start LLDP service**

LLDP Service can be started using the below command

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```bash
Supermicro SONiC Configuration Guide v1.4 144```
### Start LLDP Service

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>systemctl start lldp</code></td>
<td>Starts LLDP service</td>
</tr>
<tr>
<td>2</td>
<td><code>show lldp table</code></td>
<td>Displays the LLDP neighbors in tabular format</td>
</tr>
<tr>
<td>3</td>
<td><code>show lldp neighbors</code></td>
<td>Displays the LLDP neighbors</td>
</tr>
</tbody>
</table>

#### 3.3.2.5  Stop LLDP service

LLDP Service can be stopped using the below command

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>systemctl stop lldp</code></td>
<td>Stops LLDP service</td>
</tr>
<tr>
<td>2</td>
<td><code>show lldp table</code></td>
<td>Displays the LLDP neighbors in tabular format</td>
</tr>
<tr>
<td>3</td>
<td><code>show lldp neighbors</code></td>
<td>Displays the LLDP neighbors</td>
</tr>
</tbody>
</table>
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

```bash
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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>config feature state dhcp_relay enabled</td>
<td>Enable the dhcp relay.</td>
</tr>
</tbody>
</table>
| 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
+-----------------+-----------------+---------+---------------+-----------+-------------------+
<table>
<thead>
<tr>
<th>VLAN ID</th>
<th>IP Address</th>
<th>Ports</th>
<th>Port Tagging</th>
<th>Proxy ARP</th>
<th>DHCP Helper Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>192.168.100.1</td>
<td>Ethernet0</td>
<td>Untagged</td>
<td>disabled</td>
<td>192.168.200.20</td>
</tr>
</tbody>
</table>
+-----------------+-----------------+---------+---------------+-----------+-------------------+

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
+-----------------+-----------------+---------+---------------+-----------+-------------------+
<table>
<thead>
<tr>
<th>VLAN ID</th>
<th>IP Address</th>
<th>Ports</th>
<th>Port Tagging</th>
<th>Proxy ARP</th>
<th>DHCP Helper Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>192.168.100.1</td>
<td>Ethernet0</td>
<td>Untagged</td>
<td>disabled</td>
<td>192.168.200.20</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>192.168.200.22</td>
</tr>
</tbody>
</table>
+-----------------+-----------------+---------+---------------+-----------+-------------------+

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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     | 192.168.100.20   | 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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>config feature state dhcp_relay enabled</code></td>
<td>Enable the dhcp relay.</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>config vlan dhcp_relay add &lt;vlan_id&gt; &lt;dhcp_relay_destination_ip&gt;</code></td>
<td>Add a DHCP Relay Destination IP address to the VLAN. <code>&lt;vlan_id&gt;</code> - may be any vlan number, <code>&lt;dhcp_relay_destination_ip&gt;</code> - IPv6 address.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>show vlan brief</code></td>
<td>Displays the configured VLAN information.</td>
</tr>
<tr>
<td>Step 4</td>
<td><code>sudo config save -y</code></td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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
```
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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>config vlan dhcp_relay del &lt;vlan_id&gt; &lt;dhcp_relay_destination_ip&gt;</code></td>
<td>Delete a configured DHCP Relay Destination IP address from a VLAN interface. &lt;br&gt;vlan_id - may be any vlan number &lt;br&gt;dhcp_relay_destination_ip – IPv6 address</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>show vlan brief</code></td>
<td>Displays the configured VLAN information.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>sudo config save -y</code></td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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: ~$ sudo config interface ip add Vlan100 192.168.100.10 /24
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: ~$ 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: ~$ show ip interface
<table>
<thead>
<tr>
<th>Interface</th>
<th>Master</th>
<th>IPv4 address/mask</th>
<th>Admin/Oper</th>
<th>BGP Neighbor</th>
<th>Neighbor IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet53</td>
<td></td>
<td>10.0.0.106/31</td>
<td>up/up</td>
<td>ARISTA27T0</td>
<td>10.0.0.107</td>
</tr>
<tr>
<td>Loopback0</td>
<td></td>
<td>10.1.0.1/32</td>
<td>up/up</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Vlan100</td>
<td></td>
<td>192.168.100.10/24</td>
<td>up/down</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>docker0</td>
<td></td>
<td>240.127.1.1/24</td>
<td>up/down</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>lo</td>
<td></td>
<td>127.0.0.1/16</td>
<td>up/up</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>
```
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.

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

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
+--------------------------+--------------------------+--------------------------+--------------------------+--------------------------+--------------------------+
<table>
<thead>
<tr>
<th>VLAN ID</th>
<th>IP Address</th>
<th>Ports</th>
<th>Port Tagging</th>
<th>Proxy ARP</th>
<th>DHCP Helper Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td>disabled</td>
<td>192.168.100.20</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Interface</th>
<th>Master</th>
<th>IPv4 address/mask</th>
<th>Admin/Oper</th>
<th>BGP Neighbor</th>
<th>Neighbor IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet53</td>
<td>up/up</td>
<td>10.0.0.106/31</td>
<td></td>
<td>ARISTA27T0</td>
<td>10.0.0.107</td>
</tr>
<tr>
<td>Loopback0</td>
<td>up/up</td>
<td>10.1.0.1/32</td>
<td></td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>
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 add 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

```bash
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

```bash
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: ~$ sudo show ip interface (Note: Truncated output is added here)
```

