XFLOW RESEARCH

SONiC Utilities Build Guide

Revision History

Revision No.	Description	Editor	Date
1.0	SONiC Utilities Build Guide	Muhammad Danish Rida Hanif	Mar 13, 2023

Table of Contents

Introduction

This document will serve as a guide to test and validate custom changes to sonic-utilities package – in particular the changes related to the CLI. Command-line utilities code is packaged inside a python wheel file that can be deployed in SONiC.

Building a wheel package containing customised source code, and running it into SONiC is trivial, but this guide will help to run unit tests related to SONiC CLI to validate changes. Running unit tests require a lot of dependencies to be met, those are tedious to keep track of manually.

A convenient alternative is to let the build process of sonic-buildimage repo take care of that for users. During the build process, the SONiC build process will take care of all the necessary dependencies and install them inside of a container (known as sonic-slave container). The idea is to stay inside the container once the build process starts to make the utilities wheel file and run unit tests inside this environment.

Intended Audience

The intended audience for this document are software developers and engineers who are making code changes to the sonic-utilities package, specifically the changes related to the Command Line Interface (CLI).

The intended audience should have knowledge of Python programming and be familiar with the SONiC platform. They should also have some understanding of building and deploying software packages using wheel files and containers.

The document will guide them through the process of testing and validating their changes by running unit tests within the sonic-slave container, which is created during the build process of the sonic-buildimage repository.

Environment

OS: Ubuntu 20.04

Dependencies

1. Install git, pip3 and jinja in host build machine
   sudo apt install git
   sudo apt install -y python3-pip
   sudo pip3 install j2cli

2. Install Docker and follow post-installation steps to allow running the 'docker' command without root privileges.
   Validate that docker can run without ‘sudo’ through the command line
   sudo gpasswd -a ${USER} docker
   docker run hello-world

Build sonic-utilities wheel package

Build locally

1. Clone the sonic-buildimage repository and navigate to the local repo through the command line

   git clone https://github.com/sonic-net/sonic-buildimage

   cd sonic-buildimage

2. To clone own fork of sonic-net repository, user can edit the .gitmodules file inside the root directory

   vi .gitmodules

Edit “URL” field of submodule for which user wishes to clone own fork of the repository. Here, Users will add the url field of their own sonic-utilities submodule.

Note: The init process will clone all sonic repositories in order to set up the sonic build process. These all repositories are used by the sonic-utilities container.

3. Initialise the repository through the command

   make init

Note that this step may take hours depending on your internet connection, so please be patient.

This step will clone all of the submodules listed in the .gitmodules file inside the src/directory. After the command completes, head into src/sonic-utilities directory and inspect that your own repository is available. If code changes are in another branch switch the branch by using following command:

   git checkout <branch_name>

4. Configure the build environment for an ASIC type (any platform will do here for sonic-utilities)

   make configure PLATFORM=generic

5. Make the sonic-utilities wheel file while keeping the Bullseye slave container alive to run unit tests in it.

   make NOSTRETCH=1 NOBUSTER=1 KEEP_SLAVE_ON=yes target/python-wheels/bullseye/sonic_utilities-1.2-py3-none-any.whl

6. When the build finishes, your prompt will change indicating that users are inside the sonic-slave container. Navigate to sonic-utilities directory inside the src folder.

   cd src/sonic-utilities

7. Now users can make changes inside VS Code or their preferred code editor inside the sonic-buildimage/src/sonic-utilities directory.

8. To test changes, run the command inside the sonic-slave container
   python3 setup.py test
   The above command will run all the unit tests associated with SONiC CLI. To run an **individual test** use the command

   pytest-3 tests/<name_of_test_file>

   Useful options to above command

   -vv: Verbose output. Shows all of the individual functions run within the file.

   -rP: Show standard output while running the test.

   e.g. pytest-3 -rP -vv tests/vlan_test.py

9. Some tests might fail while running tests/building sonic-utilities in a local environment, they are:

- FAILED tests/disk_check_test.py::TestDiskCheck::test_readonly
- FAILED tests/drops_group_test.py::TestDropCounters::test_show_counts
- FAILED tests/drops_group_test.py::TestDropCounters::test_show_counts_with_group
- FAILED tests/drops_group_test.py::TestDropCounters::test_show_counts_with_type

   These tests pass in Azure pipelines under the same piece of code, so users can safely ignore them in a local build if they appear as FAILED.

