



Cumulus RMP 2.5 ESR

User Guide

Table of Contents

Welcome to Cumulus Networks	7
Extended Support Release	8
Contents	9
Quick Start Guide	10
Contents	10
What's New in Cumulus RMP 2.5.12	10
Open Source Contributions	10
Prerequisites	11
Supported Hardware	11
Setting up a Cumulus RMP Switch	11
Upgrading Cumulus RMP	11
Configuring Cumulus RMP	11
Login Credentials	12
Serial Console Management	12
Wired Ethernet Management	12
In-Band Ethernet Management	12
Configuring the Hostname and Time Zone	13
Testing Cable Connectivity	13
Configuring Switch Ports	14
Layer 2 Port Configuration	14
Layer 3 Port Configuration	15
Configuring a Loopback Interface	16
System Management	18
Setting Date and Time	18
Contents	18
Commands	18
Setting the Time Zone	18
Setting the Date and Time	19
Setting Time Using NTP	19
Configuration Files	20
Useful Links	20
Authentication, Authorization, and Accounting	20
SSH for Remote Access	20
User Accounts	22
Using sudo to Delegate Privileges	23
LDAP Authentication and Authorization	29
Configuring switchd	39

Contents	39
The switchd File System	39
Configuring switchd Parameters	41
Restarting switchd	41
Commands	42
Configuration Files	42
Installation, Upgrading and Package Management	42
Managing Cumulus RMP Disk Images	42
Adding and Updating Packages	60
Zero Touch Provisioning	66

Configuring and Managing Network Interfaces 71

Contents	71
Commands	71
Man Pages	72
Configuration Files	72
Basic Commands	72
Bringing All auto Interfaces Up or Down	73
ifupdown Behavior with Child Interfaces	74
ifupdown2 Interface Dependencies	75
ifup Handling of Upper (Parent) Interfaces	78
Configuring IP Addresses	79
Purging Existing IP Addresses on an Interface	80
Specifying User Commands	80
Sourcing Interface File Snippets	80
Using Globs for Port Lists	81
Using Templates	81
Adding Descriptions to Interfaces	82
Caveats and Errata	83
Useful Links	83
Configuring Switch Port Attributes	84
Contents	84
Commands	84
Man Pages	84
Configuration Files	84
Interface Types	85
Settings	85
Verification and Troubleshooting Commands	88
Useful Links	90

Layer 1 and Layer 2 Features 91

Spanning Tree and Rapid Spanning Tree	91
Contents	91
Commands	91
PVST/PVRST	92

Creating a Bridge and Configuring STP	92
Configuring Spanning Tree Parameters	94
Bridge Assurance	102
BPDU Guard	103
BPDU Filter	105
Configuration Files	106
Man Pages	106
Useful Links	106
Caveats and Errata	106
Link Layer Discovery Protocol	106
Contents	107
Commands	107
Man Pages	107
Configuring LLDP	107
Example lldpcli Commands	108
Enabling the SNMP Subagent in LLDP	112
Configuration Files	112
Useful Links	112
Caveats and Errata	112
Prescriptive Topology Manager - PTM	113
Contents	113
Supported Features	113
Configuring PTM	114
Bidirectional Forwarding Detection (BFD)	118
Scripts	119
Using ptmd Service Commands	119
Using ptmctl Commands	120
Configuration Files	123
Useful Links	124
Bonding - Link Aggregation	124
Contents	124
Example: Bonding 4 Slaves	125
Hash Distribution	127
Configuration Files	127
Useful Links	127
Caveats and Errata	127
Ethernet Bridging - VLANs	128
Contents	128
VLAN Tagging	140
VLAN-aware Bridge Mode for Large-scale Layer 2 Environments	148
Routing	158
Contents	158
Commands	158
Configuring Static Routing	158

Persistently Adding a Static Route	158
Useful Links	159
Monitoring and Troubleshooting	160
Contents	160
Commands	160
Using the Serial Console	160
Configuring the Serial Console on PowerPC Switches	160
Configuring the Serial Console on x86 Switches	161
Diagnostics Using cl-support	162
Configuration Files	163
Next Steps	163
Single User Mode - Boot Recovery	163
Contents	164
Entering Single User Mode on a PowerPC Switch	164
Entering Single User Mode on an x86 Switch	164
Monitoring Interfaces and Transceivers Using ethtool	165
Contents	165
Commands	166
Monitoring Interfaces Using ethtool	166
Monitoring Switch Port SFP/QSFP Using ethtool	168
Resource Diagnostics Using cl-resource-query	169
Monitoring System Hardware	170
Contents	171
Commands	171
Monitoring Hardware Using decode-syseeprom	171
Monitoring Hardware Using sensors	173
Monitoring Switch Hardware Using SNMP	174
Monitoring System Units Using smond	176
Keeping the Switch Alive Using the Hardware Watchdog	176
Configuration Files	177
Useful Links	177
Monitoring System Statistics and Network Traffic with sFlow	177
Contents	178
Installing hsflowd	178
Configuring sFlow	178
Configuring sFlow Visualization Tools	180
Configuration Files	180
Useful Links	180
Understanding and Decoding the cl-support Output File	180
The cl-support command generates a tar archive of useful information for troubleshooting that can be auto-generated or manually created. To manually create it, run the cl-support command. The cl-support file is automatically generated when:	180
Understanding the File Naming Scheme	181
Decoding the Output	181

Troubleshooting Log Files	183
Troubleshooting the etc Directory	185
Troubleshooting the support Directory	195
Managing Application Daemons	196
Contents	196
Identifying Active Listener Ports for IPv4 and IPv6	196
Identifying Daemons Currently Active or Stopped	197
Identifying Boot Time State of a Specific Daemon	197
Disabling or Enabling a Specific Daemon	198
Troubleshooting Network Interfaces	199
Contents	199
Enabling Logging for Networking	199
Using ifquery to Validate and Debug Interface Configurations	200
Debugging Mako Template Errors	201
ifdown Cannot Find an Interface that Exists	202
MTU Set on a Logical Interface Fails with Error: "Numerical result out of range"	203
Interpreting iproute2 batch Command Failures	203
Understanding the "RTNETLINK answers: Invalid argument" Error when Adding a Port to a Bridge ..	203
Network Troubleshooting	204
Contents	204
Commands	204
Checking Reachability Using ping	204
Printing Route Trace Using traceroute	205
Manipulating the System ARP Cache	205
Traffic Generation Using mz	206
Counter ACL	207
SPAN and ERSPAN	208
Configuration Files	211
Useful Links	211
Caveats and Errata	211

Index	212
--------------------	------------

Welcome to Cumulus Networks

We are transforming networking with Cumulus Linux, the industry's first, full-featured Linux operating system for networking hardware. Cumulus RMP is a network operating system solution that enables out-of-band management use cases. It provides an open platform for customers and system integrators to use as is or build rack management applications on top.

Cumulus RMP shares the same architecture, foundation and user experience with Cumulus Linux. The feature sets are also customized to the needs of out-of-band management.



Cumulus® RMP™
Rack Management Platform

This documentation is current as of November 28, 2016 for version 2.5.11. Please visit the [Cumulus Networks Web site](#) for the most up to date documentation.

Read the [release notes](#) for new features and known issues in this release.

Extended Support Release

This version of Cumulus RMP is an Extended Support Release (ESR). Cumulus Linux 2.5 ESR started with Cumulus RMP 2.5.6 and all future releases in the 2.5 product family will all be ESR releases. To learn about ESR, please read [this article](#).

Cumulus Linux Version	Release Notes
2.5.12	Release notes
2.5.11	Release notes
2.5.10	Release notes
2.5.9	Release notes
2.5.8	Release notes
2.5.7	Release notes
2.5.6	Release notes

Contents

- [Quick Start Guide \(see page 10\)](#)
- [Installation, Upgrading and Package Management \(see page 42\)](#)
- [System Management \(see page 18\)](#)
- [Configuring and Managing Network Interfaces \(see page 71\)](#)
- [Layer 1 and Layer 2 Features \(see page 91\)](#)
- [Routing \(see page 158\)](#)

Quick Start Guide

This chapter helps you get up and running with Cumulus RMP quickly and easily.

Contents

(Click to expand)

- [Contents \(see page 10\)](#)
- [What's New in Cumulus RMP 2.5.12 \(see page 10\)](#)
- [Open Source Contributions \(see page 10\)](#)
- [Prerequisites \(see page 11\)](#)
- [Supported Hardware \(see page 11\)](#)
- [Setting up a Cumulus RMP Switch \(see page 11\)](#)
- [Upgrading Cumulus RMP \(see page 11\)](#)
- [Configuring Cumulus RMP \(see page 11\)](#)
 - [Login Credentials \(see page 12\)](#)
 - [Serial Console Management \(see page 12\)](#)
 - [Wired Ethernet Management \(see page 12\)](#)
 - [In-Band Ethernet Management \(see page 12\)](#)
 - [Configuring the Hostname and Time Zone \(see page 13\)](#)
 - [Testing Cable Connectivity \(see page 13\)](#)
- [Configuring Switch Ports \(see page 14\)](#)
 - [Layer 2 Port Configuration \(see page 14\)](#)
 - [Layer 3 Port Configuration \(see page 15\)](#)
- [Configuring a Loopback Interface \(see page 16\)](#)

What's New in Cumulus RMP 2.5.12

Cumulus RMP 2.5.12 is part of Cumulus RMP 2.5 ESR and as such, contains bug fixes only. The [release notes](#) contain information about the release as well as the fixed and known issues.

Open Source Contributions

Cumulus Networks has forked various software projects, like CFEngine, Netdev and some Puppet Labs packages in order to implement various Cumulus RMP features. The forked code resides in the Cumulus Networks [GitHub repository](#).

Cumulus Networks developed and released as open source some new applications as well.

The list of open source projects is on the [open source software](#) page.

Prerequisites

Prior intermediate Linux knowledge is assumed for this guide. You should be familiar with basic text editing, Unix file permissions, and process monitoring. A variety of text editors are pre-installed, including `vi` and `nano`.

You must have access to a Linux or UNIX shell. If you are running Windows, you should use a Linux environment like [Cygwin](#) as your command line tool for interacting with Cumulus RMP.



If you're a networking engineer but are unfamiliar with Linux concepts, use [this reference guide](#) to see examples of the Cumulus RMP CLI and configuration options, and their equivalent Cisco Nexus 3000 NX-OS commands and settings for comparison. You can also [watch a series of short videos](#) introducing you to Linux in general and some Cumulus Linux-specific concepts in particular.

Supported Hardware

You can find the most up to date list of supported switches [here](#). Use this page to confirm that your switch model is supported by Cumulus Networks. The page is updated regularly, listing products by port configuration, manufacturer, and SKU part number.

Setting up a Cumulus RMP Switch

Setting up a Cumulus RMP switch is simple and straightforward. It involves:

1. Racking the switch and connecting it to power.
2. Cabling all the ports.
3. Logging in and changing the default password.
4. Configuring switch ports and a loopback interface, if needed.

This quick start guide walks you through the steps necessary for getting your Cumulus RMP switch up and running after you remove it from the box.

Upgrading Cumulus RMP

If you already have Cumulus RMP installed on your switch and are upgrading to a maintenance release (X.Y.Z, like 2.5.7) from an earlier release in the same major and minor release family **only** (like 2.5.4 to 2.5.7), you can use various methods, including `apt-get`, to upgrade to the new version instead. See [Upgrading Cumulus RMP](#) (see page 42) for details.

Configuring Cumulus RMP

When bringing up Cumulus RMP for the first time, the management port makes a DHCPv4 request. To determine the IP address of the switch, you can cross reference the MAC address of the switch with your DHCP server. The MAC address should be located on the side of the switch or on the box in which the unit was shipped.

Login Credentials

The default installation includes one system account, *root*, with full system privileges, and one user account, *cumulus*, with sudo privileges. The *root* account password is set to null by default (which prohibits login), while the *cumulus* account is configured with this default password:

```
CumulusLinux!
```

In this quick start guide, you will use the *cumulus* account to configure Cumulus RMP.



For best security, you should change the default password (using the `passwd` command) before you configure Cumulus RMP on the switch.

All accounts except root are permitted remote SSH login; sudo may be used to grant a non-root account root-level access. Commands which change the system configuration require this elevated level of access.

For more information about sudo, read [Using sudo to Delegate Privileges](#) (see page 23).

Serial Console Management

Users are encouraged to perform management and configuration over the network, either in band or out of band. Use of the serial console is fully supported; however, many customers prefer the convenience of network-based management.

Typically, switches will ship from the manufacturer with a mating DB9 serial cable. Switches with ONIE are always set to a 115200 baud rate.

Wired Ethernet Management

Switches supported in Cumulus RMP contain a number of dedicated Ethernet management ports, the first of which is named *eth0*. These interfaces are geared specifically for out-of-band management use. The management interface uses DHCPv4 for addressing by default. While it is generally recommended to **not** assign an address to *eth0*, you can set a static IP address in the `/etc/network/interfaces` file:

```
auto eth0
iface eth0
    address 192.0.2.42/24
    gateway 192.0.2.1
```

In-Band Ethernet Management

All traffic that goes to the RMP switch via an interface called *vlan.1* is marked for in-band management. DHCP is enabled on this interface by default, and you can confirm the IP address at the command line. However, if you want to set a static IP address, change the configuration for *vlan.1* in `/etc/network/interfaces`:

```
auto vlan.1
iface vlan.1
    address 10.0.1.1/24
    gateway 10.0.2.1
```

Configuring the Hostname and Time Zone

To change the hostname, modify the `/etc/hostname` and `/etc/hosts` files with the desired hostname and reboot the switch. First, edit `/etc/hostname`:

```
cumulus@switch:~$ sudo vi /etc/hostname
```

Then replace the 127.0.1.1 IP address in `/etc/hosts` with the new hostname:

```
cumulus@switch:~$ sudo vi /etc/hosts
```

Reboot the switch:

```
cumulus@switch:~$ sudo reboot
```

To update the time zone, update the `/etc/timezone` file with the [correct timezone](#), run `dpkg-reconfigure --frontend noninteractive tzdata`, then reboot the switch:

```
cumulus@switch:~$ sudo vi /etc/timezone
cumulus@switch:~$ sudo dpkg-reconfigure --frontend noninteractive tzdata
cumulus@switch:~$ sudo reboot
```



It is possible to change the hostname without a reboot via a script available on [Cumulus Networks GitHub site](#).

Testing Cable Connectivity

By default, all data plane ports and the management interface are enabled.

To test cable connectivity, administratively enable a port using `ip link set <interface> up`:

```
cumulus@switch:~$ sudo ip link set swp1 up
```

To view link status, use `ip link show`. The following examples show the output of a port in "admin down", "down" and "up" mode, respectively:

```
# Administratively Down
swp1: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast state DOWN mode
DEFAULT qlen 1000

# Administratively Up but Layer 2 protocol is Down
swp1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state
DOWN mode DEFAULT qlen 500

# Administratively Up, Layer 2 protocol is Up
swp1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
mode DEFAULT qlen 500
```

Configuring Switch Ports

Layer 2 Port Configuration

By default, all the front panel ports (swp1 through swp52) are members of a bridge called *vlan*, as seen in `/etc/network/interfaces` below. The `glob` keyword is used to put the complete range of ports into the bridge:

```
# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp

auto vlan
iface vlan
    bridge-vlan-aware yes
    # needs to scale to large port count
    bridge-ports glob swp1-52
    bridge-stp on

auto vlan.1
# update with v6 configuration
iface vlan.1 inet dhcp
```

If you modify the configuration at all, you need to activate or apply the configuration to the kernel:

```
# First, check for typos:
cumulus@switch:~$ sudo ifquery -a

# Then activate the change if no errors are found:
cumulus@switch:~$ sudo ifup -a
```

To view the changes in the kernel, use the `brctl` command:

```
cumulus@switch:~$ brctl show
bridge name      bridge id        STP enabled      interfaces
br0              8000.089e01cedcc2  yes              swp1
```



A script is available to generate a configuration that [places all physical ports in a single bridge](#).

Layer 3 Port Configuration

To configure a front panel port or bridge interface as a Layer 3 port, edit the `/etc/network/interfaces` file.

In the following configuration example, the front panel port swp1 is configured a Layer 3 access port:

```
auto swp1
iface swp1
    address 10.1.1.1/30
```

To add an IP address to a bridge interface, remove the DHCP statement from the existing interface and include the address under the `iface` configuration in `/etc/network/interfaces`:

```
auto vlan.1
iface vlan.1
    address 10.2.2.1/24
```

To activate or apply the configuration to the kernel:

```
# First check for typos:
cumulus@switch:~$ sudo ifquery -a

# Then activate the change if no errors are found:
cumulus@switch:~$ sudo ifup -a
```

To view the changes in the kernel use the `ip addr show` command:

```
vlan.1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
link/ether 00:02:00:00:00:28 brd ff:ff:ff:ff:ff:ff
inet 10.2.2.1/24 scope global br0

swp1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
link/ether 44:38:39:00:6e:fe brd ff:ff:ff:ff:ff:ff
inet 10.1.1.1/30 scope global swp1
```

Configuring a Loopback Interface

Cumulus RMP has a loopback preconfigured in `/etc/network/interfaces`. When the switch boots up, it has a loopback interface, called `lo`, which is up and assigned an IP address of 127.0.0.1.

To see the status of the loopback interface (`lo`), use the `ip addr show lo` command:


```
cumulus@switch:~$ ip addr show lo
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
```

Note that the loopback is up and is assigned an IP address of 127.0.0.1.

To add an IP address to a loopback interface, add it directly under the `iface lo inet loopback` definition in `/etc/network/interfaces`:

```
auto lo
iface lo inet loopback
    address 10.1.1.1
```



If an IP address is configured without a mask, as shown above, the IP address becomes a /32. So, in the above case, 10.1.1.1 is actually 10.1.1.1/32.

Multiple loopback addresses can be configured by adding additional `address` lines:

```
auto lo
iface lo inet loopback
    address 10.1.1.1
    address 172.16.2.1/24
```

System Management

Setting Date and Time

Setting the time zone, date and time requires root privileges; use `sudo`.

Contents

(Click to expand)

- [Contents \(see page 18\)](#)
- [Commands \(see page 18\)](#)
- [Setting the Time Zone \(see page 18\)](#)
- [Setting the Date and Time \(see page 19\)](#)
- [Setting Time Using NTP \(see page 19\)](#)
- [Configuration Files \(see page 20\)](#)
- [Useful Links \(see page 20\)](#)

Commands

- `date`
- `dpkg-reconfigure tzdata`
- `hwclock`
- `ntpd (daemon)`
- `ntpq`

Setting the Time Zone

To see the current time zone, list the contents of `/etc/timezone`:

```
cumulus@switch:~$ cat /etc/timezone
US/Eastern
```

To set the time zone, run `dpkg-reconfigure tzdata` as root:

```
cumulus@switch:~$ sudo dpkg-reconfigure tzdata
```

Then navigate the menus to enable the time zone you want. The following example selects the US/Pacific time zone:

```
cumulus@switch:~$ sudo dpkg-reconfigure tzdata

Configuring tzdata
-----

Please select the geographic area in which you live. Subsequent
configuration
questions will narrow this down by presenting a list of cities, representing
the time zones in which they are located.

    1. Africa          4. Australia   7. Atlantic   10. Pacific   13. Etc
    2. America        5. Arctic     8. Europe    11. SystemV
    3. Antarctica     6. Asia       9. Indian    12. US
Geographic area: 12

Please select the city or region corresponding to your time zone.

    1. Alaska         4. Central    7. Indiana-Starke  10. Pacific
    2. Aleutian       5. Eastern    8. Michigan        11. Pacific-New
    3. Arizona        6. Hawaii     9. Mountain        12. Samoa
Time zone: 10

Current default time zone: 'US/Pacific'
Local time is now:      Mon Jun 17 09:27:45 PDT 2013.
Universal Time is now:  Mon Jun 17 16:27:45 UTC 2013.
```

For more info see the Debian [System Administrator's Manual – Time](#).

Setting the Date and Time

The switch contains a battery backed hardware clock that maintains the time while the switch is powered off and in between reboots. When the switch is running, the Cumulus RMP operating system maintains its own software clock.

During boot up, the time from the hardware clock is copied into the operating system's software clock. The software clock is then used for all timekeeping responsibilities. During system shutdown the software clock is copied back to the battery backed hardware clock.

You can set the date and time on the software clock using the `date` command. See `man date(1)` for details.

You can set the date and time on the hardware clock using the `hwclock` command. See `man hwclock(8)` for details.

A good overview of the software and hardware clocks can be found in the Debian [System Administrator's Manual – Time](#), specifically the section [Setting and showing hardware clock](#).

Setting Time Using NTP

The `ntpd` daemon running on the switch implements the NTP protocol. It synchronizes the system time with time servers listed in `/etc/ntp.conf`. It is started at boot by default. See `man ntpd(8)` for `ntpd` details.

By default, `/etc/ntp.conf` contains some default time servers. Edit `/etc/ntp.conf` to add or update time server information. See `man ntp.conf(5)` for details on configuring `ntpd` using `ntp.conf`.

To set the initial date and time via NTP before starting the `ntpd` daemon, use `ntpd -q` (This is same as `ntpdate`, which is to be retired and not available).



`ntpd -q` can hang if the time servers are not reachable.

To verify that `ntpd` is running on the system:

```
cumulus@switch:~$ ps -ef | grep ntp
ntp          4074      1  0 Jun20 ?           00:00:33 /usr/sbin/ntpd -p /var/run
/ntp.pid -g -u 101:102
```

Configuration Files

- `/etc/default/ntp` — `ntpd` `init.d` configuration variables
- `/etc/ntp.conf` — default NTP configuration file
- `/etc/init.d/ntp` — `ntpd` `init` script

Useful Links

- Debian System Administrator's Manual – Time
- <http://www.ntp.org>
- http://en.wikipedia.org/wiki/Network_Time_Protocol
- <http://wiki.debian.org/NTP>

Authentication, Authorization, and Accounting

- SSH for Remote Access (see page 20)
- User Accounts (see page 22)
- Using `sudo` to Delegate Privileges (see page 23)
- PAM and NSS (see page 29)

SSH for Remote Access

You use SSH to securely access a Cumulus RMP switch remotely.

Contents

(Click to expand)

- [Contents \(see page 21\)](#)
- [Access Using Passkey \(Basic Setup\) \(see page 21\)](#)
 - [Completely Passwordless System \(see page 22\)](#)
- [Useful Links \(see page 22\)](#)

Access Using Passkey (Basic Setup)

Cumulus RMP uses the openSSH package to provide SSH functionality. The standard mechanisms of generating passwordless access just applies. The example below has the cumulus user on a machine called management-station connecting to a switch called *cumulus-switch1*.

First, on management-station, generate the SSH keys:

```
cumulus@management-station:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/cumulus/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/cumulus/.ssh/id_rsa.
Your public key has been saved in /home/cumulus/.ssh/id_rsa.pub.
The key fingerprint is:
8c:47:6e:00:fb:13:b5:07:b4:1e:9d:f4:49:0a:77:a9 cumulus@management-
station
The key's randomart image is:
+--[ RSA 2048 ]-----+
|      .  . = o o .    |
|      o  . O * ..    |
|      . o = =.o      |
|      . O oE         |
|      + S           |
|      +             |
|                    |
|                    |
|                    |
+-----+

```

Next, append the public key in `~/.ssh/id_rsa.pub` into `~/.ssh/authorized_keys` in the target user's home directory:

```
cumulus@management-station:~$ scp .ssh/id_rsa.pub cumulus@cumulus-switch1:.  
ssh/authorized_keys  
Enter passphrase for key '/home/cumulus/.ssh/id_rsa':  
id_rsa.pub
```



Remember, you cannot use the root account to SSH to a switch in Cumulus RMP.

Completely Passwordless System

When generating the passphrase and its associated keys, as in the first step above, do not enter a passphrase. Follow all the other instructions.

Useful Links

- <http://www.debian-administration.org/articles/152>

User Accounts

By default, Cumulus RMP has two user accounts: *root* and *cumulus*.

The *cumulus* account:

- Default password is *CumulusLinux!*
- Is a user account in the *sudo* group with sudo privileges
- User can log in to the system via all the usual channels like console and SSH (see page 20)

The *root* account:

- Default password is disabled by default
- Has the standard Linux root user access to everything on the switch
- Disabled password prohibits login to the switch by SSH, telnet, FTP, and so forth

For best security, you should change the default password (using the `passwd` command) before you configure Cumulus Linux on the switch.

You can enable a valid password for the root account using the `sudo passwd root` command and can install an SSH key for the root account if needed. Enabling a password for the root account allows the root user to log in directly to the switch. The Cumulus Linux default root account behavior is consistent with Debian.

You can add more user accounts as needed. Like the *cumulus* account, these accounts must use sudo to execute privileged commands (see page 23), so be sure to include them in the *sudo* group.

To access the switch without any password requires booting into a single shell/user mode. [Here are the instructions \(see page 163\)](#) on how to do this using PowerPC and x86 switches.

Using sudo to Delegate Privileges

By default, Cumulus RMP has two user accounts: *root* and *cumulus*. The *cumulus* account is a normal user and is in the group *sudo*.

You can add more user accounts as needed. Like the *cumulus* account, these accounts must use `sudo` to execute privileged commands.

Contents

(Click to expand)

- [Contents \(see page 23\)](#)
- [Commands \(see page 23\)](#)
- [Using sudo \(see page 23\)](#)
- [sudoers Examples \(see page 24\)](#)
- [Configuration Files \(see page 29\)](#)
- [Useful Links \(see page 29\)](#)

Commands

- `sudo`
- `visudo`

Using sudo

`sudo` allows you to execute a command as superuser or another user as specified by the security policy. See `man sudo(8)` for details.

The default security policy is *sudoers*, which is configured using `/etc/sudoers`. Use `/etc/sudoers.d/` to add to the default *sudoers* policy. See `man sudoers(5)` for details.



Use `visudo` only to edit the `sudoers` file; do not use another editor like `vi` or `emacs`. See `man visudo(8)` for details.

Errors in the `sudoers` file can result in losing the ability to elevate privileges to root. You can fix this issue only by power cycling the switch and booting into single user mode. Before modifying `sudoers`, enable the root user by setting a password for the root user.

By default, users in the *sudo* group can use `sudo` to execute privileged commands. To add users to the *sudo* group, use the `useradd(8)` or `usermod(8)` command. To see which users belong to the *sudo* group, see `/etc/group` (`man group(5)`).

Any command can be run as `sudo`, including `su`. A password is required.

The example below shows how to use `sudo` as a non-privileged user *cumulus* to bring up an interface:

```
cumulus@switch:~$ ip link show dev swp1
3: swp1: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master br0 state
```

```

DOWN mode DEFAULT qlen 500
link/ether 44:38:39:00:27:9f brd ff:ff:ff:ff:ff:ff

cumulus@switch:~$ ip link set dev swp1 up
RTNETLINK answers: Operation not permitted

cumulus@switch:~$ sudo ip link set dev swp1 up
Password:

cumulus@switch:~$ ip link show dev swp1
3: swp1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master
br0 state UP mode DEFAULT qlen 500
link/ether 44:38:39:00:27:9f brd ff:ff:ff:ff:ff:ff

```

sudoers Examples

The following examples show how you grant as few privileges as necessary to a user or group of users to allow them to perform the required task. For each example, the system group *noc* is used; groups are prefixed with an %.

When executed by an unprivileged user, the example commands below must be prefixed with `sudo`.

Category	Privilege	Example Command	sudoers Entry
Monitoring	Switch port info	<code>ethtool -m swp1</code>	<code>%noc ALL=(ALL) NOPASSWD: /sbin/ethtool</code>
Monitoring	System diagnostics	<code>cl-support</code>	<code>%noc ALL=(ALL) NOPASSWD: /usr /cumulus/bin/cl-support</code>
Monitoring	Routing diagnostics	<code>cl-resource-query</code>	<code>%noc ALL=(ALL) NOPASSWD: /usr /cumulus/bin/cl-resource-query</code>
Image management	Install images		

Category	Privilege	Example Command	sudoers Entry
		<pre>cl-img-install http://lab /install.bin</pre>	<pre>%noc ALL=(ALL) NOPASSWD:/usr /cumulus/bin/cl-img-install</pre>
Image management	Swapping slots	<pre>cl-img-select 1</pre>	<pre>%noc ALL=(ALL) NOPASSWD:/usr /cumulus/bin/cl-img-select</pre>
Image management	Clearing an overlay	<pre>cl-img-clear- overlay 1</pre>	<pre>%noc ALL=(ALL) NOPASSWD:/usr /cumulus/bin/cl-img-clear- overlay</pre>
Package management	Any apt-get command	<pre>apt-get update or apt-get install</pre>	<pre>%noc ALL=(ALL) NOPASSWD:/usr /bin/apt-get</pre>
Package management	Just apt-get update	<pre>apt-get update</pre>	<pre>%noc ALL=(ALL) NOPASSWD:/usr /bin/apt-get update</pre>
Package management	Install packages	<pre>apt-get install mtr-tiny</pre>	<pre>%noc ALL=(ALL) NOPASSWD:/usr /bin/apt-get install *</pre>
Package management	Upgrading	<pre>apt-get upgrade</pre>	

Category	Privilege	Example Command	sudoers Entry
			<pre>%noc ALL=(ALL) NOPASSWD: /usr /bin/apt-get upgrade</pre>
L1 + 2 features	Any LLDP command	<pre>lldpcli show neighbors / configure</pre>	<pre>%noc ALL=(ALL) NOPASSWD: /usr /sbin/lldpcli</pre>
L1 + 2 features	Just show neighbors	<pre>lldpcli show neighbors</pre>	<pre>%noc ALL=(ALL) NOPASSWD: /usr /sbin/lldpcli show neighbours*</pre>
Interfaces	Modify any interface	<pre>ip link set dev swp1 {up down}</pre>	<pre>%noc ALL=(ALL) NOPASSWD: /sbin/ip link set *</pre>
Interfaces	Up any interface	<pre>ifup swp1</pre>	<pre>%noc ALL=(ALL) NOPASSWD: /sbin/ifup</pre>
Interfaces	Down any interface	<pre>ifdown swp1</pre>	<pre>%noc ALL=(ALL) NOPASSWD: /sbin/ifdown</pre>
Interfaces	Up/down only swp2	<pre>ifup swp2 / ifdown swp2</pre>	

Category	Privilege	Example Command	sudoers Entry
			<pre>%noc ALL=(ALL) NOPASSWD: /sbin/ifup swp2,/sbin /ifdown swp2</pre>
Interfaces	Any IP address chg	<pre>ip addr {add del} 192.0.2.1/30 dev swp1</pre>	<pre>%noc ALL=(ALL) NOPASSWD: /sbin/ip addr *</pre>
Interfaces	Only set IP address	<pre>ip addr add 192.0.2.1/30 dev swp1</pre>	<pre>%noc ALL=(ALL) NOPASSWD: /sbin/ip addr add *</pre>
Ethernet bridging	Any bridge command	<pre>brctl addbr br0 / brctl delif br0 swp1</pre>	<pre>%noc ALL=(ALL) NOPASSWD: /sbin/brctl</pre>
Ethernet bridging	Add bridges and ints	<pre>brctl addbr br0 / brctl addif br0 swp1</pre>	<pre>%noc ALL=(ALL) NOPASSWD: /sbin/brctl addbr *,/sbin /brctl addif *</pre>
Spanning tree	Set STP properties	<pre>mstpctl setmaxage br2 20</pre>	<pre>%noc ALL=(ALL) NOPASSWD: /sbin/mstpctl</pre>

Category	Privilege	Example Command	sudoers Entry
Troubleshooting	Restart switchd	<pre>service switchd restart</pre>	<pre>%noc ALL=(ALL) NOPASSWD:/usr /sbin/service switchd *</pre>
Troubleshooting	Restart any service	<pre>service switchd cron</pre>	<pre>%noc ALL=(ALL) NOPASSWD:/usr /sbin/service</pre>
Troubleshooting	Packet capture	<pre>tcpdump</pre>	<pre>%noc ALL=(ALL) NOPASSWD:/usr /sbin/tcpdump</pre>
L3	Add static routes	<pre>ip route add 10.2.0.0/16 via 10.0.0.1</pre>	<pre>%noc ALL=(ALL) NOPASSWD:/bin /ip route add *</pre>
L3	Delete static routes	<pre>ip route del 10.2.0.0/16 via 10.0.0.1</pre>	<pre>%noc ALL=(ALL) NOPASSWD:/bin /ip route del *</pre>
L3	Any static route chg	<pre>ip route *</pre>	<pre>%noc ALL=(ALL) NOPASSWD:/bin /ip route *</pre>
L3	Any iproute command	<pre>ip *</pre>	<pre>%noc ALL=(ALL) NOPASSWD:/bin /ip</pre>

Category	Privilege	Example Command	sudoers Entry
----------	-----------	-----------------	---------------

Configuration Files

- /etc/sudoers - default security policy
- /etc/sudoers.d/ - default security policy

Useful Links

- [sudo](#)
- [Adding Yourself to sudoers](#)

LDAP Authentication and Authorization

Cumulus RMP uses Pluggable Authentication Modules (PAM) and Name Service Switch (NSS) for user authentication.

NSS specifies the order of information sources used to resolve names for each service. Using this with authentication and authorization, it provides the order and location used for user lookup and group mapping on the system. PAM handles the interaction between the user and the system, providing login handling, session setup, authentication of users and authorization of a user actions.

NSS enables PAM to use LDAP for providing user authentication, group mapping and information for other services on the system.

Contents

(Click to expand)

- [Configuring LDAP Authentication \(see page 30\)](#)
- [Installing libnss-Idapd \(see page 30\)](#)
- [Configuring nslcd.conf \(see page 31\)](#)
 - [Connection \(see page 31\)](#)
 - [Search Function \(see page 32\)](#)
 - [Search Filters \(see page 32\)](#)
 - [Attribute Mapping \(see page 32\)](#)
 - [Example Configuration \(see page 33\)](#)
- [Troubleshooting \(see page 33\)](#)
 - [Using nslcd Debug Mode \(see page 33\)](#)
 - [Common Problems \(see page 34\)](#)
- [Configuring LDAP Authorization \(see page 36\)](#)
- [Active Directory Configuration \(see page 36\)](#)
- [LDAP Verification Tools \(see page 36\)](#)
 - [Identifying a User with the id Command \(see page 36\)](#)
 - [Using getent \(see page 37\)](#)
 - [Using LDAP search \(see page 37\)](#)

- LDAP Browsers (see page 38)
- References (see page 39)

Configuring LDAP Authentication

There are 3 common ways of configuring LDAP authentication on Linux:

- `libnss-ldap`
- `libnss-ldapd`
- `libnss-sss`

This chapter covers using `libnss-ldapd` only. From internal testing, this library worked best with Cumulus RMP and was the easiest to configure, automate and troubleshoot.

Installing `libnss-ldapd`



The `libnss-ldapd` and `ldap-utils` packages are not available in the Cumulus Networks repository. You must install them from the Debian repository. You need to configure the switch to reference the Debian repository. To do so, edit the `/etc/apt/sources.list` file and adding the following line:

```
deb http://ftp.us.debian.org/debian/ wheezy main
```

If *nested group support* is required, `libnss-ldapd` must be version 0.9 or higher. For Cumulus RMP 2.x, you should add the [wheezy-backports](#) repo instead of the wheezy repo:

```
deb http://ftp.us.debian.org/debian/ wheezy-backports main
```

Then run `apt-get update` to sync with the Debian repo.

Once you reference the Debian repository, install `libnss-ldapd`, `libpam-ldapd` and `ldap-utils`. Run:

```
cumulus@switch:~$ sudo apt-get install libnss-ldapd libpam-ldapd ldap-utils
```

This brings up an interactive prompt asking questions about the LDAP URI, search base distinguished name (DN) and services that should have LDAP lookups enabled. This creates a very basic LDAP configuration, using anonymous bind, and initiating the search for a user under the base DN specified.



Alternatively, these parameters can be pre-seeded using the `debconf-utils`. To use this method, run `apt-get install debconf-utils` and create the pre-seeded parameters using `debconf-set-selections` with the appropriate answers. Run `debconf-show <pkg>` to check the settings. Here is an [example of how to preseed answers to the installer questions using `debconf-set-selections`](#).

Once the install is complete, the *name service LDAP caching daemon* (`ns1cd`) will be running. This is the service that handles all of the LDAP protocol interactions, and caches the information returned from the LDAP server. In `/etc/nsswitch.conf`, `ldap` has been appended and is the secondary information source for `passwd`, `group` and `shadow`. The local files (`/etc/passwd`, `/etc/groups` and `/etc/shadow`) are used first, as specified by the `compat` source.

```
passwd: compat ldap
group:  compat ldap
shadow: compat ldap
```



You are strongly advised to keep `compat` as the first source in NSS for `passwd`, `group` and `shadow`. This prevents you from getting locked out of the system.

Configuring `ns1cd.conf`

You need to update the main configuration file (`/etc/ns1cd.conf`) after installation to accommodate the expected LDAP server settings. The [ns1cd.conf man page](#) details all the available configuration options. Some of the more important options are related to security and how the queries are handled.

Connection

The LDAP client starts a session by connecting to the LDAP server, by default, on TCP and UDP port 389, or on port 636 for LDAPS. Depending on the configuration, this connection may be unauthenticated (anonymous bind); otherwise, the client must provide a bind user and password. The variables used to define the connection to the LDAP server are the URI and bind credentials.

The URI is mandatory, and specifies the LDAP server location using the FQDN or IP address. It also designates whether to use `ldap://` for clear text transport, or `ldaps://` for SSL/TLS encrypted transport. Optionally, an alternate port may also be specified in the URI. Typically, in production environments, it is best to utilize the LDAPS protocol. Otherwise all communications are clear text and not secure.

After the connection to the server is complete, the BIND operation authenticates the session. The BIND credentials are optional, and if not specified, an anonymous bind is assumed. This is typically not allowed in most production environments. Configure authenticated (Simple) BIND by specifying the user (`binddn`) and password (`bindpw`) in the configuration. Another option is to use SASL (Simple Authentication and Security Layer) BIND, which provides authentication services using other mechanisms, like Kerberos. Contact your LDAP server administrator for this information since it depends on the configuration of the LDAP server and what credentials are created for the client device.

```
# The location at which the LDAP server(s) should be reachable.
uri ldaps://ldap.example.com
# The DN to bind with for normal lookups.
binddn cn=CLswitch,ou=infra,dc=example,dc=com
bindpw CuMuLuS
```

Search Function

When an LDAP client requests information about a resource, it must connect and bind to the server. Then it performs one or more resource queries depending on what it is looking up. All search queries sent to the LDAP server are created using the configured search *base*, *filter*, and the desired entry (*uid=myuser*) being searched for. If the LDAP directory is large, this search may take a significant amount of time. It is a good idea to define a more specific search base for the common *maps* (*passwd* and *group*).

```
# The search base that will be used for all queries.
base dc=example,dc=com
# Mapped search bases to speed up common queries.
base passwd ou=people,dc=example,dc=com
base group ou=groups,dc=example,dc=com
```

Search Filters

It is also common to use search filters to specify criteria used when searching for objects within the directory. This is used to limit the search scope when authenticating users. The default filters applied are:

```
filter passwd (objectClass=posixAccount)
filter group (objectClass=posixGroup)
```

Attribute Mapping

The *map* configuration allows for overriding the attributes pushed from LDAP. To override an attribute for a given *map**, specify the attribute name and the new value. One example of how this is useful is ensuring the shell is *bash* and the home directory is */home/cumulus*:

```
map    passwd homeDirectory "/home/cumulus"
map    passwd shell "/bin/bash"
```



*In LDAP, the **map** refers to one of the supported maps specified in the manpage for `nsldap.conf` (such as `passwd` or `group`).

Example Configuration

Here is an [example configuration](#) using Cumulus RMP.

Troubleshooting

Using `nsldap` Debug Mode

When setting up LDAP authentication for the first time, Cumulus Networks recommends you turn off this service using `service nsldap stop` and run it in debug mode. Debug mode works whether you are using LDAP over SSL (port 636) or an unencrypted LDAP connection (port 389).

```
cumulus@switch:~$ sudo service nsldap stop
cumulus@switch:~$ sudo nsldap -d
```

Once you enable debug mode, run the following command to test LDAP queries:

```
cumulus@switch:~$ sudo getent myuser
```

If LDAP is configured correctly, the following messages appear after you run the `getent` command:

```
nsldap: DEBUG: accept() failed (ignored): Resource temporarily unavailable
nsldap: [8elf29] DEBUG: connection from pid=11766 uid=0 gid=0
nsldap: [8elf29] <passwd(all)> DEBUG: myldap_search(base="dc=example,
dc=com", filter="(objectClass=posixAccount)")
nsldap: [8elf29] <passwd(all)> DEBUG: ldap_result(): uid=myuser,ou=people,
dc=example,dc=com
nsldap: [8elf29] <passwd(all)> DEBUG: ldap_result(): ... 152 more results
nsldap: [8elf29] <passwd(all)> DEBUG: ldap_result(): end of results (162
total)
```

In the output above, `<passwd(all)>` indicates that the entire directory structure was queried.

