

## Table of Contents

- [Revision history](#)
- [Scope](#)
- [SONiC Base image Debian Upgrade](#)
  - [Debian Release Schedule](#)
  - [Timeline](#)
  - [Procedure](#)
- [SONiC Container Debian Upgrade](#)
  - [Guidelines](#)
  - [Procedure](#)
  - [Timeline](#)
- [SONiC Debian support](#)
  - [Guidelines](#)
  - [Procedure](#)
  - [Timeline](#)
- [SONiC Debian Deprecation](#)

## Revision history

Rev	Date	Authors	Change Description	
-----	-----	-----	-----	0.1   01/01/2024
		Pavan Naregundi, Saikrishna Arcot	Initial version	

## Scope

SONiC is a free and open source network operating system based on Debian based Linux. In order to keep SONiC updated with new features, bug fixes, and security updates, we want to make sure that both the SONiC base image and all of the containers are based on the most recent version of Debian. Scope of this document is to describe process and cadence for following,

- SONiC Base image Debian Upgrade
- SONiC Container Debian Upgrade
- SONiC Debian Support
- SONiC Debian Deprecation

## SONiC Base image Debian Upgrade

This section describes SONiC Base image Debian upgrade process and cadence.

### Debian Release Schedule

Debian does not have any fixed release schedule officially. Debian Community will release a new version when it's ready. Unofficially, based on

the [releases](#) since Debian Stretch (in 2017), new Debian versions have come out every 2 years, around June-August, with Bookworm released in June 2023. Based on this schedule, it's reasonable to assume that Debian Trixie may get released around June-August 2025.

## Timeline

Given Debian release info, and the fact that the current SONiC release trend is to have a release in May and November, the goal should be to target the base image upgrade to the new Debian version for the November release. If, however, the release schedule changes such that there are less than 3 months between the Debian release and the SONiC branch cutoff, it would be recommend to push back the base image upgrade to the next SONiC release.

If needed, some work (such as creating the slave container, new kernel migration) can be done at the full freeze of Debian release (which will likely be about 2-3 weeks prior to the release). This gives a little bit of extra room in the schedule. However, do note that there is a chance of issues in the new version being present during this time, so please expect potential changes to packages during this time.

## Procedure

To accomplish this, there are some changes that should be present in sonic-net/sonic-buildimage, either in the master branch or in a development branch. Specifically, this is the slave container for the new Debian version. This is required for sonic-net/sonic-linux-kernel to be able to build the new kernel version. Hence, the following tasks need to be done first, and merged into the master branch or in a development branch in sonic-net/sonic-buildimage:

1. Create the slave container for the new Debian version. a. This doesn't need to be the final/official slave container that will be used. For now, it needs to be able to build the new kernel, which should be fairly easy to accomplish.
2. Make changes to Makefile and Makefile.work to be able to build the new slave container.

After that is done, a new Azure pipeline will need to be created to build the slave container and publish it to the container registry, so that the sonic-net/sonic-linux-kernel build can use it.

Then, work on both upgrading the kernel to the new version (and disabling patches/configs as necessary to get the build to succeed) as well as building a VS/KVM image, but with the kernel build disabled (if needed), can begin. This part can proceed in parallel, since for the userspace applications in the VS build, there shouldn't be any hard dependency on the specific kernel that we build. Note that getting the kernel build done will likely take less time

than getting the userspace build done, depending on the changes that are in the new version of Debian.

On the userspace side, depending on the number of changes needed and depending on what is being built, it may be easier to disable the build of that application/package for now to get an image built. In some cases, a version upgrade may end up being all that is needed. In other cases, patches or actual code may need to be updated.

Note also that there may be changes needed in submodules. In most cases, changes done in submodules must not break the build on the current Debian version. This is because they may be installed and used in both the base image and in some container. There are two submodules (that I'm currently aware of) that only get installed on the base image. In these cases, it may be reasonable to have a separate development branch in those submodules to make any needed changes (including breaking changes, if needed) and use that until the final merge into master branch. These submodules are:

- src/sonic-linux-kernel
- platform/broadcom/saibcm-modules-dnx

The src/sonic-host-services submodule is used in the docker-sonic-vs container, meaning it needs to (largely) stay compatible with the current Debian version and the new Debian version. Because of this, there may need to be breaking changes in this repo, and this repo may need a new debian specific branch as well.

For all other submodules, any changes needed there should be done in a way that doesn't break anything on the current Debian version. They should be merged into the master branch of that submodule, which will eventually get picked up in a submodule update to the master branch of sonic-net/sonic-buildimage.

Once the VS image is built, and it boots up, at this point, it should be possible to build images for the individual platforms. Note that kernel modules that are built as part of that platform would need to be updated or disabled. In addition, applications/packages that were disabled earlier can now be fixed up and built into the image. During this time, regular code syncs from the master branch should be happening, so as to find any breaking changes in master branch and fix them sooner rather than later. I recommend doing a git rebase of the development branch on top of the master branch, so as to keep the git history for the development branch cleaner.