10. If the rest of the tests pass, the user can be sure that the changes are valid and will not cause the SONiC CLI to break. After validating the changes, user can run build the utilities wheel file through the command

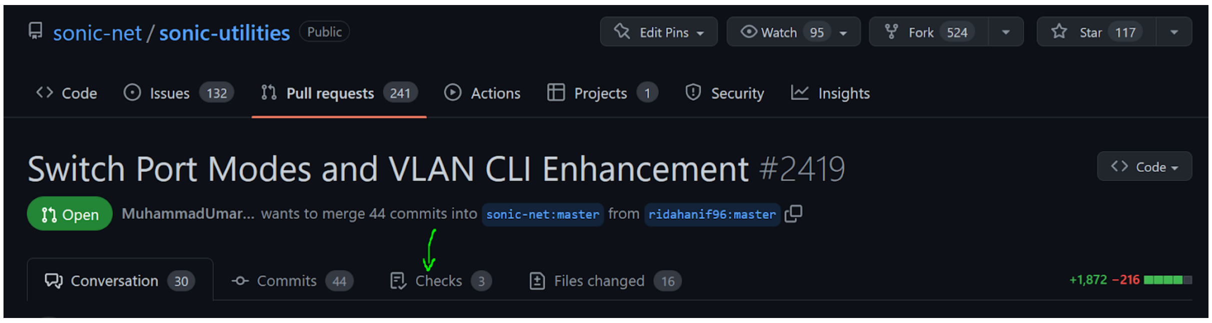
    python3 setup.py bdist_wheel

    The wheel file will be created under sonic-utilities/dist directory.

Build through CI/CD Azure SONiC Pipelines

Users can also create a draft PR on the sonic-utilities repository on GitHub. The changes will automatically pass through the Azure CI Pipelines upon pushing a commit.

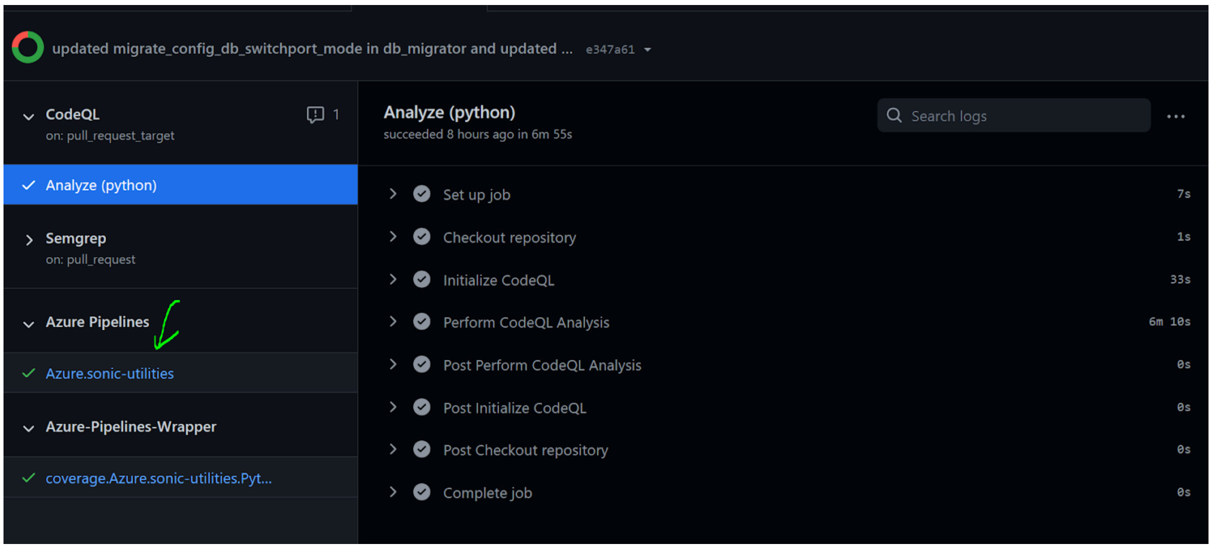
To see the output of the tests or to monitor the tests as they are running, user can navigate to the 'Checks' section. I'm using sonic-utilities#2419 as an example PR.