A specific user can be queried using the command:

```
cumulus@switch:~$ sudo getent passwd myuser
```

You can replace *myuser* with any username on the switch. The following debug output indicates that user *myuser* exists:

```
nsld: DEBUG: add_uri(ldap://10.50.21.101)
nsld: version 0.8.10 starting
nsld: DEBUG: unlink() of /var/run/nsld/socket failed (ignored): No such
file or directory
nsld: DEBUG: setgroups(0,NULL) done
nsld: DEBUG: setgid(110) done
nsld: DEBUG: setuid(107) done
nsld: accepting connections
nsld: DEBUG: accept() failed (ignored): Resource temporarily unavailable
nsld: [8b4567] DEBUG: connection from pid=11369 uid=0 gid=0
nsld: [8b4567] <passwd="myuser"> DEBUG: myldap_search(base="
dc=cumulusnetworks,dc=com", filter="(&(objectClass=posixAccount)
(uid=myuser))")
nsld: [8b4567] <passwd="myuser"> DEBUG: ldap_initialize
(ldap://<ip_address>)
nsld: [8b4567] <passwd="myuser"> DEBUG: ldap_set_rebind_proc()
nsld: [8b4567] <passwd="myuser"> DEBUG: ldap_set_option
(LDAP_OPT_PROTOCOL_VERSION,3)
nsld: [8b4567] <passwd="myuser"> DEBUG: ldap_set_option(LDAP_OPT_DEREF,0)
nsld: [8b4567] <passwd="myuser"> DEBUG: ldap_set_option(LDAP_OPT_TIMELIMIT,
0)
nsld: [8b4567] <passwd="myuser"> DEBUG: ldap_set_option(LDAP_OPT_TIMEOUT,0)
nsld: [8b4567] <passwd="myuser"> DEBUG: ldap_set_option
(LDAP_OPT_NETWORK_TIMEOUT,0)
nsld: [8b4567] <passwd="myuser"> DEBUG: ldap_set_option(LDAP_OPT_REFERRALS,
LDAP_OPT_ON)
nsld: [8b4567] <passwd="myuser"> DEBUG: ldap_set_option(LDAP_OPT_RESTART,
LDAP_OPT_ON)
nsld: [8b4567] <passwd="myuser"> DEBUG: ldap_simple_bind_s(NULL,NULL)
(uri="ldap://<ip_address>")
nsld: [8b4567] <passwd="myuser"> DEBUG: ldap_result(): end of results (0
total)
```

Notice how the `<passwd="myuser">` shows that the specific *myuser* user was queried.

Common Problems

SSL/TLS

- The FQDN of the LDAP server URI does not match the FQDN in the CA-signed server certificate exactly.

- `ns1cd` cannot read the SSL certificate, and will report a "Permission denied" error in the debug during server connection negotiation. Check the permission on each directory in the path of the root SSL certificate. Ensure that it is readable by the `ns1cd` user.

NSCD

- If the `nscd` cache daemon is also enabled and you make some changes to the user from LDAP, you may want to clear the cache using the commands:

```
nscd --invalidate = passwd
nscd --invalidate = group
```

- The `nscd` package works with `ns1cd` to cache name entries returned from the LDAP server. This may cause authentication failures. To work around these issues:

1. Disable `nscd` by running:

```
cumulus@switch:~$ sudo nscd -K
```

2. Restart the `ns1cd` service:

```
cumulus@switch:~$ sudo service ns1cd restart
```

3. Try the authentication again.

LDAP

- The search filter returns wrong results. Check for typos in the search filter. Use `ldapsearch` to test your filter.
- Optionally, configure the basic LDAP connection and search parameters in `/etc/ldap/ldap.conf`.

```
# ldapsearch -D 'cn=CLadmin' -w 'CuMuLuS' "(&
(ObjectClass=inetOrgUser)(uid=myuser))"
```

- When a local username also exists in the LDAP database, the order of the information sources in `/etc/nsswitch` can be updated to query LDAP before the local user database. This is generally not recommended. For example, the configuration below ensures that LDAP is queried before the local database.

```
# /etc/nsswitch.conf
passwd:          ldap compat
```

Configuring LDAP Authorization

Linux uses the *sudo* command to allow non-administrator users — like the default *cumulus* user account — to perform privileged operations. To control the users authorized to use *sudo*, the */etc/sudoers* file and files located in the */etc/sudoers.d/* directory have a series of rules defined. Typically, the rules are based on groups, but can also be defined for specific users. Therefore, *sudo* rules can be added using the group names from LDAP. For example, if a group of users were associated with the group *netadmin*, a rule can be added to give those users *sudo* privileges. Refer to the *sudoers* manual (*man sudoers*) for a complete usage description. Here's an illustration of this in */etc/sudoers*:

```
# The basic structure of a user specification is "who where = (as_whom)
what".
%sudo ALL=(ALL:ALL) ALL
%netadmin ALL=(ALL:ALL) ALL
```

Active Directory Configuration

Active Directory (AD) is a fully featured LDAP-based NIS server created by Microsoft. It offers unique features that classic OpenLDAP servers lack. Therefore, it can be more complicated to configure on the client and each version of AD is a little different in how it works with Linux-based LDAP clients. Some more advanced configuration examples, from testing LDAP clients on Cumulus RMP with Active Directory (AD /LDAP), are available in our [knowledge base](#).

LDAP Verification Tools

Typically, password and group information is retrieved from LDAP and cached by the LDAP client daemon. To test the LDAP interaction, these command line tools can be used to trigger an LDAP query from the device. This helps to create the best filters and verify the information sent back from the LDAP server.

Identifying a User with the *id* Command

The *id* command performs a username lookup by following the lookup information sources in NSS for the *passwd* service. This simply returns the user ID, group ID and the group list retrieved from the information source. In the following example, the user *cumulus* is locally defined in */etc/passwd*, and *myuser* is on LDAP. The NSS configuration has the *passwd* map configured with the sources *compat ldap*:

```
cumulus@switch:~$ id cumulus
uid=1000(cumulus) gid=1000(cumulus) groups=1000(cumulus),4(adm),27(sudo)
```

```
cumulus@switch:~$ id myuser
uid=1230(myuser) gid=3000(Development) groups=3000(Development),500
(Employees),27(sudo)
```

Using getent

The `getent` command retrieves all records found via NSS for a given map. It can also get a specific entry under that map. Tests can be done with the `passwd`, `group`, `shadow` or any other map configured in `/etc/nsswitch.conf`. The output from this command is formatted according to the map requested. Thus, for the `passwd` service, the structure of the output is the same as the entries in `/etc/passwd`. The same can be said for the group map will output the same as `/etc/group`. In this example, looking up a specific user in the `passwd` map, the user `cumulus` is locally defined in `/etc/passwd`, and `myuser` is only in LDAP.

```
cumulus@switch:~$ getent passwd cumulus
cumulus:x:1000:1000::/home/cumulus:/bin/bash
cumulus@switch:~$ getent passwd myuser
myuser:x:1230:3000:My Test User:/home/myuser:/bin/bash
```

In the next example, looking up a specific group in the group service, the group `cumulus` is locally defined in `/etc/groups`, and `netadmin` is on LDAP.

```
cumulus@switch:~$ getent group cumulus
cumulus:x:1000:
cumulus@switch:~$ getent group netadmin
netadmin*:502:matthew,mark,luke,john
```

Running the command `getent passwd` or `getent group` without a specific request, returns **all** local and LDAP entries for the `passwd` and `group` maps, respectively.

Using LDAP search

The `ldapsearch` command performs LDAP operations directly on the LDAP server. This does not interact with NSS. This command helps display what the LDAP daemon process is receiving back from the server. The command has many options. The simplest uses anonymous bind to the host and specifies the search DN and what attribute to lookup.

```
cumulus@switch:~$ ldapsearch -H ldap://ldap.example.com -b dc=example,
dc=com -x uid=myuser
```

[Click here to expand output of command](#)

```
# extended LDIF
#
# LDAPv3
# base <dc=example,dc=com> with scope subtree
# filter: uid=myuser
# requesting: ALL
#
# myuser, people, example.com
dn: uid=myuser,ou=people,dc=example,dc=com
cn: My User
displayName: My User
gecos: myuser
gidNumber: 3000
givenName: My
homeDirectory: /home/myuser
initials: MU
loginShell: /bin/bash
mail: myuser@example.com
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
objectClass: top
shadowExpire: -1
shadowFlag: 0
shadowMax: 999999
shadowMin: 8
shadowWarning: 7
sn: User
uid: myuser
uidNumber: 1234

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

LDAP Browsers

There are some GUI LDAP clients that help to work with LDAP servers. These are free tools to help graphically show the structure of the LDAP database.

- [Apache Directory Studio](#)
- [LDAPManager](#)

References

- <https://wiki.debian.org/LDAP/PAM>
- <https://raw.githubusercontent.com/arthurdejong/nss-pam-ldapd/master/nslcd.conf>
- <http://backports.debian.org/Instructions/>

Configuring switchd

`switchd` is the daemon at the heart of Cumulus RMP. It communicates between the switch and Cumulus RMP, and all the applications running on Cumulus RMP.

The `switchd` configuration is stored in `/etc/cumulus/switchd.conf`.

Contents

(Click to expand)

- Contents (see page 39)
- The switchd File System (see page 39)
- Configuring switchd Parameters (see page 41)
- Restarting switchd (see page 41)
- Commands (see page 42)
- Configuration Files (see page 42)

The switchd File System

`switchd` also exports a file system, mounted on `/cumulus/switchd`, that presents all the `switchd` configuration options as a series of files arranged in a tree structure. You can see the contents by parsing the `switchd` tree; run `tree /cumulus/switchd`. The output below is for a switch with one switch port configured:

```
cumulus@cumulus:~# sudo tree /cumulus/switchd/
/cumulus/switchd/
|-- config
|   |-- acl
|   |   |-- non_atomic_update_mode
|   |   |-- optimize_hw
|   |-- arp
|   |   |-- next_hops
|   |-- buf_util
|   |   |-- measure_interval
|   |   |-- poll_interval
|   |-- coalesce
|   |   |-- reducer
|   |   |-- timeout
```

```

| | -- disable_internal_restart
| | -- ignore_non_swps
| | -- interface
| | | -- swp1
| | | | -- storm_control
| | | | | -- broadcast
| | | | | -- multicast
| | | | | -- unknown_unicast
| | -- logging
| | -- route
| | | -- host_max_percent
| | | -- max_routes
| | | -- table
| | -- stats
| | | -- poll_interval
|-- ctrl
| | -- acl
| | -- hal
| | | -- resync
| | -- logger
| | -- netlink
| | | -- resync
| | -- resync
| | -- sample
| | | -- ulog_channel
|-- run
| | -- route_info
| | | -- ecmp_nh
| | | | -- count
| | | | -- max
| | | | -- max_per_route
| | | -- host
| | | | -- count
| | | | -- count_v4
| | | | -- count_v6
| | | | -- max
| | | -- mac
| | | | -- count
| | | | -- max
| | | -- route
| | | | -- count_0
| | | | -- count_1
| | | | -- count_total
| | | | -- count_v4
| | | | -- count_v6

```



```
|          |-- mask_limit
|          |-- max_0
|          |-- max_1
|          |-- max_total
|-- version
```

Configuring switchd Parameters

You can use `cl-cfg` to configure many `switchd` parameters at runtime (like ACLs, interfaces, and route table utilization), which minimizes disruption to your running switch. However, some options are read only and cannot be configured at runtime.

For example, to see data related to routes, run:

```
cumulus@cumulus:~$ sudo cl-cfg -a switchd | grep route
route.table = 254
route.max_routes = 32768
route.host_max_percent = 50
cumulus@cumulus:~$
```

To modify the configuration, run `cl-cfg -w`. For example, to set the buffer utilization measurement interval to 1 minute, run:

```
cumulus@cumulus:~$ sudo cl-cfg -w switchd buf_util.measure_interval=1
```

To verify that the value changed, use `grep`:

```
cumulus@cumulus:~# cl-cfg -a switchd | grep buf
buf_util.poll_interval = 0
buf_util.measure_interval = 1
```



You can get some of this information by running `cl-resource-query`; though you cannot update the `switchd` configuration with it.

Restarting switchd

Whenever you modify your network configuration (typically changing any `*.conf` file, like `/etc/cumulus/datapath/traffic.conf`), you must restart `switchd` for the changes to take effect:

```
cumulus@switch:~$ sudo service switchd restart
```

Commands

- [cl-cfg](#)

Configuration Files

- [/etc/cumulus/switchd.conf](#)

Installation, Upgrading and Package Management

A Cumulus RMP switch can have up to two images of the operating system installed. This section discusses installing new and updating existing Cumulus RMP disk images, and configuring those images with additional applications (via packages) if desired.

A Cumulus RMP switch comes pre-installed with the operating system.

Zero touch provisioning is a way to quickly deploy and configure new switches in a large-scale environment.

- [Managing Cumulus RMP Disk Images](#) (see page 42)
- [Adding and Updating Packages](#) (see page 60)
- [Zero Touch Provisioning](#) (see page 66)

Managing Cumulus RMP Disk Images

The Cumulus RMP operating system resides on a switch as a *disk image*. Switches running Cumulus RMP can be configured with multiple disk images. This section discusses how to manage them.

Contents

(Click to expand)

- [Contents](#) (see page 42)
- [Commands](#) (see page 43)
- [Upgrading Cumulus RMP](#) (see page 43)
- [Installing a New Cumulus RMP Image](#) (see page 43)
 - [Installing a New Image when Cumulus RMP Is already Installed](#) (see page 44)
 - [Step 1: Installing the New Image](#) (see page 44)
 - [Step 2: Backing up Your Configuration Files into /mnt/persist](#) (see page 46)
 - [Step 3: Selecting the Alternate Slot for Next Boot](#) (see page 49)
 - [Step 4: Rebooting the Switch](#) (see page 49)
 - [Step 5: Copying the Files from /mnt/persist to the New Slot](#) (see page 49)
 - [Step 6: Clearing /mnt/persist](#) (see page 50)
 - [Full Installation of Cumulus RMP Using ONIE over USB](#) (see page 50)
- [Understanding Image Slots](#) (see page 53)
 - [Managing Slot Sizes](#) (see page 54)

- [Accessing the Alternate Image Slot](#) (see page 55)
- [Reprovisioning the System \(Restart Installer\)](#) (see page 56)
- [Uninstalling All Images and Removing the Configuration](#) (see page 57)
- [Bootting into Rescue Mode](#) (see page 58)
- [Inspecting Image File Contents](#) (see page 58)
- [Useful Links](#) (see page 59)

Commands

- `cl-img-install`
- `cl-img-select`
- `cl-img-pkg`

Upgrading Cumulus RMP

If you already have Cumulus RMP installed on your switch and you are upgrading to an X.Y.Z release, like 2.5.7 from an earlier release in the same major and minor release family **only** (like 2.5.4 to 2.5.7), you can use `apt-get` to upgrade to the new version. (If are upgrading to a major (X.0) or minor (X.Y) release, you must do a full image install, as described in [Installing a New Cumulus RMP Image](#) (see page 43) below.)

To upgrade to a maintenance (X.Y.Z) release using `apt-get`:

1. Run `apt-get update`.
2. Run `apt-get dist-upgrade`.
3. Reboot the switch.



While this method doesn't overwrite the target image slot, the disk image does occupy a lot of disk space used by both Cumulus RMP image slots.

Installing a New Cumulus RMP Image

Cumulus RMP comes preinstalled on your switch. However there may be instances where you need to perform a full image installation. Before you install Cumulus RMP, the switch can be in two different states:

- The switch [already has Cumulus RMP installed](#) (see page 44) on it (see [below](#) (see page 44)).
- The switch has no image on it (so the switch is only running ONIE) or a clean installation is desired. In which case, you would install Cumulus RMP in one of the following ways:
 - Using [USB](#) (see page 50) (see [below](#) (see page 50)).
 - For all other ONIE installation methods, refer to [this knowledge base article](#).



ONIE is an open source project, equivalent to PXE on servers, that allows installation of network operating systems (NOS) on bare metal switches.

Unlike Cumulus Linux, there is no license to install on a Cumulus RMP switch.

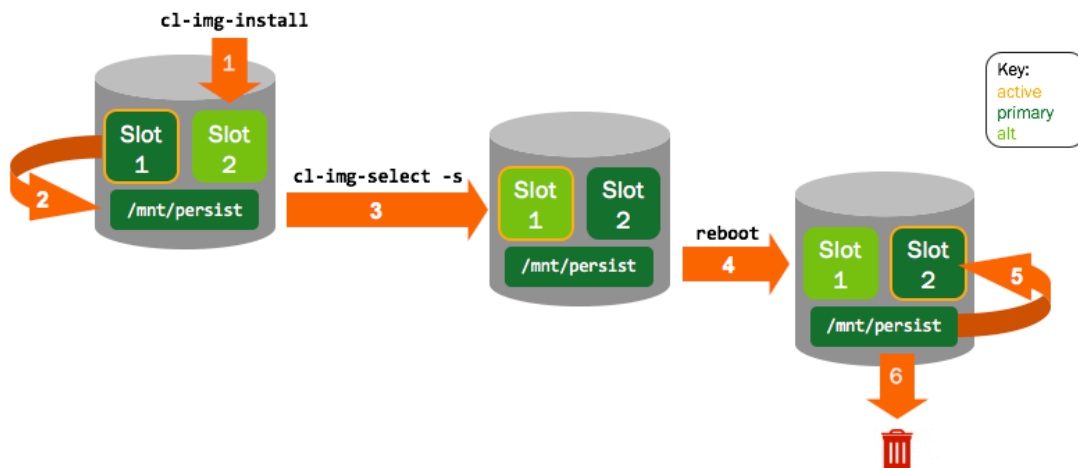
Installing a New Image when Cumulus RMP Is already Installed

Follow these upgrade steps for both major and minor releases, where:

- A major release upgrade is 2.X.X to 3.X.X (like 2.5.7 to 3.0.0)
- A minor release upgrade is X.2.X to X.3.X (like 2.5.4 to 2.5.7)

Installing a new image is a six step process:

1. Installing the new image into the alternate image slot (see [below](#) (see [page 53](#))).
2. Backing up your configuration files into `/mnt/persist`.
3. Selecting the alternate slot for next boot (that is, the slot you just installed into).
4. Rebooting the switch.
5. Copying the files from `/mnt/persist` to the new slot; this happens automatically if you follow the instructions below.
6. Clearing `/mnt/persist` out so subsequent reboots don't load `/mnt/persist`.



⚠ Installing a new image overwrites **all files** — including configuration files — on the target slot. Cumulus Networks strongly recommends you create a [persistent configuration](#) (see [page 46](#)) to back up your important files, like your configurations; see Step 2 below.

Step 1: Installing the New Image

Use the `cl-img-install` command to install a new image into the **alternate** image slot.

⚠ You can only install into the alternate slot, as it is not possible to install into the actively running slot. The system automatically determines which slot is the alternate slot (slot 2 in this case).

This example assumes the new image is located in the current directory (where the user is running the command from):

```
cumulus@switch:~$ sudo cl-img-install CumulusRMP-2.5.7-amd64.bin
```

Click to expand full output

```
cumulus@switch$ sudo cl-img-install CumulusRMP-2.5.7-amd64.bin
Defaulting to image slot 2 for install.
Dumping image info from CumulusRMP-2.5.7-amd64.bin ...
Verifying image checksum ... OK.
Preparing image archive ... OK.
Control File Contents
=====
Description: Cumulus RMP
OS-Release: 2.5.7-3b46bef-201509041633-build
Architecture: amd64
Date: Fri, 04 Sep 2015 17:10:30 -0700
Installer-Version: 1.2
Platforms: accton_as5712_54x accton_as6712_32x mlx_sx1400_i73612
dell_s6000_s1220 dell_s4000_c2338 dell_s3000_c2338 cel_redstone_xp
cel_smallstone_xp cel_pebble quanta_panther quanta_ly8_rangeley
quanta_ly6_rangeley quanta_ly9_rangeley
Homepage: http://www.cumulusnetworks.com/
Data Archive Contents
=====
-rw-r--r-- build/Development      131 2015-09-05 00:10:29 file.list
-rw-r--r-- build/Development      44 2015-09-05 00:10:29 file.list.
shal
-rw-r--r-- build/Development 140238619 2015-09-05 00:10:29 sysroot-
release.tar.gz
-rw-r--r-- build/Development      44 2015-09-05 00:10:30 sysroot-
release.tar.gz.shal
-rw-r--r-- build/Development  8094220 2015-09-05 00:10:29 vmlinuz-
initrd.tar.xz
-rw-r--r-- build/Development      44 2015-09-05 00:10:30 vmlinuz-
initrd.tar.xz.shal
Current image slot setup:
active => slot 1 (primary): 2.5.7-c4e83ad-201506011818-build
          slot 2 (alt   ): 2.5.4-727a0c6-201504132125-build
About to update image slot 2 using:
/home/cumulus/CumulusRMP-2.5.7-amd64.bin
Are you sure (y/N)? y
Verifying image checksum ... OK.
Preparing image archive ... OK.
Validating sha1 for vmlinuz-initrd.tar.xz... done.
Validating sha1 for sysroot-release.tar.gz... done.
Installing OS-Release 2.5.7-3b46bef-201509041633-build into image
slot 2 ...
Info: Copying sysroot into slot 2
Creating logical volume SYSROOT2 on volume group CUMULUS... done.
Verifying sysroot copy... OK.
Copying kernel into CLBOOT partition... done.
```

```

Verifying kernel copy... OK.
Generating grub.cfg ...
Found Cumulus RMP image: /boot/cl-vmlinuz-3.2.65-1+deb7u2+cl2.5+5-
slot-1
Found Cumulus RMP image: /boot/cl-vmlinuz-3.2.65-1+deb7u2+cl2.5+5-
slot-2
done
Success: /home/cumulus/CumulusRMP-2.5.7-amd64.bin loaded into image
slot 2.

```

Step 2: Backing up Your Configuration Files into /mnt/persist

Any files that have been modified from the factory default should be backed up to /mnt/persist.

Recommended Files to Make Persistent

Cumulus Networks recommends you consider making the following files and directories part of a persistent configuration.

Network Configuration Files

File Name and Location	Explanation	Cumulus RMP Documentation	Debian Documentation
/etc /network/	Network configuration files, most notably /etc /network /interfaces	Configuring and Managing Network Interfaces (see page 71)	wiki.debian.org /NetworkConfiguration
/etc/resolv.conf	DNS resolution	Not unique to Cumulus RMP: wiki.debian.org /NetworkConfiguration#The_resolv.conf_configuration_file	www.debian.org/doc /manuals/debian-reference/ch05.en.html
/etc /hostname	Configuration file for the hostname of the switch	Quick Start Guide#ConfiguringtheHostnameandTimeZone (see page 13)	wiki.debian.org/HowTo /ChangeHostname
/etc /cumulus /ports.conf	Breakout cable configuration file	Configuring Switch Port Attributes#ConfiguringBreakoutPorts (see page)	N/A; please read the guide on breakout cables

Additional Commonly Used Files

File Name and Location	Explanation	Cumulus RMP Documentation	Debian Documentation
/etc/motd	Message of the day	Not unique to Cumulus RMP	wiki.debian.org/motd#Wheezy
/etc/passwd	User account information	Not unique to Cumulus RMP	www.debian.org/doc/manuals/debian-reference/ch04.en.html
/etc/shadow	Secure user account information	Not unique to Cumulus RMP	www.debian.org/doc/manuals/debian-reference/ch04.en.html
/etc/lldpd.conf	Link Layer Discover Protocol (LLDP) daemon configuration	Link Layer Discovery Protocol (see page 106)	packages.debian.org/wheezy/lldpd
/etc/lldpd.d/	Configuration directory for lldpd	Link Layer Discovery Protocol (see page 106)	packages.debian.org/wheezy/lldpd
/etc/nsswitch.conf	Name Service Switch (NSS) configuration file	LDAP Authentication and Authorization (see page 29)	wiki.debian.org/LDAP/NSS
/etc/ssh/	SSH configuration files	SSH for Remote Access (see page 20)	wiki.debian.org/SSH
/etc/ldap/ldap.conf	Lightweight Directory Access Protocol configuration file	LDAP Authentication and Authorization (see page 29)	www.debian.org/doc/manuals/debian-reference/ch04.en.html

- If you are using the root user account, consider including `/root/`.
- If you have custom user accounts, consider including `/home/<username>/`.

Simple Bash Script Example

Example Bash script to automate `/mnt/persist` backup; click to expand...

The following script is a Bash script that can help grab all the above files and push them to `/mnt/persist` automatically.

```
#!/bin/bash
#network configuration files
cp -r --parents /etc/network/ /mnt/persist/
cp --parents /etc/resolv.conf /mnt/persist/
```

```

if [ -f /etc/quagga/Quagga.conf ]; then cp --parents /etc/quagga
/Quagga.conf /mnt/persist; fi
cp --parents /etc/quagga/daemons /mnt/persist
cp --parents /etc/hostname /mnt/persist
cp --parents /etc/cumulus/ports.conf /mnt/persist
#commonly used files
cp --parents /etc/motd /mnt/persist/
cp --parents /etc/passwd /mnt/persist/
cp --parents /etc/shadow /mnt/persist/
if [ -f /etc/lldpd.conf ]; then cp --parents /etc/lldpd.conf /mnt
/persist; fi
cp -r --parents /etc/lldpd.d/* /mnt/persist/
cp --parents /etc/nsswitch.conf /mnt/persist
cp -a --parents /etc/ssh/ /mnt/persist/
if [ -f /etc/ldap.conf ]; then cp --parents /etc/ldap.conf /mnt
/persist; fi

```

To run the script copy the above into a `.sh` file (for example, `sudo nano backup.sh`).

```
cumulus@switch$ bash backup.sh
```

To check if the script worked use the Linux `tree` command:

```

cumulus@switch$ tree /mnt/persist
/mnt/persist
|-- etc
|   |-- cumulus
|   |   |-- ports.conf
|   |-- hostname
|   |-- lldpd.d
|   |   |-- README.conf
|   |-- motd
|   |-- network
|   |   |-- if-down.d
|   |   |-- if-post-down.d
|   |   |-- if-post-up.d
|   |   |-- if-pre-down.d
|   |   |-- if-pre-up.d
|   |   |-- ethtool
|   |   |-- if-up.d
|   |   |   |-- ethtool
|   |   |   |-- mountnfs
|   |   |   |-- openssh-server
|   |   |-- ifupdown2
|   |   |   |-- ifupdown2.conf
|   |   |-- interfaces
|   |   |-- interfaces.d
|   |   |-- run -> /run/network
|   |-- nsswitch.conf

```



```
| -- passwd
| -- quagga
|   |-- Quagga.conf
|   |-- daemons
| -- resolv.conf
| -- shadow
|-- ssh
|   |-- moduli
|   |-- ssh_config
|   |-- ssh_host_dsa_key
|   |-- ssh_host_dsa_key.pub
|   |-- ssh_host_ecdsa_key
|   |-- ssh_host_ecdsa_key.pub
|   |-- ssh_host_rsa_key
|   |-- ssh_host_rsa_key.pub
|-- sshd_config
```

Step 3: Selecting the Alternate Slot for Next Boot

To select the slot you just installed into, either use `cl-img-select -s` to switch the primary slot to the alternate slot, or use `cl-img-select` with the number of the slot you want directly (for example, `cl-img-select 2`).

```
cumulus@switch$ sudo cl-img-select -s
Success: Primary image slot set to 2.
active => slot 1 (alt   ): 2.5.7-c4e83ad-201506011818-build
         slot 2 (primary): 2.5.6-3b46bef-201509041633-build
Reboot required to take effect.
```

Step 4: Rebooting the Switch

Reboot the switch to boot into the new primary slot.

```
cumulus@switch$ reboot
```

Step 5: Copying the Files from /mnt/persist to the New Slot

Files in `/mnt/persist` automatically are rolled into the primary image slot when the switch boots. For example, in this scenario everything in `/mnt/persist` gets automatically copied into slot 2 when the reboot is performed in step 4 above. The files in `/mnt/persist` keep their relative path after the reboot. For example, if there was a `/mnt/persist/etc/network/interfaces`, it would be copied into `/etc/network/interfaces`.

Use the `tree` command to look at the folder structure of `/mnt/`.

```
cumulus@switch$ tree /mnt/  
/mnt  
|-- persist  
    |-- etc  
        |-- network  
            |-- interfaces
```

So in this case `/mnt/persist/etc/network/interfaces` overrides the primary slot's `/etc/network/interfaces` on boot.

Step 6: Clearing `/mnt/persist`

If `/mnt/persist` is not cleared out, everything in `/mnt/persist` will overwrite any relative files in the primary slot whenever the switch boots. This can be a problem if a user modifies some files but forgets to also make the changes to `/mnt/persist`. It is best practice to clear out `/mnt/persist` so that any subsequent users can make changes and not have them overwritten the next time the switch boots.

```
cumulus@switch$ sudo rm -r /mnt/persist/*  
cumulus@switch$ ls /mnt/persist/  
cumulus@switch$
```



This is an extra reminder to clear out `/mnt/persist`. A future reboot will cause **everything** in `/mnt/persist` to overwrite the current primary slot.

Full Installation of Cumulus RMP Using ONIE over USB

Follow the steps below to conduct a full installation of Cumulus RMP. This wipes out all pre-existing configuration files that may be present on the switch.



Make sure to back up any important configuration files that you may need to restore the configuration of your switch after the installation finishes.

Preparing for USB Installation

1. Download the appropriate Cumulus RMP image for your x86 platform from the [Cumulus Downloads page](#).
2. Prepare your flash drive by formatting in one of the supported formats: FAT32, vFAT or EXT2.

Optional: Preparing a USB Drive inside Cumulus Linux



It is possible that you could severely damage your system with the following utilities, so please use caution when performing the actions below!

- a. Insert your flash drive into the USB port on the switch running Cumulus RMP and log in to the switch.
- b. Determine and note which device your flash drive can be found at using output from `cat /proc/partitions` and `sudo fdisk -l [device]`. For example, `sudo fdisk -l /dev/sdb`. These instructions assume your USB drive is the `/dev/sdb` device, which is typical. Make sure to modify the commands below to use the proper device for your USB drive.
- c. Create a new partition table on the device:

```
sudo parted /dev/sdb mklabel msdos
```

- d. Create a new partition on the device:

```
sudo parted /dev/sdb -a optimal mkpart primary 0% 100%
```

- e. Format the partition to your filesystem of choice using ONE of the examples below:

```
sudo mkfs.ext2 /dev/sdb1
sudo mkfs.msos -F 32 /dev/sdb1
sudo mkfs.vfat /dev/sdb1
```



To use `mkfs.msos` or `mkfs.vfat`, you need to install the `dosfstools` package from the [Debian software repositories](#) (step 3 here shows you how to add repositories from Debian), as they are not included by default.

- f. To continue installing Cumulus RMP, mount the USB drive in order to move files to it.

```
sudo mkdir /mnt/usb
sudo mount /dev/sdb1 /mnt/usb
```

3. Copy the image file over to the flash drive and rename the image file to `onie-installer_x86-64`.
4. Insert the USB stick into the switch, then continue with the appropriate instructions below for your x86 platform.
5. Prepare the switch for installation:
 - If the switch is offline, connect to the console and power on the switch.
 - If the switch is already online in Cumulus RMP, connect to the console and reboot the switch into the ONIE environment with the `sudo cl-img-select -i` command, followed by `sudo reboot`. Then skip to step 4 below.
 - If the switch is already online in ONIE, use the `reboot` command.

6. SSH sessions to the switch get dropped after this step. To complete the remaining instructions, connect to the console of the switch. Cumulus RMP switches display their boot process to the console, so you need to monitor the console specifically to complete the next step.

7. Monitor the console and select the ONIE option from the first GRUB screen shown below.

```
GNU GRUB  version 1.99-27+deb7u2

+-----+
|Cumulus Linux 2.5.3a-3b46bef-201509041633-build - slot 1|
|Cumulus Linux 2.5.3a-3b46bef-201509041633-build - slot 1 (recovery mode)|
|Cumulus Linux 2.5.3a-3b46bef-201509041633-build - slot 2|
|Cumulus Linux 2.5.3a-3b46bef-201509041633-build - slot 2 (recovery mode)|
|ONIE|
+-----+

Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the commands
before booting or 'c' for a command-line.
```

8. Cumulus RMP uses GRUB chainloading to present a second GRUB menu specific to the ONIE partition. No action is necessary in this menu to select the default option *ONIE: Install OS*.

```
GNU GRUB  version 2.02~beta2+e4a1fe391

+-----+
|*ONIE: Install OS|
|ONIE: Rescue|
|ONIE: Uninstall OS|
|ONIE: Update ONIE|
|ONIE: Embed ONIE|
+-----+

Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the commands
before booting or 'c' for a command-line.
```

9. At this point, the USB drive should be automatically recognized and mounted. The image file should be located and automatic installation of Cumulus RMP should begin. Here is some sample output:

```

ONIE: OS Install Mode ...

Version : quanta_common_rangeley-2014.05.05-6919d98-201410171013
Build Date: 2014-10-17T10:13+0800
Info: Mounting kernel filesystems... done.
Info: Mounting LABEL=ONIE-BOOT on /mnt/onie-boot ...
initializing eth0...
scsi 6:0:0:0: Direct-Access SanDisk Cruzer Facet 1.26 PQ: 0
ANSI: 6
sd 6:0:0:0: [sdb] 31266816 512-byte logical blocks: (16.0 GB/14.9
GiB)
sd 6:0:0:0: [sdb] Write Protect is off
sd 6:0:0:0: [sdb] Write cache: disabled, read cache: enabled,
doesn't support DPO or FUA
sd 6:0:0:0: [sdb] Attached SCSI disk

<...snip...>

ONIE: Executing installer: file://dev/sdb1/onie-installer-x86_64
Verifying image checksum ... OK.
Preparing image archive ... OK.
Dumping image info...
Control File Contents
=====
Description: Cumulus Linux
OS-Release: 2.5.7-3b46bef-201509041633-build
Architecture: amd64
Date: Fri, 04 Sep 2015 17:10:30 -0700
Installer-Version: 1.2
Platforms: accton_as5712_54x accton_as6712_32x
mlx_sxl400_i73612 dell_s6000_s1220 dell_s4000_c2338
dell_s3000_c2338 cel_redstone_xp cel_smallstone_xp cel_pebble
quanta_panther quanta_ly8_rangeley quanta_ly6_rangeley
quanta_ly9_rangeley
Homepage: http://www.cumulusnetworks.com/

```

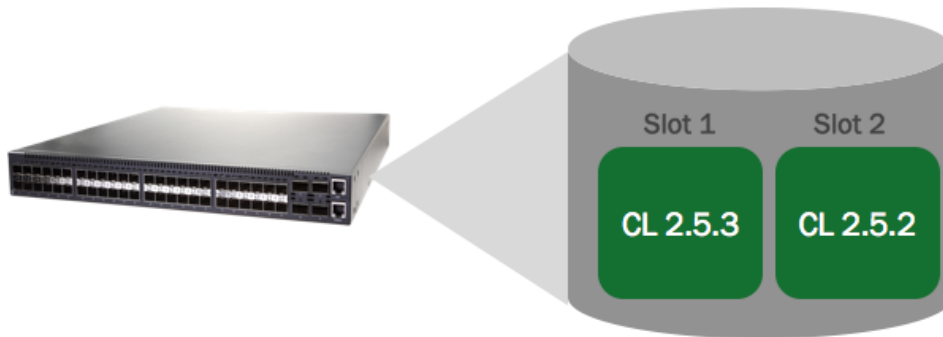
10. After installation completes, the switch automatically reboots into the newly installed instance of Cumulus RMP.

Understanding Image Slots

Cumulus RMP uses the concept of *image slots* to manage two separate Cumulus RMP images. The slots are described as follows:

- **Active image slot:** The currently running image slot.
- **Primary image slot:** The image slot that is selected for the next boot. Often this is the same as the active image slot.

- **Alternate image slot:** The inactive image slot, **not** selected for the next boot.



To identify which slot is active, which slot is the primary, and which slot is alternate use the `cl-img-select` command:

```
cumulus@switch$ sudo cl-img-select
active => slot 1 (primary): 2.5.7-c4e83ad-201506011818-build
        slot 2 (alt      ): 2.5.4-727a0c6-201504132125-build
```

The above switch is currently running 2.5.7 as indicated by the **active**. When the switch is rebooted, it will boot into slot 1, as indicated by **primary**. The **alternate** slot is running Cumulus RMP 2.5.4 and won't be booted into unless the user selects it.

Each slot is a logical volume in the physical partition, which you can manage with [LVM](#). When Cumulus RMP is installed on an x86 switch, the following entities are created on the disk:

- A disk partition using an ext4 file system that contains three logical volumes: two logical volumes named `sysroot1` and `sysroot2`, and the `/mnt/persist` logical volume. The logical volumes represent the Cumulus RMP image slots, so `sysroot1` is slot 1 and `sysroot2` is slot 2. `/mnt/persist` is where you store your [persistent configuration](#) (see [page 46](#)).
- A boot partition, shared by the logical volumes. Each volume mounts this partition as `/boot`.

Managing Slot Sizes

As space in a slot is used, you may need to increase the size of the root filesystem by increasing the size of the corresponding logical volume. This section shows you how to check current utilization and expand the filesystem as needed.

1. Check utilization on the root filesystem with the `df` command. In the following example, filesystem utilization is 16%:

```
cumulus@switch$ df -h /
Filesystem                                Size
Used Avail Use% Mounted on
/dev/disk/by-uuid/64650289-cebf-4849-91ae-a34693fce2f1 4.0G
579M  3.2G  16% /
```

2. To increase available space in the root filesystem, first use the `vgs` command to check the available space in the volume group. In this example, there is 6.34 Gigabytes of free space available in the volume group CUMULUS:

```
cumulus@switch$ sudo vgs
VG          #PV #LV #SN Attr   VSize  VFree
CUMULUS      1   3   0 wz--n- 14.36g 6.34g
```

3. Once you confirm the available space, determine the number of the currently active slot using `cl-img-select`.

```
cumulus@switch$ sudo cl-img-select | grep active
active => slot 1 (primary): 2.5.7-199c587-201501081931-build
```

`cl-img-select` indicates slot number 1 is active.

4. Resize the slot with the `lvresize` command. The following example increases slot size by 20 percent of total available space. Replace the "#" character in the example with the active slot number from the last step.

```
cumulus@switch$ sudo lvresize -l +20%FREE CUMULUS/SYSROOT#
Extending logical volume SYSROOT# to 5.27 GiB
Logical volume SYSROOT# successfully resized
```



The use of + is very important with the `lvresize` command. Issuing `lvresize` without the + results in the logical volume size being set directly to the specified size, rather than extended.

5. Once the slot has been extended, use the `resize2fs` command to expand the filesystem to fit the new space in the slot. Again, replace the "#" character in the example with the active slot number.

```
cumulus@switch$ sudo resize2fs /dev/CUMULUS/SYSROOT#
resize2fs 1.42.5 (29-Jul-2012)
Filesystem at /dev/CUMULUS/SYSROOT# is mounted on /; on-line
resizing required
old_desc_blocks = 1, new_desc_blocks = 1
Performing an on-line resize of /dev/CUMULUS/SYSROOT# to 1381376
(4k) blocks.
The filesystem on /dev/CUMULUS/SYSROOT# is now 1381376 blocks long.
```

Accessing the Alternate Image Slot

It may be useful to access the content of the alternate slot to retrieve configuration or logs.



`cl-img-install` fails while the alternate slot is mounted. It is important to unmount the alternate slot as shown in step 4 below when done.

1. Determine which slot is the alternate with `cl-img-select`.

```
cumulus@switch$ sudo cl-img-select | grep alt
slot 2 (alt    ): 2.5.0-199c587-201501081931-build
```

This output indicates slot 2 is the alternate slot.

2. Create a mount point for the alternate slot:

```
cumulus@switch$ sudo mkdir /mnt/alt
```

3. Mount the alternate slot to the mount point:

```
cumulus@switch$ sudo mount /dev/mapper/CUMULUS-SYSROOT# /mnt/alt
```

Where `#` is the number of the alternate slot.

The alternate slot is now accessible under `/mnt/alt`.

4. Unmount the mount point `/mnt/alt` when done.