<table>
<thead>
<tr>
<th>Interface</th>
<th>IPv4 address/mask</th>
<th>Admin/Oper</th>
<th>BGP Neighbor</th>
<th>Neighbor IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet53</td>
<td>10.0.0.106/31</td>
<td>up/up</td>
<td>ARISTA27T0</td>
<td>10.0.0.107</td>
</tr>
<tr>
<td>Loopback0</td>
<td>10.1.0.1/32</td>
<td>up/up</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Vlan100</td>
<td>192.168.100.30/32</td>
<td>up/up</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Vlan200</td>
<td>192.168.100.40/32</td>
<td>up/up</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>docker0</td>
<td>240.127.1.1/24</td>
<td>up/down</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>lo</td>
<td>127.0.0.1/16</td>
<td>up/</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

```
admin@sonic: ~$ sudo 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,
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

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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>show arp [OPTIONS] [IPADDRESS]</td>
<td>Displays the arp table Options:</td>
</tr>
</tbody>
</table>
### Show ARP

The following example shows the ARP entries in the ARP table.

<table>
<thead>
<tr>
<th>Address</th>
<th>MacAddress</th>
<th>Iface</th>
<th>Vlan</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.86.1</td>
<td>28:bd:89:25:3e:0a</td>
<td>eth0</td>
<td>-</td>
</tr>
</tbody>
</table>

Total number of entries 1

admin@sonic: ~$ show arp

### Show ARP with Interface

<table>
<thead>
<tr>
<th>Address</th>
<th>MacAddress</th>
<th>Iface</th>
<th>Vlan</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.86.1</td>
<td>28:bd:89:25:3e:0a</td>
<td>eth0</td>
<td>-</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Address</th>
<th>MacAddress</th>
<th>Iface</th>
<th>Vlan</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.7.1.2</td>
<td>0c:c4:7a:14:fd:4e</td>
<td>Ethernet0</td>
<td>10</td>
</tr>
<tr>
<td>7.7.2.2</td>
<td>0c:c4:7a:15:0f:ae</td>
<td>Ethernet1</td>
<td>11</td>
</tr>
<tr>
<td>7.7.3.3</td>
<td>0c:c4:7a:15:0f:ae</td>
<td>Ethernet1</td>
<td>11</td>
</tr>
<tr>
<td>172.30.0.1</td>
<td>00:25:90:01:d4:44</td>
<td>eth0</td>
<td>-</td>
</tr>
<tr>
<td>172.30.0.253</td>
<td>3c:ec:ef:48:86:6b</td>
<td>eth0</td>
<td>-</td>
</tr>
</tbody>
</table>

Total number of entries 10

root@test:~# show arp -if Ethernet1

<table>
<thead>
<tr>
<th>Address</th>
<th>MacAddress</th>
<th>Iface</th>
<th>Vlan</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP Status</td>
<td>Active</td>
</tr>
<tr>
<td>Synchronization</td>
<td>Disabled</td>
</tr>
<tr>
<td>Preference</td>
<td>None</td>
</tr>
<tr>
<td>Peer</td>
<td>None</td>
</tr>
<tr>
<td>Connection retry time</td>
<td>120 seconds</td>
</tr>
</tbody>
</table>
### BGP Configuration

#### 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.**

   ```json
   "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"
       },
   
   "device_metadata": {
       "dev": {
           "role": "switch",
           "system": {
               "platform": "arm64-supermicro_sse_g3748-r0"
           }
       }
   }
   },
   
   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:**
   ```bash
   admin@sonic:~$ sudo -i
   root@sonic:~# vtysh
   Hello, this is FRRouting (version 7.5.1-sonic).
   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
   ```

---

**Figure IP-3: BGP topology**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hold time</td>
<td>180 seconds</td>
</tr>
<tr>
<td>Keep alive</td>
<td>60 seconds</td>
</tr>
<tr>
<td>Route Advertisement Interval</td>
<td>30 seconds</td>
</tr>
<tr>
<td>EBGP Multihop</td>
<td>Disable</td>
</tr>
<tr>
<td>AS Number</td>
<td>65100</td>
</tr>
<tr>
<td>Router ID</td>
<td>None</td>
</tr>
</tbody>
</table>

---

[Image of BGP topology with Switch A and Switch B connected via Ethernet9]
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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>sudo vtysh</td>
<td>Enter FRRouting</td>
</tr>
<tr>
<td>Step 2</td>
<td>configure terminal</td>
<td>Enters the configuration mode</td>
</tr>
<tr>
<td>Step 3</td>
<td>router bgp &lt;AS no (1-65535)&gt;</td>
<td>Enable BGP and configure the AS number of the BGP Speaker</td>
</tr>
<tr>
<td>Step 4</td>
<td>End</td>
<td>Exits Configuration mode.</td>
</tr>
</tbody>
</table>

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

### 4.5.12.4 BGP Peer

Follow the steps below to configure BGP Peer.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>sudo vtysh</td>
<td>Enter FRRouting</td>
</tr>
<tr>
<td>Step 2</td>
<td>configure terminal</td>
<td>Enters the configuration mode</td>
</tr>
<tr>
<td>Step 3</td>
<td>router bgp &lt;AS no (1-65535)&gt;</td>
<td>Enable BGP and configure the AS number of the BGP Speaker</td>
</tr>
<tr>
<td>Step 4</td>
<td>bgp router-id &lt;bgp router id (ip-address)&gt;</td>
<td>Configures the BGP Identifier of the BGP Speaker.</td>
</tr>
<tr>
<td>Step 5</td>
<td>neighbor &lt;ip-address&gt; remote-as &lt;AS no (1-65535)&gt;</td>
<td>Creates a Peer and initiates the connection to the peer.</td>
</tr>
<tr>
<td>Step 6</td>
<td>neighbor &lt;ip-address&gt; {advertisement-interval &lt;0-600 seconds&gt;}</td>
<td>(Optional) Configures neighbor interval.</td>
</tr>
<tr>
<td>Step 7</td>
<td>neighbor &lt;ip-address&gt; timers {keepalive &lt;seconds&gt;</td>
<td>holdtime &lt;seconds&gt;}</td>
</tr>
</tbody>
</table>
### 4.5.12.5 Attributes

Follow the steps below to configure BGP Attributes.