Switch Port Modes and VLAN CLI Enhancement #2419

Open MuhammadUmar... wants to merge 44 commits into sonic-net:master from ridahanif96:master

Navigate to Azure Pipelines tests section

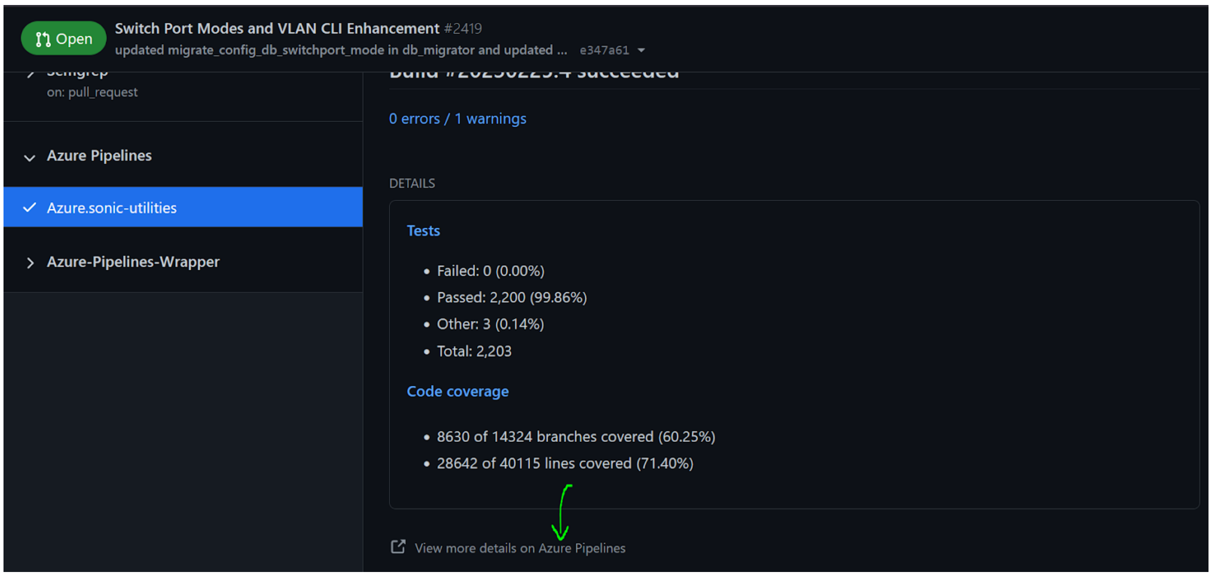


updated migrate_config_db_switchport_mode in db_migrator and updated ... e347a61

Analyze (python)
succeeded 8 hours ago in 6m 55s

```
Set up job                 7s
Checkout repository        1s
Initialize CodeQL          33s
Perform CodeQL Analysis    6m 10s
Post Perform CodeQL Analysis  0s
Post Initialize CodeQL     0s
Post Checkout repository   0s
Complete job               0s
```

scroll down now to find the link to the Azure Pipelines


Open Switch Port Modes and VLAN CLI Enhancement #2419
updated migrate_config_db_switchport_mode in db_migrator and updated ... e347a61

on: pull_request

Azure Pipelines
Azure.sonic-utilities
Azure-Pipelines-Wrapper

0 errors / 1 warnings

DETAILS

Tests

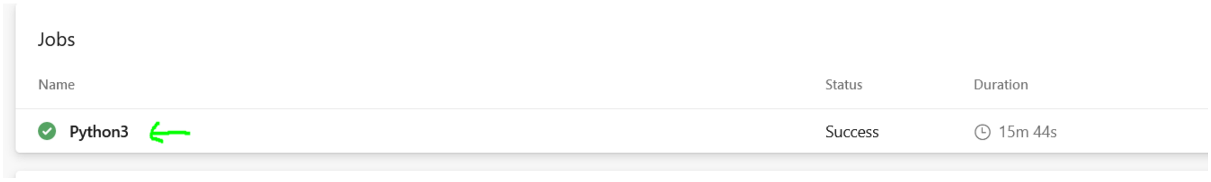
- Failed: 0 (0.00%)
- Passed: 2,200 (99.86%)
- Other: 3 (0.14%)
- Total: 2,203