```
cumulus@switch$ cd /
cumulus@switch$ sudo umount /mnt/alt/
```

Reprovisioning the System (Restart Installer)

You can reprovision the system, wiping out the contents of both image slots and `/mnt/persist`.

To initiate the provisioning and installation process, use `cl-img-select -i`:

```
cumulus@switch:~$ sudo cl-img-select -i
WARNING:
WARNING: Operating System install requested.
WARNING: This will wipe out all system data.
WARNING:
```



```
Are you sure (y/N)? y
Enabling install at next reboot...done.
Reboot required to take effect.
```



A reboot is required for the reinstall to begin.



If you change your mind, you can cancel a pending reinstall operation by using `cl-img-select -c`:

```
cumulus@switch:~$ sudo cl-img-select -c
Cancelling pending install at next reboot...done.
```

Uninstalling All Images and Removing the Configuration

To remove all installed images and configurations, returning the switch to its factory defaults, use `cl-img-select -k`:

```
cumulus@switch:~$ sudo cl-img-select -k
WARNING:
WARNING: Operating System uninstall requested.
WARNING: This will wipe out all system data.
WARNING:
Are you sure (y/N)? y
Enabling uninstall at next reboot...done.
Reboot required to take effect.
```



A reboot is required for the uninstall to begin.



If you change your mind you can cancel a pending uninstall operation by using `cl-img-select -c`:

```
cumulus@switch:~$ sudo cl-img-select -c
Cancelling pending uninstall at next reboot...done.
```

Booting into Rescue Mode

If your system becomes broken in some way, you may be able to correct things by booting into ONIE rescue mode. In rescue mode, the file systems are unmounted and you can use various Cumulus RMP utilities to try and fix the problem.

To reboot the system into the ONIE rescue mode, use `cl-img-select -r`:

```
cumulus@switch:~$ sudo cl-img-select -r
WARNING:
WARNING: Rescue boot requested.
WARNING:
Are you sure (y/N)? y
Enabling rescue at next reboot...done.
Reboot required to take effect.
```



A reboot is required to boot into rescue mode.



If you change your mind you can cancel a pending rescue boot operation by using `cl-img-select -c`:

```
cumulus@switch:~$ sudo cl-img-select -c
Cancelling pending rescue at next reboot...done.
```

Inspecting Image File Contents

From a running system you can display the contents of a Cumulus RMP image file using `cl-img-pkg -d`:

```
cumulus@switch:~$ sudo cl-img-pkg -d /var/lib/cumulus/installer/onie-
installer
Verifying image checksum ... OK.
Preparing image archive ... OK.
Control File Contents
=====
Description: Cumulus RMP
OS-Release: 2.1.0-0556262-201406101128-NB
Architecture: amd64
```

```
Date: Tue, 10 Jun 2014 11:44:28 -0700
Installer-Version: 1.2
Platforms: im_n29xx_t40n mlx_sx1400_i73612 dell_s6000_s1220
Homepage: http://www.cumulusnetworks.com/

Data Archive Contents
=====
    128 2014-06-10 18:44:26 file.list
    44 2014-06-10 18:44:27 file.list.shal
104276331 2014-06-10 18:44:27 sysroot-internal.tar.gz
    44 2014-06-10 18:44:27 sysroot-internal.tar.gz.shal
5391348 2014-06-10 18:44:26 vmlinuz-initrd.tar.xz
    44 2014-06-10 18:44:27 vmlinuz-initrd.tar.xz.shal
cumulus@switch:~$
```

You can also extract the image files to the current directory with the `-e` option:

```
cumulus@switch:~$ sudo cl-img-pkg -e /var/lib/cumulus/installer/onie-
installer
Verifying image checksum ... OK.
Preparing image archive ... OK.
file.list
file.list.shal
sysroot-internal.tar.gz
sysroot-internal.tar.gz.shal
vmlinuz-initrd.tar.xz
vmlinuz-initrd.tar.xz.shal
Success: Image files extracted OK.

cumulus@switch:~$ sudo ls -l
total 107120
-rw-r--r-- 1 1063 3000      128 Jun 10 18:44 file.list
-rw-r--r-- 1 1063 3000      44 Jun 10 18:44 file.list.shal
-rw-r--r-- 1 1063 3000 104276331 Jun 10 18:44 sysroot-internal.tar.gz
-rw-r--r-- 1 1063 3000      44 Jun 10 18:44 sysroot-internal.tar.gz.shal
-rw-r--r-- 1 1063 3000 5391348 Jun 10 18:44 vmlinuz-initrd.tar.xz
-rw-r--r-- 1 1063 3000      44 Jun 10 18:44 vmlinuz-initrd.tar.xz.shal
```

Useful Links

- [Open Network Install Environment \(ONIE\) Home Page](#)

Adding and Updating Packages

You use the Advanced Packaging Tool (APT) to manage additional applications (in the form of packages) and to install the latest updates.

Contents

(Click to expand)

- [Contents \(see page 60\)](#)
- [Commands \(see page 60\)](#)
- [Updating the Package Cache \(see page 60\)](#)
- [Listing Available Packages \(see page 61\)](#)
- [Adding a Package \(see page 62\)](#)
- [Listing Installed Packages \(see page 63\)](#)
- [Upgrading to Newer Versions of Installed Packages \(see page 64\)](#)
 - [Upgrading a Single Package \(see page 64\)](#)
 - [Upgrading All Packages \(see page 64\)](#)
- [Adding Packages from Another Repository \(see page 64\)](#)
- [Configuration Files \(see page 65\)](#)
- [Useful Links \(see page 65\)](#)

Commands

- `apt-get`
- `apt-cache`
- `dpkg`

Updating the Package Cache

To work properly, APT relies on a local cache of the available packages. You must populate the cache initially, and then periodically update it with `apt-get update`:

```
cumulus@switch:~$ sudo apt-get update
Ign https://repo.cumulusnetworks.com CumulusRMP-2.5.3 Release.gpg
Get:1 https://repo.cumulusnetworks.com CumulusRMP-2.5.3 Release [9027 B]
Get:2 https://repo.cumulusnetworks.com CumulusRMP-2.5.3/main powerpc
Packages [105 kB]
Get:3 https://repo.cumulusnetworks.com CumulusRMP-2.5.3/extras powerpc
Packages [20 B]
Get:4 https://repo.cumulusnetworks.com CumulusRMP-2.5.3/updates powerpc
Packages [20 B]
Get:5 https://repo.cumulusnetworks.com CumulusRMP-2.5.3/security-updates
```

```
powerpc Packages [20 B]
Ign https://repo.cumulusnetworks.com CumulusRMP-2.5.3/extras Translation-en
Ign https://repo.cumulusnetworks.com CumulusRMP-2.5.3/main Translation-en
Ign https://repo.cumulusnetworks.com CumulusRMP-2.5.3/security-updates
Translation-en
Ign https://repo.cumulusnetworks.com CumulusRMP-2.5.3/updates Translation-en
Fetched 115 kB in 2s (56.3 kB/s)
Reading package lists... Done
```

Listing Available Packages

Once the cache is populated, use `apt-cache` to search the cache to find the packages you are interested in or to get information about an available package. Here are examples of the `search` and `show` sub-commands:

```
cumulus@switch:~$ apt-cache search tcp
fakeroot - tool for simulating superuser privileges
libwrap0 - Wietse Venema's TCP wrappers library
libwrap0-dev - Wietse Venema's TCP wrappers library, development files
netbase - Basic TCP/IP networking system
nmap - The Network Mapper
openbsd-inetd - OpenBSD Internet Superserver
openssh-client - secure shell (SSH) client, for secure access to remote
machines
openssh-server - secure shell (SSH) server, for secure access from remote
machines
rsyslog - reliable system and kernel logging daemon
socat - multipurpose relay for bidirectional data transfer
tcpd - Wietse Venema's TCP wrapper utilities
tcpdump - command-line network traffic analyzer
tcpreplay - Tool to replay saved tcpdump files at arbitrary speeds
tcpstat - network interface statistics reporting tool
tcptrace - Tool for analyzing tcpdump output
tcpextract - extracts files from network traffic based on file signatures
quagga - BGP/OSPF/RIP routing daemon
jdoo - utility for monitoring and managing daemons or similar programs

cumulus@switch:~$ apt-cache show tcpreplay
Package: tcpreplay
Version: 3.4.3-2+wheezy1
Architecture: powerpc
Maintainer: Noël Köthe <noel@debian.org>
Installed-Size: 984
Depends: libc6 (>= 2.7), libpcap0.8 (>= 0.9.8)
```

```
Homepage: http://tcpreplay.synfin.net/
Priority: optional
Section: net
Filename: pool/main/t/tcpreplay/tcpreplay_3.4.3-2+wheezy1_powerpc.deb
Size: 435904
SHA256: 03dc29057cb608d2ddf08207aedef18d47988ed6c23db0af69d30746768a639ae
SHA1: 8eelb9b02dacd0c48a474844f4466eb54c7e1568
MD5sum: cf20bec7282ef77a091e79372a29fe1e
Description: Tool to replay saved tcpdump files at arbitrary speeds
Tcpreplay is aimed at testing the performance of a NIDS by
replaying real background network traffic in which to hide
attacks. Tcpreplay allows you to control the speed at which the
traffic is replayed, and can replay arbitrary tcpdump traces. Unlike
programmatically-generated artificial traffic which doesn't
exercise the application/protocol inspection that a NIDS performs,
and doesn't reproduce the real-world anomalies that appear on
production networks (asymmetric routes, traffic bursts/lulls,
fragmentation, retransmissions, etc.), tcpreplay allows for exact
replication of real traffic seen on real networks.
cumulus@switch:~$
```



The search commands look for the search terms not only in the package name but in other parts of the package information. Consequently, it will match on more packages than you would expect.

Adding a Package

In order to add a new package, first ensure the package is not already installed in the system:

```
cumulus@switch:~$ dpkg -l | grep {name of package}
```

If the package is installed already, ensure it's the version you need. If it's an older version, then update the package from the Cumulus RMP repository:

```
cumulus@switch:~$ sudo apt-get update
```

If the package is not already on the system, add it by running `apt-get install`. This retrieves the package from the Cumulus RMP repository and installs it on your system together with any other packages that this package might depend on.

For example, the following adds the package `tcpreplay` to the system:

```
cumulus@switch:~$ sudo apt-get install tcpreplay
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
tcpreplay
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 436 kB of archives.
After this operation, 1008 kB of additional disk space will be used.
Get:1 https://repo.cumulusnetworks.com/ CumulusRMP-2.5.3/main tcpreplay
powerpc 3.4.3-2+wheezy1 [436 kB]
Fetched 436 kB in 0s (1501 kB/s)
Selecting previously unselected package tcpreplay.
(Reading database ... 15930 files and directories currently installed.)
Unpacking tcpreplay (from .../tcpreplay_3.4.3-2+wheezy1_powerpc.deb) ...
Processing triggers for man-db ...
Setting up tcpreplay (3.4.3-2+wheezy1) ...
cumulus@switch:~$
```

Listing Installed Packages

The APT cache contains information about all the packages available on the repository. To see which packages are actually installed on your system, use `dpkg`. The following example lists all the packages on the system that have "tcp" in their package names:

```
cumulus@switch:~$ dpkg -l \*tcp\*
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-
pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name Version Architecture Description
+++-=====
=====
ii tcpcd 7.6.q-24 powerpc Wietse Venema's TCP wrapper
utili
ii tcpdump 4.3.0-1 powerpc command-line network traffic
anal
ii tcpreplay 3.4.3-2+whee powerpc Tool to replay saved tcpdump
file
cumulus@switch:~$
```

Upgrading to Newer Versions of Installed Packages

Upgrading a Single Package

A single package can be upgraded by simply installing that package again with `apt-get install`. You should perform an update first so that the APT cache is populated with the latest information about the packages.

To see if a package needs to be upgraded, use `apt-cache show <pkgname>` to show the latest version number of the package. Use `dpkg -l <pkgname>` to show the version number of the installed package.

Upgrading All Packages

You can update all packages on the system with `apt-get update`. This upgrades all installed versions with their latest versions but will not install any new packages.

Adding Packages from Another Repository

As shipped, Cumulus RMP searches the Cumulus RMP repository for available packages. You can add additional repositories to search by adding them to the list of sources that `apt-get` consults. See `man sources.list` for more information.



For several packages, Cumulus Networks has added features or made bug fixes and these packages must not be replaced with versions from other repositories. Cumulus RMP has been configured to ensure that the packages from the Cumulus RMP repository are always preferred over packages from other repositories.

If you want to install packages that are not in the Cumulus RMP repository, the procedure is the same as above with one additional step.



Packages not part of the Cumulus RMP repository have generally not been tested, and may not be supported by Cumulus RMP support.

Installing packages outside of the Cumulus RMP repository requires the use of `apt-get`, but, depending on the package, `easy-install` and other commands can also be used.

To install a new package, please complete the following steps:

1. First, ensure package is not already installed in the system. Use the `dpkg` command:

```
cumulus@switch:~$ dpkg -l | grep {name of package}
```

2. If the package is installed already, ensure it's the version you need. If it's an older version, then update the package from the Cumulus RMP repository:


```
cumulus@switch:~$ sudo apt-get update
cumulus@switch:~$ sudo apt-get install {name of package}
```

3. If the package is not on the system, then most likely the package source location is also **not** in the `/etc/apt/sources.list` file. If the source for the new package is **not** in `sources.list`, please edit and add the appropriate source to the file. For example, add the following if you wanted a package from the Debian repository that is **not** in the Cumulus RMP repository:

```
deb http://http.us.debian.org/debian wheezy main
deb http://security.debian.org/ wheezy/updates main
```

Otherwise, the repository may be listed in `/etc/apt/sources.list` but is commented out, as can be the case with the testing repository:

```
#deb http://repo.cumulusnetworks.com CumulusRMP-VERSION
testing
```

To uncomment the repository, remove the `#` at the start of the line, then save the file:

```
deb http://repo.cumulusnetworks.com CumulusRMP-VERSION testing
```

4. Run `apt-get update` then install the package:

```
cumulus@switch:~$ sudo apt-get update
cumulus@switch:~$ sudo apt-get install {name of package}
```

Configuration Files

- `/etc/apt/apt.conf`
- `/etc/apt/preferences`
- `/etc/apt/sources.list`

Useful Links

- [Debian GNU/Linux FAQ, Ch 8 Package management tools](#)
- [man pages for apt-get, dpkg, sources.list, apt_preferences](#)

Zero Touch Provisioning

Zero touch provisioning allows devices to be quickly deployed in large-scale environments. Data center engineers only need to rack and stack the switch, connect it to the management network, then install Cumulus RMP via ONIE; the initial configuration gets invoked via ZTP. Alternatively, you can insert a USB stick with the configuration so the provisioning process can start automatically.

The provisioning framework allows for a one-time, user-provided script to be executed. This script can be used to add the switch to a configuration management (CM) platform such as [puppet](#), [Chef](#), [CFEngine](#), or even a custom, home-grown tool.

In addition, you can use the `autoprovision` command in Cumulus RMP to invoke your provisioning script.

Provisioning initially takes place over the management network and is initiated via a DHCP hook. A DHCP option is used to specify a configuration script. This script is then requested from the Web server and executed locally on the switch.

Contents

(Click to expand)

- [Contents \(see page 66\)](#)
- [Commands \(see page 66\)](#)
- [Zero Touch Provisioning Process \(see page 66\)](#)
- [Specifying DHCP Option 239 \(see page 67\)](#)
- [HTTP Headers \(see page 67\)](#)
- [Script Requirements \(see page 68\)](#)
- [Example Scripts \(see page 68\)](#)
- [Using the autoprovision Command \(see page 69\)](#)
- [Notes \(see page 70\)](#)
- [Configuration Files \(see page 70\)](#)

Commands

- `autoprovision`

Zero Touch Provisioning Process

The zero touch provisioning process involves these steps:

1. The first time you boot Cumulus RMP, eth0 is configured for DHCP and makes a DHCP request.
2. The DHCP server offers a lease to the switch.
3. If option 239 is present in the response, the zero touch provisioning process itself will start.
4. The zero touch provisioning process requests the contents of the script from the URL, sending additional [HTTP headers \(see page 67\)](#) containing details about the switch.
5. The script's contents are parsed to ensure it contains the `CUMULUS-AUTOPROVISIONING` flag.
6. If the `CUMULUS-AUTOPROVISIONING` flag is present, then the script executes locally on the switch.

7. The return code of the script gets examined. If it is 0, then the provisioning state is marked as complete.

Specifying DHCP Option 239

During the DHCP process over `eth0`, Cumulus RMP will request DHCP option 239. This option is used to specify the custom provisioning script.

For example, the `dhcpd.conf` file for an ISC DHCP server could look like:

```
option cumulus-provision-url code 239 = text;

subnet 192.168.0.0 netmask 255.255.255.0 {
    range 192.168.0.100 192.168.0.200;
    option cumulus-provision-url "http://192.168.0.2/demo.sh";
}
```

Additionally, the hostname of the switch can be specified via the `host-name` option:

```
subnet 192.168.0.0 netmask 255.255.255.0 {
    range 192.168.0.100 192.168.0.200;
    option cumulus-provision-url "http://192.168.0.2/demo.sh";
    host dcl-tor-sw1 { hardware ethernet 44:38:39:00:1a:6b; fixed-address
192.168.0.101; option host-name "dcl-tor-sw1"; }
}
```

HTTP Headers

The following HTTP headers are sent in the request to the Web server to retrieve the provisioning script:

Header	Value	Example
-----	-----	-----
User-Agent		CumulusRMP-AutoProvision
/0.4		
CUMULUS-ARCH	CPU architecture	powerpc
CUMULUS-BUILD		2.5.3-5c6829a-
201309251712-final		
CUMULUS-LICENSE-INSTALLED	Either 0 or 1	1
CUMULUS-MANUFACTURER		dni
CUMULUS-PRODUCTNAME		et-7448bf
CUMULUS-SERIAL		XYZ123004
CUMULUS-VERSION		2.5.3

CUMULUS-PROV-COUNT	0
CUMULUS-PROV-MAX	32

Script Requirements

The script contents must contain the CUMULUS-AUTOPROVISIONING flag. This can be in a comment or remark and does not need to be echoed or written to `stdout`.

The script can be written in any language currently supported by Cumulus RMP, such as:

- Perl
- Python
- Ruby
- Shell

The script must return an exit code of 0 upon success, as this triggers the provisioning process to be marked as complete.

Example Scripts

Here is a simple script to install `puppet`:

```
#!/bin/bash
function error() {
    echo -e "\e[0;33mERROR: The Zero Touch Provisioning script failed while
running the command $BASH_COMMAND at line $BASH_LINENO.\e[0m" >&2
    exit 1
}
trap error ERR
apt-get update -y
apt-get upgrade -y
apt-get install puppet -y
sed -i /etc/default/puppet -e 's/START=no/START=yes/'
sed -i /etc/puppet/puppet.conf -e 's/\[main\]/\[main\]\npluginsync=true/'
service puppet restart
# CUMULUS-AUTOPROVISIONING
exit 0
```

This script illustrates how to specify an internal APT mirror and `puppet` master:

```
#!/bin/bash
function error() {
    echo -e "\e[0;33mERROR: The Zero Touch Provisioning script failed while
running the command $BASH_COMMAND at line $BASH_LINENO.\e[0m" >&2
    exit 1
}
```

```

}
trap error ERR
sed -i /etc/apt/sources.list -e 's/repo.cumulusnetworks.com/labrepo.
mycompany.com/'
apt-get update -y
apt-get upgrade -y
apt-get install puppet -y
sed -i /etc/default/puppet -e 's/START=no/START=yes/'
sed -i /etc/puppet/puppet.conf -e 's/\[main\]/\[main\]\npluginsync=true/'
sed -i /etc/puppet/puppet.conf -e 's/\[main\]/\[main\]\nserver=labpuppet.
mycompany.com/'
service puppet restart
# CUMULUS-AUTOPROVISIONING
exit 0

```

Now puppet can take over management of the switch, configuration authentication, changing the default root password, and setting up interfaces and routing protocols.

Using the autoprovision Command

You can directly invoke an your provisioning script by running the `autoprovision` command. You can use this command to enable and disable zero touch provisioning on the switch. Be sure to specify the full path to the command, as in the examples below.

To enable zero touch provisioning, use the `-e` option:

```
cumulus@switch:~$ sudo /usr/lib/cumulus/autoprovision -e
```

To run the provisioning script, use the `-u` option and include the URL to the script:

```
cumulus@switch:~$ sudo /usr/lib/cumulus/autoprovision -u http://192.168.0.1
/ztp.sh
```

To disable zero touch provisioning, use the `-x` option:

```
cumulus@switch:~$ sudo /usr/lib/cumulus/autoprovision -x
```

To enable startup discovery mode, without relying on DHCP when you boot the switch, use the `-s` option:

```
cumulus@switch:~$ sudo /usr/lib/cumulus/autoprovision -s
```

Notes

- During the development of a provisioning script, the switch may need to be reset.
- You can use the Cumulus RMP `c1-img-select -i` command to cause the switch to reprovision itself and install a network operating system again using ONIE.
- You can trigger the zero touch provisioning process when eth0 is set to use DHCP and one of the following events occur:
 - Booting the switch
 - Plugging a cable into or unplugging it from the eth0 port
 - Disconnecting then reconnecting the switch's power cord

Configuration Files

- `/var/lib/cumulus/autoprovision.conf`

Configuring and Managing Network Interfaces

`ifupdown` is the network interface manager for Cumulus RMP. Cumulus RMP uses an updated version of this tool, `ifupdown2`.

For more information on network interfaces, see [Configuring Switch Port Attributes](#) (see page 84).



By default, `ifupdown` is quiet; use the verbose option `-v` when you want to know what is going on when bringing an interface down or up.

Contents

(Click to expand)

- [Contents](#) (see page 71)
- [Commands](#) (see page 71)
- [Man Pages](#) (see page 72)
- [Configuration Files](#) (see page 72)
- [Basic Commands](#) (see page 72)
- [Bringing All auto Interfaces Up or Down](#) (see page 73)
- [ifupdown Behavior with Child Interfaces](#) (see page 74)
- [ifupdown2 Interface Dependencies](#) (see page 75)
 - [ifup Handling of Upper \(Parent\) Interfaces](#) (see page 78)
- [Configuring IP Addresses](#) (see page 79)
 - [Purging Existing IP Addresses on an Interface](#) (see page 80)
- [Specifying User Commands](#) (see page 80)
- [Sourcing Interface File Snippets](#) (see page 80)
- [Using Globs for Port Lists](#) (see page 81)
- [Using Templates](#) (see page 81)
- [Adding Descriptions to Interfaces](#) (see page 82)
- [Caveats and Errata](#) (see page 83)
- [Useful Links](#) (see page 83)

Commands

- `ifdown`
- `ifquery`

- ifreload
- ifup
- mako-render

Man Pages

- man ifdown(8)
- man ifquery(8)
- man ifreload
- man ifup(8)
- man ifupdown-addons-interfaces(5)
- man interfaces(5)

Configuration Files

- /etc/network/interfaces

Basic Commands

To bring up an interface or apply changes to an existing interface, run:

```
cumulus@switch:~$ sudo ifup <ifname>
```

To bring down a single interface, run:

```
cumulus@switch:~$ sudo ifdown <ifname>
```

Runtime Configuration (Advanced)



A runtime configuration is non-persistent, which means the configuration you create here does not persist after you reboot the switch.

To administratively bring an interface up or down, run:

```
cumulus@switch:~$ sudo ip link set dev swp1 {up|down}
```


If you specified *manual* as the address family, you must bring up that interface manually using `ifconfig`. For example, if you configured a bridge like this:

```
auto bridge01
iface bridge01 inet manual
```

You can only bring it up by running `ifconfig bridge01 up`.



`ifdown` always deletes logical interfaces after bringing them down. Use the `--admin-state` option if you only want to administratively bring the interface up or down.

To see the link and administrative state, use the `ip link show` command:

```
cumulus@switch:~$ ip link show dev swp13: swp1: <BROADCAST,MULTICAST,
UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT qlen 500
link/ether 44:38:39:00:03:c1 brd ff:ff:ff:ff:ff:ff
```

In this example, `swp1` is administratively UP and the physical link is UP (`LOWER_UP` flag). More information on interface administrative state and physical state can be found in [this knowledge base article](#).

Bringing All auto Interfaces Up or Down

You can easily bring up or down all interfaces marked `auto` in `/etc/network/interfaces`. Use the `-a` option. For further details, see individual man pages for `ifup(8)`, `ifdown(8)`, `ifreload(8)`.

To administratively bring up all interfaces marked `auto`, run:

```
cumulus@switch:~$ sudo ifup -a
```

To administratively bring down all interfaces marked `auto`, run:

```
cumulus@switch:~$ sudo ifdown -a
```

To reload all network interfaces marked `auto`, use the `if reload` command, which is equivalent to running `ifdown` then `ifup`, the one difference being that `ifreload` skips any configurations that didn't change):

```
cumulus@switch:~$ sudo ifreload -a
```

ifupdown Behavior with Child Interfaces

By default, `ifupdown` recognizes and uses any interface present on the system — whether a VLAN, bond or physical interface — that is listed as a dependent of an interface. You are not required to list them in the `interfaces` file unless they need a specific configuration, for [MTU](#), [link speed](#), and so forth (see [page 84](#)). And if you need to delete a child interface, you should delete all references to that interface from the `interfaces` file.

For this example, `swp1` and `swp2` below do not need an entry in the `interfaces` file. The following stanzas defined in `/etc/network/interfaces` provide the exact same configuration:

With Child Interfaces Defined	Without Child Interfaces Defined
<pre> auto swp1 iface swp1 auto swp2 iface swp2 auto bridge iface bridge bridge-vlan-aware yes bridge-ports swp1 swp2 bridge-vids 1-100 bridge-pvid 1 bridge-stp on </pre>	<pre> auto bridge iface bridge bridge-vlan-aware yes bridge-ports swp1 swp2 bridge-vids 1-100 bridge-pvid 1 bridge-stp on </pre>

Bridge in Traditional Mode - Example

For this example, `swp1.100` and `swp2.100` below do not need an entry in the `interfaces` file. The following stanzas defined in `/etc/network/interfaces` provide the exact same configuration:

With Child Interfaces Defined	Without Child Interfaces Defined
<pre> auto swp1.100 iface swp1.100 auto swp2.100 iface swp2.100 auto br-100 iface br-100 address 10.0.12.2 /24 address 2001:dad: beef::3/64 bridge-ports swp1.100 swp2.100 bridge-stp on </pre>	<pre> auto br-100 iface br-100 address 10.0.12.2/2 4 address 2001:dad: beef::3/64 bridge-ports swp1.1 00 swp2.100 bridge-stp on </pre>

For more information on the bridge in traditional mode vs the bridge in VLAN-aware mode, please read [this knowledge base article](#).

ifupdown2 Interface Dependencies

`ifupdown2` understands interface dependency relationships. When `ifup` and `ifdown` are run with all interfaces, they always run with all interfaces in dependency order. When run with the interface list on the command line, the default behavior is to not run with dependents. But if there are any built-in dependents, they will be brought up or down.

To run with dependents when you specify the interface list, use the `--with-depends` option. `--with-depends` walks through all dependents in the dependency tree rooted at the interface you specify. Consider the following example configuration:

```
auto bond1
iface bond1
    address 100.0.0.2/16
    bond-slaves swp29 swp30
    bond-mode 802.3ad
    bond-miimon 100
    bond-use-carrier 1
    bond-lacp-rate 1
    bond-min-links 1
    bond-xmit-hash-policy layer3+4

auto bond2
iface bond2
    address 100.0.0.5/16
    bond-slaves swp31 swp32
    bond-mode 802.3ad
    bond-miimon 100
    bond-use-carrier 1
    bond-lacp-rate 1
    bond-min-links 1
    bond-xmit-hash-policy layer3+4

auto br2001
iface br2001
    address 12.0.1.3/24
    bridge-ports bond1.2001 bond2.2001
    bridge-stp on
```

Using `ifup --with-depends br2001` brings up all dependents of `br2001`: `bond1.2001`, `bond2.2001`, `bond1`, `bond2`, `bond1.2001`, `bond2.2001`, `swp29`, `swp30`, `swp31`, `swp32`.

```
cumulus@switch:~$ sudo ifup --with-depends br2001
```

Similarly, specifying `ifdown --with-depends br2001` brings down all dependents of `br2001`: `bond1.2001`, `bond2.2001`, `bond1`, `bond2`, `bond1.2001`, `bond2.2001`, `swp29`, `swp30`, `swp31`, `swp32`.

```
cumulus@switch:~$ sudo ifdown --with-depends br2001
```



As mentioned earlier, `ifdown2` always deletes logical interfaces after bringing them down. Use the `--admin-state` option if you only want to administratively bring the interface up or down. In terms of the above example, `ifdown br2001` deletes `br2001`.

To guide you through which interfaces will be brought down and up, use the `--print-dependency` option to get the list of dependents.

Use `ifquery --print-dependency=list -a` to get the dependency list of all interfaces:

```
cumulus@switch:~$ sudo ifquery --print-dependency=list -a
lo : None
eth0 : None
bond0 : ['swp25', 'swp26']
bond1 : ['swp29', 'swp30']
bond2 : ['swp31', 'swp32']
br0 : ['bond1', 'bond2']
bond1.2000 : ['bond1']
bond2.2000 : ['bond2']
br2000 : ['bond1.2000', 'bond2.2000']
bond1.2001 : ['bond1']
bond2.2001 : ['bond2']
br2001 : ['bond1.2001', 'bond2.2001']
swp40 : None
swp25 : None
swp26 : None
swp29 : None
swp30 : None
swp31 : None
swp32 : None
```

To print the dependency list of a single interface, use:

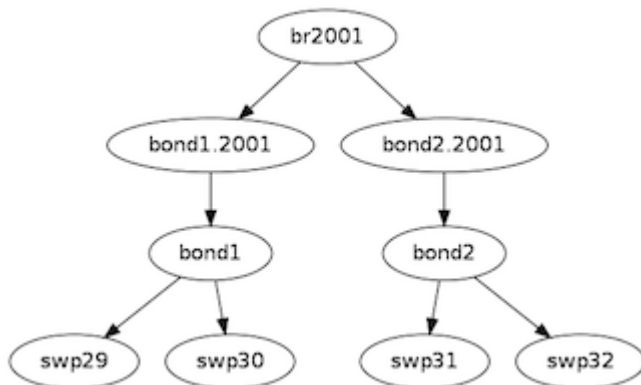
```
cumulus@switch:~$ sudo ifquery --print-dependency=list br2001
br2001 : ['bond1.2001', 'bond2.2001']
bond1.2001 : ['bond1']
bond2.2001 : ['bond2']
bond1 : ['swp29', 'swp30']
bond2 : ['swp31', 'swp32']
swp29 : None
swp30 : None
swp31 : None
```

```
swp32 : None
```

To print the dependency information of an interface in dot format:

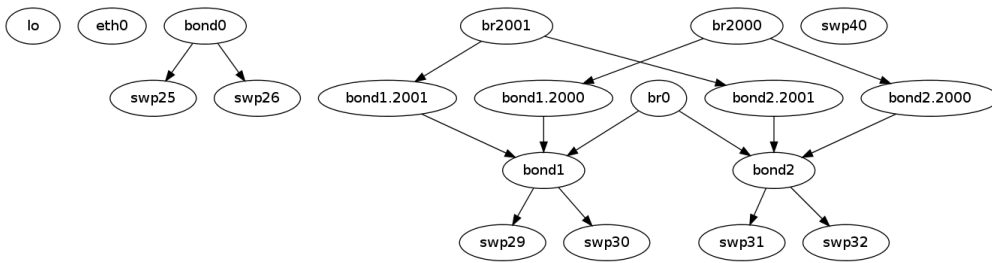
```
cumulus@switch:~$ sudo ifquery --print-dependency=dot br2001
/* Generated by GvGen v.0.9 (http://software.inl.fr/trac/wiki/GvGen)
*/
digraph G {
    compound=true;
    node1 [label="br2001"];
    node2 [label="bond1.2001"];
    node3 [label="bond2.2001"];
    node4 [label="bond1"];
    node5 [label="bond2"];
    node6 [label="swp29"];
    node7 [label="swp30"];
    node8 [label="swp31"];
    node9 [label="swp32"];
    node1->node2;
    node1->node3;
    node2->node4;
    node3->node5;
    node4->node6;
    node4->node7;
    node5->node8;
    node5->node9;
}
```

You can use dot to render the graph on an external system where dot is installed.



To print the dependency information of the entire interfaces file:

```
cumulus@switch:~$ sudo ifquery --print-dependency=dot -a >interfaces_all.dot
```



ifup Handling of Upper (Parent) Interfaces

When you run `ifup` on a logical interface (like a bridge, bond or VLAN interface), if the `ifup` resulted in the creation of the logical interface, by default it implicitly tries to execute on the interface's upper (or parent) interfaces as well. This helps in most cases, especially when a bond is brought down and up, as in the example below. This section describes the behavior of bringing up the upper interfaces.

Consider this example configuration:

```
auto br100
iface br100
    bridge-ports bond1.100 bond2.100

auto bond1
iface bond1
    bond-slaves swp1 swp2
```

If you run `ifdown bond1`, `ifdown` deletes `bond1` and the VLAN interface on `bond1` (`bond1.100`); it also removes `bond1` from the bridge `br100`. Next, when you run `ifup bond1`, it creates `bond1` and the VLAN interface on `bond1` (`bond1.100`); it also executes `ifup br100` to add the bond VLAN interface (`bond1.100`) to the bridge `br100`.

As you can see above, implicitly bringing up the upper interface helps, but there can be cases where an upper interface (like `br100`) is not in the right state, which can result in warnings. The warnings are mostly harmless.

If you want to disable these warnings, you can disable the implicit upper interface handling by setting `skip_upperifaces=1` in `/etc/network/ifupdown2/ifupdown2.conf`.

With `skip_upperifaces=1`, you will have to explicitly execute `ifup` on the upper interfaces. In this case, you will have to run `ifup br100` after an `ifup bond1` to add `bond1` back to bridge `br100`.



Although specifying a subinterface like `swp1.100` and then running `ifup swp1.100` will also result in the automatic creation of the `swp1` interface in the kernel, Cumulus Networks recommends you specify the parent interface `swp1` as well. A parent interface is one where any physical layer configuration can reside, such as `link-speed 1000` or `link-duplex full`.

It's important to note that if you only create `swp1.100` and not `swp1`, then you cannot run `ifup swp1` since you did not specify it.

Configuring IP Addresses

In `/etc/network/interfaces`, list all IP addresses as shown below under the `iface` section (see `man interfaces` for more information):

```
auto swp1
iface swp1
    address 12.0.0.1/30
    address 12.0.0.2/30
```

The address method and address family are not mandatory. They default to `inet/inet6` and `static` by default, but `inet/inet6` **must** be specified if you need to specify `dhcp` or `loopback`:

```
auto lo
iface lo inet loopback
```

You can specify both IPv4 and IPv6 addresses in the same `iface` stanza:

```
auto swp1
iface swp1
    address 192.0.2.1/30
    address 192.0.2.2/30
    address 2001:DB8::1/126
```

Runtime Configuration (Advanced)



A runtime configuration is non-persistent, which means the configuration you create here does not persist after you reboot the switch.

To make non-persistent changes to interfaces at runtime, use `ip addr add`:

```
cumulus@switch:~$ sudo ip addr add 192.0.2.1/30 dev swp1
cumulus@switch:~$ sudo ip addr add 2001:DB8::1/126 dev swp1
```

To remove an addresses from an interface, use `ip addr del`:

```
cumulus@switch:~$ sudo ip addr del 192.0.2.1/30 dev swp1
cumulus@switch:~$ sudo ip addr del 2001:DB8::1/126 dev swp1
```

See `man ip` for more details on the options available to manage and query interfaces.

To show the assigned address on an interface, use `ip addr show`:

```
cumulus@switch:~$ ip addr show dev swp1
3: swp1: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast state UP qlen 500
    link/ether 44:38:39:00:03:c1 brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.1/30 scope global swp1
    inet 192.0.2.2/30 scope global swp1
    inet6 2001:DB8::1/126 scope global tentative
        valid_lft forever preferred_lft forever
```

Purging Existing IP Addresses on an Interface

By default, `ifupdown2` purges existing IP addresses on an interface. If you have other processes that manage IP addresses for an interface, you can disable this feature including the `address-purge` setting in the interface's configuration. For example, add the following to the interface configuration in `/etc/network/interfaces`:

```
auto swp1
iface swp1
    address-purge no
```

Specifying User Commands

You can specify additional user commands in the `interfaces` file. As shown in the example below, the interface stanzas in `/etc/network/interfaces` can have a command that runs at pre-up, up, post-up, pre-down, down, and post-down:

```
auto swp1
iface swp1
    address 12.0.0.1/30
    up /sbin/foo bar
```

Any valid command can be hooked in the sequencing of bringing an interface up or down, although commands should be limited in scope to network-related commands associated with the particular interface.

For example, it wouldn't make sense to install some Debian package on `ifup` of `swp1`, even though that is technically possible. See `man interfaces` for more details.

Sourcing Interface File Snippets

Sourcing interface files helps organize and manage the `interfaces(5)` file. For example:


```
cumulus@switch:~$ cat /etc/network/interfaces
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp

source /etc/network/interfaces.d/bond0
```

The contents of the sourced file used above are:

```
cumulus@switch:~$ cat /etc/network/interfaces.d/bond0
auto bond0
iface bond0
    address 14.0.0.9/30
    address 2001::ded:beef:2::1/64
    bond-slaves swp25 swp26
    bond-mode 802.3ad
    bond-miimon 100
    bond-use-carrier 1
    bond-lacp-rate 1
    bond-min-links 1
    bond-xmit-hash-policy layer3+4
```

Using Globs for Port Lists

Some modules support globs to define port lists (that is, a range of ports). You can use the `glob` keyword to specify bridge ports and bond slaves:

```
auto br0
iface br0
    bridge-ports glob swp1-6.100

auto br1
iface br1
    bridge-ports glob swp7-9.100 swp11.100 glob swp15-18.100
```

Using Templates

`ifupdown2` supports [Mako-style templates](#). The Mako template engine is run over the `interfaces` file before parsing.

Use the template to declare cookie-cutter bridges in the `interfaces` file:

```
%for v in [11,12]:
auto vlan${v}
iface vlan${v}
    address 10.20.${v}.3/24
    bridge-ports glob swp19-20.${v}
    bridge-stp on
%endfor
```

And use it to declare addresses in the `interfaces` file:

```
%for i in [1,12]:
auto swp${i}
iface swp${i}
    address 10.20.${i}.3/24
```



Regarding Mako syntax, use square brackets (`[1,12]`) to specify a list of individual numbers (in this case, 1 and 12). Use `range(1,12)` to specify a range of interfaces.



You can test your template and confirm it evaluates correctly by running `mako-render /etc/network/interfaces`.



For more examples of configuring Mako templates, read this [knowledge base article](#).

Adding Descriptions to Interfaces

You can add descriptions to the interfaces configured in `/etc/network/interfaces` by using the `alias` keyword. For example:

```
auto swp1
iface swp1
    alias swp1 hypervisor_port_1
```

You can query interface descriptions by running `ip link show`. The alias appears on the alias line:

```
cumulus@switch$ ip link show swp1
3: swp1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast
state DOWN mode DEFAULT qlen 500
    link/ether aa:aa:aa:aa:aa:bc brd ff:ff:ff:ff:ff:ff
    alias hypervisor_port_1
```

Interface descriptions also appear in the [SNMP OID \(see page 174\)](#) IF-MIB::ifAlias.

Caveats and Errata

While `ifupdown2` supports the inclusion of multiple `iface` stanzas for the same interface, Cumulus Networks recommends you use a single `iface` stanza for each interface, if possible.

There are cases where you must specify more than one `iface` stanza for the same interface. For example, the configuration for a single interface can come from many places, like a template or a sourced file.

If you do specify multiple `iface` stanzas for the same interface, make sure the stanzas do not specify the same interface attributes. Otherwise, unexpected behavior can result.

For example, `swp1` is configured in two places:

```
cumulus@switch:~$ cat /etc/network/interfaces

source /etc/interfaces.d/speed_settings

auto swp1
iface swp1
    address 10.0.14.2/24
```

As well as `/etc/interfaces.d/speed_settings`

```
cumulus@switch:~$ cat /etc/interfaces.d/speed_settings

auto swp1
iface swp1
    link-speed 1000
    link-duplex full
```

`ifupdown2` correctly parses a configuration like this because the same attributes are not specified in multiple `iface` stanzas.

Useful Links

- <http://wiki.debian.org/NetworkConfiguration>
- <http://www.linuxfoundation.org/collaborate/workgroups/networking/bonding>

- <http://www.linuxfoundation.org/collaborate/workgroups/networking/bridge>
- <http://www.linuxfoundation.org/collaborate/workgroups/networking/vlan>

Configuring Switch Port Attributes

This chapter discusses the various network interfaces on a switch running Cumulus RMP.

Contents

(Click to expand)

- [Contents \(see page 84\)](#)
- [Commands \(see page 84\)](#)
- [Man Pages \(see page 84\)](#)
- [Configuration Files \(see page 84\)](#)
- [Interface Types \(see page 85\)](#)
- [Settings \(see page 85\)](#)
 - [Port Speed and Duplexing \(see page 85\)](#)
 - [Auto-negotiation \(see page 86\)](#)
 - [MTU \(see page 87\)](#)
- [Verification and Troubleshooting Commands \(see page 88\)](#)
 - [Statistics \(see page 88\)](#)
 - [Querying SFP Port Information \(see page 89\)](#)
- [Useful Links \(see page 90\)](#)

Commands

- [ethtool](#)
- [ip](#)

Man Pages

- [man ethtool](#)
- [man interfaces](#)
- [man ip](#)
- [man ip addr](#)
- [man ip link](#)

Configuration Files

- [/etc/network/interfaces](#)

Interface Types

Cumulus RMP exposes network interfaces for several types of physical and logical devices:

- lo, network loopback device
- ethN, switch management port(s), for out of band management only
- swpN, switch front panel ports
- (optional) brN, bridges (IEEE 802.1Q VLANs)
- (optional) bondN, bonds (IEEE 802.3ad link aggregation trunks, or port channels)

Settings

You can set the MTU, speed, duplex and auto-negotiation settings under a physical or logical interface stanza:

```
auto swp1
iface swp1
    address 10.1.1.1/24
    mtu 9000
    link-speed 10000
    link-duplex full
    link-autoneg off
```

To load the updated configuration, run the `ifreload -a` command:

```
cumulus@switch:~$ sudo ifreload -a
```

Port Speed and Duplexing

Cumulus RMP supports both half- and **full-duplex** configurations. Supported port speeds include 1G and 10G. Set the speeds in terms of Mbps, where the setting for 1G is 1000 and 10G is 10000.

You can create a persistent configuration for port speeds in `/etc/network/interfaces`. Add the appropriate lines for each switch port stanza. For example:

```
auto swp1
iface swp1
    address 10.1.1.1/24
    link-speed 10000
    link-duplex full
```



If you specify the port speed in `/etc/network/interfaces`, you must also specify the duplex mode setting along with it; otherwise, `ethtool` defaults to half duplex.

You can also configure these settings at run time, using `ethtool`.

Runtime Configuration (Advanced)



A runtime configuration is non-persistent, which means the configuration you create here does not persist after you reboot the switch.

You can use `ethtool` to configure duplexing and the speed for your switch ports. You must specify both port speed and duplexing in the `ethtool` command; auto-negotiation is optional. The following examples use `swp1`.

- To set the port speed to 1G, run:

```
ethtool -s swp1 speed 1000 duplex full
```

- To set the port speed to 10G, run:

```
ethtool -s swp1 speed 10000 duplex full
```

- To enable duplexing, run:

```
ethtool -s swp1 speed 10000 duplex full|half
```

Port Speed Limitations

Ports can be configured to one speed less than their maximum speed.

Switch port Type	Lowest Configurable Speed
1G	100 Mb
10G	1 Gigabit (1000 Mb)

Auto-negotiation

You can enable or disable [auto-negotiation](#) (that is, set it *on* or *off*) on a switch port.

