GUIDE
By: Humza Altaf, Network Engineer

Revision No.	Description	Editor	Date
1.0	Guide for Config_DB in SONiC	Humza Altaf	April 15, 2024

Adopt SONiC Successfully with Hardware Nation.

Talk with our specialists to learn about our integrated approach, which includes expert guidance, consultation, hardware selection, network design, deployment, and ongoing support.

Table of Contents

About this Document

This document provides details about how the switch configuration is handled on a SONiC device.

Definitions

| Term	Meaning
Config DB	SONiC Configuration Database
startup-config	/etc/sonic/config_db.json

Introduction to Config_DB

SONiC switch configuration is stored in a Redis database instance known as Config DB. In SONiC OS, the configuration database (Config DB) stores network settings and state information. It's typically implemented using Redis, in a structured format like **key-value pairs or JSON**. This allows for fast access and manipulation, enabling easy automation and integration with other tools. The Config DB serves as a central source of truth for network configurations, ensuring consistency and flexibility in managing the network.

In the current version of SONiC, ConfigDB is implemented as database 4 of local Redis. When the system boots, configurations will be loaded into Redis from the "/etc/sonic/config_db.json" file. ConfigDB content won't automatically be written back into /etc/sonic/config_db.json file. To do that, a config save command needs to be manually executed from CLI. Similarly, config load will trigger a force load of JSON file into DB. Generally, content in "/etc/sonic/config_db.json" can be considered as starting config, and content in redisDB running-config.

The contents of Config DB reflect most of the configuration of a SONiC switch. The contents of Config DB can be saved in a file /etc/sonic/config_db.json using the config save CLI command. During switch bootup, Config DB is populated with the intended configuration present in the file /etc/sonic/config_db.json. Throughout this document, the term startup-configuration is used to refer to /etc/sonic/config_db.json.

Network Topology

The GNS3 network topology consists of two switches SONiC (202305) named S1 and S2 and two hosts. First switch S1 is configured by editing the JSON file and then these configurations are loaded in switch S2.
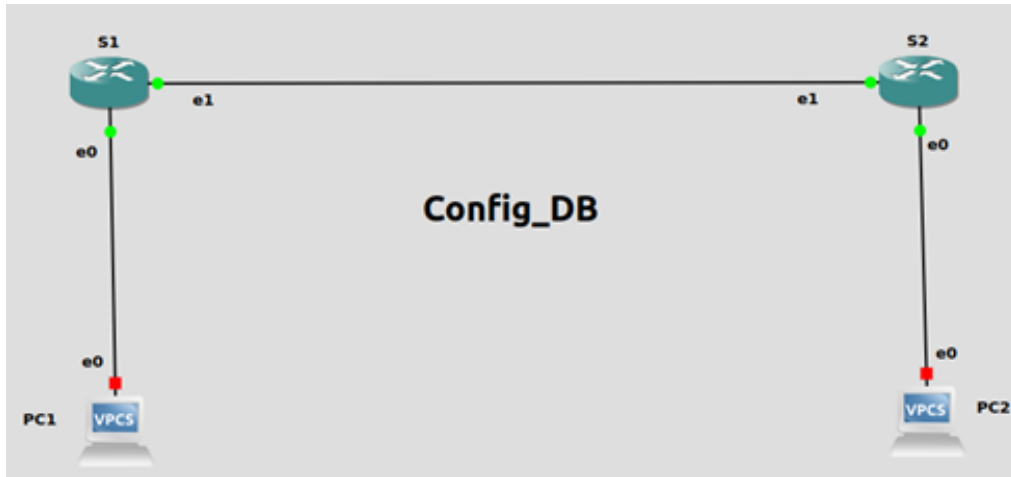
Port Mapping

GNS3	SONiC
Ethernet 0	Ethernet 0
Ethernet 1	Ethernet 4
Ethernet 2	Ethernet 8
Ethernet 3	Ethernet 12

Configurations

Command Reference guide is also available on GitHub for SONiC, whose link is given here.

Follow these steps to configure S1.