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

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

### 4.5.12.6 Network

Follow the steps below to configure Network through BGP.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>sudo vtysh</code></td>
<td>Enter FRRouting</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>configure terminal</code></td>
<td>Enters the configuration mode</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>router bgp &lt;AS no (1-65535)&gt;</code></td>
<td>Enable BGP and configure the AS number of the BGP Speaker</td>
</tr>
<tr>
<td>Step 4</td>
<td><code>no bgp ebgp-requires-policy</code></td>
<td>Disable Require policy on EBGP</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>sudo vtysh</code></td>
<td>Enter FRRouting</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>configure terminal</code></td>
<td>Enters the configuration mode</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>router bgp &lt;AS no (1-65535)&gt;</code></td>
<td>Enable BGP and configure the AS number of the BGP Speaker</td>
</tr>
<tr>
<td>Step 4</td>
<td><code>no bgp ebgp-requires-policy</code></td>
<td>Disable Require policy on EBGP</td>
</tr>
<tr>
<td>Step 5</td>
<td><strong>address-family ipv4 unicast</strong></td>
<td>Declare neighbors with whom need to exchange normal &quot;IPv4 unicast&quot; routes</td>
</tr>
<tr>
<td>Step 6</td>
<td><strong>redistribute connected</strong></td>
<td>Redistributes connected routes to internal and external BGP peers</td>
</tr>
<tr>
<td>Step 7</td>
<td><strong>exit-address-family</strong></td>
<td>Exits address-family mode</td>
</tr>
<tr>
<td>Step 8</td>
<td><strong>Exit</strong></td>
<td>Exits BGP Router Mode</td>
</tr>
<tr>
<td>Step 9</td>
<td><strong>End</strong></td>
<td>Exits Configuration mode.</td>
</tr>
</tbody>
</table>

---

The command “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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>sudo vtysh</code></td>
<td>Enter FRRouting</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>configure terminal</code></td>
<td>Enters the configuration mode</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>router bgp &lt;AS no (1-65535)&gt;</code></td>
<td>Enable BGP and configure the AS number of the BGP Speaker</td>
</tr>
<tr>
<td>Step 4</td>
<td><code>no bgp ebgp-requires-policy</code></td>
<td>Disable Require policy on EBGP</td>
</tr>
<tr>
<td>Step 5</td>
<td><strong>address-family ipv4 unicast</strong></td>
<td>Declare neighbors with whom need to exchange normal &quot;IPv4 unicast&quot; routes</td>
</tr>
<tr>
<td>Step 6</td>
<td><strong>redistribute connected</strong></td>
<td>Redistributes connected routes to internal and external BGP peers</td>
</tr>
<tr>
<td>Step 7</td>
<td><strong>exit-address-family</strong></td>
<td>Exits address-family mode</td>
</tr>
<tr>
<td>Step 8</td>
<td><strong>Exit</strong></td>
<td>Exits BGP Router Mode</td>
</tr>
<tr>
<td>Step 9</td>
<td><strong>End</strong></td>
<td>Exits Configuration mode.</td>
</tr>
<tr>
<td>Step</td>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Step 1</td>
<td><code>sudo vtysh</code></td>
<td>Enter FRRouting</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>configure terminal</code></td>
<td>Enters the configuration mode</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>router bgp &lt;AS no (1-65535)&gt;</code></td>
<td>Enable BGP and configure the AS number of the BGP Speaker</td>
</tr>
<tr>
<td>Step 4</td>
<td><code>no bgp ebgp-require-policy</code></td>
<td>Disable Require policy on EBGP</td>
</tr>
<tr>
<td>Step 5</td>
<td><code>address-family ipv4 unicast</code></td>
<td>Declare neighbors with whom need to exchange normal &quot;IPv4 unicast&quot; routes</td>
</tr>
<tr>
<td>Step 6</td>
<td><code>redistribute static</code></td>
<td>Redistributes static routes to internal and external BGP peers</td>
</tr>
<tr>
<td>Step 7</td>
<td><code>exit-address-family</code></td>
<td>Exits address-family mode</td>
</tr>
<tr>
<td>Step 8</td>
<td><code>exit</code></td>
<td>Exits BGP Router Mode</td>
</tr>
<tr>
<td>Step 9</td>
<td><code>End</code></td>
<td>Exits Configuration mode.</td>
</tr>
</tbody>
</table>

**4.5.13 BGP Configuration Example**

This section shows a sample BGP configuration.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>sudo vtysh</code></td>
<td>Enter FRRouting</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>configure</code></td>
<td>Enter configuration mode</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>no route bgp &lt; (1-4294967295) AS number&gt;</code></td>
<td>Remove default router (65100)</td>
</tr>
<tr>
<td>Step 4</td>
<td><code>router bgp &lt; (1-4294967295) AS number&gt;</code></td>
<td>Add new router</td>
</tr>
<tr>
<td>Step 5</td>
<td><code>bgp router-id &lt;A.B.C.D&gt;</code></td>
<td>Manually configure router identifier</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Router-id – Ipv4/Ipv6 address</td>
</tr>
<tr>
<td>Step 6</td>
<td><code>neighbor &lt;A.B.C.D Neighbor address&gt; remote-as &lt; (1-4294967295) AS number&gt;</code></td>
<td>Manually configure neighbor address and remote-as</td>
</tr>
<tr>
<td>Step 7</td>
<td><code>bgp default local-preference (0-4294967295)</code></td>
<td>Configure default local preference value</td>
</tr>
<tr>
<td>Step 8</td>
<td><code>neighbor &lt;A.B.C.D Neighbor address&gt; ebgp-multihop (1-255)</code></td>
<td>Configure ebgp-multihop</td>
</tr>
</tbody>
</table>