```
auto swp1
iface swp1
    link-autoneg off
```

Runtime Configuration (Advanced)



A runtime configuration is non-persistent, which means the configuration you create here does not persist after you reboot the switch.

You can use `ethtool` to configure auto-negotiation for your switch ports. The following example use `swp1`:

- To enable or disable auto-negotiation, run:

```
ethtool -s swp1 speed 10000 duplex full autoneg on|off
```

MTU

Interface MTU applies to the management port, front panel port, bridge, VLAN subinterfaces and bonds.

```
auto swp1
iface swp1
    mtu 9000
```

Runtime Configuration (Advanced)



A runtime configuration is non-persistent, which means the configuration you create here does not persist after you reboot the switch.

To set `swp1` to Jumbo Frame MTU=9000, use `ip link set`:

```
cumulus@switch:~$ sudo ip link set dev swp1 mtu 9000
cumulus@switch:~$ ip link show dev swp1
3: swp1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc
pfifo_fast state UP mode DEFAULT qlen 500
    link/ether 44:38:39:00:03:c1 brd ff:ff:ff:ff:ff:ff
```



You must take care to ensure there are no MTU mismatches in the conversation path. MTU mismatches will result in dropped or truncated packets, degrading or blocking network performance.

When you are configuring MTU for a bridge, its MTU setting is the lowest MTU setting of any interface that is a member of that bridge (that is, every interface specified in `bridge-ports` in the bridge configuration in the `interfaces` file). Consider this bridge configuration:

```
auto br0
iface br0
    bridge-ports bond1 bond2 bond3 bond4 peer5
    bridge-vlan-aware yes
    bridge-vids 100-110
    bridge-stp on
```

In order for `br0` to have an MTU of 9000, set the MTU for each of the member interfaces (`bond1` to `bond 4`, and `peer5`), to 9000 at minimum.

```
auto peer5
iface peer5
    bond-slaves swp3 swp4
    bond-mode 802.3ad
    bond-miimon 100
    bond-lacp-rate 1
    bond-min-links 1
    bond-xmit_hash_policy layer3+4
    mtu 9000
```

To show MTU, use `ip link show`:

```
cumulus@switch:~$ ip link show dev swp1
3: swp1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP mode DEFAULT qlen 500
    link/ether 44:38:39:00:03:c1 brd ff:ff:ff:ff:ff:ff
```

Verification and Troubleshooting Commands

Statistics

High-level interface statistics are available with the `ip -s link` command:

```
cumulus@switch:~$ ip -s link show dev swp1
3: swp1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP mode DEFAULT qlen 500
    link/ether 44:38:39:00:03:c1 brd ff:ff:ff:ff:ff:ff
```


RX: bytes	packets	errors	dropped	overrun	mcast
21780	242	0	0	0	242
TX: bytes	packets	errors	dropped	carrier	collsns
1145554	11325	0	0	0	0

Low-level interface statistics are available with `ethtool`:

```
cumulus@switch:~$ sudo ethtool -S swp1
NIC statistics:
HwIfInOctets: 21870
HwIfInUcastPkts: 0
HwIfInBcastPkts: 0
HwIfInMcastPkts: 243
HwIfOutOctets: 1148217
HwIfOutUcastPkts: 0
HwIfOutMcastPkts: 11353
HwIfOutBcastPkts: 0
HwIfInDiscards: 0
HwIfInL3Drops: 0
HwIfInBufferDrops: 0
HwIfInAclDrops: 0
HwIfInBlackholeDrops: 0
HwIfInDot3LengthErrors: 0
HwIfInErrors: 0
SoftInErrors: 0
SoftInDrops: 0
SoftInFrameErrors: 0
HwIfOutDiscards: 0
HwIfOutErrors: 0
HwIfOutQDrops: 0
HwIfOutNonQDrops: 0
SoftOutErrors: 0
SoftOutDrops: 0
SoftOutTxFifoFull: 0
HwIfOutQLen: 0
```

Querying SFP Port Information

You can verify SFP settings using `ethtool -m`. The following example shows the output for 1G and 10G modules:

```
cumulus@switch:~# sudo ethtool -m | egrep '(swp|RXPower :|TXPower :|EthernetComplianceCode)'
```

```
swp1: SFP detected
      EthernetComplianceCodes : 1000BASE-LX
      RXPower : -10.4479dBm
      TXPower : 18.0409dBm
```

```
swp3: SFP detected
      10GEthernetComplianceCode : 10G Base-LR
      RXPower : -3.2532dBm
      TXPower : -2.0817dBm
```

Useful Links

- <http://wiki.debian.org/NetworkConfiguration>
- <http://www.linuxfoundation.org/collaborate/workgroups/networking/vlan>
- <http://www.linuxfoundation.org/collaborate/workgroups/networking/bridge>
- <http://www.linuxfoundation.org/collaborate/workgroups/networking/bonding>

Layer 1 and Layer 2 Features

Spanning Tree and Rapid Spanning Tree

Spanning tree protocol (STP) is always recommended in layer 2 topologies, as it prevents bridge loops and broadcast radiation on a bridged network.

`mstpd` is a daemon that implements IEEE802.1D 2004 and IEEE802.1Q 2011. Currently, STP is disabled by default on the bridge in Cumulus RMP.

To enable STP, configure `brctl stp <bridge> on`.



The STP modes Cumulus RMP supports vary depending upon which [bridge driver mode](#) (see [page 128](#)) is in use. For a bridge configured in *traditional* mode, STP, RSTP, PVST and PVRST are supported; with the default set to PVRST. *VLAN-aware* (see [page 148](#)) bridges only operate in RSTP mode.

If a bridge running RSTP (802.1w) receives a common STP (802.1D) BPDU, it will automatically fall back to 802.1D operation.

You can configure `mstpd` to be in common STP mode only, by setting `setforcevers` to `STP`.

Contents

(Click to expand)

- [Contents](#) (see page 91)
- [Commands](#) (see page 91)
- [PVST/PVRST](#) (see page 92)
- [Creating a Bridge and Configuring STP](#) (see page 92)
- [Configuring Spanning Tree Parameters](#) (see page 94)
 - [Understanding Spanning Tree Parameters](#) (see page 94)
- [Bridge Assurance](#) (see page 102)
- [BPDU Guard](#) (see page 103)
 - [Configuring BPDU Guard](#) (see page 103)
 - [Recovering a Port Disabled by BPDU Guard](#) (see page 104)
- [BPDU Filter](#) (see page 105)
- [Configuration Files](#) (see page 106)
- [Man Pages](#) (see page 106)
- [Useful Links](#) (see page 106)
- [Caveats and Errata](#) (see page 106)

Commands

- `brctl`

- mstpctl

mstpctl is a utility to configure STP. mstpd is started by default on bootup. mstpd logs and errors are located in `/var/log/daemon.log`.

PVST/PVRST

Per VLAN Spanning Tree (PVST) creates a spanning tree instance for a bridge. Rapid PVST (PVRST) supports RSTP enhancements for each spanning tree instance. You must create a bridge corresponding to the untagged native/access VLAN, and all the physical switch ports must be part of the same VLAN. When connected to a switch that has a native VLAN configuration, the native VLAN **must** be configured to be VLAN 1 only.

Cumulus RMP supports the RSTP/PVRST/PVST modes of STP natively when the bridge is configured in [traditional mode](#) (see page 128).

Creating a Bridge and Configuring STP

To create a bridge, configure the bridge stanza under `/etc/network/interfaces`. More information on configuring bridges [can be found here](#) (see page 128). To enable STP on the bridge, include the keyword `bridge-stp on`.

```
auto br2
iface br2
    bridge-ports swp1.101 swp4.101 swp5.101
    bridge-stp on
```

To enable the bridge, run `ifreload -a`:

```
cumulus@switch:~$ sudo ifreload -a
```

Runtime Configuration (Advanced)

You use `brctl` to create the bridge, add bridge ports in the bridge and configure STP on the bridge. `mstpctl` is used only when an admin needs to change the default configuration parameters for STP:

```
cumulus@switch:~$ sudo brctl addbr br2

cumulus@switch:~$ sudo brctl addif br2 swp1.101 swp4.101 swp5.101

cumulus@switch:~$ sudo brctl stp br2 on

cumulus@switch:~$ sudo ifconfig br2 up
```

To get the bridge state, use:

```
cumulus@switch:~$ sudo brctl show
bridge name      bridge id          STP enabled      interfaces
br2              8000.001401010100  yes              swp1.101
                  swp4.101
                  swp5.101
```

To get the mstpd bridge state, use:

```
cumulus@switch:~$ sudo mstpctl showbridge br2
br2 CIST info
enabled          yes
bridge id        F.000.00:14:01:01:01:00
designated root   F.000.00:14:01:01:01:00
regional root    F.000.00:14:01:01:01:00
root port        none
path cost        0          internal path cost  0
max age          20          bridge max age      20
forward delay    15          bridge forward delay 15
tx hold count    6          max hops            20
hello time       2          ageing time          200
force protocol version rstp
time since topology change 90843s
topology change count      4
topology change            no
topology change port       swp4.101
last topology change port  swp5.101
```

To get the mstpd bridge port state, use:

```
cumulus@switch:~$ sudo mstpctl showport br2
E swp1.101 8.001 forw F.000.00:14:01:01:01:00 F.000.00:14:01:01:01:00
8.001 Desg
    swp4.101 8.002 forw F.000.00:14:01:01:01:00 F.000.00:14:01:01:01:00
8.002 Desg
    E swp5.101 8.003 forw F.000.00:14:01:01:01:00 F.000.00:14:01:01:01:00
8.003 Desg

cumulus@switch:~$ sudo mstpctl showportdetail br2 swp1.101
br2:swp1.101 CIST info
enabled          yes          role          Designated
```

port id	8.001	state	forwarding
external port cost	2000	admin external cost	0
internal port cost	2000	admin internal cost	0
designated root	F.000.00:14:01:01:01:00	dsgn external cost	0
dsgn regional root	F.000.00:14:01:01:01:00	dsgn internal cost	0
designated bridge	F.000.00:14:01:01:01:00	designated port	8.001
admin edge port	no	auto edge port	yes
oper edge port	yes	topology change ack	no
point-to-point	yes	admin point-to-point	auto
restricted role	no	restricted TCN	no
port hello time	2	disputed	no
bpdu guard port	no	bpdu guard error	no
network port	no	BA inconsistent	no
Num TX BPDU	45772	Num TX TCN	4
Num RX BPDU	0	Num RX TCN	0
Num Transition FWD	2	Num Transition BLK	2

Configuring Spanning Tree Parameters

The persistent configuration for a bridge is set in `/etc/network/interfaces`. The configuration below shows every possible option configured. There is no requirement to configure any of these options:

```

auto br2
iface br2 inet static
    bridge-ports swp1 swp2 swp3
    bridge-stp on
    mstpctl-maxage 20
    mstpctl-fdelay 15
    mstpctl-maxhops 20
    mstpctl-txholdcount 6
    mstpctl-forcevers rstp
    mstpctl-treeprio 32768
    mstpctl-hello 2
    mstpctl-portpathcost swp1=0 swp2=0
    mstpctl-portadmedge swp1=no swp2=no
    mstpctl-portautoedge swp1=yes swp2=yes
    mstpctl-portp2p swp1=no swp2=no
    mstpctl-portrestrrole swp1=no swp2=no
    mstpctl-bpduguard swp1=no swp2=no
    mstpctl-portrestrtcn swp1=no swp2=no
    mstpctl-portnetwork swp1=no
    mstpctl-treeportprio swp3=128
  
```

Understanding Spanning Tree Parameters

The spanning tree parameters are defined in the IEEE 802.1D, 802.1Q specifications and in the table below.

While configuring spanning tree in a persistent configuration, as described above, is the preferred method, you can also use `mstpctl(8)` to configure spanning tree protocol parameters at runtime.




A runtime configuration is non-persistent, which means the configuration you create here does not persist after you reboot the switch.


The `mstp` daemon is an open source project that some network engineers may be unfamiliar with. For example, many incumbent vendors use the keyword `portfast` to describe a port that is automatically set to forwarding when the port is brought up. The `mstp` equivalent is `mstpctl-portadmedge`. For more comparison [please read this knowledge base article](#).

Examples are included below:

Parameter	Description
maxage	<p>Sets the bridge's <i>maximum age</i> to <code><max_age></code> seconds. The default is 20.</p> <p>The maximum age must meet the condition $2 * (\text{Bridge Forward Delay} - 1 \text{ second}) \geq \text{Bridge Max Age}$.</p> <p>To set this parameter persistently, configure it under the bridge stanza:</p> <pre>mstpctl-maxage 24</pre> <p>To set this parameter at runtime, use:</p> <pre>mstpctl setmaxage <bridge> <max_age></pre> <pre>cumulus@switch:~\$ sudo mstpctl setmaxage br2 24</pre>
ageing	<p>Sets the Ethernet (MAC) address <i>ageing time</i> in <code><time></code> seconds for the bridge when the running version is STP, but not RSTP/MSTP. The default is 300.</p> <p>To set this parameter persistently, configure it under the bridge stanza:</p> <pre>mstpctl-ageing 240</pre> <p>To set this parameter at runtime, use:</p> <pre>mstpctl setageing <bridge> <time></pre>

Parameter	Description
	<pre>cumulus@switch:~\$ sudo mstpctl setageing br2 240</pre>
fdelay	<p>Sets the bridge's <i>bridge forward delay</i> to <time> seconds. The default is 15.</p> <p>The bridge forward delay must meet the condition $2 * (\text{Bridge Forward Delay} - 1 \text{ second}) \geq \text{Bridge Max Age}$.</p> <p>To set this parameter persistently, configure it under the bridge stanza:</p> <pre>mstpctl-fdelay 15</pre> <p>To set this parameter at runtime, use:</p> <pre>mstpctl setfdelay <bridge> <time></pre> <pre>cumulus@switch:~\$ sudo mstpctl setfdelay br2 15</pre>
maxhops	<p>Sets the bridge's <i>maximum hops</i> to <max_hops>. The default is 20.</p> <p>To set this parameter persistently, configure it under the bridge stanza:</p> <pre>mstpctl-maxhops 24</pre> <p>To set this parameter at runtime, use:</p> <pre>mstpctl setmaxhops <bridge> <max_hops></pre> <pre>cumulus@switch:~\$ sudo mstpctl setmaxhops br2 24</pre>
txholdcount	<p>Sets the bridge's <i>bridge transmit hold count</i> to <tx_hold_count>. The default is 6.</p> <p>To set this parameter persistently, configure it under the bridge stanza:</p> <pre>mstpctl-txholdcount 6</pre> <p>To set this parameter at runtime, use:</p>

Parameter	Description
	<pre>mstpctl settxholdcount <bridge> <tx_hold_count></pre> <pre>cumulus@switch:~\$ sudo mstpctl settxholdcount br2 5</pre>
forcevers	<p>Sets the bridge's <i>force STP version</i> to either RSTP/STP. MSTP is not supported currently. The default is <i>RSTP</i>.</p> <p>To set this parameter persistently, configure it under the bridge stanza:</p> <pre>mstpctl-forcevers rstp</pre> <p>To set this parameter at runtime, use:</p> <pre>mstpctl setforcevers <bridge> {mstp rstp stp}</pre> <pre>cumulus@switch:~\$ sudo mstpctl setforcevers br2 rstp</pre>
treeprio	<p>Sets the bridge's <i>tree priority</i> to <priority> for an MSTI instance. The priority value is a number between 0 and 65535 and must be a multiple of 4096. The bridge with the lowest priority is elected the <i>root bridge</i>. The default is 32768.</p> <div style="border: 1px solid red; padding: 5px; margin: 10px 0;">  For <code>msti</code>, only 0 is supported currently. </div> <p>To set this parameter persistently, configure it under the bridge stanza:</p> <pre>mstpctl-treeprio 8192</pre> <p>To set this parameter at runtime, use:</p> <pre>mstpctl settreeprio <bridge> <mstid> <priority></pre> <pre>cumulus@switch:~\$ sudo mstpctl settreeprio br2 0 8192</pre>
treeportprio	

Parameter	Description
	<p>Sets the <i>priority</i> of port <port> to <priority> for the MSTI instance. The priority value is a number between 0 and 240 and must be a multiple of 16. The default is 128.</p> <div>  For msti, only 0 is supported currently. </div> <p>To set this parameter persistently, configure it under the bridge stanza:</p> <pre>mstpctl-treeportprio swp4.101 64</pre> <p>To set this parameter at runtime, use:</p> <pre>mstpctl settreeportprio <bridge> <port> <mstid> <priority></pre> <pre>cumulus@switch:~\$ sudo mstpctl settreeportprio br2 swp4.101 0 64</pre>
hello	<p>Sets the bridge's <i>bridge hello time</i> to <time> seconds. The default is 2.</p> <p>To set this parameter persistently, configure it under the bridge stanza:</p> <pre>mstpctl-hello 20</pre> <p>To set this parameter at runtime, use:</p> <pre>mstpctl sethello <bridge> <time></pre> <pre>cumulus@switch:~\$ sudo mstpctl sethello br2 20</pre>
portpathcost	<p>Sets the <i>port cost</i> of the port <port> in bridge <bridge> to <cost>. The default is 0. mstpd supports only long mode; that is, 32 bits for the path cost.</p> <p>To set this parameter persistently, configure it under the bridge stanza:</p> <pre>mstpctl-portpathcost swp1.101=10</pre>

Parameter	Description
	<p>To set this parameter at runtime, use:</p> <pre>mstpctl setportpathcost <bridge> <port> <cost></pre> <pre>cumulus@switch:~\$ sudo mstpctl setportpathcost br2 swp1.101 10</pre>
portadminedge	<p>Enables/disables the <i>initial edge state</i> of the port <port> in bridge <bridge>. The default is <i>no</i>.</p> <p>To set this parameter persistently, configure it under the bridge stanza:</p> <pre>mstpctl-portadminedge swp1.101=yes</pre> <p>To set this parameter at runtime, use:</p> <pre>mstpctl setportadminedge <bridge> <port> {yes no}</pre> <pre>cumulus@switch:~\$ sudo mstpctl setportadminedge br2 swp1.101 yes</pre>
portautoedge	<p>Enables/disables the <i>auto transition</i> to/from the edge state of the port <port> in bridge <bridge>. The default is <i>yes</i>.</p> <p>To set this parameter persistently, configure it under the bridge stanza:</p> <pre>mstpctl-portautoedge swp1.101=no</pre> <p>To set this parameter at runtime, use:</p> <pre>mstpctl setportautoedge <bridge> <port> {yes no}</pre> <pre>cumulus@switch:~\$ sudo mstpctl setportautoedge br2 swp1.101 no</pre>

portp2p	<p>Enables/disables the <i>point-to-point detection mode</i> of the port <port> in bridge <bridge>. The default is <i>auto</i>.</p> <p>To set this parameter persistently, configure it under the bridge stanza:</p> <pre>mstpctl-portp2p swp1.101=no</pre> <p>To set this parameter at runtime, use:</p> <pre>mstpctl setportp2p <bridge> <port> {yes no auto}</pre> <pre>cumulus@switch:~\$ sudo mstpctl setportp2p br2 swp1.101 no</pre>
portrestrrole	<p>Enables/disables the ability of the port <port> in bridge <bridge> to take the <i>root role</i>. The default is <i>no</i>.</p> <p>To set this parameter persistently, configure it under the bridge stanza:</p> <pre>mstpctl-portrestrrole swp1.101=no</pre> <p>To set this parameter at runtime, use:</p> <pre>mstpctl setportrestrrole <bridge> <port> {yes no}</pre> <pre>cumulus@switch:~\$ sudo mstpctl setportrestrrole br2 swp1.101 yes</pre>
portrestrtcn	<p>Enables/disables the ability of the port <port> in bridge <bridge> to propagate <i>received topology change notifications</i>. The default is <i>no</i>.</p> <p>To set this parameter persistently, configure it under the bridge stanza:</p> <pre>mstpctl-portrestrtcn swp1.101=yes</pre> <p>To set this parameter at runtime, use:</p>

Parameter	Description
	<pre>mstpctl setportrestrtcn <bridge> <port> {yes no}</pre> <pre>cumulus@switch:~\$ sudo mstpctl setportrestrtcn br2 swp1.101 yes</pre>
portnetwork	<p>Enables/disables the <i>bridge assurance capability</i> for a network port <port> in bridge <bridge>. The default is <i>no</i>.</p> <p>To set this parameter persistently, configure it under the bridge stanza:</p> <pre>mstpctl-portnetwork swp4.101=yes</pre> <p>To set this parameter at runtime, use:</p> <pre>mstpctl setportnetwork <bridge> <port> {yes no}</pre> <pre>cumulus@switch:~\$ sudo mstpctl setportnetwork br2 swp4.101 yes</pre>
bpduguard	<p>Enables/disables the <i>BPDU guard configuration</i> of the port <port> in bridge <bridge>. The default is <i>no</i>.</p> <p>To set this parameter persistently, configure it under the bridge stanza:</p> <pre>mstpctl-bpduguard swp1=no</pre> <p>To set this parameter at runtime, use:</p> <pre>mstpctl setbpduguard <bridge> <port> {yes no}</pre> <pre>cumulus@switch:~\$ sudo mstpctl setbpduguard br2 swp1.101 yes</pre>
portbpdufilter	

Parameter	Description
	<p>Enables/disables the <i>BPDU filter</i> functionality for a port <port> in bridge <bridge>. The default is <i>no</i>.</p> <p>To set this parameter persistently, configure it under the bridge stanza:</p> <pre>mstpctl-portbpdufilter swp4.101=yes</pre> <p>To set this parameter at runtime, use:</p> <pre>mstpctl setportbpdufilter <bridge> <port> {yes no}</pre> <pre>cumulus@switch:~\$ sudo mstpctl setportbpdufilter br2 swp4.101 yes</pre>

Bridge Assurance

On a point-to-point link where RSTP is running, if you want to detect unidirectional links and put the port in a discarding state (in error), you can enable bridge assurance on the port by enabling a port type network. The port would be in a bridge assurance inconsistent state until a BPDU is received from the peer. You need to configure the port type network on both the ends of the link in order for bridge assurance to operate properly.

The default setting for bridge assurance is off. This means that there is no difference between disabling bridge assurance on an interface and not configuring bridge assurance on an interface.

To enable bridge assurance on an interface, edit `/etc/network/interfaces` and add a line similar to the example below to the bridge configuration:

```
mstpctl-portnetwork swp1=no
```

You can monitor logs for bridge assurance messages by doing the following:

```
cumulus@switch:~$ sudo grep -in assurance /var/log/syslog | grep mstp
1365:Jun 25 18:03:17 mstpd: br1007:swp1.1007 Bridge assurance
inconsistent
```

To load the new configuration from `/etc/network/interfaces`, run `ifreload -a`:

```
cumulus@switch:~$ sudo ifreload -a
```

Runtime Configuration (Advanced)



A runtime configuration is non-persistent, which means the configuration you create here does not persist after you reboot the switch.

To enable bridge assurance at runtime, run `mstpctl`:

```
cumulus@switch:~$ sudo mstpctl setportnetwork br1007 swp1.1007 yes
cumulus@switch:~$ sudo mstpctl showportdetail br1007 swp1.1007 | grep
network
network port          yes                      BA inconsistent      yes
```

BPDU Guard

To protect the spanning tree topology from unauthorized switches affecting the forwarding path, you can configure *BPDU guard* (Bridge Protocol Data Unit). One very common example is when someone hooks up a new switch to an access port off of a leaf switch. If this new switch is configured with a low priority, it could become the new root switch and affect the forwarding path for the entire Layer 2 topology.

Configuring BPDU Guard

You configure BPDU guard under the bridge stanza in `/etc/network/interfaces`:

```
auto br2
iface br2 inet static
    bridge-ports swp1 swp2 swp3 swp4 swp5 swp6
    bridge-stp on
    mstpctl-bpduguard swp1=yes swp2=yes swp3=yes swp4=yes
```

To load the new configuration, run `ifreload -a`:

```
cumulus@switch:~$ sudo ifreload -a
```

Non-Persistent Configuration

You can also configure BPDU guard on an individual port using a runtime configuration.

Runtime Configuration (Advanced)



A runtime configuration is non-persistent, which means the configuration you create here does not persist after you reboot the switch.

```
cumulus@switch:~$ sudo mstpcctl setbpduguard br2 swp1 yes
cumulus@switch:~$ sudo mstpcctl setbpduguard br2 swp2 yes
cumulus@switch:~$ sudo mstpcctl setbpduguard br2 swp3 yes
cumulus@switch:~$ sudo mstpcctl setbpduguard br2 swp4 yes
```

Recovering a Port Disabled by BPDU Guard

If a BPDU is received on the port, STP will bring down the port and log an error in `/var/log/syslog`. The following is a sample error:

```
mstpd: error, MSTP_IN_rx_bpdu: bridge:bond0 Recvd BPDU on BPDU Guard
Port - Port Down
```

To determine whether BPDU guard is configured, or if a BPDU has been received, run `mstpcctl showportdetail <bridge name>`:

```
cumulus@switch:~$ sudo mstpcctl showportdetail br2 swp1 | grep guard
bpdu guard port      yes                bpdu guard error    yes
```

The only way to recover a port that has been placed in the disabled state is to manually un-shut or bring up the port with `sudo ifup [port]`, as shown in the example below:



Bringing up the disabled port does not fix the problem if the configuration on the connected end-station has not been rectified.

```
cumulus@leaf2$ mstpcctl showportdetail bridge bond0
bridge:bond0 CIST info
  enabled          no          role
Disabled
  port id          8.001          state
discarding
  external port cost 305          admin external cost 0
  internal port cost 305          admin internal cost 0
  designated root    8.000.6C:64:1A:00:4F:9C dsgn external cost 0
  dsgn regional root 8.000.6C:64:1A:00:4F:9C dsgn internal cost 0
  designated bridge  8.000.6C:64:1A:00:4F:9C designated port      8.00
1
  admin edge port    no          auto edge port      yes
  oper edge port     no          topology change ack no
  point-to-point     yes         admin point-to-point auto
  restricted role     no          restricted TCN       no
  port hello time    10         disputed            no
  bpdu guard port     yes         bpdu guard error     yes
  network port       no          BA inconsistent      no
  Num TX BPDU        3          Num TX TCN           2
```



```

Num RX BPDU          488
Num Transition FWD    1
bpdufilter port      no
clag ISL              no
clag role             unknown
0:0:0:0
clag remote portID F.FFF
0:0:0:0

Num RX TCN           2
Num Transition BLK    2
clag ISL Oper UP     no
clag dual conn mac    0:0:
clag system mac       0:0:

cumulus@leaf2$ sudo ifup bond0

cumulus@leaf2$ mstpctl showportdetail bridge bond0
bridge:bond0 CIST info
  enabled              yes
  port id              8.001
  forwarding
    external port cost 305
    internal port cost 305
    designated root    8.000.6C:64:1A:00:4F:9C
    dsgn regional root 8.000.6C:64:1A:00:4F:9C
    designated bridge  8.000.6C:64:1A:00:4F:9C
1
  admin edge port      no
  oper edge port       no
  point-to-point       yes
  restricted role      no
  port hello time      2
  bpdu guard port      no
  network port         no
  Num TX BPDU          3
  Num RX BPDU          43
  Num Transition FWD    1
  bpdufilter port      no
  clag ISL              no
  clag role             unknown
0:0:0:0
clag remote portID F.FFF
0:0:0:0

  role                 Root
  state
  admin external cost  0
  admin internal cost  0
  dsgn external cost   0
  dsgn internal cost   0
  designated port      8.00
  auto edge port       yes
  topology change ack  no
  admin point-to-point auto
  restricted TCN        no
  disputed              no
  bpdu guard error     no
  BA inconsistent       no
  Num TX TCN           2
  Num RX TCN           1
  Num Transition BLK    0
  clag ISL Oper UP     no
  clag dual conn mac    0:0:
  clag system mac       0:0:

```

BPDU Filter

You can enable `bpdufilter` on a switch port, which filters BPDUs in both directions. This effectively disables STP on the port.

To enable it, add the following to `/etc/network/interfaces` under the `bridge port iface` section example:

```
auto br100
```

```
iface br100
    bridge-ports swp1.100 swp2.100
    mstpctl-portbpdufilter swp1=yes swp2=yes
```

To load the new configuration from `/etc/network/interfaces`, run `ifreload -a`:

```
cumulus@switch:~$ sudo ifreload -a
```

For more information, see `man(5) ifupdown-addons-interfaces`.

Runtime Configuration (Advanced)



A runtime configuration is non-persistent, which means the configuration you create here does not persist after you reboot the switch.

To enable BPDU filter at runtime, run `mstpctl`:

```
cumulus@switch:~$ sudo mstpctl setportbpdufilter br100 swp1.100=yes swp2.
100=yes
```

Configuration Files

- `/etc/network/interfaces`

Man Pages

- `brctl(8)`
- `bridge-utils-interfaces(5)`
- `ifupdown-addons-interfaces(5)`
- `mstpctl(8)`
- `mstpctl-utils-interfaces(5)`

Useful Links

The source code for `mstpd/mstpctl` was written by [Vitalii Demianets](#) and is hosted at the sourceforge URL below.

- <https://sourceforge.net/projects/mstpd/>
- http://en.wikipedia.org/wiki/Spanning_Tree_Protocol

Caveats and Errata

- MSTP is not supported currently. However, interoperability with MSTP networks can be accomplished using PVRSTP or PVSTP.

Link Layer Discovery Protocol

The `lldpd` daemon implements the IEEE802.1AB (Link Layer Discovery Protocol, or LLDP) standard. LLDP allows you to know which ports are neighbors of a given port. By default, `lldpd` runs as a daemon and is started at system boot. `lldpd` command line arguments are placed in `/etc/default/lldpd`. `lldpd` configuration options are placed in `/etc/lldpd.conf` or under `/etc/lldpd.d/`.

For more details on the command line arguments and config options, please see `man lldpd(8)`.

`lldpd` supports CDP (Cisco Discovery Protocol, v1 and v2). `lldpd` logs by default into `/var/log/daemon.log` with an `lldpd` prefix.

`lldpcli` is the CLI tool to query the `lldpd` daemon for neighbors, statistics and other running configuration information. See `man lldpcli(8)` for details.

Contents

(Click to expand)

- [Contents \(see page 107\)](#)
- [Commands \(see page 107\)](#)
- [Man Pages \(see page 107\)](#)
- [Configuring LLDP \(see page 107\)](#)
- [Example lldpcli Commands \(see page 108\)](#)
- [Enabling the SNMP Subagent in LLDP \(see page 112\)](#)
- [Configuration Files \(see page 112\)](#)
- [Useful Links \(see page 112\)](#)
- [Caveats and Errata \(see page 112\)](#)

Commands

- `lldpd` (daemon)
- `lldpcli` (interactive CLI)

Man Pages

- `man lldpd`
- `man lldpcli`

Configuring LLDP

You configure `lldpd` settings in `/etc/lldpd.conf` or `/etc/lldpd.d/`. Such a configuration persists when you reboot the switch.

Here is an example configuration:

```
cumulus@switch:~$ sudo cat /etc/lldpd.conf
configure lldp tx-interval 40
configure lldp tx-hold 3
configure system interface pattern-blacklist "eth0"
```

lldpd logs to /var/log/daemon.log with the *lldpd* prefix:

```
cumulus@switch:~$ sudo tail -f /var/log/daemon.log | grep lldp
Aug  7 17:26:17 switch lldpd[1712]: unable to get system name
Aug  7 17:26:17 switch lldpd[1712]: unable to get system name
Aug  7 17:26:17 switch lldpcli[1711]: lldpd should resume operations
Aug  7 17:26:32 switch lldpd[1805]: NET-SNMP version 5.4.3 AgentX subagent
connected
```

Example lldpcli Commands

To see all neighbors on all ports/interfaces:

```
cumulus@switch:~$ sudo lldpcli show neighbors
-----
LLDP neighbors:
-----
Interface:      eth0, via: CDPv1, RID: 72, Time: 0 day, 00:33:40
Chassis:
  ChassisID:    local test-server-1
  SysName:      test-server-1
  SysDescr:     Linux running on
Linux 3.2.2+ #1 SMP Mon Jun 10 16:21:22 PDT 2013 ppc
  MgmtIP:       192.0.2.72
  Capability:   Router, on
Port:
  PortID:       ifname eth1
-----
Interface:      swp1, via: CDPv1, RID: 87, Time: 0 day, 00:36:27
nChassis:
  ChassisID:    local T1
  SysName:      T1
  SysDescr:     Linux running on
Cumulus RMP
  MgmtIP:       192.0.2.15
```

```

    Capability:    Router, on
Port:
    PortID:       ifname swp1
    PortDescr:    swp1
-----
... and more (output truncated to fit this doc)

```

To see neighbors on specific ports:

```

cumulus@switch:~$ sudo lldpcli show neighbors ports swp1,swp2
-----
Interface:      swp1, via: CDPv1, RID: 87, Time: 0 day, 00:36:27
Chassis:
    ChassisID:   local T1
    SysName:     T1
    SysDescr:    Linux running on
Cumulus RMP
    MgmtIP:      192.0.2.15
    Capability:   Router, on
Port:
    PortID:      ifname swp1
    PortDescr:   swp1
-----
Interface:      swp2, via: CDPv1, RID: 123, Time: 0 day, 00:36:27
Chassis:
    ChassisID:   local T2
    SysName:     T2
    SysDescr:    Linux running on
Cumulus RMP
    MgmtIP:      192.0.2.15
    Capability:   Router, on
Port:
    PortID:      ifname swp1
    PortDescr:   swp1

```

To see lldpd statistics for all ports:

```

cumulus@switch:~$ sudo lldpcli show statistics
-----
LLDP statistics:
-----
Interface:      eth0

```

```
Transmitted: 9423
Received:    17634
Discarded:   0
Unrecognized: 0
Ageout:      10
Inserted:    20
Deleted:     10
-----
Interface:   swp1
Transmitted: 9423
Received:    6264
Discarded:   0
Unrecognized: 0
Ageout:      0
Inserted:    2
Deleted:     0
-----
Interface:   swp2
Transmitted: 9423
Received:    6264
Discarded:   0
Unrecognized: 0
Ageout:      0
Inserted:    2
Deleted:     0
-----
Interface:   swp3
Transmitted: 9423
Received:    6265
Discarded:   0
Unrecognized: 0
Ageout:      0
Inserted:    2
Deleted:     0
-----
... and more (output truncated to fit this document)
```

To see lldpd statistics summary for all ports:

```
cumulus@switch:~$ sudo lldpcli show statistics summary
-----
LLDP Global statistics:
-----
```

Summary of stats:

```
Transmitted: 648186
Received:    437557
Discarded:   0
Unrecognized: 0
Ageout:      10
Inserted:    38
Deleted:     10
```

To see the lldpd running configuration:

```
cumulus@switch:~$ sudo lldpcli show running-configuration
```

Global configuration:

Configuration:

```
Transmit delay: 1
Transmit hold: 4
Receive mode: no
Pattern for management addresses: (none)
Interface pattern: (none)
Interface pattern for chassis ID: (none)
Override description with: (none)
Override platform with: (none)
Advertise version: yes
Disable LLDP-MED inventory: yes
LLDP-MED fast start mechanism: yes
LLDP-MED fast start interval: 1
```

Runtime Configuration (Advanced)



A runtime configuration does not persist when you reboot the switch — all changes are lost.

To configure active interfaces:

```
lldpcli configure system interface pattern "swp*"
```

To configure inactive interfaces:

```
lldpcli configure system interface pattern-blacklist "eth0"
```



The active interface list always overrides the inactive interface list.

To reset any interface list to none:

```
lldpcli configure system interface pattern-blacklist ""
```

Enabling the SNMP Subagent in LLDP

LLDP does not enable the SNMP subagent by default. You need to edit `/etc/default/lldpd` and enable the `-x` option.