Code coverage

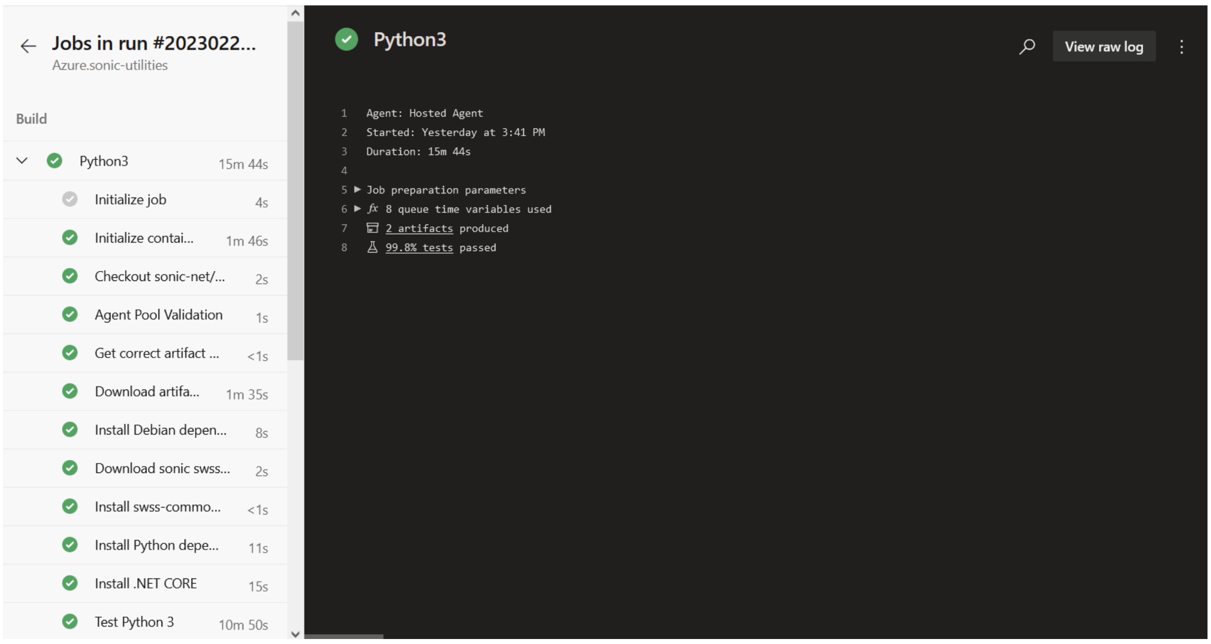
- 8630 of 14324 branches covered (60.25%)
- 28642 of 40115 lines covered (71.40%)

View more details on Azure Pipelines

Click on the link that’ll redirect to dev.azure.com pipelines page. User ’ll find the output on the tests under the Jobs section


Jobs

Name	Status	Duration
Python3	Success	15m 44s


Jobs in run #2023022...
Azure.sonic-utilities

Build

Python3	15m 44s
Initialize job	4s
Initialize contai...	1m 46s
Checkout sonic-net/...	2s
Agent Pool Validation	1s
Get correct artifact ...	<1s
Download artifa...	1m 35s
Install Debian depen...	8s
Download sonic swss...	2s
Install swss-commo...	<1s
Install Python depe...	11s
Install .NET CORE	15s
Test Python 3	10m 50s

Python3 View raw log

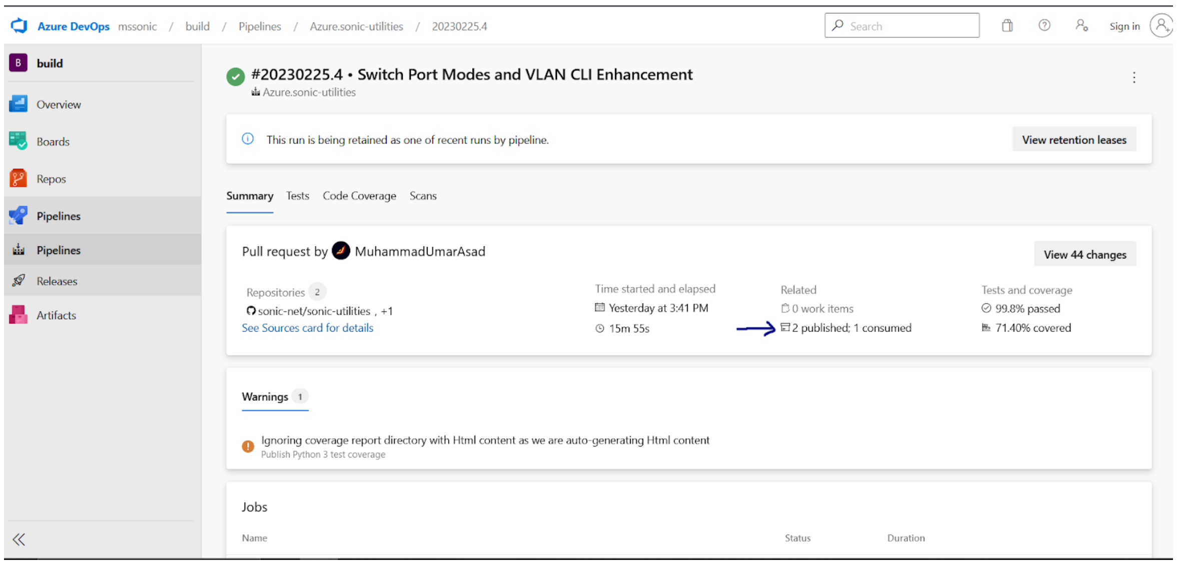
```
Agent: Hosted Agent
Started: Yesterday at 3:41 PM
Duration: 15m 44s

▶ Job preparation parameters
▶ fx 8 queue time variables used
2 artifacts produced
99.8% tests passed
```