An estimate of how much time is needed for each task is given below:

Task	Time Estimate
Create slave container	1 week
Update Makefile and Makefile.work to be able to build the new slave container	2 days
Merge into master (or dev) branch of sonic-net/sonic-buildimage	2 days
Define new Azure pipeline to build the new slave container	2 days
Update kernel to build the new version	1.5 weeks
Update slave.mk to build images for the new Debian version	1 day
Make changes in sonic-buildimage to build VS image (with modules	

disabled as needed) | 2.5 week | | Update platform modules and python scripts for different platforms for the new Debian release | 6 week | | Fix up issues/TODOs added when building the VS image | In parallel with previous task. Should be within 2-3 weeks, depending on the scope of issues. | | Ensure that the kernel is stable, and that all images are building and functional | 1 week |

This gives a total estimate of about 10.5-11 weeks (assuming that tasks that can be done in parallel are done in parallel).

## **SONiC Container Debian Upgrade**

This section describes SONiC container Debian upgrade process and cadence.

As explained in previous section base SONiC will be first upgraded to latest Debian release and first SONiC release of Debian upgrade cycle will only target this.

Guidelines and procedure of container Debian upgrade as follows,

### **Guidelines**

- Container upgrade will be targeted from the next release after the base SONiC Debian upgrade. Let us call this release as 'second' release of Debian upgrade cycle.
  - Ex: Bookworm base Debian upgrade in 202311, Container upgrade will be targeted from release 202405
- Following are the list if container which needs upgrade. Further, list is broken into Phase1 and 2. All Phase 1 containers are enabled by default in 'rules/config' or built by default. Also, some of the Phase 2 containers are used for specific use cases. ##### Phase 1
  - database
  - swss and orchagent
  - teamd
  - pmon
  - lldp
  - snmp
  - syncd/saiserver /syncd-rpc
  - frr
  - radvd
  - nat
  - eventd
  - dhcp-relay
  - telemetry
  - macsec
  - sflow
  - mux

## Phase 2

- p4rt
  - gbsyncd
  - iccpd
  - restapi
  - dhcp-server
  - sonic-sdk
  - ptf
  - mgmt-framework
  - PDE
- Phase 1 list container upgrades should be covered in 'second' release. It is also recommended to upgrade Phase 2 containers in 'second' release, but will be best effort only.
    - Ex: Phase 1 list container should be targeted for upgrade to bookworm in 202405 release. Container from phase 2 list can be included in 202405 releases if PR is raised within the time.

## Procedure

- Create docker-base-\ and docker-config-engine-\\.
- Create docker-swss-layer-\\.
  - orchagent, teamd, frr, nat, sflow are using this swss-layer.
- Upgrade Dockerfile.j2 of each container to point to latest Debian.
  - Sometimes this may need packaging updates in some containers.

## Timeline

Task	Time estimate	Description
----- ----- -----		-----
docker	1 week	Create base docker and config-engine
migration for each container	-	Task for respective owners to upgrade the containers.

## SONiC Debian support

This section describes SONiC Debian support process to update SONiC with latest fixes/CVEs from Debian community.

Debian community has three active stable releases named stable, oldstable and oldoldstable([Debian Releases](#)). SONiC releases mapping to these Debian releases needs active support.

This document is targeted at packages which needs manual update in sonic-buildimage. Below is the guidelines and procedure,

## Guidelines

- Target Debian source which needs active support is currently limited to list below. Minor version of these software package will be updated to the latest available from Debian community.
  - Linux Kernel
- Target timelines for Linux Kernel minor version upgrades on different SONiC branches.
  - Branches based on Debian 'Stable' - 6 months.
  - Branches based on Debian 'OldStable' - 6 months.
  - Branches Based on Debian 'OldOldStable' - Based on requirement.
  - Ex: As per the current status below is the mapping of SONiC branches to Debian releases.
    - OldOldStable (Buster) - 202006, 202012, 202106
    - OldStable (Bullseye) - 202111, 202205, 202211, 202305, 202311
    - Stable (Bookworm) - master, 202405(planned)
- Other third-party list below will be updated based on requirement.
  - bash
  - isc-dhcp-relay
  - iproute2
  - iptables
  - kdump-tools
  - ntp
  - protobuf
  - snmpd
  - socat
  - lm-sensors
  - redis
  - FRR
  - ifupdown2
  - libnl3
  - libteam
  - monit
  - openssh
  - ptf
  - libyang
  - ldpd
  - thrift
  - dockerd

## Procedure

- Linux Kernel minor version update.
  - Update the minor version in sonic-linux-kernel.
    - Vendors may need to update or remove the patches.
  - Update sonic-buildimage
    - Update installer file with latest minor version.

- Update makefile related to kernel module from different vendors.
- Vendors may need to update drivers.
- Change needs to be pushed to all target branches.

## Timeline

Task	Time estimate	sonic-linux-kernel changes	sonic-buildimage changes	Backport changes to target branches

## SONiC Debian Deprecation

Deprecation defines when to stop Debian based support for SONiC release/branch.

Deprecation of the older SONiC branch for Debian support will happen after EOL of LTS Debian version. After deprecation of SONiC branch Debian source list will point to use last stable LTS snapshot archive for continued build.