**Info**

no redistribute static stops the static routes to internal and external BGP peers
<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td><code>neighbor &lt;A.B.C.D Neighbor address&gt; timers (0-65535) connect</code></td>
<td>Configure timers - Keepalive interval and BGP connect timer.</td>
</tr>
<tr>
<td>10</td>
<td><code>neighbor &lt;A.B.C.D Neighbor address&gt; advertisement-interval (0-600)</code></td>
<td>Configure advertisement interval and time in seconds.</td>
</tr>
<tr>
<td>11</td>
<td><code>end</code></td>
<td>Exit configure mode in FRRouting.</td>
</tr>
<tr>
<td>12</td>
<td><code>show ip bgp neighbors</code></td>
<td>Displays configured BGP neighbor.</td>
</tr>
<tr>
<td>13</td>
<td><code>show ip bgp summary</code></td>
<td>Displays configured BGP details.</td>
</tr>
<tr>
<td>14</td>
<td><code>exit</code></td>
<td>Exit FRRouting.</td>
</tr>
<tr>
<td>15</td>
<td><code>sudo config save -y</code></td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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).
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)# neighbor 192.168.100.2 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).
```
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

<table>
<thead>
<tr>
<th></th>
<th>Sent</th>
<th>Rcvd</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>19</td>
<td>22</td>
</tr>
</tbody>
</table>

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  COA86402  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: ~$ sudo vtysh
Hello, this is FRRouting (version 7.5.1-sonic).

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

<table>
<thead>
<tr>
<th>Neighbor</th>
<th>V</th>
<th>AS</th>
<th>MsgRcvd</th>
<th>MsgSent</th>
<th>TblVer</th>
<th>InQ</th>
<th>OutQ</th>
<th>Up/Down</th>
<th>State/PfxRcd</th>
<th>PfxSnt</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.100.2</td>
<td>4</td>
<td>300</td>
<td>69</td>
<td>36</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>00:11:24</td>
<td>(Policy)</td>
<td>(Policy)</td>
</tr>
</tbody>
</table>

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
  
<table>
<thead>
<tr>
<th>Sent</th>
<th>Rcvd</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>15</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>28</td>
</tr>
</tbody>
</table>

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 FFFFFFF FFFFFFF 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::ec4:7a:fe2:8675
Nexthop local: fe80::ec4:7ade:fe2:8675
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).

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

<table>
<thead>
<tr>
<th>Neighbor</th>
<th>V</th>
<th>AS</th>
<th>MsgRcvd</th>
<th>MsgSent</th>
<th>TblVer</th>
<th>InQ</th>
<th>OutQ</th>
<th>Up/Down</th>
<th>State/PfxRcd</th>
<th>PfxSnt</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.100.1</td>
<td>4</td>
<td>200</td>
<td>36</td>
<td>69</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>00:15:24</td>
<td>(Policy)</td>
<td>(Policy)</td>
</tr>
</tbody>
</table>

Total number of neighbors 1
sonic#

### 4.5.14 Route Reflector Configuration

This section shows a sample BGP configuration for route reflector.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>sudo vtysh</td>
<td>Enter FRRouting</td>
</tr>
<tr>
<td>Step 2</td>
<td>configure</td>
<td>Enter configuration mode</td>
</tr>
<tr>
<td>Step 3</td>
<td>router bgp &lt; (1-4294967295) AS number&gt;</td>
<td>Add new router</td>
</tr>
<tr>
<td>Step 4</td>
<td>neighbor &lt;A.B.C.D Neighbor address&gt; remote-as &lt; (1-4294967295) AS number&gt;</td>
<td>Configure neighbor address and remote-as</td>
</tr>
<tr>
<td>Step 5</td>
<td>neighbor &lt;A.B.C.D Neighbor address&gt; route-reflector-client</td>
<td>Configure neighbor as route-reflector-client.</td>
</tr>
<tr>
<td>Step 6</td>
<td>end</td>
<td>Exit configure mode in FRRouting.</td>
</tr>
<tr>
<td>Step 7</td>
<td>write</td>
<td>Optional step - saves this configuration to be part of bgpd.conf.</td>
</tr>
<tr>
<td>Step 8</td>
<td>exit</td>
<td>Exit FRRouting</td>
</tr>
<tr>
<td>Step 9</td>
<td>sudo config save -y</td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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).

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>sudo vtysh</td>
<td>Enter FRRouting</td>
</tr>
<tr>
<td>Step 2</td>
<td>configure</td>
<td>Enter configuration mode</td>
</tr>
<tr>
<td>Step 3</td>
<td>router bgp &lt; (1-4294967295) AS number&gt;</td>
<td>Add new router</td>
</tr>
<tr>
<td>Step 4</td>
<td>neighbor &lt;Neighbor’s IPv6 address&gt; remote-as &lt; (1-4294967295) AS number&gt;</td>
<td>Configure neighbor’s IPv6 address and remote-as</td>
</tr>
<tr>
<td>Step 5</td>
<td>neighbor &lt;Neighbor’s IPv6 address&gt; route-reflector-client</td>
<td>Configure neighbor as route-reflector-client.</td>
</tr>
<tr>
<td>Step 6</td>
<td>end</td>
<td>Exit configure mode in FRRouting.</td>
</tr>
<tr>
<td>Step 7</td>
<td>write</td>
<td>Optional step - saves this configuration to be part of bgpd.conf.</td>
</tr>
<tr>
<td>Step 8</td>
<td>exit</td>
<td>Exit FRRouting</td>
</tr>
<tr>
<td>Step 9</td>
<td>sudo config save -y</td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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.
<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>sudo vtysh</code></td>
<td>Enter FRRouting</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>configure</code></td>
<td>Enter configuration mode</td>
</tr>
</tbody>
</table>
| 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 |
### 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] [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).
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.

- **Drop**
  - The switch drops all packets matching this ACL

- **Accept**
  - The switch accepts all packets matching this ACL

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>config acl add table [OPTIONS] &lt;table_name&gt; &lt;table_type&gt; [-d &lt;description&gt;] [-p &lt;ports&gt;] [-s (ingress</td>
<td>egress)]</td>
</tr>
<tr>
<td>Step 2</td>
<td>show acl table</td>
<td>Displays the ACL table.</td>
</tr>
<tr>
<td>Step 3</td>
<td>sudo config save -y</td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

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
```

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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.

```json
{
    "acl": {
        "acl-sets": {
            "acl-set": {
                "ACL_Rules_1": {
                    "acl-entries": {
                        "acl-entry": {
                            "1": {
                                "actions": {
                                    "config": {
```
Below is the ACL configuration to drop the packets with source IP address 172.31.0.19.

```json
{
   "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.