```
cumulus@switch:~$ sudo nano /etc/default/lldpd
# Uncomment to start SNMP subagent and enable CD
P, SONMP and EDP protocol
#DAEMON_ARGS="-x -c -s -e"

# Enable CDP by default
DAEMON_ARGS="-c"
DAEMON_ARGS="-x"
```

Configuration Files

- `/etc/lldpd.conf`
- `/etc/lldpd.d`
- `/etc/default/lldpd`

Useful Links

- <http://vincentbernat.github.io/lldpd/>
- http://en.wikipedia.org/wiki/Link_Layer_Discovery_Protocol

Caveats and Errata

- Annex E (and hence Annex D) of IEEE802.1AB (lldp) is not supported.

Prescriptive Topology Manager - PTM

In data center topologies, right cabling is a time-consuming endeavor and is error prone. Prescriptive Topology Manager (PTM) is a dynamic cabling verification tool to help detect and eliminate such errors. It takes a graphviz-DOT specified network cabling plan (something many operators already generate), stored in a `topology.dot` file, and couples it with runtime information derived from LLDP to verify that the cabling matches the specification. The check is performed on every link transition on each node in the network. It also detects forwarding path failures using Bidirectional Forwarding Detection (BFD).

You can customize the `topology.dot` file to control `ptmd` at both the global/network level and the node /port level.

PTM runs as a daemon, named `ptmd`.

For more information, see `man ptmd(8)`.

Contents

(Click to expand)

- [Contents \(see page 113\)](#)
- [Supported Features \(see page 113\)](#)
- [Configuring PTM \(see page 114\)](#)
 - [Configuration Parameters \(see page 114\)](#)
- [Bidirectional Forwarding Detection \(BFD\) \(see page 118\)](#)
 - [Configuring BFD \(see page 118\)](#)
 - [Echo Function \(see page 118\)](#)
- [Scripts \(see page 119\)](#)
- [Using ptmd Service Commands \(see page 119\)](#)
- [Using ptmctl Commands \(see page 120\)](#)
 - [ptmctl Examples \(see page 120\)](#)
 - [ptmctl Error Outputs \(see page 123\)](#)
- [Configuration Files \(see page 123\)](#)
- [Useful Links \(see page 124\)](#)

Supported Features

- Topology verification using LLDP. `ptmd` creates a client connection to the LLDP daemon, `lldpd`, and retrieves the neighbor relationship between the nodes/ports in the network and compares them against the prescribed topology specified in the `topology.dot` file.
- Only physical interfaces, like `swp1` or `eth0`, are currently supported. Cumulus RMP does not support specifying virtual interfaces like bonds or subinterfaces like `eth0.200` in the topology file.
- Forwarding path failure detection using [Bidirectional Forwarding Detection \(BFD\)](#); however, demand mode is not supported. For more information on how BFD operates in Cumulus RMP, [see below \(see page 118\)](#) and see `man ptmd(8)`.

- Client management: `ptmd` creates an abstract named socket `/var/run/ptmd.socket` on startup. Other applications can connect to this socket to receive notifications and send commands.
- Event notifications: see Scripts below.
- User configuration via a `topology.dot` file; see below (see page 114).

Configuring PTM

`ptmd` verifies the physical network topology against a DOT-specified network graph file, `/etc/ptm.d/topology.dot`. This file must be present or else `ptmd` will not start. You can specify an alternate file using the `-c` option.

At startup, `ptmd` connects to `lldpd`, the LLDP daemon, over a Unix socket and retrieves the neighbor name and port information. It then compares the retrieved port information with the configuration information that it read from the topology file. If there is a match, then it is a PASS, else it is a FAIL.



PTM performs its LLDP neighbor check using the PortID ifname TLV information. Previously, it used the PortID port description TLV information.

PTM also supports **undirected graphs**:

```
graph G {
    node [shape=record];
    graph [hostidtype="hostname", version="1:0", date="04/12/2013"];
    edge [dir=none, len=1, headport=center, tailport=center];
    //R1's connections - R1 is top-tier spine
    "R1":"swp1" -- "R3":"swp3";
    "R1":"swp2" -- "R4":"swp3";
}
```



It's a good idea to always wrap the hostname in double quotes, like `"www.example.com"`. Otherwise, `ptmd` can fail if you specify a fully-qualified domain name as the hostname and do not wrap it in double quotes.

Further, to avoid errors when starting the `ptmd` process, make sure that `/etc/hosts` and `/etc/hostname` both reflect the hostname you are using in the `topology.dot` file.

Configuration Parameters

You can configure `ptmd` parameters in the topology file. The parameters are classified as host-only, global, per-port/node and templates.

Host-only Parameters

Host-only parameters apply to the entire host on which PTM is running. You can include the `hostnametype` host-only parameter, which specifies whether PTM should use only the host name (`hostname`) or the fully-qualified domain name (`fqdn`) while looking for the `self-node` in the graph file. For example, in the graph file below, PTM will ignore the FQDN and only look for `switch04`, since that is the host name of the switch it's running on:

```
graph G {
    hostnametype="hostname"
    BFD="upMinTx=150,requiredMinRx=250"
    "cumulus":swp44 -- "switch04.cumulusnetworks.com":swp20
    "cumulus":swp46 -- "switch04.cumulusnetworks.com":swp22
}
```

However, in this next example, PTM will compare using the FQDN and look for `switch05.cumulusnetworks.com`, which is the FQDN of the switch it's running on:

```
graph G {
    hostnametype="fqdn"
    "cumulus":swp44 -- "switch05.cumulusnetworks.com":swp20
    "cumulus":swp46 -- "switch05.cumulusnetworks.com":swp22
}
```

Global Parameters

Global parameters apply to every port listed in the topology file. There are two global parameters: LLDP and BFD. LLDP is enabled by default; if no keyword is present, default values are used for all ports. However, BFD is disabled if no keyword is present, unless there is a per-port override configured. For example:

```
graph G {
    LLDP=" "
    BFD="upMinTx=150,requiredMinRx=250,afi=both"
    "cumulus":swp44 -- "qct-ly2-04":swp20
    "cumulus":swp46 -- "qct-ly2-04":swp22
}
```

Per-port Parameters

Per-port parameters provide finer-grained control at the port level. These parameters override any global or compiled defaults. For example:

```
graph G {
    LLDP=" "
    BFD="upMinTx=300,requiredMinRx=100"
    "cumulus":swp44 -- "qct-ly2-04":swp20 [BFD="upMinTx=150,
requiredMinRx=250,afi=both"]
    "cumulus":swp46 -- "qct-ly2-04":swp22
}
```

Templates

Templates provide flexibility in choosing different parameter combinations and applying them to a given port. A template instructs `ptmd` to reference a named parameter string instead of a default one. There are two parameter strings `ptmd` supports:

- `bfdtmpl`, which specifies a custom parameter tuple for BFD.
- `lldptmpl`, which specifies a custom parameter tuple for LLDP.

For example:

```
graph G {
    LLDP=" "
    BFD="upMinTx=300,requiredMinRx=100"
    BFD1="upMinTx=200,requiredMinRx=200"
    BFD2="upMinTx=100,requiredMinRx=300"
    LLDP1="match_type=ifname"
    LLDP2="match_type=portdescr"
    "cumulus":swp44 -- "qct-ly2-04":swp20 [BFD="bfdtmpl=BFD1", LLDP="
lldptmpl=LLDP1"]
    "cumulus":swp46 -- "qct-ly2-04":swp22 [BFD="bfdtmpl=BFD2", LLDP="
lldptmpl=LLDP2"]
    "cumulus":swp46 -- "qct-ly2-04":swp22
}
```

In this template, LLDP1 and LLDP2 are templates for LLDP parameters while BFD1 and BFD2 are template for BFD parameters.

Supported BFD and LLDP Parameters

`ptmd` supports the following BFD parameters:

- `upMinTx`: the minimum transmit interval, which defaults to 300ms, specified in milliseconds.
- `requiredMinRx`: the minimum interval between received BFD packets, which defaults to 300ms, specified in milliseconds.
- `detectMult`: the detect multiplier, which defaults to 3, and can be any non-zero value.

- `afi`: the address family to be supported for the edge. The address family must be one of the following:
 - `v4`: BFD sessions will be built for only IPv4 connected peer. This is the default value.
 - `v6`: BFD sessions will be built for only IPv6 connected peer.
 - `both`: BFD sessions will be built for both IPv4 and IPv6 connected peers.

The following is an example of a topology with BFD applied at the port level:

```
graph G {
    "cumulus-1":swp44 -- "cumulus-2":swp20 [BFD="upMinTx=300,
requiredMinRx=100,afi=v6"]
    "cumulus-1":swp46 -- "cumulus-2":swp22 [BFD="detectMult=4"]
}
```

`ptmd` supports the following LLDP parameters:

- `match_type`, which defaults to the interface name (`ifname`), but can accept a port description (`portdescr`) instead if you want `lldpd` to compare the topology against the port description instead of the interface name. You can set this parameter globally or at the per-port level.
- `match_hostname`, which defaults to the host name (`hostname`), but enables PTM to match the topology using the fully-qualified domain name (`fqdn`) supplied by LLDP.

The following is an example of a topology with LLDP applied at the port level:

```
graph G {
    "cumulus-1":swp44 -- "cumulus-2":swp20 [LLDP="match_hostname=fqdn"]
    "cumulus-1":swp46 -- "cumulus-2":swp22 [LLDP="
match_type=portdescr"]
}
```



When you specify `match_hostname=fqdn`, `ptmd` will match the entire FQDN, like `cumulus-2.domain.com` in the example below. If you do not specify anything for `match_hostname`, `ptmd` will match based on hostname only, like `cumulus-3` below, and ignore the rest of the URL:

```
graph G {
    "cumulus-1":swp44 -- "cumulus-2.domain.com":swp20 [LLDP="
match_hostname=fqdn"]
    "cumulus-1":swp46 -- "cumulus-3":swp22 [LLDP="
match_type=portdescr"]
}
```

Bidirectional Forwarding Detection (BFD)

BFD provides low overhead and rapid detection of failures in the paths between two network devices. It provides a unified mechanism for link detection over all media and protocol layers. Use BFD to detect failures for IPv4 and IPv6 single or multihop paths between any two network devices, including unidirectional path failure detection.



BFD requires an IP address for any interface for which it is configured. The neighbor IP address for a single hop BFD session must be in the ARP table before BFD can start sending control packets.



You cannot specify BFD multihop sessions in the `topology.dot` file since you cannot specify the source and destination IP address pairs in that file.

Configuring BFD

You configure BFD by specifying the configuration in the `topology.dot` file. However, the topology file has some limitations:

- The `topology.dot` file supports creating BFD IPv4 and IPv6 single hop sessions only; you cannot specify IPv4 or IPv6 multihop sessions in the topology file.
- The topology file supports BFD sessions for only link-local IPv6 peers; BFD sessions for global IPv6 peers discovered on the link will not be created.

Echo Function

Cumulus RMP supports the *echo function* for IPv4 single hops only, and with the a synchronous operating mode only (Cumulus RMP does not support demand mode).

You use the echo function primarily to test the forwarding path on a remote system. To enable the echo function, set `echoSupport` to 1 in the topology file.

Once the echo packets are looped by the remote system, the BFD control packets can be sent at a much lower rate. You configure this lower rate by setting the `slowMinTx` parameter in the topology file to a non-zero value of milliseconds.

You can use more aggressive detection times for echo packets since the round-trip time is reduced because they are accessing the forwarding path. You configure the detection interval by setting the `echoMinRx` parameter in the topology file to a non-zero value of milliseconds; the minimum setting is 50 milliseconds. Once configured, BFD control packets are sent out at this required minimum echo Rx interval. This indicates to the peer that the local system can loop back the echo packets. Echo packets are transmitted if the peer supports receiving echo packets.

About the Echo Packet

BFD echo packets are encapsulated into UDP packets over destination and source UDP port number 3785. The BFD echo packet format is vendor-specific and has not been defined in the RFC. BFD echo packets that originate from Cumulus RMP are 8 bytes long and have the following format:

0	1	2	3
Version	Length	Reserved	
My Discriminator			

Where:

- **Version** is the version of the BFD echo packet.
- **Length** is the length of the BFD echo packet.
- **My Discriminator** is a non-zero value that uniquely identifies a BFD session on the transmitting side. When the originating node receives the packet after being looped back by the receiving system, this value uniquely identifies the BFD session.

Transmitting and Receiving Echo Packets

BFD echo packets are transmitted for a BFD session only when the peer has advertised a non-zero value for the required minimum echo Rx interval (the `echoMinRx` setting) in the BFD control packet when the BFD session starts. The transmit rate of the echo packets is based on the peer advertised echo receive value in the control packet.

BFD echo packets are looped back to the originating node for a BFD session only if locally the `echoMinRx` and `echoSupport` are configured to a non-zero values.

Using Echo Function Parameters

You configure the echo function by setting the following parameters in the topology file at the global, template and port level:

- **echoSupport:** Enables and disables echo mode. Set to 1 to enable the echo function. It defaults to 0 (disable).
- **echoMinRx:** The minimum interval between echo packets the local system is capable of receiving. This is advertised in the BFD control packet. When the echo function is enabled, it defaults to 50. If you disable the echo function, this parameter is automatically set to 0, which indicates the port or the node cannot process or receive echo packets.
- **slowMinTx:** The minimum interval between transmitting BFD control packets when the echo packets are being exchanged.

Scripts

ptmd executes scripts at `/etc/ptm.d/if-topo-pass` and `/etc/ptm.d/if-topo-fail` for each interface that goes through a change, running `if-topo-pass` when an LLDP or BFD check passes and running `if-topo-fails` when the check fails. The scripts receive an argument string that is the result of the `ptmctl` command, described in ptmd Commands below.

You should modify these default scripts as needed.

Using ptmd Service Commands

PTM sends client notifications in CSV format.

cumulus@switch:~\$ `sudo service ptmd start|restart|force-reload`: Starts or restarts the ptmd service. The `topology.dot` file must be present in order for the service to start.

cumulus@switch:~\$ `sudo service ptmd reconfig`: Instructs ptmd to read the `topology.dot` file again without restarting, applying the new configuration to the running state.

cumulus@switch:~\$ `sudo service ptmd stop`: Stops the ptmd service.

cumulus@switch:~\$ `sudo service ptmd status`: Retrieves the current running state of ptmd.

Using ptmctl Commands

ptmctl is a client of ptmd; it retrieves the daemon's operational state. It connects to ptmd over a Unix socket and listens for notifications. ptmctl parses the CSV notifications sent by ptmd.

See `man ptmctl` for more information.

ptmctl Examples

For basic output, use ptmctl without any options:

```
cumulus@switch:~$ sudo ptmctl
```

```
-----
port  cbl      BFD      BFD      BFD      BFD
      status status peer      local  type
-----
swp1  pass     pass     11.0.0.2  N/A      singlehop
swp2  pass     N/A      N/A      N/A      N/A
swp3  pass     N/A      N/A      N/A      N/A
```

For more detailed output, use the `-d` option:

```
cumulus@switch:~$ sudo ptmctl -d
```

```
-----
-----
-----
port  cbl      exp      act      sysname  portID  portDescr  match  last
BFD   BFD      BFD      BFD      det_mult tx_timeout rx_timeout
echo_tx_timeout echo_rx_timeout max_hop_cnt
      status nbr      nbr      on      upd
Type  state  peer  DownDiag
-----
-----
-----
swp45 pass   h1:swp1 h1:swp1  h1      swp1      swp1      IfName 5m: 5s  N
```


/A	N/A	N/A	N/A	N/A	N/A	N/A	N
/A		N/A		N/A			
swp46	fail	h2:swp1	h2:swp1	h2	swp1	swp1	IfName 5m: 5s N
/A	N/A	N/A	N/A	N/A	N/A	N/A	N
/A		N/A		N/A			

To return information on active BFD sessions `ptmd` is tracking, use the `-b` option:

```
cumulus@switch:~$ sudo ptmctl -b
```

port	peer	state	local	type	diag
swp1	11.0.0.2	Up	N/A	singlehop	N/A
N/A	12.12.12.1	Up	12.12.12.4	multihop	N/A

To return LLDP information, use the `-l` option. It returns only the active neighbors currently being tracked by `ptmd`.

```
cumulus@switch:~$ sudo ptmctl -l
```

port	sysname	portID	port descr	match on	last upd
swp45	h1	swp1	swp1	IfName	5m:59s
swp46	h2	swp1	swp1	IfName	5m:59s

To return detailed information on active BFD sessions `ptmd` is tracking, use the `-b` and `-d` options (results are for an IPv6-connected peer):

```
cumulus@switch:~$ sudo ptmctl -b -d
```

port	peer	state	local	type	diag	det	tx_timeout
rx_timeout	echo	echo	max	rx_ctrl	tx_ctrl	rx_echo	
tx_echo							

```

mult                                tx_timeout  rx_timeout
hop_cnt
-----
-----
-----
swp1  fe80::202:ff:fe00:1  Up      N/A      singlehop  N/A      3      300
900      0      0      N/A      187172    185986    0
0
swp1  3101:abc:bcad::2    Up      N/A      singlehop  N/A      3      300
900      0      0      N/A      501      533      0
0

```

To output `ptmctl` data in JSON format, use the `-j` option:

```

cumulus@switch:~$ sudo ptmctl -j

{
  "0": {
    "BFD status": "pass",
    "BFD type": "singlehop",
    "cbl status": "pass",
    "BFD peer": "11.0.0.2",
    "BFD local": "N/A",
    "port": "swp1"
  },
  "1": {
    "BFD status": "N/A",
    "BFD type": "N/A",
    "cbl status": "pass",
    "BFD peer": "N/A",
    "BFD local": "N/A",
    "port": "swp2"
  },
  "2": {
    "BFD status": "N/A",
    "BFD type": "N/A",
    "cbl status": "pass",
    "BFD peer": "N/A",
    "BFD local": "N/A",
    "port": "swp3"
  }
}

```

ptmctl Error Outputs

If there are errors in the topology file or there isn't a session, PTM will return appropriate outputs. Typical error strings are:

```
Topology file error [/etc/ptm.d/topology.dot] [cannot find node cumulus] -
please check /var/log/ptmd.log for more info

Topology file error [/etc/ptm.d/topology.dot] [cannot open file (errno 2)] -
please check /var/log/ptmd.log for more info

No Hostname/MgmtIP found [Check LLDPD daemon status] -
please check /var/log/ptmd.log for more info

No BFD sessions . Check connections

No LLDP ports detected. Check connections

Unsupported command
```

For example:

```
cumulus@switch:~$ sudo ptmctl

-----
cmd          error
-----
get-status   Topology file error [/etc/ptm.d/topology.dot] [cannot open file
(errno 2)] - please check /var/log/ptmd.log for more info
```



If you encounter errors with the `topology.dot` file, you can use `dot` (included in the Graphviz package) to validate the syntax of the topology file.

By simply opening the topology file with Graphviz, you can ensure that it is readable and that the file format is correct.

If you edit `topology.dot` file from a Windows system, be sure to double check the file formatting; there may be extra characters that keep the graph from working correctly.

Configuration Files

- `/etc/ptm.d/topology.dot`
- `/etc/ptm.d/if-topo-pass`

- [/etc/ptm.d/if-topo-fail](#)

Useful Links

- [Bidirectional Forwarding Detection \(BFD\)](#)
- [Graphviz](#)
- [LLDP on Wikipedia](#)
- [PTMd GitHub repo](#)

Bonding - Link Aggregation

Linux bonding provides a method for aggregating multiple network interfaces (the slaves) into a single logical bonded interface (the bond). Cumulus RMP bonding supports the IEEE 802.3ad link aggregation mode. Link aggregation allows one or more links to be aggregated together to form a *link aggregation group* (LAG), such that a media access control (MAC) client can treat the link aggregation group as if it were a single link. The benefits of link aggregation are:

- Linear scaling of bandwidth as links are added to LAG
- Load balancing
- Failover protection

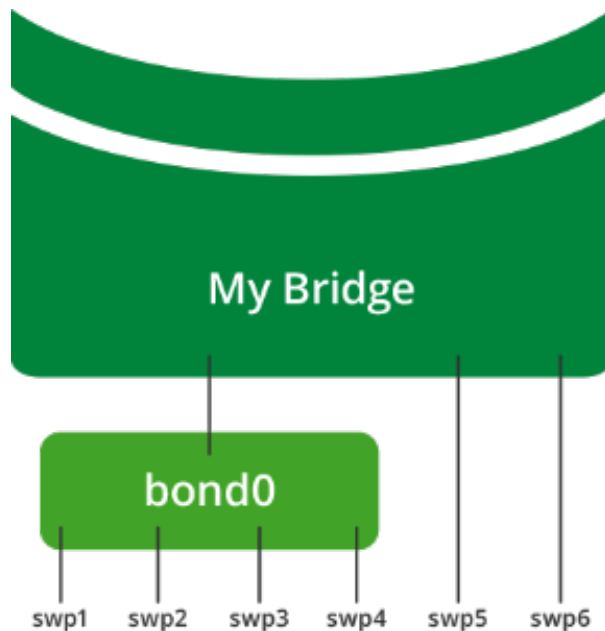
The Cumulus RMP LAG control protocol is LACP version 1.

Contents

(Click to expand)

- [Contents \(see page 124\)](#)
- [Example: Bonding 4 Slaves \(see page 125\)](#)
- [Hash Distribution \(see page 127\)](#)
- [Configuration Files \(see page 127\)](#)
- [Useful Links \(see page 127\)](#)
- [Caveats and Errata \(see page 127\)](#)

Example: Bonding 4 Slaves



In this example, front panel port interfaces swp1-swp4 are slaves in bond0 (swp5 and swp6 are not part of bond0). The name of the bond is arbitrary as long as it follows Linux interface naming guidelines, and is unique within the switch. The only bonding mode supported in Cumulus RMP is 802.3ad. There are several 802.3ad settings that can be applied to each bond:

- `bond-slave`: The list of slaves in bond.
- `bond-mode`: **Must** be set to `802.3ad`.
- `bond-miimon`: How often the link state of each slave is inspected for link failures. It defaults to `0`, but `100` is the recommended value.



`bond-miimon` **must** be defined in `/etc/network/interfaces`.

- `bond-use-carrier`: How to determine link state.
- `bond-xmit-hash-policy`: Hash method used to select the slave for a given packet; **must** be set to `layer3+4`.
- `bond-lacp-rate`: Rate to ask link partner to transmit LACP control packets.
- `bond-min-links`: Specifies the minimum number of links that must be active before asserting carrier on the bond. Minimum value is `1`, but a value greater than `1` is useful if higher level services need to ensure a minimum of aggregate bandwidth before putting the bond in service.



`bond-min-links` **must** be defined in `/etc/network/interfaces` and it cannot be set to `0`. See also [this release note](#).

See Useful Links below for more details on settings.

To configure the bond, edit `/etc/network/interfaces` and add a stanza for `bond0`:

```
auto bond0
iface bond0
    address 10.0.0.1/30
    bond-slaves swp1 swp2 swp3 swp4
    bond-mode 802.3ad
    bond-miimon 100
    bond-use-carrier 1
    bond-lacp-rate 1
    bond-min-links 1
    bond-xmit-hash-policy layer3+4
```

However, if you are intending that the bond become part of a bridge, you don't need to specify an IP address. The configuration would look like this:

```
auto bond0
iface bond0
    bond-slaves glob swp1-4
    bond-mode 802.3ad
    bond-miimon 100
    bond-use-carrier 1
    bond-lacp-rate 1
    bond-min-links 1
    bond-xmit-hash-policy layer3+4
```

See `man interfaces` for more information on `/etc/network/interfaces`.

Here the link state sampling rate is 1/10 sec, and the LACP transmit rate is set to high. `min_links` is set to 1 to indicate the bond must have at least one active member for bond to assert carrier. If the number of active members drops below `min_links`, the bond will appear to upper-level protocols as *link-down*. When the number of active links returns to greater than or equal to `min_links`, the bond will become *link-up*.

When networking is started on switch, `bond0` is created as MASTER and interfaces `swp1-swp4` come up in SLAVE mode, as seen in the `ip link show` command:

```
3: swp1: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
master bond0 state UP mode DEFAULT qlen 500
    link/ether 44:38:39:00:03:c1 brd ff:ff:ff:ff:ff:ff
4: swp2: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
master bond0 state UP mode DEFAULT qlen 500
    link/ether 44:38:39:00:03:c1 brd ff:ff:ff:ff:ff:ff
5: swp3: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
master bond0 state UP mode DEFAULT qlen 500
```

```
link/ether 44:38:39:00:03:c1 brd ff:ff:ff:ff:ff:ff
6: swp4: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
master bond0 state UP mode DEFAULT qlen 500
link/ether 44:38:39:00:03:c1 brd ff:ff:ff:ff:ff:ff
```

And

```
55: bond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc noqueue
state UP mode DEFAULT
link/ether 44:38:39:00:03:c1 brd ff:ff:ff:ff:ff:ff
```



All slave interfaces within a bond will have the same MAC address as the bond. Typically, the first slave added to the bond donates its MAC address for the bond. The other slaves' MAC addresses are set to the bond MAC address. The bond MAC address is used as source MAC address for all traffic leaving the bond, and provides a single destination MAC address to address traffic to the bond.

Hash Distribution

Egress traffic through a bond is distributed to a slave based on a packet hash calculation. This distribution provides load balancing over the slaves. The hash calculation uses packet header data to pick which slave to transmit the packet. For IP traffic, IP header source and destination fields are used in the calculation. For IP + TCP/UDP traffic, source and destination ports are included in the hash calculation. Traffic for a given conversation flow will always hash to the same slave. Many flows will be distributed over all the slaves to load balance the total traffic. In a failover event, the hash calculation is adjusted to steer traffic over available slaves.

Configuration Files

- /etc/network/interfaces

Useful Links

- <http://www.linuxfoundation.org/collaborate/workgroups/networking/bonding>
- 802.3ad (Accessible writeup)
- Link aggregation from Wikipedia

Caveats and Errata

- An interface cannot belong to multiple bonds.
- Slave ports within a bond should all be set to the same speed/duplex, and should match the link partner's slave ports.

- A bond cannot enslave VLAN subinterfaces. A bond can have subinterfaces, but not the other way around.

Ethernet Bridging - VLANs

Ethernet bridges provide a means for hosts to communicate at layer 2. Bridge members can be individual physical interfaces, bonds or logical interfaces that traverse an 802.1Q VLAN trunk.

Cumulus RMP has two modes for configuring bridges: *VLAN-aware* (see page 148) and *traditional*. The bridge driver in Cumulus RMP is capable of VLAN filtering, which allows for configurations that are similar to incumbent network devices. While Cumulus RMP supports Ethernet bridges in traditional mode Cumulus Networks recommends using *VLAN-aware* mode.

For a comparison of traditional and VLAN-aware modes, read [this knowledge base article](#).



You can configure both VLAN-aware and traditional mode bridges on the same network in Cumulus RMP; however you should not have more than one VLAN-aware bridge on a given switch.

Contents

(Click to expand)

- Contents (see page 128)
 - Configuration Files (see page 128)
 - Commands (see page 129)
 - Creating a Bridge between Physical Interfaces (see page 129)
 - Creating the Bridge and Adding Interfaces (see page 129)
 - Showing and Verifying the Bridge Configuration (see page 131)
 - Bridge Interface MAC Address and MTU (see page 131)
 - Examining MAC Addresses (see page 131)
 - Multiple Bridges (see page 132)
 - Configuring an SVI (Switch VLAN Interface) (see page 135)
 - Showing and Verifying the Bridge Configuration (see page 136)
 - Using Trunks in Traditional Bridging Mode (see page 137)
 - Trunk Example (see page 138)
 - Showing and Verifying the Trunk (see page 139)
 - Additional Examples (see page 139)
 - Configuration Files (see page 139)
 - Useful Links (see page 140)
 - Caveats and Errata (see page 140)

Configuration Files

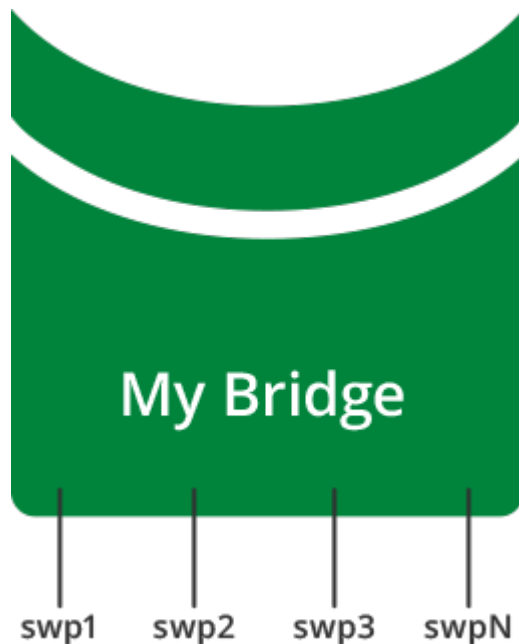
- /etc/network/interfaces

Commands

- brctl
- bridge
- ip addr
- ip link

Creating a Bridge between Physical Interfaces

The basic use of bridging is to connect all of the physical and logical interfaces in the system into a single layer 2 domain.



Creating the Bridge and Adding Interfaces

You statically manage bridge configurations in `/etc/network/interfaces`. The following configuration snippet details an example bridge used throughout this chapter, explicitly enabling [spanning tree](#) (see [page 91](#)) and setting the bridge MAC address ageing timer. First, create a bridge with a descriptive name of 15 characters or fewer. Then add the logical interfaces (bond0) and physical interfaces (swp5, swp6) to assign to that bridge.

```
auto my_bridge
iface my_bridge
    bridge-ports bond0 swp5 swp6
    bridge-ageing 150
    bridge-stp on
```

Keyword	Explanation
bridge-ports	List of logical and physical ports belonging to the logical bridge.
bridge-ageing	Maximum amount of time before a MAC addresses learned on the bridge expires from the bridge MAC cache. The default value is 300 seconds.
bridge-stp	Enables spanning tree protocol on this bridge. The default spanning tree mode is Per VLAN Rapid Spanning Tree Protocol (PVRST). For more information on spanning-tree configurations see the configuration section: Spanning Tree and Rapid Spanning Tree (see page 91) .

To bring up the bridge `my_bridge`, use the `ifreload` command:

```
cumulus@switch:~$ sudo ifreload -a
```

Runtime Configuration (Advanced)



A runtime configuration is non-persistent, which means the configuration you create here does not persist after you reboot the switch.

To create the bridge and interfaces on the bridge, run:

```
cumulus@switch:~$ sudo brctl addbr my_bridge

cumulus@switch:~$ sudo brctl addif my_bridge bond0 swp5 swp6

cumulus@switch:~$ sudo brctl show
bridge name      bridge id        STP enabled      interfaces
my_bridge        8000.44383900129b  yes              bond0
                                                           swp5
                                                           swp6
```

```
cumulus@switch:~$ sudo ip link set up dev my_bridge
```

```
cumulus@switch:~$ sudo ip link set up dev bond0
```

```
cumulus@switch:~$ sudo for I in {5..6}; do ip link set up dev swp$I; done
```

Showing and Verifying the Bridge Configuration

```
cumulus@switch:~$ ip link show my_bridge
56: my_bridge: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
state UP mode DEFAULT
    link/ether 44:38:39:00:12:9b brd ff:ff:ff:ff:ff:ff
```

Using netshow to Display Bridge Information

netshow is an add-on tool that is not installed in Cumulus RMP by default. Refer to [this knowledge base article](#) for steps to install it.

```
cumulus@switch$ netshow interface bridge
```

	Name	Speed	Mtu	Mode	Summary
--	-----	-----	-----	-----	-----
UP	my_bridge	N/A	1500	Bridge/L2	Untagged: bond0, swp5-6 Root Port: bond0 VlanID: Untagged



Do not try to bridge the management port, eth0, with any switch ports (like swp0, swp1, and so forth). For example, if you created a bridge with eth0 and swp1, it will **not** work.

Bridge Interface MAC Address and MTU

A bridge is a logical interface with a MAC address and an MTU (maximum transmission unit). The bridge MTU is the minimum MTU among all its members. The bridge's MAC address is inherited from the first interface that is added to the bridge as a member. The bridge MAC address remains unchanged until the member interface is removed from the bridge, at which point the bridge will inherit from the next member interface, if any. The bridge can also be assigned an IP address, as discussed below.

Examining MAC Addresses

A bridge forwards frames by looking up the destination MAC address. A bridge learns the source MAC address of a frame when the frame enters the bridge on an interface. After the MAC address is learned, the bridge maintains an age for the MAC entry in the bridge table. The age is refreshed when a frame is seen again with the same source MAC address. When a MAC is not seen for greater than the MAC ageing time, the MAC address is deleted from the bridge table.

The following shows the MAC address table of the example bridge. Notice that the `is local?` column indicates if the MAC address is the interface's own MAC address (`is local` is `yes`), or if it is learned on the interface from a packet's source MAC (where `is local` is `no`):

```
cumulus@switch:~$ sudo brctl showmacs my_bridge
```

port	name	mac addr	is local?	ageing timer
swp4		06:90:70:22:a6:2e	no	19.47
swp1		12:12:36:43:6f:9d	no	40.50
bond0		2a:95:22:94:d1:f0	no	1.98
swp1		44:38:39:00:12:9b	yes	0.00
swp2		44:38:39:00:12:9c	yes	0.00
swp3		44:38:39:00:12:9d	yes	0.00
swp4		44:38:39:00:12:9e	yes	0.00
bond0		44:38:39:00:12:9f	yes	0.00
swp2		90:e2:ba:2c:b1:94	no	12.84
swp2		a2:84:fe:fc:bf:cd	no	9.43

You can use the `bridge fdb` command to display the MAC address table as well:

```
cumulus@en-sw2$ bridge fdb show
70:72:cf:9d:4e:36 dev swp2 VLAN 0 master bridge-A permanent
70:72:cf:9d:4e:35 dev swp1 VLAN 0 master bridge-A permanent
70:72:cf:9d:4e:38 dev swp4 VLAN 0 master bridge-B permanent
70:72:cf:9d:4e:37 dev swp3 VLAN 0 master bridge-B permanent
```

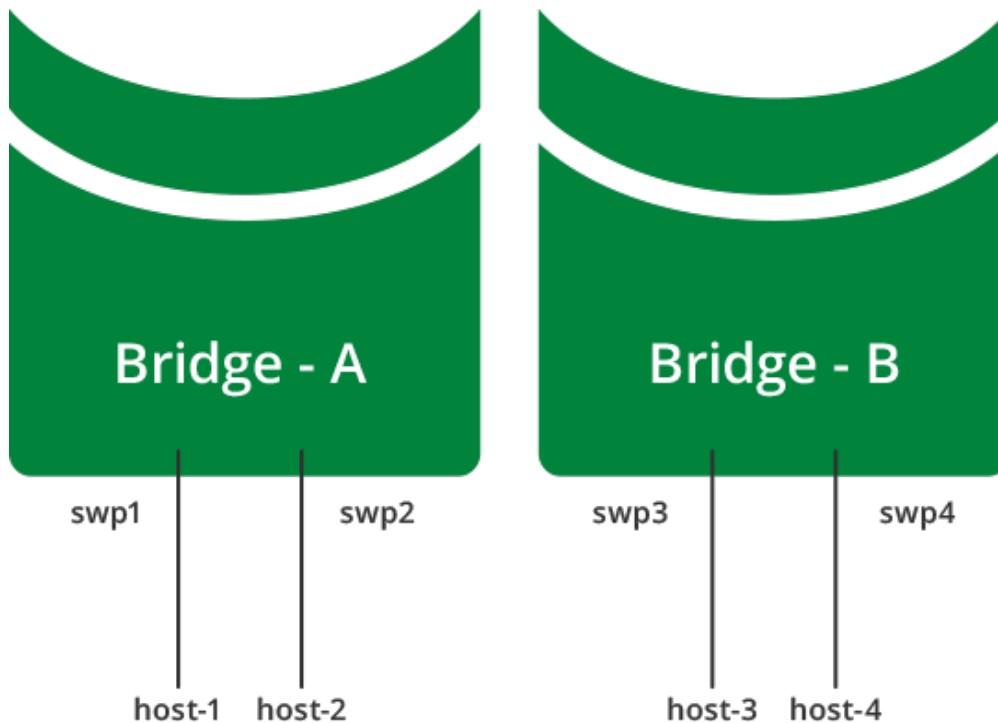


You can clear a MAC address from the table using the `bridge fdb` command:

```
cumulus@switch:~$ sudo bridge fdb del 90:e2:ba:2c:b1:94 dev swp2
```

Multiple Bridges

Sometimes it is useful to logically divide a switch into multiple layer 2 domains, so that hosts in one domain can communicate with other hosts in the same domain but not in other domains. You can achieve this by configuring multiple bridges and putting different sets of interfaces in the different bridges. In the following example, host-1 and host-2 are connected to the same bridge (bridge-A), while host-3 and host-4 are connected to another bridge (bridge-B). host-1 and host-2 can communicate with each other, so can host-3 and host-4, but host-1 and host-2 cannot communicate with host-3 and host-4.



To configure multiple bridges, edit `/etc/network/interfaces`:

```
auto bridge-A
iface bridge-A
    bridge-ports swp1 swp2
    bridge-stp on

auto my_bridge
iface my_bridge
    bridge-ports swp3 swp4
    bridge-stp on
```

To bring up the bridges bridge-A and bridge-B, use the `ifreload` command:

```
cumulus@switch:~$ sudo ifreload -a
```

Runtime Configuration (Advanced)



A runtime configuration is non-persistent, which means the configuration you create here does not persist after you reboot the switch.

```

cumulus@switch:~$ sudo brctl addbr bridge-A

cumulus@switch:~$ sudo brctl addif bridge-A swp1 swp2

cumulus@switch:~$ sudo brctl addbr bridge-B

cumulus@switch:~$ sudo brctl addif bridge-B swp3 swp4

cumulus@switch:~$ sudo for I in {1..4}; do ip link set up dev swp$I; done

cumulus@switch:~$ sudo ip link set up dev bridge-A

cumulus@switch:~$ sudo ip link set up dev bridge-B

cumulus@switch:~$ sudo brctl show
  bridge name      bridge id                STP enabled    interfaces
  bridge-A         8000.44383900129b        yes            swp1
                                                           swp2
  bridge-B         8000.44383900129d        yes            swp3
                                                           swp4

cumulus@en-sw2$ ip link show bridge-A
97: bridge-A: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
state UP mode DEFAULT
    link/ether 70:72:cf:9d:4e:35 brd ff:ff:ff:ff:ff:ff
cumulus@en-sw2$ ip link show bridge-B
98: bridge-B: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
state UP mode DEFAULT
    link/ether 70:72:cf:9d:4e:37 brd ff:ff:ff:ff:ff:ff

```

Using netshow to Display the Bridges

netshow is an add-on tool that is not installed in Cumulus RMP by default. Refer to [this knowledge base article](#) for steps to install it.

```

cumulus@switch$ netshow interface bridge

```

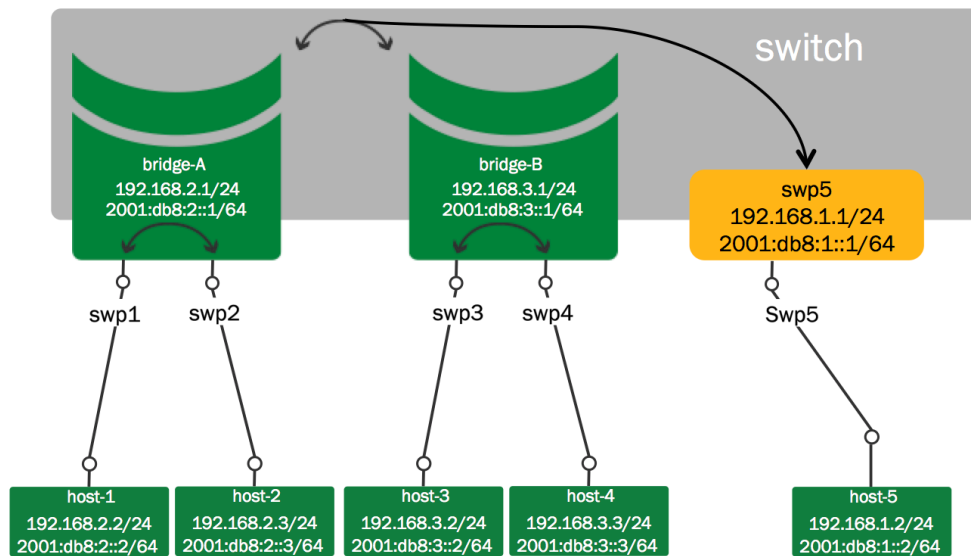
	Name	Speed	Mtu	Mode	Summary
UP	bridge-A	N/A	1500	Bridge/L2	Untagged: swp1-2 Root Port: swp2 VlanID: Untagged
UP	bridge-B	N/A	1500	Bridge/L2	Untagged: swp3-4 Root Port: swp3 VlanID: Untagged

Configuring an SVI (Switch VLAN Interface)

A bridge creates a layer 2 forwarding domain for hosts to communicate. A bridge can be assigned an IP address — typically of the same subnet as the hosts that are members of the bridge — and participate in routing topologies. This enables hosts within a bridge to communicate with other hosts outside the bridge through layer 3 routing.



When an interface is added to a bridge, it ceases to function as a router interface, and the IP address on the interface, if any, becomes reachable.



The configuration for the two bridges example looks like the following:

```
auto swp5
iface swp5
    address 192.168.1.2/24
    address 2001:DB8:1::2/64
auto bridge-A
iface bridge-A
    address 192.168.2.1/24
    address 2001:DB8:2::1/64
    bridge-ports swp1 swp2
    bridge-stp on
auto bridge-B
iface bridge-B
    address 192.168.3.1/24
    address 2001:DB8:3::1/64
    bridge-ports swp3 swp4
    bridge-stp on
```

To bring up swp5 and bridges bridge-A and bridge-B, use the `ifreload` command:

```
cumulus@switch:~$ sudo ifreload -a
```

Showing and Verifying the Bridge Configuration

```
cumulus@switch$ ip addr show bridge-A
106: bridge-A: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
state UP
    link/ether 70:72:cf:9d:4e:35 brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.1/24 scope global bridge-A
    inet6 2001:db8:2::1/64 scope global
    valid_lft forever preferred_lft forever
    inet6 fe80::7272:cfff:fe9d:4e35/64 scope link
    valid_lft forever preferred_lft forever
```

```
cumulus@switch$ ip addr show bridge-B
107: bridge-B: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
state UP
    link/ether 70:72:cf:9d:4e:37 brd ff:ff:ff:ff:ff:ff
    inet 192.168.3.1/24 scope global bridge-B
    inet6 2001:db8:3::1/64 scope global
    valid_lft forever preferred_lft forever
    inet6 fe80::7272:cfff:fe9d:4e37/64 scope link
    valid_lft forever preferred_lft forever
```

To see all the routes on the switch use the `ip route show` command:

```
cumulus@switch$ ip route show
192.168.1.0/24 dev swp5 proto kernel scope link src 192.168.1.2 dead
192.168.2.0/24 dev bridge-A proto kernel scope link src 192.168.2.1
192.168.3.0/24 dev bridge-B proto kernel scope link src 192.168.3.1
```

Runtime Configuration (Advanced)



A runtime configuration is non-persistent, which means the configuration you create here does not persist after you reboot the switch.

To add an IP address to a bridge:

```
cumulus@switch:~$ sudo ip addr add 192.0.2.101/24 dev bridge-A

cumulus@switch:~$ sudo ip addr add 192.0.2.102/24 dev bridge-B
```

Using netshow to Display the SVI

netshow is an add-on tool that is not installed in Cumulus RMP by default. Refer to [this knowledge base article](#) for steps to install it.