Step 1

First check the status of IP addresses by using the command "show ip interfaces".

```
admin@sonic:~$ show ip interfaces
/usr/local/lib/python3.9/dist-packages/sonic_platform_base/sonic_xcvr/fields/xcvr_fie
  assert (bitpos >= 0 and (bitpos+self.size-1 < 8), "bitpos must be within one byte")
Interface    Master    IPv4 address/mask    Admin/Oper    BGP Neighbor    Neighbor IP
----------   --------   ------------------   -----------   -------------   -----------
Ethernet0              10.0.0.0/31          up/up         ARISTA01T2      10.0.0.1
Ethernet4              10.0.0.2/31          up/up         ARISTA02T2      10.0.0.3
Ethernet8              10.0.0.4/31          up/up         ARISTA03T2      10.0.0.5
Ethernet12             10.0.0.6/31          up/up         ARISTA04T2      10.0.0.7
```

In the above figure, default IP addresses are assigned. Now we will change the IP address of interface "Ethernet0" by editing the JSON file "config_db.json" which is present in the directory "/etc/sonic/".

```
admin@sonic:~$ cd /etc/sonic/
admin@sonic:/etc/sonic$ ls
asic_config_checksum    fast-reboot_order          sonic-environment
config_db.json          frr                        sonic_release
constants.yml           generated_services.conf    sonic_version.yml
copp_cfg.json           init_cfg.json              swss_dependent
core_analyzer.rc.json   macsec_reconcile           updategraph.conf
dhcp_relay_reconcile    snmp.yml                   warm-reboot_order
```

Before editing the file, check the contents of this file by using the command "cat config_db.json".

```
admin@sonic:/etc/sonic$ cat config_db.json
{
    "DEVICE_METADATA": {
        "localhost": {
            "hwsku": "Force10-S6000",
            "platform": "x86_64-kvm_x86_64-r0",
            "mac": "0c:37:e0:ab:00:00",
            "hostname": "sonic",
            "type": "LeafRouter",
            "bgp_asn": "65100"
        }
    },
    "PORT": {
        "Ethernet0": {
            "lanes": "25,26,27,28",
            "alias": "fortyGigE0/0",
            "index": "0",
            "speed": "40000",
            "admin_status": "up",
            "mtu": "9100"
        },
```

Step 2

Now edit the "config_db.json" file by using the command "sudo vi /etc/sonic/config_db.json". After pressing i, edit the IP address of Ethernet0 and assign "192.168.10.2/24".

Reload it by using the command "sudo config reload -y".

Note: After editing the JSON file, refrain from using the 'config save' command as it will overwrite the current content with the previous data. Instead, after editing, reload it using the 'config reload' command. If the switch is configured through the CLI, avoid using the 'config reload' command. Instead, utilize the 'config save' command to preserve configurations in the config_db.

Now check the status of IP address.

As we can see in the above figure, the IP address has been assigned by editing a JSON file.

Step 3

Now configure VLANs by editing config_db. Check the status of VLAN table by using command "show vlan brief".

```
admin@sonic:~$ show vlan brief
+-----------+-------------+----------+--------------+-----------+----------------------+
| VLAN ID   | IP Address  | Ports    | Port Tagging | Proxy ARP | DHCP Helper Address  |
+===========+=============+==========+==============+===========+======================+
+-----------+-------------+----------+--------------+-----------+----------------------+
```

The interesting thing here is that VLAN10 will be created using CLI while VLAN20 and VLAN30 will be created by editing the JSON file.

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

Since VLAN10 has been created using CLI, "config save" command must be used to show this VLAN in the config_db.

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

Now check the status of config_db.

```
},
"VLAN": {
    "Vlan10": {
        "vlanid": "10"
    }
}
}
```