```json
{
    "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.

```json
{
    "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": 182  
                }  
              }  
            }  
          }  
        }  
      }  
    }  
  }  
}
```
"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.

```json
{
    "acl": {
        "acl-sets": {
            "acl-set": {
                "ACL_Rules_1": {
                    "acl-entries": {
                        "acl-entry": {
                            "1": {
                                "actions": {
                                    "config": {
                                        "forwarding-action": "DROP"
                                    }
                                }
                            }
                        }"sequence-id":1
                    }
                }
            }
        }
    }
}
```
5.1.1.5 Sample ACL Configuration Based on Destination Port

Below is the ACL configuration to accept the packets with destination port 179.

```json
{
  "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.

```json
{
  "acl": {
    "acl-sets": {
      "acl-set": {
        "ACL_Rules_1": {
          "acl-entries": {
            "acl-entry": {
            }
          }
        }
      }
    }
  }
}
```
Below is the ACL configuration to accept the packets with destination port range from 179 to 182.

```json
{
    "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": {

```
Below is the ACL configuration to drop the TCP packets.

```json
{
    "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.

```json
{
    "acl": {
        "acl-sets": {
            "acl-set": {
                "ACL_Rules_1": {
                    "acl-entries": {
                        "acl-entry": {
                            "1": {
                                "actions": {
                                    "config": {
                                        "forwarding-action": "ACCEPT"
                                    }
                                },
                                "config": {
                                    "sequence-id": 1
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

Below is the ACL configuration to drop the UDP packets.

```json
{
    "acl": {
        "acl-sets": {
            "acl-set": {
                "ACL_Rules_1": {
                    "acl-entries": {
                        "acl-entry": {
                            "1": {
                                "actions": {
                                    "config": {
                                        "forwarding-action": "DROP"
                                    }
                                },
                                "config": {
                                    "sequence-id": 1
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```
Below is the ACL configuration to accept the TCP packets with acknowledgement flag set.

```json
{
    "acl": {
        "acl-sets": {
            "acl-set": {
                "ACL_Rules_1": {
                    "acl-entries": {
                        "acl-entry": {
                            "5": {
                                "actions": {
                                    "config": {
                                        "forwarding-action": "ACCEPT"
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```
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.

```json
{
    "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"
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```
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.

```json
{
    "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"  
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```
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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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>
<thead>
<tr>
<th>Table</th>
<th>Rule</th>
<th>Priority</th>
<th>Action</th>
<th>Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACL_RULES_1</td>
<td>RULE_1</td>
<td>9999</td>
<td>DROP</td>
<td>ETHER_TYPE: 2048</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SRC_IPV6: fe80::ec4:7aff:fe2e:1000/124</td>
</tr>
<tr>
<td>ACL_RULES_1</td>
<td>RULE_2</td>
<td>9998</td>
<td>DROP</td>
<td>ETHER_TYPE: 2048</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>L4_SRC_PORT: 179</td>
</tr>
</tbody>
</table>

```
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`.

```json
{
    "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.

```json
{
    "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.

```json
{
    "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 destination IP address fe80::ec4:7aff:fe2e:2000/124.

```json

    "acl-entry": {
        "1": {
            "actions":{
                "config":{
                    "forwarding-action":"ACCEPT"
                }
            },
            "config":{
                "sequence-id":1
            },
            "ip":{
                "config":{
                    "destination-ip-address":"fe80::ec4:7aff:fe2e:1000/124"
                }
            }
        }
    }

```


5.1.2.4 Sample ACL Configuration Based on Source Port

Below is the ACL configuration to accept the packets with source port 179.

```json
{
    "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.

```json
{
    "acl": {
        "acl-sets": {
            "acl-set": {
                "ACL_Rules_2": {
                    "acl-entries": {
                        "acl-entry": {
                            "1": {
                                "actions": {
                                    "config": {
                                        "forwarding-action": "DROP"
                                    }
                                },
                                "config": {
                                    "sequence-id": 1
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```
Below is the ACL configuration to accept the packets with source port range from 179 to 182.

```json
{
    "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.

```json
{
   "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.

```json
{
   "acl": {
      "acl-sets": {
```

---

Supermicro SONiC Configuration Guide v1.4
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"
                           }
                        }
                     }
                  }
               }
            }
         }
      }
   }
}
```
Below is the ACL configuration to accept the packets with destination port range from 179 to 182.

```json
{
    "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.

```json
{
    "acl": {
        "acl-sets": {
            "acl-set": {
                "ACL_Rules_2": {
                    "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
                            }
                        }
                    }
                }
            }
        }
    }
}
```
Below is the ACL configuration to drop the TCP packets.

```json
{
    "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.

```json
{
    "acl": {
        "acl-sets": {

```

Supermicro SONiC Configuration Guide v1.4 207
Below is the ACL configuration to drop the UDP packets.

```json
{
  "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.

```json
{
    "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.

```json
{
    "acl": {
        "acl-sets": {

```
Below is the ACL configuration to accept the UDP packets with source IP address fe80::ec4:7aff:fe2e:1000/124.

```json
{
    "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"
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```
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.