```
cumulus@switch$ netshow interface bridge
```

	Name	Speed	Mtu	Mode	Summary
UP	bridge-A	N/A	1500	Bridge/L3	IP: 192.168.2.1/24, 2001:db8:2::1/64 Untagged: swp1-2 Root Port: swp2 VlanID: Untagged
UP	bridge-B	N/A	1500	Bridge/L3	IP: 192.168.3.1/24, 2001:db8:3::1/64 Untagged: swp3-4 Root Port: swp3 VlanID: Untagged

Using Trunks in Traditional Bridging Mode

The IEEE standard for trunking is 802.1Q. The 802.1Q specification adds a 4 byte header within the Ethernet frame that identifies the VLAN of which the frame is a member.

802.1Q also identifies an *untagged* frame as belonging to the *native* VLAN (most network devices default their native VLAN to 1). The concept of native, non-native, tagged or untagged has generated confusion due to mixed terminology and vendor-specific implementations. Some clarification is in order:

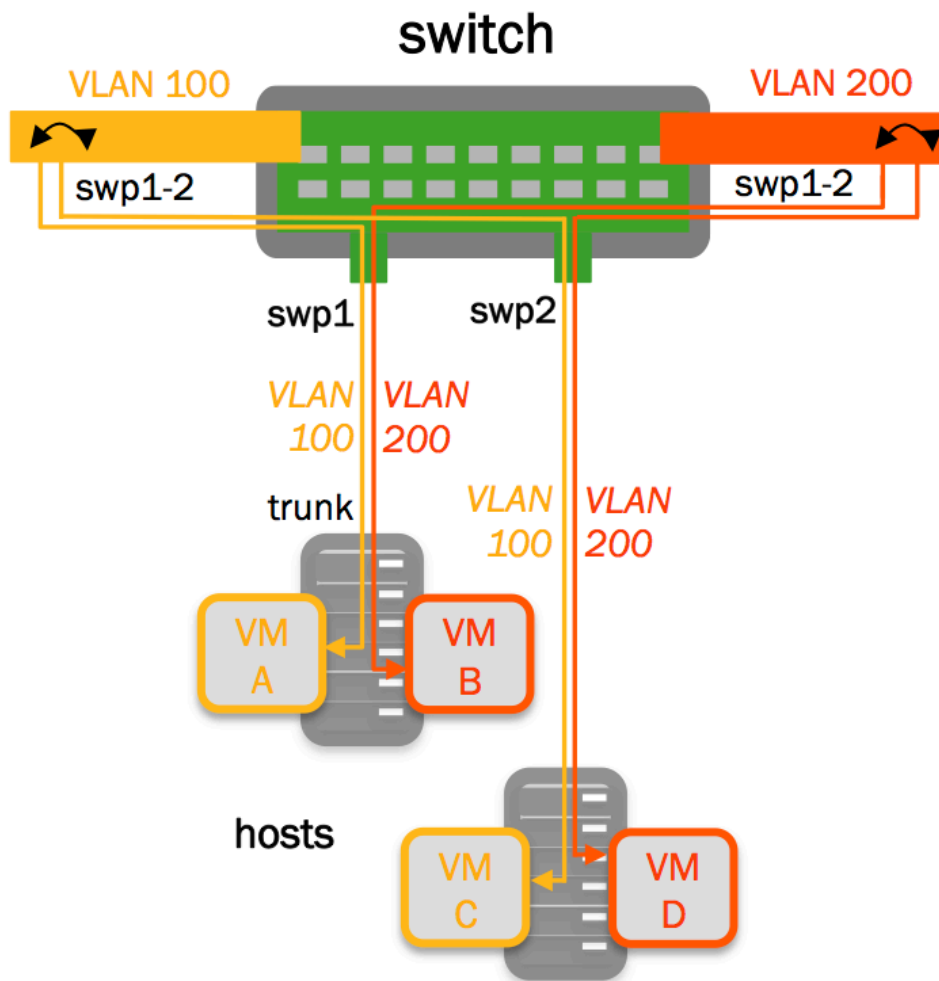
- A *trunk port* is a switch port configured to send and receive 802.1Q tagged frames.
- A switch sending an untagged (bare Ethernet) frame on a trunk port is sending from the native VLAN defined on the trunk port.
- A switch sending a tagged frame on a trunk port is sending to the VLAN identified by the 802.1Q tag.
- A switch receiving an untagged (bare Ethernet) frame on a trunk port places that frame in the native VLAN defined on the trunk port.
- A switch receiving a tagged frame on a trunk port places that frame in the VLAN identified by the 802.1Q tag.

A bridge in traditional mode has no concept of trunks, just tagged or untagged frames. With a trunk of 200 VLANs, there would need to be 199 bridges, each containing a tagged physical interface, and one bridge containing the native untagged VLAN. See the examples below for more information.



The interaction of tagged and un-tagged frames on the same trunk often leads to undesired and unexpected behavior. A switch that uses VLAN 1 for the native VLAN may send frames to a switch that uses VLAN 2 for the native VLAN, thus merging those two VLANs and their spanning tree state.

Trunk Example



Configure the following in `/etc/network/interfaces`:

```
auto br-VLAN100
iface br-VLAN100
    bridge-ports swp1.100 swp2.100
    bridge-stp on
```

```
auto br-VLAN200
iface br-VLAN200
    bridge-ports swp1.200 swp2.200
    bridge-stp on
```

To bring up br-VLAN100 and br-VLAN200, use the `ifreload` command:

```
cumulus@switch:~$ sudo ifreload -a
```

Showing and Verifying the Trunk

```
cumulus@en-sw2$ brctl show
bridge name bridge id STP enabled interfaces
br-VLAN100 8000.7072cf9d4e35 no swp1.100
    swp2.100
br-VLAN200 8000.7072cf9d4e35 no swp1.200
    swp2.200
```

Using `netshow` to Display the Trunk

`netshow` is an add-on tool that is not installed in Cumulus RMP by default. Refer to [this knowledge base article](#) for steps to install it.

```
cumulus@switch$ netshow interface bridge
```

	Name	Speed	Mtu	Mode	Summary
UP	br-VLAN100	N/A	1500	Bridge/L2	Tagged: swp1-2 STP: rootSwitch(32768) VlanID: 100
UP	br-VLAN200	N/A	1500	Bridge/L2	Tagged: swp1-2 STP: rootSwitch(32768) VlanID: 200

Additional Examples

You can find additional examples of VLAN tagging in [this chapter](#) (see page 140).

Configuration Files

- `/etc/network/interfaces`
- `/etc/network/interfaces.d/`

- [/etc/network/if-down.d/](#)
- [/etc/network/if-post-down.d/](#)
- [/etc/network/if-pre-up.d/](#)
- [/etc/network/if-up.d/](#)

Useful Links

- <http://www.linuxfoundation.org/collaborate/workgroups/networking/bridge>
- <http://www.linuxfoundation.org/collaborate/workgroups/networking/vlan>
- <http://www.linuxjournal.com/article/8172>

Caveats and Errata

- The same bridge cannot contain multiple subinterfaces of the **same** port as members. Attempting to apply such a configuration will result in an error.

VLAN Tagging

This article shows two examples of VLAN tagging (see [page 140](#)), one basic and one more advanced. They both demonstrate the streamlined interface configuration from `ifupdown2`. For more information, see [Configuring and Managing Network Interfaces](#) (see [page 71](#)).

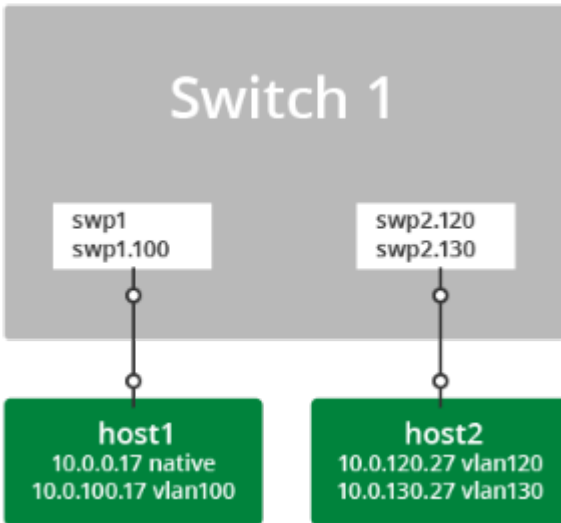
Contents

(Click to expand)

- [Contents](#) (see [page 140](#))
- [VLAN Tagging, a Basic Example](#) (see [page 140](#))
 - [Persistent Configuration](#) (see [page 141](#))
- [VLAN Tagging, an Advanced Example](#) (see [page 141](#))
 - [Persistent Configuration](#) (see [page 142](#))
 - [VLAN Translation](#) (see [page 147](#))

VLAN Tagging, a Basic Example

A simple configuration demonstrating VLAN tagging involves two hosts connected to a switch.



- *host1* connects to swp1 with both untagged frames and with 802.1Q frames tagged for *vlan100*.
- *host2* connects to swp2 with 802.1Q frames tagged for *vlan120* and *vlan130*.

Persistent Configuration

To configure the above example persistently, configure `/etc/network/interfaces` like this:

```

# Config for host1

auto swp1
iface swp1

auto swp1.100
iface swp1.100

# Config for host2
# swp2 must exist to create the .1Q subinterfaces, but it is not assigned
an address

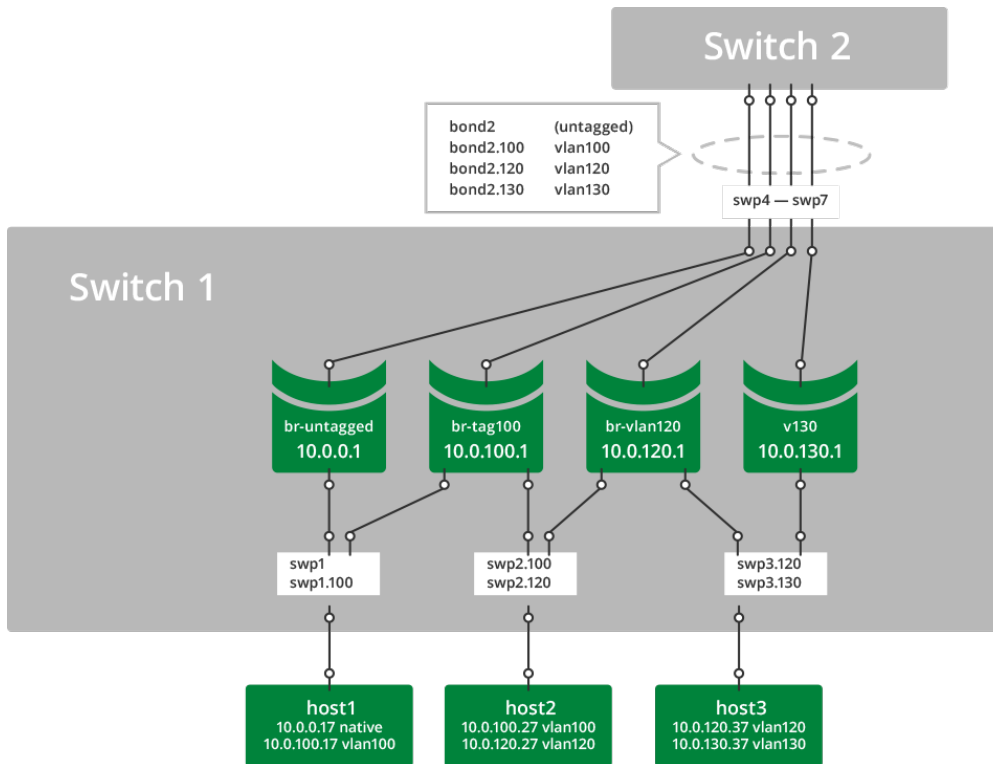
auto swp2
iface swp2

auto swp2.120
iface swp2.120

auto swp2.130
iface swp2.130
  
```

VLAN Tagging, an Advanced Example

This example of VLAN tagging is more complex, involving three hosts and two switches, with a number of bridges and a bond connecting them all.



- *host1* connects to bridge *br-untagged* with bare Ethernet frames and to bridge *br-tag100* with 802.1q frames tagged for *vlan100*.
- *host2* connects to bridge *br-tag100* with 802.1q frames tagged for *vlan100* and to bridge *br-vlan120* with 802.1q frames tagged for *vlan120*.
- *host3* connects to bridge *br-vlan120* with 802.1q frames tagged for *vlan120* and to bridge *v130* with 802.1q frames tagged for *vlan130*.
- *bond2* carries tagged and untagged frames in this example.

Although not explicitly designated, the bridge member ports function as 802.1Q *access ports* and *trunk ports*. In the example above, comparing Cumulus RMP with a traditional Cisco device:

- *swp1* is equivalent to a trunk port with untagged and *vlan100*.
- *swp2* is equivalent to a trunk port with *vlan100* and *vlan120*.
- *swp3* is equivalent to a trunk port with *vlan120* and *vlan130*.
- *bond2* is equivalent to an EtherChannel in trunk mode with untagged, *vlan100*, *vlan120*, and *vlan130*.
- Bridges *br-untagged*, *br-tag100*, *br-vlan120*, and *v130* are equivalent to SVIs (switched virtual interfaces).

Persistent Configuration

From `/etc/network/interfaces`:

```
# Config for host1 - - - - -
```

```

- - - - -

# swp1 does not need an iface section unless it has a specific setting,
# it will be picked up as a dependent of swp1.100.
# And swp1 must exist in the system to create the .1q subinterfaces..
# but it is not applied to any bridge..or assigned an address.

auto swp1.100
iface swp1.100

# Config for host2
# swp2 does not need an iface section unless it has a specific setting,
# it will be picked up as a dependent of swp2.100 and swp2.120.
# And swp2 must exist in the system to create the .1q subinterfaces..
# but it is not applied to any bridge..or assigned an address.

auto swp2.100
iface swp2.100

auto swp2.120
iface swp2.120

# Config for host3
# swp3 does not need an iface section unless it has a specific setting,
# it will be picked up as a dependent of swp3.120 and swp3.130.
# And swp3 must exist in the system to create the .1q subinterfaces..
# but it is not applied to any bridge..or assigned an address.

auto swp3.120
iface swp3.120

auto swp3.130
iface swp3.130

# Configure the bond - - - - -
- - - - -

auto bond2
iface bond2
    bond-slaves glob swp4-7
    bond-mode 802.3ad
    bond-miimon 100
    bond-use-carrier 1
    bond-lacp-rate 1
    bond-min-links 1

```

```

bond-xmit-hash-policy layer3+4

# configure the bridges - - - - -

- - - - -

auto br-untagged
iface br-untagged
    address 10.0.0.1/24
    bridge-ports swp1 bond2
    bridge-stp on

auto br-tag100
iface br-tag100
    address 10.0.100.1/24
    bridge-ports swp1.100 swp2.100 bond2.100
    bridge-stp on

auto br-vlan120
iface br-vlan120
    address 10.0.120.1/24
    bridge-ports swp2.120 swp3.120 bond2.120
    bridge-stp on

auto vl30
iface vl30
    address 10.0.130.1/24
    bridge-ports swp2.130 swp3.130 bond2.130
    bridge-stp on

# - - - - -

```

To verify:

```

cumulus@switch:~$ sudo mstpctl showbridge br-tag100
br-tag100 CIST info
  enabled          yes
  bridge id        8.000.44:38:39:00:32:8B
  designated root  8.000.44:38:39:00:32:8B
  regional root    8.000.44:38:39:00:32:8B
  root port        none
  path cost        0          internal path cost    0
  max age          20          bridge max age      20
  forward delay    15          bridge forward delay 15

```



```

tx hold count 6          max hops          20
hello time      2        ageing time       300
force protocol version  rstp
time since topology change 333040s
topology change count    1
topology change          no
topology change port      swp2.100
last topology change port None

cumulus@switch:~$ sudo mstpcctl showportdetail br-tag100 | grep -B 2 state
br-tag100:bond2.100 CIST info
    enabled          yes          role          Designated
    port id          8.003        state         forwarding
--
br-tag100:swp1.100 CIST info
    enabled          yes          role          Designated
    port id          8.001        state         forwarding
--
br-tag100:swp2.100 CIST info
    enabled          yes          role          Designated
    port id          8.002        state         forwarding

cumulus@switch:~$ cat /proc/net/vlan/config
VLAN Dev name      | VLAN ID
Name-Type: VLAN_NAME_TYPE_RAW_PLUS_VID_NO_PAD
bond2.100          | 100   | bond2
bond2.120          | 120   | bond2
bond2.130          | 130   | bond2
swp1.100           | 100   | swp1
swp2.100           | 100   | swp2
swp2.120           | 120   | swp2
swp3.120           | 120   | swp3
swp3.130           | 130   | swp3

cumulus@switch:~$ cat /proc/net/bonding/bond2
Ethernet Channel Bonding Driver: v3.7.1 (April 27, 2011)

Bonding Mode: IEEE 802.3ad Dynamic link aggregation
Transmit Hash Policy: layer3+4 (1)
MII Status: up
MII Polling Interval (ms): 100
Up Delay (ms): 0
Down Delay (ms): 0

```

```
802.3ad info
LACP rate: fast
Min links: 0
Aggregator selection policy (ad_select): stable
Active Aggregator Info:
    Aggregator ID: 3
    Number of ports: 4
    Actor Key: 33
    Partner Key: 33
    Partner Mac Address: 44:38:39:00:32:cf
```

```
Slave Interface: swp4
MII Status: up
Speed: 10000 Mbps
Duplex: full
Link Failure Count: 0
Permanent HW addr: 44:38:39:00:32:8e
Aggregator ID: 3
Slave queue ID: 0
```

```
Slave Interface: swp5
MII Status: up
Speed: 10000 Mbps
Duplex: full
Link Failure Count: 0
Permanent HW addr: 44:38:39:00:32:8f
Aggregator ID: 3
Slave queue ID: 0
```

```
Slave Interface: swp6
MII Status: up
Speed: 10000 Mbps
Duplex: full
Link Failure Count: 0
Permanent HW addr: 44:38:39:00:32:90
Aggregator ID: 3
Slave queue ID: 0
```

```
Slave Interface: swp7
MII Status: up
Speed: 10000 Mbps
Duplex: full
Link Failure Count: 0
Permanent HW addr: 44:38:39:00:32:91
Aggregator ID: 3
```

Slave queue ID: 0



A single bridge cannot contain multiple subinterfaces of the **same** port as members. Attempting to apply such a configuration will result in an error:

```
cumulus@switch:~$ sudo brctl addbr another_bridge
cumulus@switch:~$ sudo brctl addif another_bridge swp9 swp9.100
bridge cannot contain multiple subinterfaces of the same port: swp9,
swp9.100
```

VLAN Translation

By default, Cumulus RMP does not allow VLAN subinterfaces associated with different VLAN IDs to be part of the same bridge. Base interfaces are not explicitly associated with any VLAN IDs and are exempt from this restriction:

```
cumulus@switch:~$ sudo brctl addbr br_mix

cumulus@switch:~$ sudo ip link add link swp10 name swp10.100 type vlan id
100
cumulus@switch:~$ sudo ip link add link swp11 name swp11.200 type vlan id
200

cumulus@switch:~$ sudo brctl addif br_mix swp10.100 swp11.200
can't add swp11.200 to bridge br_mix: Invalid argument
```

In some cases, it may be useful to relax this restriction. For example, two servers may be connected to the switch using VLAN trunks, but the VLAN numbering provisioned on the two servers are not consistent. You can choose to just bridge two VLAN subinterfaces of different VLAN IDs from the servers. You do this by enabling the `sysctl net.bridge.bridge-allow-multiple-vlans`. Packets entering a bridge from a member VLAN subinterface will egress another member VLAN subinterface with the VLAN ID translated.



A bridge in **VLAN-aware mode** (see page 148) cannot have VLAN translation enabled for it; only bridges configured in traditional mode can utilize VLAN translation.

The following example enables the VLAN translation `sysctl`:

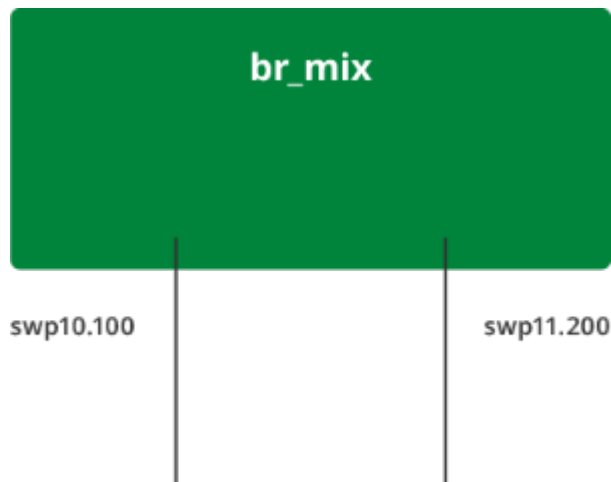
```
cumulus@switch:~$ echo net.bridge.bridge-allow-multiple-vlans = 1 | sudo
tee /etc/sysctl.d/multiple_vlans.conf
net.bridge.bridge-allow-multiple-vlans = 1
cumulus@switch:~$ sudo sysctl -p /etc/sysctl.d/multiple_vlans.conf
net.bridge.bridge-allow-multiple-vlans = 1
```

If the `sysctl` is enabled and you want to disable it, run the above example, setting the `sysctl net.bridge.bridge-allow-multiple-vlans` to 0.

Once the `sysctl` is enabled, ports with different VLAN IDs can be added to the same bridge. In the following example, packets entering the bridge `br-mix` from `swp10.100` will be bridged to `swp11.200` with the VLAN ID translated from 100 to 200:

```
cumulus@switch:~$ sudo brctl addif br_mix swp10.100 swp11.200

cumulus@switch:~$ sudo brctl show br_mix
bridge name      bridge id        STP enabled      interfaces
br_mix           8000.4438390032bd  yes              swp10.100
                  swp11.200
```



VLAN-aware Bridge Mode for Large-scale Layer 2 Environments

The Cumulus RMP bridge driver supports two configuration modes, one that is VLAN-aware, and one that follows a more traditional Linux bridge model.

For traditional Linux bridges, the kernel supports VLANs in the form of VLAN subinterfaces. Enabling bridging on multiple VLANs means configuring a bridge for each VLAN and, for each member port on a bridge, creating one or more VLAN subinterfaces out of that port. This mode poses scalability challenges in terms of configuration size as well as boot time and run time state management, when the number of ports times the number of VLANs becomes large.

The VLAN-aware mode in Cumulus RMP implements a configuration model for large-scale L2 environments, with **one single instance** of [Spanning Tree](#) (see page 91). Each physical bridge member port is configured with the list of allowed VLANs as well as its port VLAN ID (either PVID or native VLAN — see below). MAC address learning, filtering and forwarding are *VLAN-aware*. This significantly reduces the configuration size, and eliminates the large overhead of managing the port/VLAN instances as subinterfaces, replacing them with lightweight VLAN bitmaps and state updates.



You can configure both VLAN-aware and traditional mode bridges on the same network in Cumulus RMP; however you should not have more than one VLAN-aware bridge on a given switch.

Contents

(Click to expand)

- [Creating the Bridge](#) (see page 149)
- [Defining VLAN Memberships](#) (see page 149)
- [Configuring Router Interfaces](#) (see page 149)
- [Using the Show Commands](#) (see page 150)
- [Configuring a VLAN-aware Bridge](#) (see page 151)
 - [Example Basic Configuration](#) (see page 152)
 - [Example Configuration with Access Ports and Pruned VLANs](#) (see page 153)
 - [Example Configuration with Bonds](#) (see page 154)
- [Caveats and Errata](#) (see page 157)

Creating the Bridge

You need to configure only one VLAN-aware bridge, and you need to add only physical ports or bonds to the bridge. Use `ifupdown2` to create the configuration.

Defining VLAN Memberships

With the VLAN-aware bridge mode, VLAN membership is defined for each bridge member interface. This includes the allowed VLAN list and the PVID of the interface (that is, native or default VLAN). In the code below, `bond0` and `bond1` are trunk ports with native VLAN of 10 and allowed VLAN list of 1-1000, 1010-1020. `swp5` is an access port with access VLAN of 10.

Configuring Router Interfaces

In case L3 termination of any VLANs is required, you can configure a router interface as a VLAN subinterface of the bridge device itself.

To continue with the previous example, say VLAN 10 and VLAN 1000 are layer 3 routed. You can create the router interfaces by running:

```
cumulus@switch:~$ sudo ip link add link br name br.10 type vlan id 10
cumulus@switch:~$ sudo ip link add link br name br.1000 type vlan id 1000
```

Then you use the `ip addr add` command to assign an IP address to each interface. Note that in order for the bridge to pass routed traffic on these two VLANs, you need to assign the VLANs in the bridge's VLAN list. To do this, run:

```
cumulus@switch:~$ sudo bridge vlan add vid 10 dev br self
cumulus@switch:~$ sudo bridge vlan add vid 1000 dev br self
```

Using the Show Commands

To show all bridge VLANs:

```
cumulus@switch:~$ bridge vlan show
port      vlan ids

bond0      10 PVID Egress Untagged
           1-9
           11-1000
           1010-1020

bond1      10 PVID Egress Untagged
           1-9
           11-1000
           1010-1020

swp5       10 PVID Egress Untagged

br         10
           1000
```

To show membership of a particular VLAN:

```
cumulus@switch:~$ sudo bridge vlan show vlan 10
VLAN 10:
      bond0 bond1 swp5 br0
```

To show MAC addresses, do one of the following:

```
cumulus@switch:~$ sudo brctl showmacs br | grep -v yes
port name mac addr          vlan  is local? ageing timer
bond0      00:e0:ec:25:2f:5b      10    no  39.47
```

```
cumulus@switch:~$ sudo bridge fdb show | grep -v perm
00:e0:ec:25:2f:5b dev bond0 vlan 10 port 0
```

Configuring a VLAN-aware Bridge

To configure a VLAN-aware bridge, include the `bridge-vlan-aware` attribute, setting it to `yes`. Name the bridge `bridge` to help ensure it is the only VLAN-aware bridge on the switch. The following attributes are useful for configuring VLAN-aware bridges:

- `bridge-vlan-aware`: set to `yes` to indicate that the bridge is VLAN-aware.
- `bridge-access`: declares the access port.
- `bridge-pvid`: specifies native VLANs if the ID is other than 1.
- `bridge-vids`: declares the VLANs associated with this bridge.

For a definitive list of bridge attributes, run `ifquery --syntax-help` and look for the entries under **bridge**, **bridgevlan** and **mstpctl**.

A basic configuration for a VLAN-aware bridge configured for STP that contains two switch ports looks like this:

```
auto bridge
iface bridge
    bridge-vlan-aware yes
    bridge-ports swp1 swp2
    bridge-stp on
```

By default, the bridge port inherits the bridge VLANs. You can have a port override the bridge VLANs by specifying port-specific VLANs, using the `bridge-ports` attribute.

As with traditional bridges, the bridge port membership and bridge attributes remain under bridge configuration. But bridge port attributes reside under the ports themselves.

When configuring the VLAN attributes for the bridge, put the layer 2 attributes in a separate stanza using this special VLAN interface: `<bridge>.<vlanid/range>`. You can specify a range of VLANs as well. For example:

```
auto bridge.4094
vlan bridge.4094
    address 172.16.101.100
    hwaddress 44:38:39:ff:00:00
    bridge-igmp-querier-src 172.16.101.1
```

Or:

```
auto bridge.[4094-4096]
vlan bridge.[4094-4096]
    ATTRIBUTE VALUE
```

For switched virtual interface configurations, specify a regular bridge.vlanid device with the `address` attribute:

```
auto bridge.4094
iface bridge.4094
    address <ipaddr>
    hwaddress <mac>
```

VLAN-aware bridges are backwards compatible with traditional bridge configurations.

Example Basic Configuration

The following is a basic example illustrating how to configure a VLAN-aware bridge using `ifupdown2` (see [page 71](#)). Add this persistent configuration to `/etc/network/interfaces`.

Note the attributes used in the stanza:

- The `bridge-vlan-aware` is set to `yes`, indicating the bridge is VLAN-aware.
- The `glob` keyword referenced in the `bridge-ports` attribute indicates that `swp1` through `swp52` are part of the bridge, instead of enumerating them one by one.
- STP (see [page 91](#)) is enabled on the bridge.
- The `bridge-vids` attribute declares the VLANs associated with the bridge.

```
#
# vlan-aware bridge simple example
#
# 'bridge' is a vlan aware bridge with all ports (swp1-52).
# native vlan is by default 1
#
# 'bridge-vids' attribute is used to declare vlans.
# 'bridge-pvid' attribute is used to specify native vlans if other than 1
# 'bridge-access' attribute is used to declare access port
#
#
# ports swp1-swp52 are trunk ports which inherit vlans from 'bridge'
# ie vlans 310 700 707 712 850 910
```



```
#
# the following is a vlan aware bridge with ports swp1-swp52
# It has stp on
#
auto bridge
iface bridge
    bridge-vlan-aware yes
    bridge-ports glob swp1-52
    bridge-stp on
    bridge-vids 310 700 707 712 850 910
```

Example Configuration with Access Ports and Pruned VLANs

The following example contains an access port and a switch port that is *pruned*; that is, it only sends and receives traffic tagged to and from a specific set of VLANs declared by the `bridge-vids` attribute. It also contains other switch ports that send and receive traffic from all the defined VLANs.

```
#
# vlan-aware bridge access ports and pruned vlan example
#
# 'bridge' is a vlan aware bridge with all ports (swp1-52).
# native vlan is by default 1
#
# 'bridge-vids' attribute is used to declare vlans.
# 'bridge-pvid' attribute is used to specify native vlans if other than 1
# 'bridge-access' attribute is used to declare access port
#
#

# The following is an access port to vlan 310, no trunking
auto swp1
iface swp1
    bridge-access 310
    mstpctl-portadmindedge yes
    mstpctl-bpduguard yes

# The following is a trunk port that is "pruned".
# native vlan is 1, but only .1q tags of 707, 712, 850 are
# sent and received
#
auto swp2
iface swp2
    bridge-vids 707 712 850
    mstpctl-portadmindedge yes
```

```

mstpctl-bpduguard yes

# The following port is the trunk uplink and inherits all vlans
# from 'bridge'; bridge assurance is enabled using 'portnetwork' attribute
auto swp49
iface swp49
    mstpctl-portpathcost 10
    mstpctl-portnetwork yes

# The following port is the trunk uplink and inherits all vlans
# from 'bridge'; bridge assurance is enabled using 'portnetwork' attribute
auto swp50
iface swp50
    mstpctl-portpathcost 0
    mstpctl-portnetwork yes

#
# ports swp3-swp48 are trunk ports which inherit vlans from the 'bridge'
# ie vlans 310,700,707,712,850,910

#
# the following is a vlan aware bridge with ports swp1-swp52
# It has stp on
#
auto bridge
iface bridge
    bridge-vlan-aware yes
    bridge-ports glob swp1-52
    bridge-stp on
    bridge-vids 310 700 707 712 850 910

```

Example Configuration with Bonds

This configuration demonstrates a VLAN-aware bridge with a large set of bonds. The bond configurations are generated from a [Mako](#) template.

```

#
# vlan-aware bridge with bonds example
#
# uplink1, peerlink and downlink are bond interfaces.
# 'bridge' is a vlan aware bridge with ports uplink1, peerlink
# and downlink (swp2-20).
#
# native vlan is by default 1

```

```
#
# 'bridge-vids' attribute is used to declare vlans.
# 'bridge-pvid' attribute is used to specify native vlans if other than 1
# 'bridge-access' attribute is used to declare access port
#
auto lo
iface lo

auto eth0
iface eth0 inet dhcp

# bond interface
auto uplink1
iface uplink1
    bond-slaves swp32
    bond-mode 802.3ad
    bond-miimon 100
    bond-use-carrier 1
    bond-lacp-rate 1
    bond-min-links 1
    bond-xmit-hash-policy layer3+4
    bridge-vids 2000-2079

# bond interface
auto peerlink
iface peerlink
    bond-slaves swp30 swp31
    bond-mode 802.3ad
    bond-miimon 100
    bond-use-carrier 1
    bond-lacp-rate 1
    bond-min-links 1
    bond-xmit-hash-policy layer3+4
    bridge-vids 2000-2079 4094

# bond interface
auto downlink
iface downlink
    bond-slaves swp1
    bond-mode 802.3ad
    bond-miimon 100
    bond-use-carrier 1
    bond-lacp-rate 1
    bond-min-links 1
    bond-xmit-hash-policy layer3+4
```

```
bridge-vids 2000-2079

#
# Declare vlans for all swp ports
# swp2-20 get vlans from 2004 to 2022.
# The below uses mako templates to generate iface sections
# with vlans for swp ports
#
%for port, vlanid in zip(range(2, 20), range(2004, 2022)) :
    auto swp${port}
    iface swp${port}
        bridge-vids ${vlanid}

%endfor

# svi vlan 4094
auto bridge.4094
iface bridge.4094
    address 11.100.1.252/24

# 12 attributes for vlan 4094
auto bridge.4094
vlan bridge.4094
    bridge-igmp-querier-src 172.16.101.1

#
# vlan-aware bridge
#
auto bridge
iface bridge
    bridge-vlan-aware yes
    bridge-ports uplink1 peerlink downlink glob swp2-20
    bridge-stp on

# svi peerlink vlan
auto peerlink.4094
iface peerlink.4094
    address 192.168.10.1/30
    broadcast 192.168.10.3
```

Caveats and Errata

- **STP:** Because [Spanning Tree and Rapid Spanning Tree \(see page 91\)](#) (STP) are enabled on a per-bridge basis, VLAN-aware mode essentially supports a single instance of STP across all VLANs. A common practice when using a single STP instance for all VLANs is to define all every VLAN on each switch in the spanning tree instance. `mstpd` continues to be the user space protocol daemon, and Cumulus RMP supports RSTP.
- **Reserved VLAN range:** For hardware data plane internal operations, the switching silicon requires VLANs for every physical port, Linux bridge, and layer 3 subinterface. Cumulus RMP reserves a range of 700 VLANs by default; this range is 3300-3999. In case any of your user-defined VLANs conflict with the default reserved range, you can modify the range, as long as the new range is a contiguous set of VLANs with IDs anywhere between 2 and 4094, and the minimum size of the range is 300 VLANs:
 1. Edit `/etc/cumulus/switchd.conf`, uncomment `resv_vlan_range` and specify the new range.
 2. Restart `switchd` (`sudo service switchd restart`) for the new range to take effect.

 While restarting `switchd`, all running ports will flap and forwarding will be interrupted.
- **VLAN translation:** A bridge in VLAN-aware mode cannot have VLAN translation enabled for it; only bridges configured in [traditional mode \(see page 128\)](#) can utilize VLAN translation.

Routing

This chapter discusses routing on switches running Cumulus RMP.

Contents

(Click to expand)

- [Contents \(see page 158\)](#)
- [Commands \(see page 158\)](#)
- [Configuring Static Routing \(see page 158\)](#)
 - [Persistently Adding a Static Route \(see page 158\)](#)
- [Useful Links \(see page 159\)](#)

Commands

- [ip route](#)

Configuring Static Routing

The `ip route` command allows manipulating the kernel routing table directly from the Linux shell. See `man ip(8)` for details.

To display the routing table:

```
cumulus@switch:~$ ip route show
default via 10.0.1.2 dev eth0
default via 10.42.0.1 dev eth0
10.42.0.0/22 dev eth0 proto kernel scope link src 10.42.0.65
10.168.2.0/24 dev swp2 proto kernel scope link src 10.168.2.1
10.168.26.0/24 dev swp26 proto kernel scope link src 10.168.26.1
```

Persistently Adding a Static Route

A static route can be persistently added by adding up `ip route add ..` into `/etc/network/interfaces`. For example:

```
cumulus@switch:~$ cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5), ifup(8)
#
```

```
# Please see /usr/share/doc/python-ifupdown2/examples/ for examples
#
#

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp

auto swp2
iface swp2 inet static
    address 10.168.2.1/24
    up ip route add 203.0.113.0/24 via 10.168.2.2

auto swp26
iface swp26 inet static
    address 10.168.26.1/24
```

Configuration Files

- /etc/network/interfaces

Useful Links

- <http://linux-ip.net/html/tools-ip-route.html>

Monitoring and Troubleshooting

This chapter introduces monitoring and troubleshooting Cumulus RMP.

Contents

(Click to expand)

- [Contents \(see page 160\)](#)
- [Commands \(see page 160\)](#)
- [Using the Serial Console \(see page 160\)](#)
 - [Configuring the Serial Console on PowerPC Switches \(see page 160\)](#)
 - [Configuring the Serial Console on x86 Switches \(see page 161\)](#)
- [Diagnostics Using cl-support \(see page 162\)](#)
- [Configuration Files \(see page 163\)](#)
- [Next Steps \(see page 163\)](#)

Commands

- [cl-support](#)
- [fw_setenv](#)

Using the Serial Console

The serial console can be a useful tool for debugging issues, especially when you find yourself rebooting the switch often or if you don't have a reliable network connection.

The default serial console baud rate is 115200, which is the baud rate [ONIE](#) uses.

Configuring the Serial Console on PowerPC Switches

On PowerPC switches, the U-Boot environment variable `baudrate` identifies the baud rate of the serial console. To change the `baudrate` variable, use the `fw_setenv` command:

```
cumulus@switch:~$ sudo fw_setenv baudrate 9600
Updating environment variable: `baudrate'
Proceed with update [N/y]? y
```

You must reboot the switch for the `baudrate` change to take effect.

The valid values for `baudrate` are:

- 300

- 600
- 1200
- 2400
- 4800
- 9600
- 19200
- 38400
- 115200

Configuring the Serial Console on x86 Switches

On x86 switches, you configure serial console baud rate by editing `grub`. The valid values for the baud rate are:

- 300
- 600
- 1200
- 2400
- 4800
- 9600
- 19200
- 38400
- 115200

To change the serial console baud rate:

1. Edit `/etc/default/grub`. The two relevant lines in `/etc/default/grub` are as follows; replace the `115200` value with a valid value specified above in the `--speed` variable in the first line and in the console variable in the second line:

```
GRUB_SERIAL_COMMAND="serial --port=0x2f8 --speed=115200 --word=8 --
parity=no --stop=1"
GRUB_CMDLINE_LINUX="console=ttyS1,115200n8
cl_platform=accton_as5712_54x"
```

2. After you save your changes to the grub configuration, type the following at the command prompt:

```
cumulus@switch:~$ update-grub
```

3. If you plan on accessing your switch's BIOS over the serial console, you need to update the baud rate in the switch BIOS. For more information, see [this knowledge base article](#).
4. Reboot the switch.

Diagnostics Using cl-support

You can use `cl-support` to generate a single export file that contains various details and the configuration from a switch. This is useful for remote debugging and troubleshooting.

You should run `cl-support` before you submit a support request to Cumulus Networks as this file helps in the investigation of issues:

```
cumulus@switch:~$ sudo cl-support -h
Usage: cl-support [-h] [-s] [-t] [-v] [reason]...

Args:
[reason]: Optional reason to give for invoking cl-support.
          Saved into tarball's cmdline.args file.

Options:
-h: Print this usage statement
-s: Security sensitive collection
-t: User filename tag
-v: Verbose
-e MODULES: Enable modules. Comma separated module list (run with -e help
for module names)
-d MODULES: Disable modules. Comma separated module list (run with -d help
for module names)
```

Example output:

```
cumulus@switch:~$ ls /var/support
cl_support_20130806_032720.tar.xz
```

The directory structure is compressed using LZMA2 compression and can be extracted using the `unxz` command:

```
cumulus@switch:~$ cd /var/support
cumulus@switch:~$ sudo unxz cl_support_20130729_140040.tar.xz
cumulus@switch:~$ sudo tar xf cl_support_20130729_140040.tar
cumulus@switch:~$ ls -l cl_support_20130729_140040/

-rwxr-xr-x  1 root root 7724 Jul 29 14:00 cl-support
-rw-r--r--  1 root root   52 Jul 29 14:00 cmdline.args
drwxr-xr-x  2 root root 4096 Jul 29 14:00 core
drwxr-xr-x 64 root root 4096 Jul 29 13:51 etc
drwxr-xr-x  4 root root 4096 Jul 29 14:00 proc
```

```
drwxr-xr-x  2 root root 4096 Jul 29 14:01 support
drwxr-xr-x  3 root root 4096 Jul 29 14:00 sys
drwxr-xr-x  3 root root 4096 Aug  8 15:22 var
```

The directory contains the following elements:

Directory	Description
core	Contains the core files generated from Cumulus RMP HAL process, <code>switchd</code> .
etc	Is a replica of the switch's <code>/etc</code> directory. <code>/etc</code> contains all the general Linux configuration files, as well as configurations for the system's network interfaces, <code>jdoe</code> , and other packages.
log	Is a replica of the switch's <code>/var/log</code> directory. Most Cumulus RMP log files are located in this directory. Notable log files include <code>switchd.log</code> and <code>daemon.log</code> log files, and <code>syslog</code> . For more information, read this knowledge base article .
proc	Is a replica of the switch's <code>/proc</code> directory. In Linux, <code>/proc</code> contains runtime system information (like system memory, devices mounted, and hardware configuration). These files are not actual files but the current state of the system.
support	Is a set of files containing further system information, which is obtained by <code>cl-support</code> running commands such as <code>ps -aux</code> , <code>netstat -i</code> , and so forth — even the routing tables.

`cl-support`, when untarred, contains a `reason.txt` file. This file indicates what reason triggered it. When contacting Cumulus Networks technical support, please attach the `cl-support` file if possible. For more information about `cl-support`, please read [Understanding and Decoding the cl-support Output File](#) (see page 180).

Configuration Files

- `/etc/cumulus/switchd.conf`

Next Steps

The links listed under Child Pages below discuss more specific monitoring topics.

Single User Mode - Boot Recovery

Use single user mode to assist in troubleshooting system boot issues or for password recovery. Entering single user mode is [platform-specific](#), so follow the appropriate steps for your x86 or PowerPC switch.

Contents

(Click to expand)

- [Contents \(see page 164\)](#)
- [Entering Single User Mode on a PowerPC Switch \(see page 164\)](#)
- [Entering Single User Mode on an x86 Switch \(see page 164\)](#)

Entering Single User Mode on a PowerPC Switch

1. From the console, boot the switch, interrupting the U-Boot countdown to enter the U-Boot prompt. Enter the following:

```
=> setenv lbootargs init=/bin/sh
=> boot
```

2. After the system boots, the shell command prompt appears. In this mode, you can change the root password or test a boot service that is hanging the boot process.
3. Reboot the system.