If all the tests pass, user will have a downloadable artifact containing the wheel file to run on SONiC

Download the Artifact

Under the summary section, find the list of produced artifacts and download them on your local machine.



#20230225.4 • Switch Port Modes and VLAN CLI Enhancement
Azure.sonic-utilities

This run is being retained as one of recent runs by pipeline.

Pull request by MuhammadUmarAsad

Repositories 2	Time started and elapsed	Related	Tests and coverage
sonic-net/sonic-utilities , +1	Yesterday at 3:41 PM	0 work items	99.8% passed
See Sources card for details	15m 55s	2 published; 1 consumed	71.40% covered

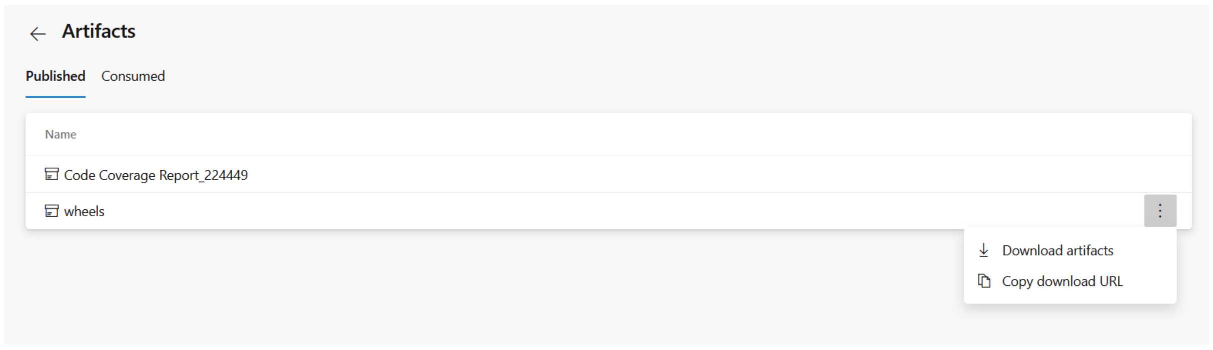
Warnings 1

Ignoring coverage report directory with Html content as we are auto-generating Html content
Publish Python 3 test coverage

Jobs

Name	Status	Duration

Us can download the **"wheels"** artifacts that would be a zip file containing the sonic-utilities wheel file produced as a result of the build run.



Extract the wheel file into a file into a folder and follow the next instructions to run it on SONiC.

Note: The build run is cleaned after a few days, so the artifacts generated through this method are available for a limited time. They would be re-run if a new commit is pushed thereby generating a fresh artifact.

Run wheel file in SONiC

To incorporate the changes into SONiC, users need to replace the existing sonic-utilities package inside the OS with your newly created package.

1. Start a SONiC instance and log in with your credentials.

2. Delete the current CLI package through pip

   sudo pip uninstall sonic_utilities

3. Navigate to the directory where the newly created wheel file is located. Run a python web server inside the directory to install it into SONiC through the web.

   python3 -m http.server <port> --bind <ip_address>

   Example: python3 -m http.server 8000 –bind 192.168.1.64

4. Install the wheel file inside SONiC through the command

   sudo pip install http://<ip_addr>:<port>/sonic_utilities-1.2-py3-none-any.whl

5. Voila! Your changes are now present inside SONiC.