```json
{
    "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": {
```
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

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Binding</th>
<th>Description</th>
<th>Stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACL_RULES_1</td>
<td>L3</td>
<td>Ethernet15</td>
<td>External access rules.</td>
<td>egress</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet16</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet17</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet18</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet19</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACL_RULES_2</td>
<td>L3</td>
<td>Ethernet26</td>
<td>Finance/accounting dept.</td>
<td>ingress</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet27</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet28</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet29</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet30</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Example-2:
```
root@sonic:~# show acl table

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Binding</th>
<th>Description</th>
<th>Stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACL_RULES_1</td>
<td>L3V6</td>
<td>Ethernet15</td>
<td>External access rules.</td>
<td>egress</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet16</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet17</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet18</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet19</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACL_RULES_2</td>
<td>L3V6</td>
<td>Ethernet26</td>
<td>Finance/accounting dept.</td>
<td>ingress</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet27</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ethernet28</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
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
```

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
```

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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. |
<table>
<thead>
<tr>
<th>Step 2</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td><code>config mirror_session remove [OPTIONS] &lt;session_name&gt;</code></td>
<td>Deletes a mirror session</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td><code>show mirror_session</code></td>
<td>Displays the configured VLAN information.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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
```

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
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
```
<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>DST Port</th>
<th>SRC Port</th>
<th>Direction</th>
<th>Queue</th>
<th>Policer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>config portchannel add PortChannel01</td>
<td>config portchannel add PortChannel01</td>
</tr>
<tr>
<td>config portchannel add PortChannel02</td>
<td>config portchannel add PortChannel02</td>
</tr>
<tr>
<td>config portchannel member add PortChannel01 Ethernet48</td>
<td>config portchannel member add PortChannel01 Ethernet48</td>
</tr>
<tr>
<td>config portchannel member add PortChannel02 Ethernet49</td>
<td>config portchannel member add PortChannel02 Ethernet49</td>
</tr>
<tr>
<td>config portchannel member add PortChannel03 Ethernet53</td>
<td>config portchannel member add PortChannel03 Ethernet53</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>config int ip add PortChannel01 10.10.10.1/24</td>
<td>config int ip add PortChannel01 10.10.10.1/24</td>
</tr>
<tr>
<td>config int ip add PortChannel02 20.20.20.1/24</td>
<td>config int ip add PortChannel02 20.20.20.1/24</td>
</tr>
<tr>
<td>config int ip add PortChannel03 192.168.10.1/24</td>
<td>config int ip add PortChannel03 192.168.10.2/24</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>config mclag add [OPTIONS] &lt;domain_id&gt; &lt;source_ip_addr&gt; &lt;peer_ip_addr&gt; &lt;peer_ifname&gt;</td>
<td>Add MCLAG domain. domain_id – The MCLAG domain id. source_ip_addr – IP address of the port-channel.</td>
</tr>
</tbody>
</table>
Step 2: 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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>config mclag member add 1 PortChannel01</code></td>
<td><code>config mclag member add 1 PortChannel01</code></td>
</tr>
<tr>
<td><code>config mclag member add 1 PortChannel02</code></td>
<td><code>config mclag member add 1 PortChannel02</code></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>config route add prefix 10.10.0.24</td>
<td>config route add prefix 10.10.0.24</td>
</tr>
<tr>
<td>nexthop 192.168.10.2</td>
<td>nexthop 192.168.10.1</td>
</tr>
<tr>
<td>config route add prefix 20.20.0.24</td>
<td>config route add prefix 20.20.0.24</td>
</tr>
<tr>
<td>nexthop 192.168.10.2</td>
<td>nexthop 192.168.10.1</td>
</tr>
</tbody>
</table>

### 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.

```bash
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
session 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.

```bash
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   | 20.20.0.10     | ac:1f:6b:38:73:52 | PortChannel02   | R    |
```
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

<table>
<thead>
<tr>
<th>No.</th>
<th>TYPE</th>
<th>MAC</th>
<th>VID</th>
<th>DEV</th>
<th>ORIGIN-DEV</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>D</td>
<td>ac:1f:6b:38:6e:f9</td>
<td>10</td>
<td>PortChannel01</td>
<td>PortChannel01</td>
<td>L</td>
</tr>
<tr>
<td>2</td>
<td>D</td>
<td>ac:1f:6b:38:6e:fa</td>
<td>10</td>
<td>PortChannel01</td>
<td>PortChannel01</td>
<td>P</td>
</tr>
<tr>
<td>3</td>
<td>D</td>
<td>ac:1f:6b:59:39:6c</td>
<td>10</td>
<td>PortChannel01</td>
<td>PortChannel01</td>
<td>L</td>
</tr>
<tr>
<td>4</td>
<td>D</td>
<td>ac:1f:6b:1b:7d:e3</td>
<td>20</td>
<td>PortChannel02</td>
<td>PortChannel02</td>
<td>L</td>
</tr>
<tr>
<td>5</td>
<td>D</td>
<td>ac:1f:6b:38:73:52</td>
<td>20</td>
<td>PortChannel02</td>
<td>PortChannel02</td>
<td>L</td>
</tr>
<tr>
<td>6</td>
<td>D</td>
<td>ac:1f:6b:38:73:53</td>
<td>20</td>
<td>PortChannel02</td>
<td>PortChannel02</td>
<td>P</td>
</tr>
<tr>
<td>7</td>
<td>S</td>
<td>88:5a:85:fa:2a:d1</td>
<td>100</td>
<td>PortChannel03</td>
<td>PortChannel03</td>
<td>L</td>
</tr>
</tbody>
</table>

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
```

```
------------------------------------------------------------
<p>| Ifindex: 13                                          |
| Type: Ethernet                                       |
| PortName: Ethernet1                                  |
| State: Up                                            |</p>
<table>
<thead>
<tr>
<th>VlanList:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>-------------------------------------------------------</td>
</tr>
</tbody>
</table>
-------------------------------------------------------
-------------------------------------------------------
-------------------------------------------------------
```

```
<p>| Ifindex: 12                                          |
| Type: Ethernet                                       |
| PortName: Ethernet0                                  |
| State: Up                                            |</p>
<table>
<thead>
<tr>
<th>VlanList:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>-------------------------------------------------------</td>
</tr>
</tbody>
</table>
```

root@sonic:~#
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
PortchannelsUp: 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
PortchannelsUp: 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
PortchannelsUp: 1
IsIsolateWithPeerlink: No
IsTrafficDisable: No
VlanList: 10 20 100

Supermicro SONiC Configuration Guide v1.4
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.