```
cumulus@switch:~$ sudo reboot -f
Restarting the system.
```

Entering Single User Mode on an x86 Switch

From the console, boot the switch. At the GRUB menu, select the image slot you wish to boot into with a password:

```
GNU GRUB  version 1.99-27+deb7u2
+-----+
|Cumulus RMP 2.5.3-be24dc3-201412021541-build - slot 1          |
|Cumulus RMP 2.5.3-be24dc3-201412021541-build - slot 1 (recovery mode) |
|Cumulus RMP 2.5.3-b1bb3b7-201412090640-build - slot 2          |
|Cumulus RMP 2.5.3-b1bb3b7-201412090640-build - slot 2 (recovery mode) |
|ONIE                                                         |
+-----+
```

In this example, you are selecting the slot2 image. Under the `linux` option, add `init=/bin/bash`:

```
GNU GRUB  version 1.99-27+deb7u2
+-----+
```

```
| insmod part_gpt
|^
| insmod ext2
| set root='(hd0,gpt3)'
| search --no-floppy --fs-uuid --set=root c42be287-5321-4e77-975f-54e237a\
| d72b0
| echo 'Loading Linux ...'
| linux /cl-vmlinuz-3.2.60-1+deb7u1+cl2.5-slot-2 root=UUID=f01a2d40-d2fe-\
| 435b-b3d1-7edc1eb0c42f console=ttyS0,115200n8 cl_platform=dell_s6000_s1\
| 220 quiet active=2 init=/bin/bash
| echo 'Loading initial ramdisk ...' A
| initrd /cl-initrd.img-3.2.60-1+deb7u1+cl2.5-slot-2
|
|
+-----+
```

Type Ctrl+x or F10 to boot with this change.

When you are done making changes as a single user, run `reboot -f` to boot the switch back to a normal state:

```
Begin: Running /scripts/init-bottom ... done.
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
cumulus@switch:/# sudo reboot -f
```

Monitoring Interfaces and Transceivers Using ethtool

The `ethtool` command enables you to query or control the network driver and hardware settings. It takes the device name (like `swp1`) as an argument. When the device name is the only argument to `ethtool`, it prints the current settings of the network device. See `man ethtool(8)` for details. Not all options are currently supported on switch port interfaces.

Contents

(Click to expand)

- [Contents \(see page 165\)](#)
- [Commands \(see page 166\)](#)
- [Monitoring Interfaces Using ethtool \(see page 166\)](#)
 - [Viewing and Clearing Interface Counters \(see page 167\)](#)
- [Monitoring Switch Port SFP/QSFP Using ethtool \(see page 168\)](#)

Commands

- `cl-netstat`
- `ethtool`

Monitoring Interfaces Using *ethtool*

To check the status of an interface using `ethtool`:

```
cumulus@switch:~$ ethtool swp1
Settings for swp1:
    Supported ports: [ TP ]
    Supported link modes:   10baseT/Half 10baseT/Full
                           100baseT/Half 100baseT/Full
                           1000baseT/Full
    Supported pause frame use: Symmetric Receive-only
    Supports auto-negotiation: Yes
    Advertised link modes:  10baseT/Half 10baseT/Full
                           100baseT/Half 100baseT/Full
                           1000baseT/Half 1000baseT/Full
    Advertised pause frame use: Symmetric
    Advertised auto-negotiation: No
    Speed: 1000Mb/s
    Duplex: Full
    Port: FIBRE
    PHYAD: 0
    Transceiver: external
    Auto-negotiation: on
    Current message level: 0x00000000 (0)

    Link detected: yes
```

To query interface statistics:

```
cumulus@switch:~$ sudo ethtool -S swp1
NIC statistics:
    HwIfInOctets: 1435339
    HwIfInUcastPkts: 11795
    HwIfInBcastPkts: 3
    HwIfInMcastPkts: 4578
    HwIfOutOctets: 14866246
    HwIfOutUcastPkts: 11791
```

```

HwIfOutMcastPkts: 136493
HwIfOutBcastPkts: 0
HwIfInDiscards: 0
HwIfInL3Drops: 0
HwIfInBufferDrops: 0
HwIfInAclDrops: 28
HwIfInDot3LengthErrors: 0
HwIfInErrors: 0
SoftInErrors: 0
SoftInDrops: 0
SoftInFrameErrors: 0
HwIfOutDiscards: 0
HwIfOutErrors: 0
HwIfOutQDrops: 0
HwIfOutNonQDrops: 0
SoftOutErrors: 0
SoftOutDrops: 0
SoftOutTxFifoFull: 0
HwIfOutQLen: 0

```

Viewing and Clearing Interface Counters

Interface counters contain information about an interface. You can view this information when you run `cl-netstat`, `ifconfig`, or `cat /proc/net/dev`. You can also use `cl-netstat` to save or clear this information:

```

cumulus@switch:~# sudo cl-netstat
Kernel Interface table

```

Iface	MTU	Met	RX_OK	RX_ERR	RX_DRP	RX_OVR	TX_OK
TX_ERR	TX_DRP	TX_OVR	Flg				
eth0	1500	0	8391	0	0	0	
9694	0	0		0	BMRU		
lo	16436	0	1693	0	0	0	
1693	0	0		0	LRU		
swp1	1500	0	11914	0	8948	0	
20854	0	9338		0	BMRU		
swp2	1500	0	20734	0	17969	0	
12033	0	13142		0	BMRU		

```

cumulus@switch:~# sudo :~# cl-netstat -c
Cleared counters

```

Option	Description
-c	Copies and clears statistics. It does not clear counters in the kernel or hardware.
-d	Deletes saved statistics, either the <code>uid</code> or the specified tag.
-D	Deletes all saved statistics.
-j	Display in JSON format.
-l	Lists saved tags.
-r	Displays raw statistics (unmodified output of <code>cl-netstat</code>).
-t <tag name>	Saves statistics with <tag name>.
-v	Prints <code>cl-netstat</code> version and exits.

Monitoring Switch Port SFP/QSFP Using ethtool

The `ethtool -m` command provides switch port SFP information. It shows connector information, vendor data, and more:

```
cumulus@switch:~$ sudo ethtool -m swp49
Identifier                               : 0xff (reserved or
unknown)
Optical diagnostics support              : Yes
Laser bias current                       : 130.046 mA
Laser output power                       : 6.5025 mW / 8.13 dBm
Receiver signal average optical power    : 6.5535 mW / 8.16 dBm
Module temperature                       : 0.00 degrees C / 32.00
degrees F
Module voltage                           : 6.5282 V
Alarm/warning flags implemented          : Yes
Laser bias current high alarm            : On
Laser bias current low alarm             : On
Laser bias current high warning          : On
Laser bias current low warning           : On
Laser output power high alarm            : On
Laser output power low alarm             : On
Laser output power high warning          : On
Laser output power low warning           : On
Module temperature high alarm            : On
```



```

Module temperature low alarm           : On
Module temperature high warning        : On
Module temperature low warning         : On
Module voltage high alarm              : On
Module voltage low alarm               : On
Module voltage high warning            : On
Module voltage low warning             : On
Laser rx power high alarm              : On
Laser rx power low alarm               : On
Laser rx power high warning            : On
Laser rx power low warning             : On
Laser bias current high alarm threshold : 130.046 mA
Laser bias current low alarm threshold : 130.046 mA
Laser bias current high warning threshold : 130.046 mA
Laser bias current low warning threshold : 130.046 mA
Laser output power high alarm threshold : 6.5025 mW / 8.13 dBm
Laser output power low alarm threshold : 6.5025 mW / 8.13 dBm
Laser output power high warning threshold : 6.5025 mW / 8.13 dBm
Laser output power low warning threshold : 6.5025 mW / 8.13 dBm
Module temperature high alarm threshold : -1.00 degrees C / 30.20
degrees F
Module temperature low alarm threshold : 0.00 degrees C / 32.00
degrees F
Module temperature high warning threshold : 0.00 degrees C / 32.00
degrees F
Module temperature low warning threshold : 0.00 degrees C / 32.00
degrees F
Module voltage high alarm threshold     : 6.5282 V
Module voltage low alarm threshold      : 6.5282 V
Module voltage high warning threshold   : 6.5282 V
Module voltage low warning threshold    : 6.5282 V
Laser rx power high alarm threshold     : 6.5535 mW / 8.16 dBm
Laser rx power low alarm threshold      : 6.5535 mW / 8.16 dBm
Laser rx power high warning threshold   : 6.5535 mW / 8.16 dBm
Laser rx power low warning threshold    : 6.5535 mW / 8.16 dBm

```

Resource Diagnostics Using cl-resource-query

You can use `cl-resource-query` to retrieve information about host entries, MAC entries, L2 and L3 routes, and ingress and degrees ACL counters and entries that are in use. This is especially useful because Cumulus RMP syncs routes between the kernel and the switching silicon. If the required resource pools in hardware fill up, new kernel routes can cause existing routes to move from being fully allocated to being partially allocated.

In order to avoid this, routes in the hardware should be monitored and kept below the ASIC limits. For example on a Cumulus RMP system, the limits are as follows:

```

routes: 8092 <<<< if all routes are IPv6, or 16384 if all routes are IPv4
long mask routes 2048 <<<< these are routes with a mask longer than the
route mask limit
route mask limit 64
host_routes: 8192
ecmp_nhs: 16346
ecmp_nhs_per_route: 52

```

You can monitor this in Cumulus RMP with the `cl-resource-query` command.

```

cumulus@switch:~$ sudo cl-resource-query
Host entries:                5,    0% of maximum value    2048
IPv4 neighbors:              1
IPv6 neighbors:              2
IPv4/IPv6 entries:          13,    3% of maximum value    412
Long IPv6 entries:           3,    1% of maximum value    256
IPv4 Routes:                 9
IPv6 Routes:                 5
Total Routes:                14,    0% of maximum value   32768
ECMP nexthops:               0,    0% of maximum value     1
MAC entries:                 0,    0% of maximum value   16384
Ingress ACL entries:         463,   22% of maximum value   2048
Ingress ACL counters:        56,    2% of maximum value   2048
Ingress ACL meters:          10,    0% of maximum value   2048
Ingress ACL slices:          3,   37% of maximum value     8
Egress ACL entries:          36,    7% of maximum value   512
Egress ACL counters:         36,    7% of maximum value   512
Egress ACL meters:           18,    3% of maximum value   512
Egress ACL slices:           2,   50% of maximum value     4

```

Monitoring System Hardware

You monitor system hardware in these ways, using:

- decode-syseeprom
- sensors
- smond
- Net-SNMP

- [watchdog](#)

Contents

(Click to expand)

- [Contents \(see page 171\)](#)
- [Commands \(see page 171\)](#)
- [Monitoring Hardware Using decode-syseeprom \(see page 171\)](#)
 - [Command Options \(see page 172\)](#)
 - [Related Commands \(see page 172\)](#)
- [Monitoring Hardware Using sensors \(see page 173\)](#)
 - [Command Options \(see page 173\)](#)
- [Monitoring Switch Hardware Using SNMP \(see page 174\)](#)
 - [Public Community Disabled \(see page 175\)](#)
- [Monitoring System Units Using smond \(see page 176\)](#)
 - [Command Options \(see page 176\)](#)
- [Keeping the Switch Alive Using the Hardware Watchdog \(see page 176\)](#)
- [Configuration Files \(see page 177\)](#)
- [Useful Links \(see page 177\)](#)

Commands

- [decode-syseeprom](#)
- [dmidecode](#)
- [lshw](#)
- [sensors](#)
- [smond](#)

Monitoring Hardware Using decode-syseeprom

The `decode-syseeprom` command enables you to retrieve information about the switch's EEPROM. If the EEPROM is writable, you can set values on the EEPROM.

For example:

```
cumulus@switch:~# decode-syseeprom
TlvInfo Header:
  Id String:      TlvInfo
  Version:        1
  Total Length: 159
TLV Name          Code Len Value
-----
Product Name      0x21   6 Pebble
```

```

Part Number      0x22  14  R0854-G0008-02
Serial Number    0x23  19  D2070023918PE000012
Manufacture Date 0x25  19  01/15/2015 14:30:00
Device Version   0x26   1  2
Label Revision   0x27   6  Pebble
Platform Name    0x28   6  Pebble
MAC Addresses    0x2A   2  73
Manufacturer     0x2B   9  CELESTICA
Manufacture Country 0x2C   3  CHN
Vendor Name      0x2D   9  CELESTICA
Diag Version     0x2E   5  1.0.0
Service Tag      0x2F   2  LB
Vendor Extension 0xFD   1  0x62
Base MAC Address 0x24   6  44:38:39:00:89:DD
ONIE Version     0x29  13  2014.11.0.0.2
CRC-32          0xFE   4  0x19AFD83A
(checksum valid)

```

Command Options

Usage: /usr/cumulus/bin/decode-syseeprom [-a][-r][-s [args]][-t]

Option	Description
-h, --help	Displays the help message and exits.
-a	Prints the base MAC address for switch interfaces.
-r	Prints the number of MACs allocated for switch interfaces.
-s	Sets the EEPROM content if the EEPROM is writable. args can be supplied in command line in a comma separated list of the form ' <field>=<value> , ...'. ' ', ',' and '=' are illegal characters in field names and values. Fields that are not specified will default to their current values. If args are supplied in the command line, they will be written without confirmation. If args is empty, the values will be prompted interactively.
-t TARGET	Selects the target EEPROM (board , psu2 , psu1) for the read or write operation; default is board .
-e, --serial	Prints the device serial number.
-m	Prints the base MAC address for management interfaces.

Related Commands

You can also use the `dmidecode` command to retrieve hardware configuration information that's been populated in the BIOS.

You can use `apt-get` to install the `lshw` program on the switch, which also retrieves hardware configuration information.

Monitoring Hardware Using sensors

The `sensors` command provides a method for monitoring the health of your switch hardware, such as power, temperature and fan speeds. This command executes `lm-sensors`.

For example:

```
cumulus@switch:~$ sensors
coretemp-isa-0000
Adapter: ISA adapter
Core 0:          +32.0 C  (high = +110.0 C, crit = +110.0 C)
Core 2:          +35.0 C  (high = +110.0 C, crit = +110.0 C)

lm75-i2c-0-4a
Adapter: SMBus I801 adapter at e000
temp1:           +27.0 C  (high = +60.0 C, hyst = +25.0 C)
```

Command Options

Usage: `sensors` [OPTION]... [CHIP]...

Option	Description
-c, -config-file	Specify a config file; use - after -c to read the config file from <code>stdin</code> ; by default, <code>sensors</code> references the configuration file in <code>/etc/sensors.d/</code> .
-s, -set	Executes set statements in the config file (root only); <code>sensors -s</code> is run once at boot time and applies all the settings to the boot drivers.
-f, -fahrenheit	Show temperatures in degrees Fahrenheit.
-A, -no-adapter	Do not show the adapter for each chip.
-bus-list	Generate bus statements for <code>sensors.conf</code> .

If [CHIP] is not specified in the command, all chip info will be printed. Example chip names include:

- `lm78-i2c-0-2d *-i2c-0-2d`

- `lm78-i2c-0-* *-i2c-0-*`
- `lm78-i2c-* -2d *-i2c-* -2d`
- `lm78-i2c-*-* *-i2c-*-*`
- `lm78-isa-0290 *-isa-0290`
- `lm78-isa-* *-isa-*`
- `lm78-*`

Monitoring Switch Hardware Using SNMP

Cumulus RMP ships with Net-SNMP v5.4.3. However, it is disabled by default. To enable Net-SNMP, use `jdoo`, which is the fork of `monit` version 5.2.5.



`jdoo` and `monit` are mutually exclusive, so the `monit` package is not installed on Cumulus RMP. If you would prefer to use `monit`, it will uninstall `jdoo` from Cumulus RMP. However, Cumulus Networks will not provide support for issues with `monit`.

1. Edit `/etc/default/snmpd` and verify that `SNMPDRUN=yes`.
2. In order to use `jdoo` on SNMPD, you need to add a configuration like the following to your `/etc/jdoo/jdoo.rc` file:

```
check process snmpd with pidfile /var/run/snmpd.pid
    every 6 cycles
    group networking
    start program = "/etc/init.d/snmpd start"
    stop program = "/etc/init.d/snmpd stop"
```

3. Then reload `jdoo`:

```
# sudo jdoo reload
```

4. Start `snmpd`:

```
# sudo jdoo start snmpd
```

5. Optionally, if you don't want to monitor SNMPD, you can just start it natively:

```
# service snmpd start
```

Once enabled, you can use SNMP to manage various components on the switch. The supported MIBs include many publicly used MIBs as well as some MIBs developed by Cumulus Networks for Cumulus RMP:

- [SNMP-FRAMEWORK](#)
- [SNMP-MPD](#)
- [SNMP-USER-BASED-SM](#)
- [SNMP-VIEW-BASED-ACM](#)
- [SNMPv2](#)
- [IP \(includes ICMP\)](#)
- [TCP](#)
- [UDP](#)
- [UCD-SNMP](#) (For information on exposing CPU and memory information via SNMP, see this [knowledge base article](#).)
- [IF-MIB](#)
- [LLDP](#)
- [LM-SENSORS MIB](#)
- [NET-SNMP-EXTEND-MIB](#) (See also [this knowledge base article](#) on extending NET-SNMP in Cumulus RMP to include data from power supplies, fans and temperature sensors.)
- Resource utilization: Cumulus RMP includes its own resource utilization MIB, which is similar to using `cl-resource-query`. It monitors L3 entries by host, route, nexthops, and L2 MAC/BDPU entries. The MIB is defined in `/usr/share/snmp/Cumulus-Resource-Query-MIB.txt`.
- Discard counters: Cumulus RMP also includes its own counters MIB, defined in `/usr/share/snmp/Cumulus-Counters-MIB.txt`.
- The overall Cumulus RMP MIB is defined in `/usr/share/snmp/Cumulus-Snmp-MIB.txt`.

Public Community Disabled

Public community is disabled by default in Cumulus RMP. While it is disabled, `/etc/snmp/snmpd.conf` will have its public community entry commented out, like this:

```
#rocommunity public default -V systemonly
```

If the comment is removed, an agent can query the switch with this:

```
rocommunity public default -V systemonly
```

After you make any change to `snmpd.conf`, you must restart `snmpd` using `service snmpd restart` for the new configuration to take effect.

To define the desired community configuration, use:

```
rocommunity <any community> default -V systemonly
```

Monitoring System Units Using smond

The `smond` daemon monitors these system units: power, board, temp, fan and volt. It updates their corresponding LEDs, and logs the change in the state. Changes in system unit state are detected via the `cp1d` registers. `smond` utilizes these registers to read all sources, which impacts the health of the system unit, determines the unit's health, and updates the system LEDs.

Use `smonctl` to display sensor information for the various system units:

```
cumulus@switch:~$ smonctl
Board                               : OK
Fan                                 : OK
PSU1                               : OK
PSU2                               : BAD
Temp1   (Networking ASIC Die Temp Sensor) : OK
Temp10  (Right side of the board)         : OK
Temp2   (Near the CPU (Right))            : OK
Temp3   (Top right corner)                : OK
Temp4   (Right side of Networking ASIC)    : OK
Temp5   (Middle of the board)              : OK
Temp6   (P2020 CPU die sensor)            : OK
Temp7   (Left side of the board)          : OK
Temp8   (Left side of the board)          : OK
Temp9   (Right side of the board)         : OK
```

Command Options

Usage: `smonctl` [OPTION]... [CHIP]...

Option	Description
<code>-j, --json</code>	Generates JSON output.
<code>-s SENSOR, --sensor SENSOR</code>	Displays data for the specified sensor.
<code>-v, --verbose</code>	Displays detailed hardware sensors data.

For more information, read `man smond` and `man smonctl`.

Keeping the Switch Alive Using the Hardware Watchdog

Cumulus RMP includes a simplified version of the `wd_keepalive(8)` daemon from the standard Debian package `watchdog`. `wd_keepalive` writes to a file called `/dev/watchdog` periodically to keep the switch from resetting, at least once per minute. Each write delays the reboot time by another minute. After one minute of inactivity where `wd_keepalive` doesn't write to `/dev/watchdog`, the switch resets itself.

The watchdog is enabled by default on QuantaMesh BMS T1048-LB9 switches only; you must enable the watchdog on all other switch platforms. When enabled, it starts when you boot the switch, before `switchd` starts.

To enable the hardware watchdog, edit the `/etc/watchdog.d/<your_platform>` file and set `run_watchdog` to `1`:

```
run_watchdog=1
```

To disable the watchdog, edit the `/etc/watchdog.d/<your_platform>` file and set `run_watchdog` to `0`:

```
run_watchdog=0
```

Then stop the daemon:

```
cumulus@switch:~$ sudo service wd_keepalive stop
```

You can modify the settings for the watchdog — like the timeout setting and scheduler priority — in its configuration file, `/etc/watchdog.conf`.

Configuration Files

- `/etc/cumulus/switchd.conf`
- `/etc/cumulus/sysledcontrol.conf`
- `/etc/sensors.d/<switch>.conf` - sensor configuration file (do **not** edit it!)
- `/etc/watchdog.conf`

Useful Links

- <http://packages.debian.org/search?keywords=lshw>
- <http://lm-sensors.org>
- Net-SNMP tutorials

Monitoring System Statistics and Network Traffic with sFlow

sFlow is a monitoring protocol that samples network packets, application operations, and system counters. sFlow enables you to monitor your network traffic as well as your switch state and performance metrics. An outside server, known as an *sFlow collector*, is required to collect and analyze this data.

`hsflowd` is the daemon that samples and sends sFlow data to configured collectors. `hsflowd` is not included in the base Cumulus RMP installation. After installation, `hsflowd` will automatically start when the switch boots up.

Contents

(Click to expand)

- [Contents \(see page 178\)](#)
- [Installing hsflowd \(see page 178\)](#)
- [Configuring sFlow \(see page 178\)](#)
 - [Configuring sFlow via DNS-SD \(see page 178\)](#)
 - [Manually Configuring /etc/hsflowd.conf \(see page 179\)](#)
- [Configuring sFlow Visualization Tools \(see page 180\)](#)
- [Configuration Files \(see page 180\)](#)
- [Useful Links \(see page 180\)](#)

Installing hsflowd

To download and install the `hsflowd` package, use `apt-get`:

```
cumulus@switch:~$ sudo apt-get update
cumulus@switch:~$ sudo apt-get install -y hsflowd
```

Configuring sFlow

You can configure `hsflowd` to send to the designated collectors via two methods:

- [DNS service discovery \(DNS-SD\)](#)
- [Manually configuring /etc/hsflowd.conf](#)

Configuring sFlow via DNS-SD

With this method, you need to configure your DNS zone to advertise the collectors and polling information to all interested clients. Add the following content to the zone file on your DNS server:

```
_sflow._udp SRV 0 0 6343 collector1
_sflow._udp SRV 0 0 6344 collector2
_sflow._udp TXT (
    "txtvers=1"
    "sampling.1G=2048"
    "sampling.10G=4096"
    "polling=20"
)
```

The above snippet instructs `hsflowd` to send sFlow data to collector1 on port 6343 and to collector2 on port 6344. `hsflowd` will poll counters every 20 seconds and sample 1 out of every 2048 packets.

After the initial configuration is ready, bring up the sFlow daemon by running:

```
cumulus@switch:~$ sudo service hsflowd start
```

No additional configuration is required in `/etc/hsflowd.conf`.

Manually Configuring `/etc/hsflowd.conf`

With this method you will set up the collectors and variables on each switch.

Edit `/etc/hsflowd.conf` and change `DNSSD = on` to `DNSSD = off`:

```
DNSSD = off
```

Then set up your collectors and sampling rates in `/etc/hsflowd.conf`:

```
# Manual Configuration (requires DNSSD=off above)
#####

# Typical configuration is to send every 30 seconds
polling = 20

sampling.1G=2048
sampling.10G=4096

collector {
    ip = 192.0.2.100
    udpport = 6343
}

collector {
    ip = 192.0.2.200
    udpport = 6344
}
```

This configuration polls the counters every 20 seconds, samples 1 of every 2048 packets and sends this information to a collector at 192.0.2.100 on port 6343 and to another collector at 192.0.2.200 on port 6344.



Some collectors require each source to transmit on a different port, others may listen on only one port. Please refer to the documentation for your collector for more information.

Configuring sFlow Visualization Tools

For information on configuring various sFlow visualization tools, read this [Help Center article](#).

Configuration Files

- `/etc/hsflowd.conf`

Useful Links

- [sFlow Collectors](#)
- [sFlow Wikipedia page](#)

Understanding and Decoding the cl-support Output File

The `cl-support` command generates a tar archive of useful information for troubleshooting that can be auto-generated or manually created. To manually create it, run the `cl-support` command. The `cl-support` file is automatically generated when:

- There is a [core file dump](#) of any application (not specific to Cumulus RMP, but something all Linux distributions support)
- Memory usage surpasses 90% of the total system memory (memory usage > 90% for 1 cycle)
- The `loadavg` over 15 minutes has on average greater than 2 (`loadavg (15min) > 2`)

All of these conditions are triggered by `monit`, located at `/etc/monit/monitrc`.

The Cumulus Networks support team may request you submit the output from `cl-support` to help with the investigation of issues you might experience with Cumulus RMP.

```
cumulus@switch:~$ sudo cl-support -h
Usage: cl-support [-h] [reason]...
Args:
[reason]: Optional reason to give for invoking cl-support.
          Saved into tarball's reason.txt file.
Options:
-h: Print this usage statement
```

Example output:

```
cumulus@switch:~$ ls /var/support
cl_support__switch_20141204_203833
```

(Click to expand)

- The `cl-support` command generates a tar archive of useful information for troubleshooting that can be auto-generated or manually created. To manually create it, run the `cl-support` command. The `cl-support` file is automatically generated when: (see page 180)
- Understanding the File Naming Scheme (see page 181)
- Decoding the Output (see page 181)

Understanding the File Naming Scheme

The `cl-support` command generates a file under `/var/support` with the following naming scheme. The following example describes the file called `cl_support__switch_20141204_203833.tar.xz`.

cl_support	switch	20141204	203833
This is always prepended to the <code>tar.gz</code> output.	This is the hostname of the switch where <code>cl-support</code> was executed.	The date in year, month, day; so 20141204 is December, 4th, 2014.	The time in hours, minutes, seconds; so 203833 is 20, 38, 33 (20:38:33) or the equivalent to 8:38:33 PM.

Decoding the Output

Decoding a `cl_support` file is a simple process performed using the `tar` command. The following example illustrates extracting the `cl_support` file:

```
tar -xf cl_support__switch_20141204_203834.tar.xz
```

The `-xf` options are defined here:

Option	Description
-x	Extracts to disk from the archive.
-f	Reads the archive from the specified file.

```
cumulus@switch:~$ ls -l cl_support__switch_20141204_203834/
```

```
-rwxr-xr-x  1 root root 7724 Jul 29 14:00 cl-support
-rw-r--r--  1 root root   52 Jul 29 14:00 cmdline.args
drwxr-xr-x  2 root root 4096 Jul 29 14:00 core
drwxr-xr-x 64 root root 4096 Jul 29 13:51 etc
drwxr-xr-x  4 root root 4096 Jul 29 14:00 proc
drwxr-xr-x  2 root root 4096 Jul 29 14:01 support
drwxr-xr-x  3 root root 4096 Jul 29 14:00 sys
drwxr-xr-x  3 root root 4096 Aug  8 15:22 var
```

The `c1_support` file, when untarred, contains a `reason.txt` file. This file indicates what reason triggered the event. When contacting Cumulus Networks technical support, please attach the `c1-support` file if possible.

The directory contains the following elements:

Directory	Description
<code>cl-support</code>	This is a copy of the <code>c1-support</code> script that generated the <code>c1_support</code> file. It is copied so Cumulus Networks knows exactly which files were included and which weren't. This helps to fix future <code>c1-support</code> requests in the future.
<code>core</code>	Contains the core files generated from the Cumulus RMP HAL (hardware abstraction layer) process, <code>switchd</code> .
<code>etc</code>	<code>etc</code> is the core system configuration directory. <code>c1-support</code> replicates the switch's <code>/etc</code> directory. <code>/etc</code> contains all the general Linux configuration files, as well as configurations for the system's network interfaces, <code>quagga</code> , <code>monit</code> , and other packages.
<code>var/log</code>	<code>/var</code> is the "variable" subdirectory, where programs record runtime information. System logging, user tracking, caches and other files that system programs create and monitor go into <code>/var</code> . <code>c1-support</code> includes only the <code>log</code> subdirectory of the <code>var</code> system-level directory and replicates the switch's <code>/var/log</code> directory. Most Cumulus RMP log files are located in this directory. Notable log files include <code>switchd.log</code> and <code>daemon.log</code> log files, and <code>syslog</code> . For more information, read this knowledge base article .
<code>proc</code>	<code>proc</code> (short for processes) provides system statistics through a directory-and-file interface. In Linux, <code>/proc</code> contains runtime system information (like system memory, devices mounted, and hardware configuration). <code>c1-support</code> simply replicates the switch's <code>/proc</code> directory to determine the current state of the system.
<code>support</code>	<code>support</code> is not a replica of the Linux file system like the other folders listed above. Instead, it is a set of files containing the output of commands from the command line. Examples include the output of <code>ps -aux</code> , <code>netstat -i</code> , and so forth — even the routing tables are included.

Here is more information on the file structure:

- [Troubleshooting the etc Directory \(see page 185\)](#) — In terms of sheer numbers of files, `/etc` contains the largest number of files to send to Cumulus Networks by far. However, log files could be significantly larger in file size.
- [Troubleshooting Log Files \(see page 183\)](#) — This guide highlights the most important log files to look at. Keep in mind, `c1-support` includes all of the log files.
- [Troubleshooting the support Directory \(see page 195\)](#) — This is an explanation of the `support` directory included in the `c1-support` output.

Troubleshooting Log Files

The only real unique entity for logging on Cumulus RMP compared to any other Linux distribution is `switchd.log`, which logs the HAL (hardware abstraction layer) from hardware like the Broadcom ASIC.

This [guide on NixCraft](#) is amazing for understanding how `/var/log` works. The green highlighted rows below are the most important logs and usually looked at first when debugging.

Log	Description	Why is this important?
<code>/var/log/alternatives.log</code>	Information from the update-alternatives are logged into this log file.	
<code>/var/log/apt</code>	Information the <code>apt</code> utility can send logs here; for example, from <code>apt-get install</code> and <code>apt-get remove</code> .	
<code>/var/log/audit/</code>	Contains log information stored by the Linux audit daemon, <code>auditd</code> .	
<code>/var/log/auth.log</code>	Authentication logs.	
<code>/var/log/boot.log</code>	Contains information that is logged when the system boots.	
<code>/var/log/btmp</code>	<p>This file contains information about failed login attempts. Use the <code>last</code> command to view the <code>btmp</code> file. For example:</p> <pre>last -f /var/log/btmp more</pre>	
<code>/var/log/daemon.log</code>	Contains information logged by the various background daemons that run on the system.	
<code>/var/log/dmesg</code>	Contains kernel ring buffer information. When the system boots up, it prints number of messages on the screen that display information about the hardware devices that the kernel detects during boot process. These messages are available in the kernel ring buffer and whenever a new message arrives, the old message gets overwritten. You can also view the content of this file using the <code>dmesg</code> command.	<code>dmesg</code> is one of the few places to determine hardware errors.
<code>/var/log/dpkg.log</code>	Contains information that is logged when a package is installed or removed using the <code>dpkg</code> command.	

Log	Description	Why is this important?
/var/log/faillog	Contains failed user login attempts. Use the <code>faillog</code> command to display the contents of this file.	
/var/log/fsck/*	The <code>fsck</code> utility is used to check and optionally repair one or more Linux filesystems.	
/var/log/mail.log	Mail server logs.	
/var/log/messages	General messages and system related information.	
/var/log/monit.log	<code>monit</code> is a utility for managing and monitoring processes, files, directories and filesystems on a Unix system.	
/var/log/news/*	The <code>news</code> command keeps you informed of news concerning the system.	
/var/log/ntpstats	Logs for network configuration protocol.	
/var/log/kern.log	Kernel logs.	
/var/log/switchd.log	The HAL log for Cumulus RMP.	This is specific to Cumulus RMP. Any <code>switchd</code> crashes are logged here.
/var/log/syslog	The main system log, which logs everything except auth-related messages.	The primary log; it's easiest to <code>grep</code> this file to see what occurred during a problem.
/var/log/wtmp	Login records file.	
/var/log/yum.log	<code>apt</code> command log file.	

Troubleshooting the etc Directory

The `c1-support` (see page 180) script replicates the `/etc` directory.

Files that `c1-support` deliberately excludes are:

File	Description
<code>/etc/nologin</code>	<code>nologin</code> prevents unprivileged users from logging into the system.
<code>/etc/alternatives</code>	<code>update-alternatives</code> creates, removes, maintains and displays information about the symbolic links comprising the Debian alternatives system.

This is the alphabetical of the output from running `ls -l` on the `/etc` directory structure created by `c1-support`. The green highlighted rows are the ones Cumulus Networks finds most important when troubleshooting problems.

File	Description	Why is this important?
<code>adduser.conf</code>	The file <code>/etc/adduser.conf</code> contains defaults for the programs <code>adduser</code> , <code>addgroup</code> , <code>deluser</code> , and <code>delgroup</code> .	
<code>adjtime</code>	Corrects the time to synchronize the system clock.	
<code>apt</code>	<code>apt</code> (Advanced Package Tool) is the command-line tool for handling packages. This folder contains all the configurations.	<code>apt</code> interactions or unsupported apps can affect machine performance.
<code>audisp</code>	The directory that contains <code>audisp-remote.conf</code> , which is the file that controls the configuration of the audit remote logging subsystem.	
<code>audit</code>	The directory that contains the <code>/etc/audit/auditd.conf</code> , which contains configuration information specific to the audit daemon.	
<code>bash.bashrc</code>	<code>Bash</code> is an sh-compatible command language interpreter that executes commands read from standard input or from a file.	
<code>bash_completion</code>	This points to <code>/usr/share/bash-completion/bash_completion</code> .	
<code>bash_completion.d</code>	This folder contains app-specific code for Bash completion on Cumulus RMP, such as <code>mstpctl</code> .	

File	Description	Why is this important?
bcm.d	Broadcom-specific ASIC file structure (hardware interaction). If there are questions contact the Cumulus Networks Support team. This is unique to Cumulus RMP.	
bindresvport. blacklist	This file contains a list of port numbers between 600 and 1024, which should not be used by <code>bindresvport</code> .	
ca-certificates	The folder for <code>ca-certificates</code> . It is empty by default on Cumulus RMP; see below for more information.	
ca-certificates. conf	Each lines list the pathname of activated CA certificates under <code>/usr/share/ca-certificates</code> .	
calendar	The system-wide default calendar file .	
chef	This is an example of something that is not included by default. In this instance, <code>c1-support</code> included the chef folder for some reason.	This is not installed by default, but this tool could have been installed or configured incorrectly, which is why it's included in the <code>c1-support</code> output.
cron.d	<code>cron</code> is a daemon that executes scheduled commands .	
cron.daily	See above.	
cron.hourly	See above.	
cron.monthly	See above.	
cron.weekly	See above.	
crontab	See above.	
cumulus	This directory contains the following: <ul style="list-style-type: none"> • ACL information, stored in the <code>acl</code> directory. • <code>switchd</code> configuration file, <code>switchd.conf</code>. • <code>qos</code>, which is under the <code>datapath</code> directory. • The routing protocol process priority, <code>nice.conf</code>. 	This folder is specific to Cumulus RMP and does not exist on other Linux platforms. For example, while you can configure <code>iptables</code> , to hardware accelerate rules into the hardware you need to use

File	Description	Why is this important?
	<ul style="list-style-type: none"> The breakout cable configuration, under <code>ports.conf</code>. 	<code>cl-acltool</code> and have the rules under the <code>/etc/cumulus/acl/policy.d/<filename.rules</code>)
<code>debconf.conf</code>	Debconf is a configuration system for Debian packages.	
<code>debian_version</code>	The complete Debian version string.	
<code>debsums-ignore</code>	<code>debsums</code> verifies installed package files against their MD5 checksums. This file identifies the packages to ignore.	
<code>default</code>	This folder contains files with configurable flags for many different applications (most installed by default or added manually). For example, <code>/etc/default/networking</code> has a flag for <code>EXCLUDE_INTERFACES=</code> , which is set to nothing by default, but a user could change it to something like <code>swp3</code> .	
<code>deluser.conf</code>	The file <code>/etc/deluser.conf</code> contains defaults for the programs <code>deluser</code> and <code>delgroup</code> .	
<code>dhcp</code>	This directory contains DHCP-specific information.	
<code>dpkg</code>	The package manager for Debian.	
<code>e2fsck.conf</code>	The configuration file for <code>e2fsck</code> . It controls the default behavior of <code>e2fsck</code> while it checks <code>ext2</code> , <code>ext3</code> or <code>ext4</code> filesystems.	
<code>environment</code>	Utilized by <code>pam_env</code> for setting and unsetting environment variables.	
<code>ethertypes</code>	This file can be used to show readable characters instead of hexadecimal numbers for the protocols. For example, <code>0x0800</code> will be represented by <code>IPv4</code> .	
<code>fstab</code>	Static information about the filesystems.	
<code>fstab.d</code>	The directory that can contain additional <code>fstab</code> information; it is empty by default.	
<code>fw_env.config</code>	Configuration file utilized by U-Boot.	
<code>gai.conf</code>		

File	Description	Why is this important?
	Configuration file for sorting the return information from <code>getaddrinfo</code> .	
<code>groff</code>	The directory containing information for <code>groffer</code> , an application used for displaying <code>Unix man pages</code> .	
<code>group</code>	The <code>/etc/group</code> file is a text file that <code>defines the groups</code> on the system.	
<code>group-</code>	Backup for the <code>/etc/group</code> file.	
<code>gshadow</code>	<code>/etc/gshadow</code> contains the <code>shadowed information for group accounts</code> .	
<code>gshadow-</code>	Backup for the <code>/etc/gshadow</code> file.	
<code>host.conf</code>	<code>Resolver configuration file</code> , which contains options like <code>multi</code> that determines whether <code>/etc/hosts</code> will respond with multiple entries for DNS names.	
<code>hostname</code>	The <code>system host name</code> , such as <code>leaf1</code> , <code>spine1</code> , <code>sw1</code> .	
<code>hosts</code>	The <code>static table lookup</code> for hostnames.	
<code>hosts.allow</code>	The part of the <code>host_access</code> program for controlling a simple access control language. <code>hosts.allow=Access</code> is granted when a daemon/client pair matches an entry.	
<code>hosts.deny</code>	See <code>hosts.allow</code> above, except that access is denied when a daemon/client pair matches an entry.	
<code>init</code>	Default location of the <code>system job configuration files</code> .	
<code>init.d</code>	In order for a service to start when the switch boots, you should add the <code>necessary script</code> to the director here. The differences between <code>init</code> and <code>init.d</code> are explained well here .	
<code>inittab</code>	The format of the <code>inittab</code> file used by the <code>sysv-compatible init</code> process.	
<code>inputrc</code>	The initialization file utilized by <code>readline</code> .	

File	Description	Why is this important?
insserv	This application enables installed system init scripts ; this directory is empty by default.	
insserv.conf	Configuration file for insserv .	
insserv.conf.d	Additional directory for insserv configurations .	
iproute2	Directory containing values for the Linux command line tool ip .	
issue	<code>/etc/issue</code> is a text file that contains a message or system identification to be printed before the login prompt.	
issue.net	Identification file for telnet sessions .	
ld.so.cache	Contains a compiled list of candidate libraries previously found in the augmented library path.	
ld.so.conf	Used by the <code>ldconfig</code> tool, which configures dynamic linker run-time bindings .	
ld.so.conf.d	The directory that contains additional <code>ld.so.conf</code> configuration (see above).	
ldap	The directory containing the ldap.conf configuration file used to set the system-wide default to be applied when running LDAP clients.	
libaudit.conf	Configuration file utilized by get_auditfail_action .	
libnl-3	Directory for the configuration relating to the libnl library , which is the core library for implementing the fundamentals required to use the netlink protocol such as socket handling, message construction and parsing, and sending and receiving of data.	
lldpd.d	Directory containing configuration files whose commands are executed by <code>lldpdcli</code> at startup.	
localtime	Copy of the original data file for <code>/etc/timezone</code> .	
logcheck	Directory containing <code>logcheck.conf</code> and logfiles utilized by the <code>log check</code> program, which scans system logs for interesting lines.	

File	Description	Why is this important?
login.defs	Shadow password suite configuration.	
logrotate.conf	Rotates, compresses and mails system logs.	
logrotate.d	Directory containing additional log rotate configurations.	
lsb-release	Shows the current version of Linux on the system. Run <code>cat /etc/lsb-release</code> for output.	This shows you the version of the operating system you are running; also compare this to the output of <code>c1-img-select</code> .
magic	Used by the <code>file</code> command to determine file type. <code>magic</code> tests check for files with data in particular fixed formats.	
magic.mime	The <code>magic</code> MIME type causes the <code>file</code> command to output MIME type strings rather than the more traditional human readable ones.	
mailcap	The <code>mailcap</code> file is read by the <code>metamail</code> program to determine how to display non-text at the local site.	
mailcap.order	The order of entries in the <code>/etc/mailcap</code> file can be altered by editing the <code>/etc/mailcap.order</code> file.	
manpath.config	The <code>manpath</code> configuration file is used by the manual page utilities to assess users' manpaths at run time, to indicate which manual page hierarchies (manpaths) are to be treated as system hierarchies and to assign them directories to be used for storing cat files.	
mime.types	MIME type description file for cups.	
mke2fs.conf	Configuration file for <code>mke2fs</code> , which is a program that creates an ext, ext3 or ext4 filesystem.	
modprobe.d	Configuration directory for <code>modprobe</code> , which is a utility that can add and remove modules from the Linux kernel.	
modules	The kernel modules to load at boot time.	
monit		

File	Description	Why is this important?
	<code>monit</code> is a utility for monitoring services on a Unix system ; this directory has configuration files beneath it.	
<code>motd</code>	The contents of <code>/etc/motd</code> ("message of the day") are displayed by <code>pam_motd</code> after a successful login but just before it executes the login shell.	
<code>mtab</code>	The programs <code>mount</code> and <code>umount</code> maintain a list of currently mounted filesystems in the <code>/etc/mtab</code> file. If no arguments are given to <code>mount</code> , this list is printed.	
<code>nanorc</code>	The GNU <code>nano</code> <code>rcfile</code> .	
<code>network</code>	Contains the network interface configuration for <code>ifup</code> and <code>ifdown</code> .	The main configuration file is under <code>/etc/network/interfaces</code> . This is where you configure L2 and L3 information for all of your front panel ports (swp interfaces). Settings like MTU, link speed, IP address information, VLANs are all done here.
<code>networks</code>	Network name information.	
<code>nsswitch.conf</code>	System databases and name service switch configuration file.	
<code>ntp.conf</code>	NTP (network time protocol) server configuration file.	
<code>openvswitch</code>	The directory containing the <code>conf.db</code> file , which is used by <code>ovsdb-server</code> .	
<code>openvswitch-vtep</code>	Configuration files used for the VTEP daemon and <code>ovsdb-server</code> .	
<code>opt</code>	Host-specific configuration files for add-on applications installed in <code>/opt</code> .	
<code>os-release</code>	Operating system identification.	
<code>pam.conf</code>	The PAM (pluggable authentication module) configuration file. When a PAM-aware privilege granting application is started, it activates its	

File	Description	Why is this important?
	attachment to the PAM-API. This activation performs a number of tasks, the most important being the reading of the configuration file(s).	
pam.d	Alternate directory to configure PAM (see above).	
passwd	User account information.	
passwd-	Backup file for <code>/etc/passwd</code> .	
perl	Perl is an available scripting language. <code>/etc/perl</code> contains configuration files specific to Perl.	
profile	<code>/etc/profile</code> is utilized by <code>sysprofile</code> , a modular centralized shell configuration.	
profile.d	The directory version of the above, which contains configuration files.	
protocols	The protocols definition file , a plain ASCII file that describes the various DARPAnet protocols that are available from the TCP/IP subsystem.	
ptm.d	The directory containing scripts that are run if PTM (see page 113) passes or fails.	Cumulus RMP-specific folder for PTM (prescriptive topology manager).
python	python is an available scripting language.	
python2.6	The 2.6 version of python.	
python2.7	The 2.7 version of python.	
rc.local	The <code>/etc/rc.local</code> script is used by the system administrator to execute after all the normal system services are started , at the end of the process of switching to a multiuser runlevel. You can use it to start a custom service, for example, a server that's installed in <code>/usr/local</code> . Most installations don't need <code>/etc/rc.local</code> ; it's provided for the minority of cases where it's needed .	
rc0.d	Like <code>rc.local</code> , these scripts are booted by default, but the number of the folder represents the Linux runlevel . This folder 0 represents runlevel 0 (halt the system).	

File	Description	Why is this important?
rc1.d	This is run level 1, which is single-user/minimal mode.	
rc2.d	Runlevels 2 through 5 are multiuser modes. Debian systems (such as Cumulus RMP) come with <code>id=2</code> , which indicates that the default runlevel will be 2 when the multi-user state is entered , and the scripts in <code>/etc/rc2.d/</code> will be run.	
rc3.d	See above.	
rc4.d	See above.	
rc5.d	See above.	
rc6.d	Runlevel 6 is reboot the system.	
rcS.d	S stands for <i>single</i> and is equivalent to rc1.	
resolv.conf	Resolver configuration file , which is where DNS is set (domain, nameserver and search).	You need DNS to reach the Cumulus RMP repository.
rmt	This is not a mistake. The shell script <code>/etc/rmt</code> is provided for compatibility with other Unix-like systems, some of which have utilities that expect to find (and execute) <code>rmt</code> in the <code>/etc</code> directory on remote systems.	
rpc	The <code>rpc</code> file contains human-readable names that can be used in place of RPC program numbers.	
rsyslog.conf	The <code>rsyslog.conf</code> file is the main configuration file for <code>rsyslogd</code> , which logs system messages on *nix systems.	
rsyslog.d	The directory containing additional configuration for <code>rsyslog.conf</code> (see above).	
securetty	This file lists terminals into which the root user can log in .	
security	The <code>/etc/security</code> directory contains security-related configurations files . Whereas PAM concerns itself with the methods used to authenticate any given user, the files under <code>/etc/security</code> are concerned with just what a user can or cannot do. For example,	

File	Description	Why is this important?
	the <code>/etc/security/access.conf</code> file contains a list of which users are allowed to log in and from what host (for example, using telnet). The <code>/etc/security/limits.conf</code> file contains various system limits, such as maximum number of processes.	
selinux	NSA Security-Enhanced Linux .	
sensors.d	The directory from which the sensors program loads its configuration; this is unique for each hardware platform. See also Monitoring System Hardware (see page 170).	
sensors3.conf	The sensors.conf file describes how <code>libsensors</code> , and thus all programs using it, should translate the raw readings from the kernel modules to real-world values.	
services	<code>services</code> is a plain ASCII file providing a mapping between human-readable textual names for internet services and their underlying assigned port numbers and protocol types.	
shadow	shadow is a file that contains the password information for the system's accounts and optional aging information.	
shadow-	The backup for the <code>/etc/shadow</code> file.	
shells	The pathnames of valid login shells .	
skel	The skeleton directory (usually <code>/etc/skel</code>) is used to copy default files and also sets a umask for the creation used by <code>pam_mkhomedir</code> .	
snmp	Interface functions to the SNMP (simple network management protocol) toolkit.	
ssh	The ssh configuration.	
ssl	The OpenSSL ssl library implements the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols. This directory holds certificates and configuration.	

File	Description	Why is this important?
staff-group-for-usr-local	Use <code>cat</code> or <code>more</code> on this file to learn more information, see http://bugs.debian.org/299007 .	
sudoers	The <code>sudoers</code> policy plugin determines a user's <code>sudo</code> privileges.	
sudoers.d	The directory file containing additional <code>sudoers</code> configuration (see above).	
sysctl.conf	Configures <code>kernel parameters at boot</code> .	
sysctl.d	The directory file containing additional configuration (see above).	
systemd	<code>systemd</code> system and service manager.	
terminfo	Terminal capability database.	
timezone	If this file exists, it is read and its contents are used as the <code>time zone name</code> .	
ucf.conf	The update configuration file <code>preserves user changes</code> in configuration files.	
udev	Dynamic device management.	
ufw	Provides both a command line interface and a framework for managing a <code>netfilter firewall</code> .	
vim	Configuration file for command line tool <code>vim</code> .	
wgetrc	Configuration file for command line tool <code>wget</code> .	

Troubleshooting the support Directory

The `support` directory is unique in the fact that it is not a copy of the switch's filesystem. Actually, it is the output from various commands. For example:

File	Equivalent Command	Description
support /ip.addr		This shows you all the interfaces (including swp front panel ports), IP address information, admin state and physical state.

File	Equivalent Command	Description
	<pre>cumulus@sw\$ ip addr show</pre>	

Managing Application Daemons

You manage application daemons in Cumulus RMP in the following ways:

- Identifying active listener ports
- Identifying daemons currently active or stopped
- Identifying boot time state of a specific daemon
- Disabling or enabling a specific daemon

Contents

(Click to expand)

- [Contents \(see page 196\)](#)
- [Identifying Active Listener Ports for IPv4 and IPv6 \(see page 196\)](#)
- [Identifying Daemons Currently Active or Stopped \(see page 197\)](#)
- [Identifying Boot Time State of a Specific Daemon \(see page 197\)](#)
- [Disabling or Enabling a Specific Daemon \(see page 198\)](#)

Identifying Active Listener Ports for IPv4 and IPv6

You can identify the active listener ports under both IPv4 and IPv6 using the `lsof` command:

```
cumulus@switch:~$ sudo lsof -Pnl +M -i4
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
ntpd 1882 104 16u IPv4 3954 0t0 UDP *:123
ntpd 1882 104 18u IPv4 3963 0t0 UDP 127.0.0.1:123
ntpd 1882 104 19u IPv4 3964 0t0 UDP 192.168.8.37:123
snmpd 1987 105 8u IPv4 5423 0t0 UDP *:161
zebra 1993 103 10u IPv4 5151 0t0 TCP 127.0.0.1:2601 (LISTEN)
sshd 2496 0 3u IPv4 5809 0t0 TCP *:22 (LISTEN)
jdoo 2622 0 6u IPv4 6132 0t0 TCP 127.0.0.1:2812 (LISTEN)
sshd 31700 0 3r IPv4 187630 0t0 TCP 192.168.8.37:22->192.168.8.3:50386
```

```
(ESTABLISHED)
```

```
cumulus@switch:~$ sudo lsof -Pnl +M -i6
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
ntpd 1882 104 17u IPv6 3955 0t0 UDP *:123
ntpd 1882 104 20u IPv6 3965 0t0 UDP [::1]:123
ntpd 1882 104 21u IPv6 3966 0t0 UDP [fe80::7272:cfff:fe96:6639]:123
sshd 2496 0 4u IPv6 5811 0t0 TCP *:22 (LISTEN)
```

Identifying Daemons Currently Active or Stopped

To determine which daemons are currently active or stopped, use the `service --status-all` command, then pipe the results to `grep`, using the `-` or `+` operators:

```
cumulus@switch:~$ sudo service --status-all | grep +
[ ? ] aclinit
[ + ] arp_refresh
[ + ] auditd
...

cumulus@switch:~$ sudo service --status-all | grep -
[ - ] isc-dhcp-server
[ - ] openvswitch-vtep
[ - ] ptmd
...
```

Identifying Boot Time State of a Specific Daemon

The `ls` command can provide the boot time state of a daemon. A file link with a name starting with **S** identifies a boot-time-enabled daemon. A file link with a name starting with **K** identifies a disabled daemon.

```
cumulus@switch:~/etc$ sudo ls -l rc*.d | grep <daemon name>
```

For example:

```
cumulus@switch:~/etc$ sudo ls -l rc*.d | grep snmpd
lrwxrwxrwx 1 root root 15 Apr 4 2014 K02snmpd -> ../init.d/snmpd
lrwxrwxrwx 1 root root 15 Apr 4 2014 K02snmpd -> ../init.d/snmpd
lrwxrwxrwx 1 root root 15 Apr 4 2014 S01snmpd -> ../init.d/snmpd
lrwxrwxrwx 1 root root 15 Apr 4 2014 S01snmpd -> ../init.d/snmpd
```

```
lrwxrwxrwx 1 root root 15 Apr 4 2014 S01snmpd -> ../init.d/snmpd
lrwxrwxrwx 1 root root 15 Apr 4 2014 S01snmpd -> ../init.d/snmpd
lrwxrwxrwx 1 root root 15 Apr 4 2014 K02snmpd -> ../init.d/snmpd
```

Disabling or Enabling a Specific Daemon

To enable or disable a specific daemon, run:

```
cumulus@switch:~$ update-rc.d <daemon> disable | enable
```

For example:

```
cumulus@switch:~/etc$ sudo update-rc.d snmpd disable
update-rc.d: using dependency based boot sequencing
insserv: warning: current start runlevel(s) (empty) of script `snmpd'
overrides LSB defaults (2 3 4 5).
insserv: warning: current stop runlevel(s) (0 1 2 3 4 5 6) of script
`snmpd' overrides LSB defaults (0 1 6).
insserv: warning: current start runlevel(s) (empty) of script `snmpd'
overrides LSB defaults (2 3 4 5).
insserv: warning: current stop runlevel(s) (0 1 2 3 4 5 6) of script
`snmpd' overrides LSB defaults (0 1 6).
```

```
cumulus@switch:~/etc$ sudo ls -l rc*.d | grep snmpd
lrwxrwxrwx 1 root root 15 Apr 4 2014 K02snmpd -> ../init.d/snmpd
lrwxrwxrwx 1 root root 15 Apr 4 2014 K02snmpd -> ../init.d/snmpd
lrwxrwxrwx 1 root root 15 Feb 13 17:35 K02snmpd -> ../init.d/snmpd
lrwxrwxrwx 1 root root 15 Feb 13 17:35 K02snmpd -> ../init.d/snmpd
lrwxrwxrwx 1 root root 15 Feb 13 17:35 K02snmpd -> ../init.d/snmpd
lrwxrwxrwx 1 root root 15 Feb 13 17:35 K02snmpd -> ../init.d/snmpd
lrwxrwxrwx 1 root root 15 Apr 4 2014 K02snmpd -> ../init.d/snmpd
```

```
cumulus@switch:~/etc$ sudo update-rc.d snmpd enable
update-rc.d: using dependency based boot sequencing
```

```
cumulus@switch:~/etc$ sudo ls -l rc*.d | grep snmpd
lrwxrwxrwx 1 root root 15 Apr 4 2014 K02snmpd -> ../init.d/snmpd
lrwxrwxrwx 1 root root 15 Apr 4 2014 K02snmpd -> ../init.d/snmpd
```

```
lrwxrwxrwx 1 root root 15 Feb 13 17:35 S01snmpd -> ../init.d/snmpd
lrwxrwxrwx 1 root root 15 Feb 13 17:35 S01snmpd -> ../init.d/snmpd
lrwxrwxrwx 1 root root 15 Feb 13 17:35 S01snmpd -> ../init.d/snmpd
lrwxrwxrwx 1 root root 15 Feb 13 17:35 S01snmpd -> ../init.d/snmpd
lrwxrwxrwx 1 root root 15 Apr 4 2014 K02snmpd -> ../init.d/snmpd
```

Troubleshooting Network Interfaces

The following sections describe various ways you can troubleshoot `ifupdown2`.

Contents

(Click to expand)

- [Contents \(see page 199\)](#)
- [Enabling Logging for Networking \(see page 199\)](#)
- [Using ifquery to Validate and Debug Interface Configurations \(see page 200\)](#)
- [Debugging Mako Template Errors \(see page 201\)](#)
- [ifdown Cannot Find an Interface that Exists \(see page 202\)](#)
- [MTU Set on a Logical Interface Fails with Error: "Numerical result out of range" \(see page 203\)](#)
- [Interpreting iproute2 batch Command Failures \(see page 203\)](#)
- [Understanding the "RTNETLINK answers: Invalid argument" Error when Adding a Port to a Bridge \(see page 203\)](#)

Enabling Logging for Networking

The `/etc/default/networking` file contains two settings for logging:

- To get `ifupdown2` logs when the switch boots (stored in `syslog`)
- To enable logging when you run `service networking [start|stop|reload]`

This file also contains an option for excluding interfaces when you boot the switch or run `service networking start|stop|reload`. You can exclude any interface specified in `/etc/network/interfaces`. These interfaces do not come up when you boot the switch or start/stop/reload the networking service.

```
$cat /etc/default/networking
#
#
# Parameters for the /etc/init.d/networking script
#
#

# Change the below to yes if you want verbose logging to be enabled
VERBOSE="no"
```

```
# Change the below to yes if you want debug logging to be enabled
DEBUG="no"

# Change the below to yes if you want logging to go to syslog
SYSLOG="no"

# Exclude interfaces
EXCLUDE_INTERFACES=

# Set to 'yes' if you want to skip ifdown during system reboot
# and shutdown. This is of interest in large scale interface
# deployments where you dont want to wait for interface
# deconfiguration to speed up shutdown/reboot
SKIP_DOWN_AT_SYSRESET="yes"
```

Using ifquery to Validate and Debug Interface Configurations

You use `ifquery` to print parsed interfaces file entries.

To use `ifquery` to pretty print iface entries from the `interfaces` file, run:

```
cumulus@switch:~$ sudo ifquery swp1
auto swp1
iface swp1 inet static
    address 10.168.26.1/24
```

Use `ifquery --check` to check the current running state of an interface within the `interfaces` file. It returns exit code `0` or `1` if the configuration does not match:

```
cumulus@switch:~$ sudo ifquery --check swp26
auto swp1
iface swp1 inet static [pass]
    address 10.168.26.1/24 [pass]
```



`ifquery --check` is an experimental feature.

Use `ifquery --running` to print the running state of interfaces in the `interfaces` file format:


```
cumulus@switch:~$ sudo ifquery --running swp1
auto swp1
iface swp1
    address 10.168.26.1/24
```

`ifquery --syntax-help` provides help on all possible attributes supported in the `interfaces` file. For complete syntax on the `interfaces` file, see `man interfaces` and `man ifupdown-addons.interfaces`.

`ifquery` can dump information in JSON format:

```
cumulus@switch:~$ sudo ifquery --format=json swp1
[
  {
    "auto": true,
    "config": {
      "address": "10.168.26.1/24"
    },
    "addr_method": "static",
    "name": "swp1",
    "addr_family": "inet"
  }
]
```

You can use `ifquery --print-savedstate` to check the `ifupdown2` state database. `ifdown` works only on interfaces present in this state database.

```
cumulus@leaf1$ sudo ifquery --print-savedstate eth0
auto eth0
iface eth0 inet dhcp
```

Debugging Mako Template Errors

An easy way to debug and get details about template errors is to use the `mako-render` command on your `interfaces` template file or on `/etc/network/interfaces` itself.

```
cumulus@switch:~$ sudo mako-render /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
```

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp
#auto eth1
#iface eth1 inet dhcp

# Include any platform-specific interface configuration
source /etc/network/interfaces.d/*.if

# ssim2 added

auto swp45
iface swp45

auto swp46
iface swp46

cumulus@switch:~$ sudo mako-render /etc/network/interfaces.d
/<interfaces_stub_file>
```

ifdown Cannot Find an Interface that Exists

If you are trying to bring down an interface that you know exists, use `ifdown` with the `--use-current-config` option to force `ifdown` to check the current `/etc/network/interfaces` file to find the interface. This can solve issues where the `ifup` command issues for that interface was interrupted before it updated the state database. For example:

```
cumulus@switch:~$ sudo ifdown br0
error: cannot find interfaces: br0 (interface was probably never up ?)

cumulus@switch:~$ sudo brctl show
bridge name      bridge id        STP enabled      interfaces
br0              8000.44383900279f  yes              downlink
                                     peerlink

cumulus@switch:~$ sudo ifdown br0 --use-current-config
```

MTU Set on a Logical Interface Fails with Error: "Numerical result out of range"

This error occurs when the MTU you are trying to set on an interface is higher than the MTU of the lower interface or dependent interface. Linux expects the upper interface to have an MTU less than or equal to the MTU on the lower interface.

In the example below, the swp1.100 VLAN interface is an upper interface to physical interface swp1. If you want to change the MTU to 9000 on the VLAN interface, you must include the new MTU on the lower interface swp1 as well.

```
auto swp1.100
iface swp1.100
    mtu 9000

auto swp1
iface swp1
    mtu 9000
```

Interpreting iproute2 batch Command Failures

ifupdown2 batches iproute2 commands for performance reasons. A batch command contains `ip -force -batch -` in the error message. The command number that failed is at the end of this line: `Command failed -:1`.

Below is a sample error for the command `1: link set dev host2 master bridge`. There was an error adding the bond `host2` to the bridge named `bridge` because `host2` did not have a valid address.

```
error: failed to execute cmd 'ip -force -batch - [link set dev host2 master
bridge
addr flush dev host2
link set dev host1 master bridge
addr flush dev host1
]'(RTNETLINK answers: Invalid argument
Command failed -:1)
warning: bridge configuration failed (missing ports)
```

Understanding the "RTNETLINK answers: Invalid argument" Error when Adding a Port to a Bridge

This error can occur when the bridge port does not have a valid hardware address.

This can typically occur when the interface being added to the bridge is an incomplete bond; a bond without slaves is incomplete and does not have a valid hardware address.

Network Troubleshooting

Cumulus RMP contains a number of command line and analytical tools to help you troubleshoot issues with your network.

Contents

(Click to expand)

- [Contents \(see page 204\)](#)
- [Commands \(see page 204\)](#)
- [Checking Reachability Using ping \(see page 204\)](#)
- [Printing Route Trace Using traceroute \(see page 205\)](#)
- [Manipulating the System ARP Cache \(see page 205\)](#)
- [Traffic Generation Using mz \(see page 206\)](#)
- [Counter ACL \(see page 207\)](#)
- [SPAN and ERSPAN \(see page 208\)](#)
- [Configuration Files \(see page 211\)](#)
- [Useful Links \(see page 211\)](#)
- [Caveats and Errata \(see page 211\)](#)

Commands

- [arp](#)
- [cl-acltool](#)
- [ip](#)
- [mz](#)
- [ping](#)
- [traceroute](#)

Checking Reachability Using ping

`ping` is used to check reachability of a host. `ping` also calculates the time it takes for packets to travel the round trip. See `man ping` for details.

To test the connection to an IPv4 host:

```
cumulus@switch:~$ ping 206.190.36.45
PING 206.190.36.45 (206.190.36.45) 56(84) bytes of data.
64 bytes from 206.190.36.45: icmp_req=1 ttl=53 time=40.4 ms
64 bytes from 206.190.36.45: icmp_req=2 ttl=53 time=39.6 ms
...
```

To test the connection to an IPv6 host:

```
cumulus@switch:~$ ping6 -I swp1 fe80::202:ff:fe00:2
PING fe80::202:ff:fe00:2(fe80::202:ff:fe00:2) from fe80::202:ff:fe00:1
swp1: 56 data bytes
64 bytes from fe80::202:ff:fe00:2: icmp_seq=1 ttl=64 time=1.43 ms
64 bytes from fe80::202:ff:fe00:2: icmp_seq=2 ttl=64 time=0.927 ms
```

Printing Route Trace Using traceroute

`trace` routetracks the route that packets take from an IP network on their way to a given host. See `man traceroute` for details.

To track the route to an IPv4 host:

```
cumulus@switch:~$ traceroute www.google.com
traceroute to www.google.com (74.125.239.49), 30 hops max, 60 byte packets
 1  fw.cumulusnetworks.com (192.168.1.1)  0.614 ms  0.863 ms  0.932 ms
 2  router.hackertojo.com (157.22.42.1)  15.459 ms  16.447 ms  16.818 ms
 3  gw-cpe-hackertojo.via.net (157.22.10.97)  18.470 ms  18.473 ms  18.897 ms
 4  ge-1-5-v223.core1.uspao.via.net (157.22.10.81)  20.419 ms  20.422 ms
    21.026 ms
 5  core2-1-1-0.pao.net.google.com (198.32.176.31)  22.347 ms  22.584 ms
    24.328 ms
 6  216.239.49.250 (216.239.49.250)  24.371 ms  25.757 ms  25.987 ms
 7  72.14.232.35 (72.14.232.35)  27.505 ms  22.925 ms  22.323 ms
 8  nuq04s19-in-f17.1e100.net (74.125.239.49)  23.544 ms  21.851 ms  22.604
    ms
```

Manipulating the System ARP Cache

`arp` manipulates or displays the kernel's IPv4 network neighbor cache. See `man arp` for details.

To display the ARP cache:

```
cumulus@switch:~$ arp -a
? (11.0.2.2) at 00:02:00:00:00:10 [ether] on swp3
? (11.0.3.2) at 00:02:00:00:00:01 [ether] on swp4
? (11.0.0.2) at 44:38:39:00:01:c1 [ether] on swp1
```

To delete an ARP cache entry:

```
cumulus@switch:~$ arp -d 11.0.2.2
cumulus@switch:~$ arp -a
? (11.0.2.2) at <incomplete> on swp3
? (11.0.3.2) at 00:02:00:00:00:01 [ether] on swp4
? (11.0.0.2) at 44:38:39:00:01:c1 [ether] on swp1
```

To add a static ARP cache entry:

```
cumulus@switch:~$ arp -s 11.0.2.2 00:02:00:00:00:10
cumulus@switch:~$ arp -a
? (11.0.2.2) at 00:02:00:00:00:10 [ether] PERM on swp3
? (11.0.3.2) at 00:02:00:00:00:01 [ether] on swp4
? (11.0.0.2) at 44:38:39:00:01:c1 [ether] on swp1
```

Traffic Generation Using mz

mzis a fast traffic generator. It can generate a large variety of packet types at high speed. See `man mz` for details.

For example, to send two sets of packets to TCP port 23 and 24, with source IP 11.0.0.1 and destination 11.0.0.2, do the following:

```
cumulus@switch:~$ sudo mz swp1 -A 11.0.0.1 -B 11.0.0.2 -c 2 -v -t tcp
"dp=23-24"

Mausezahn 0.40 - (C) 2007-2010 by Herbert Haas - http://www.perihel.at/sec
/mz/
Use at your own risk and responsibility!
-- Verbose mode --

This system supports a high resolution clock.
The clock resolution is 4000250 nanoseconds.
Mausezahn will send 4 frames...
IP:  ver=4, len=40, tos=0, id=0, frag=0, ttl=255, proto=6, sum=0, SA=11.
0.0.1, DA=11.0.0.2,
    payload=[see next layer]
TCP:  sp=0, dp=23, S=42, A=42, flags=0, win=10000, len=20, sum=0,
    payload=

IP:  ver=4, len=40, tos=0, id=0, frag=0, ttl=255, proto=6, sum=0, SA=11.
0.0.1, DA=11.0.0.2,
    payload=[see next layer]
TCP:  sp=0, dp=24, S=42, A=42, flags=0, win=10000, len=20, sum=0,
```

```

payload=

IP:  ver=4, len=40, tos=0, id=0, frag=0, ttl=255, proto=6, sum=0, SA=11.
0.0.1, DA=11.0.0.2,
    payload=[see next layer]
TCP:  sp=0, dp=23, S=42, A=42, flags=0, win=10000, len=20, sum=0,
    payload=

IP:  ver=4, len=40, tos=0, id=0, frag=0, ttl=255, proto=6, sum=0, SA=11.
0.0.1, DA=11.0.0.2,
    payload=[see next layer]
TCP:  sp=0, dp=24, S=42, A=42, flags=0, win=10000, len=20, sum=0,
    payload=

```

Counter ACL

In Linux, all ACL rules are always counted. To create an ACL rule for counting purposes only, set the rule action to ACCEPT.



Always place your rules files under `/etc/cumulus/acl/policy.d/`.

To count all packets going to a Web server:

```

cumulus@switch$ cat sample_count.rules

[iptables]
-A FORWARD -p tcp --dport 80 -j ACCEPT

cumulus@switch:$ sudo cl-acltool -i -p sample_count.rules
Using user provided rule file sample_count.rules
Reading rule file sample_count.rules ...
Processing rules in file sample_count.rules ...
Installing acl policy... done.

cumulus@switch$ sudo iptables -L -v
Chain INPUT (policy ACCEPT 16 packets, 2224 bytes)
pkts bytes target      prot opt in      out     source
destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target      prot opt in      out     source
destination

```

```

2    156 ACCEPT      tcp  --  any    any    anywhere
anywhere            tcp dpt:http

```

```

Chain OUTPUT (policy ACCEPT 44 packets, 8624 bytes)
pkts bytes target      prot opt in      out     source
destination

```

SPAN and ERSPAN

SPAN (Switched Port Analyzer) provides for the mirroring of all packets coming in from or going out of an interface to a local port for monitoring. This port is referred to as a mirror-to-port (MTP). The original packet is still switched, while a mirrored copy of the packet is sent to the MTP port.

ERSPAN (Encapsulated Remote SPAN) enables the mirrored packets to be sent to a monitoring node located anywhere across the routed network. The switch finds the outgoing port of the mirrored packets by doing a lookup of the destination IP address in its routing table. The original L2 packet is encapsulated with GRE for IP delivery. The encapsulated packets have the following format:

```

-----
| MAC_HEADER | IP_HEADER | GRE_HEADER | L2_Mirrored_Packet |
-----

```

SPAN and ERSPAN are configured via `cl-acltool`. The match criteria for SPAN and ERSPAN can only be an interface; more granular match terms are not supported. The interface can be a port, a subinterface or a bond interface. Both ingress and egress interfaces can be matched.

Cumulus RMP supports a maximum of 2 SPAN destinations. Multiple rules can point to the same SPAN destination. The MTP interface can be a physical port, a subinterface, or a bond interface. The SPAN /ERSPAN action is independent of security ACL actions. If packets match both a security ACL rule and a SPAN rule, both actions will be carried out.



Always place your rules files under `/etc/cumulus/acl/policy.d/`.

To configure SPAN for all packets coming in from swp1 locally to swp3:

```

cumulus@switch$ cat span.rules

[iptables]
-A FORWARD --in-interface swp1 -j SPAN --dport swp3

cumulus@switch$ cl-acltool -i -p span.rules
Using user provided rule file span.rules
Reading rule file span.rules ...
Processing rules in file span.rules ...

```



```
Installing acl policy... done.

cumulus@switch$ sudo iptables -L -v
Chain INPUT (policy ACCEPT 18 packets, 3034 bytes)
  pkts bytes target      prot opt in      out     source
destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target      prot opt in      out     source
destination
    28  3014 SPAN          all  --  swp1    any     anywhere
anywhere                dport:swp3

Chain OUTPUT (policy ACCEPT 56 packets, 12320 bytes)
  pkts bytes target      prot opt in      out     source
destination
```

To configure SPAN for all packets going out of bond0 locally to bond1:

```
cumulus@switch$ cat span.rules

[iptables]
-A FORWARD --out-interface bond0 -j SPAN --dport bond1

cumulus@switch$ cl-acltool -i -p span.rules
Using user provided rule file span.rules
Reading rule file span.rules ...
Processing rules in file span.rules ...
Installing acl policy... done.

cumulus@switch$ sudo iptables -L -v
Chain INPUT (policy ACCEPT 57 packets, 10000 bytes)
  pkts bytes target      prot opt in      out     source
destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target      prot opt in      out     source
destination
    19  1938 SPAN          all  --  any     bond0   anywhere
anywhere                dport:bond1
```

```
Chain OUTPUT (policy ACCEPT 686 packets, 119K bytes)
  pkts bytes target    prot opt in     out     source
destination
```

To configure ERSPAN for all packets coming in from swp1 to 12.0.0.2. :

```
cumulus@switch$ cat erspan.rules

[iptables]
-A FORWARD --in-interface swp1 -j ERSPAN --src-ip 12.0.0.1 --dst-ip
12.0.0.2 --ttl 64

cumulus@switch$ sudo cl-acltool -i -p erspan.rules
Using user provided rule file erspan.rules
Reading rule file erspan.rules ...
Processing rules in file erspan.rules ...
Installing acl policy... done.

cumulus@switch$ sudo iptables -L -v
Chain INPUT (policy ACCEPT 27 packets, 5526 bytes)
  pkts bytes target    prot opt in     out     source
destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source
destination
    69 6804 ERSPAN      all  --  swp1    any     anywhere
anywhere                ERSPAN src-ip:12.0.0.1 dst-ip:12.0.0.2

Chain OUTPUT (policy ACCEPT 822 packets, 163K bytes)
  pkts bytes target    prot opt in     out     source
destination
```

The `src-ip` option can be any IP address, whether it exists in the routing table or not. The `dst-ip` option must be an IP address reachable via the routing table. The destination IP address must be reachable from a front-panel port, and not the management port. Use `ping` or `ip route get <ip>` to verify that the destination IP address is reachable. Setting the `--ttl` option is recommended.



When using [Wireshark](#) to review the ERSPAN output, Wireshark may report the message "Unknown version, please report or test to use fake ERSPAN preference", and the trace is unreadable. To resolve this, go into the General preferences for Wireshark, then go to **Protocols > ERSPAN** and check the **Force to decode fake ERSPAN frame** option.

Configuration Files

- /etc/cumulus/acl/policy.conf

Useful Links

- <http://en.wikipedia.org/wiki/Ping>
- <https://en.wikipedia.org/wiki/Traceroute>
- <http://www.perihel.at/sec/mz/mzguide.html>

Caveats and Errata

- SPAN rules cannot match outgoing subinterfaces.

Index

/

/mnt/persist [46](#)

A

access ports [142](#)

active image slot [53](#)

active listener ports [196](#)

alternate image slot [44](#), [49](#), [54](#), [56](#)

 accessing [56](#)

 installing a new image [44](#)

 selecting [49](#)

apt-get [60](#)

arp cache [205](#)

auto-negotiation [86](#)

autoprovision command [69](#)

autoprovisioning [66](#)

B

BFD [113](#), [116](#), [118](#)

 Bidirectional Forwarding Detection [113](#), [116](#)

 echo function [118](#)

Bidirectional Forwarding Detection [113](#)

bonds [124](#)

boot recovery [163](#)

bpdufilter [105](#)

 and STP [105](#)

BPDU guard [103](#)

 and STP [103](#)

brctl [15](#), [92](#), [129](#), [130](#)

 and STP [92](#)

bridge assurance [102](#)

 and STP [102](#)

bridges [128](#), [128](#), [129](#), [129](#), [130](#), [131](#), [131](#), [135](#), [137](#), [142](#), [142](#), [148](#)

 access ports [142](#)

 adding interfaces [129](#), [130](#)

 adding IP addresses [135](#)

 MAC addresses [131](#)

- MTU 131
- physical interfaces 129
- trunk ports 142
- untagged frames 137
- VLAN-aware 128, 148

C

- cable connectivity 13
- cabling 113
 - Prescriptive Topology Manager 113
- cl-acctool 207
- cl-cfg 41
- cl-img-install 44
- cl-img-pkg 58
- cl-img-select 49, 56, 57, 58
- cl-netstat 167
- cl-resource-query 41, 169
- cl-support 162
- Cumulus Linux 43, 56, 57, 157
 - installing 43
 - reprovisioning 56
 - reserved VLAN ranges 157
 - uninstalling 57
- Cumulus RMP 43
 - upgrading 43
- cumulus user 22

D

- daemons 196
- date 19
 - setting 19
- deb 64
- debugging 160
- decode-syseeprom 171
- dmidecode 173
- dpkg 62
- dpkg-reconfigure 18
- duplex interfaces 85

E

- echo function [118](#), [118](#)
 - BFD [118](#)
 - PTM [118](#)
- ERSPAN [208](#)
 - network troubleshooting [208](#)
- Ethernet management port [12](#)
- ethtool [89](#), [165](#)
 - switch ports [89](#)

G

- globs [81](#)
- Graphviz [113](#)

H

- hardware [170](#)
 - monitoring [170](#)
- hash distribution [127](#)
- host entries [169](#)
 - monitoring [169](#)
- hostname [13](#)
- hsflowd [178](#)
- hwclock [19](#)

I

- ifdown [72](#)
- ifquery [76](#), [200](#)
- ifup [72](#)
- ifupdown [71](#)
- ifupdown2 [71](#), [80](#), [140](#), [199](#), [199](#), [199](#)
 - excluding interfaces [199](#)
 - logging [199](#)
 - purging IP addresses [80](#)
 - troubleshooting [199](#)
 - VLAN tagging [140](#)
- image contents [58](#)
- image slots [53](#), [54](#), [54](#)
 - resizing [54](#)
 - x86 [54](#)
- interface counters [167](#)
- interface dependencies [75](#)

interfaces [84, 88](#)

 statistics [88](#)

IP addresses [80](#)

 purging [80](#)

iproute2 [203](#)

 failures [203](#)

J

jdoo [174](#)

L

LACP [124](#)

layer 3 access ports [15](#)

 configuring [15](#)

LDAP [30](#)

link aggregation [124](#)

Link Layer Discovery Protocol [107](#)

LLDP [107](#)

lldpcli [108](#)

lldpd [107, 114](#)

logging [199, 199](#)

 ifupdown2 [199](#)

 networking service [199](#)

loopback interface [16](#)

 configuring [16](#)

lshw [173](#)

M

MAC entries [169](#)

 monitoring [169](#)

Mako templates [81, 201](#)

 debugging [201](#)

monitoring [18, 160, 165, 169, 176, 177](#)

 hardware watchdog [176](#)

 network traffic [177](#)

mstpctl [92, 144](#)

MTU [87, 131, 203](#)

 bridges [131](#)

 failures [203](#)

multiple bridges [132](#)

mz [206](#)
 traffic generator [206](#)

N

name switch service [29](#)
Net-SNMP [174](#)
networking service [199](#)
 logging [199](#)
network interfaces [84](#)
network traffic [177](#)
 monitoring [177](#)
NSS [29](#)
 name switch service [29](#)
NTP [20](#)
 time [20](#)
ntpd [20](#)

O

ONIE [58](#)
 rescue mode [58](#)
open source contributions [10](#)

P

packages [60](#)
 managing [60](#)
PAM [29](#)
 pluggable authentication modules [29](#)
parent interfaces [78](#)
password [22](#)
 default [22](#)
passwordless access [22](#)
passwords [12](#)
persistent configurations [46](#)
Per VLAN Spanning Tree [92](#)
 PVST [92](#)
ping [204](#)
pluggable authentication modules [29](#)
port lists [81](#)
port speeds [85](#)
Prescriptive Topology Manager [113](#)

- primary image slot [53](#)
- privileged commands [23](#)
- PTM [113](#), [118](#)
 - echo function [118](#)
 - Prescriptive Topology Manager [113](#)
- ptmctl [120](#)
- ptmd [113](#)
- PTM scripts [119](#)
- public community [175](#)
- PVRST [92](#)
 - Rapid PVST [92](#)
- PVST [92](#)
 - Per VLAN Spanning Tree [92](#)

Q

- QSFP [168](#)

R

- Rapid PVST [92](#)
 - PVRST [92](#)
- recommended configuration [46](#)
- remote access [20](#)
- repositories [64](#)
 - other packages [64](#)
- rescue mode [58](#)
- reserved VLAN ranges [157](#)
- restart [41](#)
 - switchd [41](#)
- root user [12](#), [22](#)
- routes [169](#)
 - monitoring [169](#)
- RSTP [92](#)

S

- sensors command [173](#)
- serial console management [12](#)
- services [196](#)
- sFlow [177](#)
- sFlow visualization tools [180](#)
- SFP [89](#), [168](#)

- switch ports 89
- single user mode 163
- smonctl 176
- smond 176
- SNMP 174
- snmpd 174
- sources.list 64
- SPAN 208
 - network troubleshooting 208
- spanning tree parameters 95
- Spanning Tree Protocol 91, 149
 - STP 91
 - VLAN-aware bridges 149
- SSH 20
- SSH keys 21
- static routing 158
 - with ip route 158
- STP 91, 102
 - and bridge assurance 102
 - Spanning Tree Protocol 91
- sudo 22, 23
- sudoers 23, 24
 - examples 24
- switchd 39, 39, 41
 - configuring 39
 - file system 39
 - restarting 41
- switch ports 14
 - configuring 14
- system management 160

T

- templates 81
- time 19
 - setting 19
- time zone 13, 18
- topology 113
 - data center 113
- traceroute 205
- traffic distribution 127
- traffic generator 206
 - mz 206
- troubleshooting 160, 163

single user mode [163](#)
trunk ports [137](#), [142](#)
tzdata [18](#)

U

U-Boot [160](#)
untagged frames [137](#)
 bridges [137](#)
user accounts [22](#)
 cumulus [22](#)
 root [22](#)
user authentication [29](#)
user commands [80](#)
 interfaces [80](#)

V

visudo [23](#)
VLAN-aware bridges [128](#), [148](#), [149](#), [151](#), [152](#), [153](#), [154](#)
 basic example [152](#)
 configuring [151](#)
 Spanning Tree Protocol [149](#)
 with access ports and pruned VLANs [153](#)
 with bonds [154](#)
VLAN tagging [140](#), [140](#), [142](#)
 advanced example [142](#)
 basic example [140](#)
VLAN translation [147](#)

W

watchdog [176](#)
 monitoring [176](#)

Z

zero touch provisioning [66](#)
ZTP [66](#)