In the above figure, VLAN10 is present in the config_db. Now create VLAN20 and VLAN30 by editing JSON file and then reload the switch.

```
admin@sonic:~$ sudo config reload -y
Disabling container monitoring ...
Stopping SONiC target ...
Running command: /usr/local/bin/sonic-cfggen -j /etc/sonic/init_cfg.json -j /etc/sonic/config_db.json --write-to-db
Running command: /usr/local/bin/db_migrator.py -o migrate
Running command: /usr/local/bin/sonic-cfggen -d -y /etc/sonic/sonic_version.yml -t /usr/share/sonic/templates/sonic
Restarting SONiC target ...
Enabling container monitoring ...
Reloading Monit configuration ...
Reinitializing monit daemon
admin@sonic:~$ show vlan brief
+-----------+-------------+----------+--------------+-----------+----------------------+
| VLAN ID   | IP Address  | Ports    | Port Tagging | Proxy ARP | DHCP Helper Address  |
+===========+=============+==========+==============+===========+======================+
|    10     |             |          |              | disabled  |                      |
+-----------+-------------+----------+--------------+-----------+----------------------+
|    20     |             |          |              | disabled  |                      |
+-----------+-------------+----------+--------------+-----------+----------------------+
|    30     |             |          |              | disabled  |                      |
+-----------+-------------+----------+--------------+-----------+----------------------+
```

In the above figure, VLAN20 and VLAN30 are present in the table

Step 4

In SONiC, BGP runs by default with ASN 65100. First this instance must be removed to configure new one.

To configure OSPF, it's essential to start the OSPF daemon 'ospfd' beforehand. Otherwise, OSPF will not function correctly.

By editing DEVICE METADATA in the config_db, this problem can be solved. Just add "frr_mgmt_framework_config": "true" in the DEVICE METADATA.

After editing JSON file, these problems will be resolved as can be seen in the figure below.

Follow these steps to configure S2.

Step 5

Do not configure switch S2 by editing the JSON file. Instead, copy the config_db from switch S1 and paste it into the document. These configurations will then be loaded into switch S2.

First, check the status of the IP address and VLAN table before loading configurations.

```
admin@sonic:~$ show vlan brief
+-----------+-------------+---------+----------------
| VLAN ID   | IP Address  | Ports   | Port Tagging
+===========+=============+=========+================
+-----------+-------------+---------+----------------
```

```
admin@sonic:~$ show ip interfaces
Interface     Master     IPv4 address/mask     Admin/Oper
----------    -------    -------------------   ----------
Ethernet0                10.0.0.0/31           up/up
Ethernet4                10.0.0.2/31           up/up
```

Now create a new JSON file in the directory "/etc/sonic/" named "humza_config_db.json".

```
admin@sonic:~$ sudo touch /etc/sonic/humza_config_db.json
admin@sonic:~$ cd /etc/sonic/
admin@sonic:/etc/sonic$ ls
asic_config_checksum    frr                      sonic_release
config_db.json          generated_services.conf  sonic_version.yml
constants.yml           humza_config_db.json     swss_dependent
copp_cfg.json           init_cfg.json            updategraph.conf
core_analyzer.rc.json   macsec_reconcile         warm-reboot_order
dhcp_relay_reconcile    snmp.yml
fast-reboot_order       sonic-environment
admin@sonic:/etc/sonic$ cat humza_config_db.json
admin@sonic:/etc/sonic$
```

Paste these configurations in the "humza_config_db.json" file and reload it.

```
admin@sonic:~$ sudo config reload /etc/sonic/humza_config_db.json
Clear current config and reload config in config_db from the file(s) /etc/sonic/humza_config_db.json ? [y/N]: y
Disabling container monitoring ...
Stopping SONiC target ...
Running command: /usr/local/bin/sonic-cfggen -j /etc/sonic/init_cfg.json -j /etc/sonic/humza_config_db.json --write-to-db
Running command: /usr/local/bin/db_migrator.py -o migrate
Running command: /usr/local/bin/sonic-cfggen -d -y /etc/sonic/sonic_version.yml -t /usr/share/sonic/templates/sonic-envir
Restarting SONiC target ...
Enabling container monitoring ...
Reloading Monit configuration ...
Reinitializing monit daemon
```

Step 5 (Continued)

Now check the status of the VLAN table and IP address.

As can be seen in the above figures, configurations have been loaded successfully.

Summary

SONiC, an open-source network operating system, utilizes a configuration database known as config_db to manage switch configurations. This database stores settings and parameters for various network functionalities. Users can configure a SONiC switch using either the Command Line Interface (CLI) or JSON scripts. This flexibility allows for efficient management and customization of network configurations according to specific requirements.

References

https://github.com/sonic-net/sonic-utilities/blob/master/doc/Command-Reference.md
https://github.com/sonic-net/SONiC/wiki/Configuration

Connecting ideas, people, and technology.