![Layer-3 MCLAG Topology Diagram]

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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>config portchannel add PortChannel01</td>
<td>config portchannel add PortChannel01</td>
</tr>
<tr>
<td>config portchannel add PortChannel02</td>
<td>config portchannel add PortChannel02</td>
</tr>
<tr>
<td>config portchannel add PortChannel03</td>
<td>config portchannel add PortChannel03</td>
</tr>
<tr>
<td>config portchannel member add PortChannel01 Ethernet48</td>
<td>config portchannel member add PortChannel01 Ethernet48</td>
</tr>
<tr>
<td>config portchannel member add PortChannel02 Ethernet49</td>
<td>config portchannel member add PortChannel02 Ethernet49</td>
</tr>
<tr>
<td>config portchannel member add PortChannel03 Ethernet53</td>
<td>config portchannel member add PortChannel03 Ethernet53</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>config int ip add PortChannel01 2002:aaaa:bbbb:cccc::1/64</td>
<td>config int ip add PortChannel01 2002:aaaa:bbbb:cccc::1/64</td>
</tr>
<tr>
<td>config int ip add PortChannel02 2001:aaaa:bbbb:dddd::1/64</td>
<td>config int ip add PortChannel02 2001:aaaa:bbbb:dddd::1/64</td>
</tr>
<tr>
<td>config int ip add PortChannel03 192.168.10.1/24</td>
<td>config int ip add PortChannel03 192.168.10.2/24</td>
</tr>
<tr>
<td>config int ip add PortChannel03 2000:aaaa:bbbb:eeee::1/64</td>
<td>config int ip add PortChannel03 2000:aaaa:bbbb:eeee::2/64</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>config mclag add [OPTIONS] &lt;domain_id&gt; &lt;source_ip_addr&gt; &lt;peer_ip_addr&gt; &lt;peer_ifname&gt;</td>
<td>Add MCLAG domain. &lt;br&gt; domain_id – The MCLAG domain id. &lt;br&gt; source_ip_addr – IP address of the port-channel. &lt;br&gt; peer_ip_addr – Ports to bind the ACL table. &lt;br&gt; peer_ifname – ingress/egress direction.</td>
</tr>
<tr>
<td>Step 2</td>
<td>sudo config save –y</td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

Example command to create MCLAG domain is given below.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
</table>
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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>config mclag member add [OPTIONS] &lt;domain_id&gt; &lt;portchannel_names&gt;</td>
<td>Add member MCLAG interfaces.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>domain_id – The MCLAG domain id.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>portchannel_names – Name of the port-channel.</td>
</tr>
<tr>
<td>Step 2</td>
<td>sudo config save –y</td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

Example command to add MCLAG port-channels to the domain is given below.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>config mclag member add 1 PortChannel01</td>
<td>config mclag member add 1 PortChannel01</td>
</tr>
<tr>
<td>config mclag member add 1 PortChannel02</td>
<td>config mclag member add 1 PortChannel02</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>vtysh</td>
<td>vtysh</td>
</tr>
<tr>
<td>configure terminal</td>
<td>configure terminal</td>
</tr>
</tbody>
</table>
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
Session 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.

![Sample Layer-2 MCLAG Topology](image)

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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>config portchannel add PortChannel01</td>
<td>config portchannel add PortChannel01</td>
</tr>
<tr>
<td>config portchannel add PortChannel02</td>
<td>config portchannel add PortChannel02</td>
</tr>
<tr>
<td>config portchannel add PortChannel03</td>
<td>config portchannel add PortChannel03</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>config vlan add 10</td>
<td>config vlan add 10</td>
</tr>
<tr>
<td>config vlan add 30</td>
<td>config vlan add 30</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>config int ip rem Ethernet48 10.0.0.96/31</td>
<td>config int ip rem Ethernet48 10.0.0.96/31</td>
</tr>
<tr>
<td>config int ip rem Ethernet49 10.0.0.98/31</td>
<td>config int ip rem Ethernet49 10.0.0.98/31</td>
</tr>
<tr>
<td>config int ip rem Ethernet53 10.0.0.106/31</td>
<td>config int ip rem Ethernet53 10.0.0.106/31</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>config portchannel member add PortChannel01 Ethernet48</td>
<td>config portchannel member add PortChannel01 Ethernet48</td>
</tr>
<tr>
<td>config portchannel member add PortChannel02 Ethernet49</td>
<td>config portchannel member add PortChannel02 Ethernet49</td>
</tr>
<tr>
<td>config portchannel member add PortChannel03 Ethernet53</td>
<td>config portchannel member add PortChannel03 Ethernet53</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>config vlan member add 30 PortChannel03</td>
<td>config vlan member add 30 PortChannel03</td>
</tr>
<tr>
<td>config vlan member add -u 10 PortChannel01</td>
<td>config vlan member add -u 10 PortChannel01</td>
</tr>
<tr>
<td>config vlan member add -u 10 PortChannel02</td>
<td>config vlan member add -u 10 PortChannel02</td>
</tr>
<tr>
<td>config vlan member add 10 PortChannel03</td>
<td>config vlan member add 10 PortChannel03</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>config mclag add [OPTIONS] &lt;domain_id&gt; &lt;source_ip_addr&gt; &lt;peer_ip_addr&gt; &lt;peer_ifname&gt;</td>
<td>Add MCLAG domain. &lt;br&gt; domain_id – The MCLAG domain id. &lt;br&gt; source_ip_addr – IP address of the port-channel. &lt;br&gt; peer_ip_addr – Ports to bind the ACL table. &lt;br&gt; peer_ifname – ingress/egress direction.</td>
</tr>
<tr>
<td>Step 2</td>
<td>sudo config save –y</td>
<td>Optional step - saves this configuration to be part of startup configuration.</td>
</tr>
</tbody>
</table>

