

DPEN Compute Project®

Testbed in a Box

SONiC Case Study

Connect. Collaborate. Accelerate.



Agenda

- Testbed in a box
- SONiC testing, current state
- SONiC Community Test bed
- SONiC Testbed in a box
 - A peek inside
 - Configuration simplified
 - Benefits
- Demo



Testbed in a box





- SONiC successfully led the disaggregation through open NW Operating System(NOS)
- Disaggregation requires additional testing & validation
 - Testing functionality of each layer
 - Testing for interoperability between multiple vendors
 - Testing end to end performance
 - Verifying vendor's compliance
- Test Working Group developed test bed topology
- Testbed in a box is our approach to simplification



What's required to build one

- Hardware/Software
 - Multiple servers

Compute

- Root/Fan-out switches
- Multiple VMs for vEOS
- Multiple PTF containers
- Configuration
 - Multiple configuration files, formats, locations and dependencies
 - Feature rich CLI commands prone to human errors
- Building & stabilizing is time consuming





SONiC Testbed in a box

Orchestration built in to abstract complex steps from users

- Build virtual root, fan-out switches and connectivity using SDN
- Create cEOS and PTF instances
- Use lightweight cEOS instead of vEOS
- End to end workflow using ansible scripts

Testing

- Verified capacity & capabilities of the virtual instances and connectivity
- Ran the community test scripts from Github and similar results observed







A Peek Inside



What does Orchestration Look Like?

Controller

• Central entity that manages complexity of testbed orchestration, test logs, test reports, SDN etc.

Client

- Simple yet powerful user-facing CLI that can talk to the Controller
- Idiomatic commands to manage testbed, run tests, get test logs, get test reports, manage licenses etc.



Configuration Simplified



| topo start [flags] | |
|--------------------|----------------------------------|
| Flags: | |
| dut-host string | IP of DUT |
| dut-hwsku string | DUT type. |
| dut-pass string | Password for DUT |
| dut-user string | User for |
| topology string | <pre>topology type(t1, t0)</pre> |

Connect. Collaborate. Accelerate.



Benefits

 No need of multiple configuration files
Single CLI command to build the test bed connection and execute the suite

Simplification



 Avoid serial allocation of the setup

 Each dev engineer could have access to their own setup to develop
test & fix faster

Increase Efficiency & Higher productivity OCP form factor 1U box
~3X power & space

reduction

Space & Power



- Supports CI/CD
- Seamless integration into automation

Ease of Operation





Demo

Connect. Collaborate. Accelerate.