SONiC System Architecture

Kanza Latif (April 2024)

Overview

The SONiC system architecture is designed to provide modularity, scalability and efficient communication among its components. At its core, SONiC relies on docker containers and a centralized infrastructure powered by a redis-database engine.This architecture enables SONiC to effectively manage network functionalities and interactions between different subsystems.

Goals of SONiC

- Faster Technology Evolution
- Reduce Operational Burden
- Open & Modular Software
- Choices of Vendors & Platforms

Docker Containerization

In SONiC, each functional module is encapsulated within its own docker container. This approach ensures high cohesion among semantically-related components while minimizing coupling between different modules. By leveraging containerization, SONiC can efficiently manage and deploy its components across diverse network environments.

Main Functional Components

SONiC comprises of several key functional components, each hosted within dedicated docker containers:

- ## DHCP-relay

  Facilitates the DHCP relay functionality to forward DHCP requests from one subnet to DHCP servers on other subnets

- ## Pmon

  Manages sensor readings and alarms, along with fan-related state collection

- ## SNMP

  Hosts SNMP features, including snmpd server and a SONiC-specific subagent for handling SNMP queries

- ## Lidp

  Implements LLDP functionality including the LLDP demon, lldp_syncd for state upload, and lldpmgr for incremental configuration

- ## BGP

  Runs routing stacks like Quagga or FRR, handling routing state through bgpd , zebra, and fpmsyncd processes

- ## Teamd

  Manages Link Aggregation(LAG) functionality using the “teamd” open-source implementation, along with interaction with the “teamsyncd” process

- ## Database

  Hosts the Redis database engine, providing storage for various SONiC applications and subsystems

- ## SWSS

  Houses the Switch State Service, facilitating communication among SONiC modules and north-bound interaction with SONiC applications

- ### Syncd

  Responsible for synchronizing the switch's network state with its hardware/ASIC, using the Syncd process along with the SAI API and ASIC SDK

SONiC Subsystems Description

Here, I will discuss about an extensive overview of the functionality enclosed within each Docker container and key SONiC components operating from the Linux host system

Teamd Container

This container manages Link Aggregation (LAG) functionality, utilizing the "teamd" open-source implementation and interacting with the "teamsyncd" process

Pmon Container

Responsible for managing sensor readings, logging alarms,and collecting fan-related state

SNMP Container

Hosts SNMP features , including the SNMP server and the SONiC-specific SNMP-agent for feeding information to the master agent

DHCP-relay Container

Facilitates DHCP relay functionality to forward DHCP requests from one subnet to DHCP servers on other subnets

LLDP Container

Handles LLDp functionality, including the LLDP demon, lldp_syncd for state upload, and lldpmgr for incremental configuration

BGP Container

Runs routing stacks like Quagga or FRR, handling routing state through BGPd , zebra, and fpmsyncd processes

Database Container

Hosts the Redis database engine, providing storage for various SONiC applications and subsystems

SWSS Container

Houses the Switch State Service, facilitating communication among SONiC modules and north-bound interaction with SONiC applications