Example command to create MCLAG domain and configuring a unique-ip is given below.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>config mclag add 1 192.168.10.1 192.168.10.2</td>
<td>config mclag add 1 192.168.10.2 192.168.10.1</td>
</tr>
<tr>
<td>config mclag unique-ip add Vlan3</td>
<td>config mclag unique-ip add Vlan3</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>config mclag member add [OPTIONS] &lt;domain_id&gt; &lt;portchannel_names&gt;</td>
<td>Add member MCLAG interfaces.</td>
</tr>
</tbody>
</table>
### 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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>config mclag member add 1 PortChannel01</td>
<td>config mclag member add 1 PortChannel01</td>
</tr>
<tr>
<td>config mclag member add 1 PortChannel02</td>
<td>config mclag member add 1 PortChannel02</td>
</tr>
</tbody>
</table>

### 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.

![MCLAG Topology Diagram]

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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>config portchannel add PortChannel01</td>
<td>config portchannel add PortChannel01</td>
</tr>
<tr>
<td>config portchannel add PortChannel02</td>
<td>config portchannel add PortChannel02</td>
</tr>
<tr>
<td>config portchannel add PortChannel03</td>
<td>config portchannel add PortChannel03</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>config vlan add 10</td>
<td>config vlan add 10</td>
</tr>
<tr>
<td>config vlan add 20</td>
<td>config vlan add 20</td>
</tr>
<tr>
<td>config vlan add 30</td>
<td>config vlan add 30</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>config int ip rem Ethernet48 10.0.0.96/31</td>
<td>config int ip rem Ethernet48 10.0.0.96/31</td>
</tr>
<tr>
<td>config int ip rem Ethernet49 10.0.0.98/31</td>
<td>config int ip rem Ethernet49 10.0.0.98/31</td>
</tr>
<tr>
<td>config int ip rem Ethernet53 10.0.0.106/31</td>
<td>config int ip rem Ethernet53 10.0.0.106/31</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>config portchannel member add PortChannel01 Ethernet48</td>
<td>config portchannel member add PortChannel01 Ethernet48</td>
</tr>
<tr>
<td>config portchannel member add PortChannel02 Ethernet49</td>
<td>config portchannel member add PortChannel02 Ethernet49</td>
</tr>
<tr>
<td>config portchannel member add PortChannel03 Ethernet53</td>
<td>config portchannel member add PortChannel03 Ethernet53</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>config vlan member add 30 PortChannel03</td>
<td>config vlan member add 30 PortChannel03</td>
</tr>
<tr>
<td>config vlan member add -u 10 PortChannel01</td>
<td>config vlan member add -u 10 PortChannel01</td>
</tr>
<tr>
<td>config vlan member add 10 PortChannel03</td>
<td>config vlan member add 10 PortChannel03</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>config mclag add 1 192.168.10.1 192.168.10.2</td>
<td>config mclag add 1 192.168.10.1 192.168.10.2</td>
</tr>
<tr>
<td>config mclag unique-ip add Vlan3</td>
<td>config mclag unique-ip add Vlan3</td>
</tr>
</tbody>
</table>

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.
<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>config mclag member add 1 PortChannel01</td>
<td>config mclag member add 1 PortChannel01</td>
</tr>
<tr>
<td>config mclag member add 1 PortChannel02</td>
<td>config mclag member add 1 PortChannel02</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>config portchannel add PortChannel01</td>
<td>config portchannel add PortChannel01</td>
</tr>
<tr>
<td>config portchannel add PortChannel02</td>
<td>config portchannel add PortChannel02</td>
</tr>
<tr>
<td>config portchannel add PortChannel03</td>
<td>config portchannel add PortChannel03</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>config vlan add 10</td>
<td></td>
</tr>
<tr>
<td>config vlan add 20</td>
<td></td>
</tr>
<tr>
<td>config vlan add 30</td>
<td></td>
</tr>
<tr>
<td>config vlan add 10</td>
<td></td>
</tr>
<tr>
<td>config vlan add 20</td>
<td></td>
</tr>
<tr>
<td>config vlan add 30</td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>config int ip rem Ethernet48 10.0.0.96/31</td>
<td></td>
</tr>
<tr>
<td>config int ip rem Ethernet49 10.0.0.98/31</td>
<td></td>
</tr>
<tr>
<td>config int ip rem Ethernet53 10.0.0.106/31</td>
<td></td>
</tr>
<tr>
<td>config int ip rem Ethernet48 10.0.0.96/31</td>
<td></td>
</tr>
<tr>
<td>config int ip rem Ethernet49 10.0.0.98/31</td>
<td></td>
</tr>
<tr>
<td>config int ip rem Ethernet53 10.0.0.106/31</td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>config portchannel member add PortChannel01 Ethernet48</td>
<td></td>
</tr>
<tr>
<td>config portchannel member add PortChannel02 Ethernet49</td>
<td></td>
</tr>
<tr>
<td>config portchannel member add PortChannel03 Ethernet53</td>
<td></td>
</tr>
<tr>
<td>config portchannel member add PortChannel01 Ethernet48</td>
<td></td>
</tr>
<tr>
<td>config portchannel member add PortChannel02 Ethernet49</td>
<td></td>
</tr>
<tr>
<td>config portchannel member add PortChannel03 Ethernet53</td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
</table>
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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| Step 1 | `config mclag add [OPTIONS] <domain_id> <source_ip_addr> <peer_ip_addr> <peer_iface>` | 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_iface – 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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>config mclag add 1 192.168.10.1 192.168.10.2</code></td>
<td><code>config mclag add 1 192.168.10.2 192.168.10.1</code></td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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.

<table>
<thead>
<tr>
<th>MCLAG Switch - 1</th>
<th>MCLAG Switch - 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>config mclag member add 1 PortChannel01</td>
<td>config mclag member add 1 PortChannel01</td>
</tr>
<tr>
<td>config mclag member add 1 PortChannel02</td>
<td>config mclag member add 1 PortChannel02</td>
</tr>
</tbody>
</table>
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
session timeout: 15
Peer Link Mac: 0c:c4:7a:3e:18:2d
Role: Active
MCLAG Interface: PortChannel02,PortChannel01
Loglevel: NOTICE
root@sonic:~#