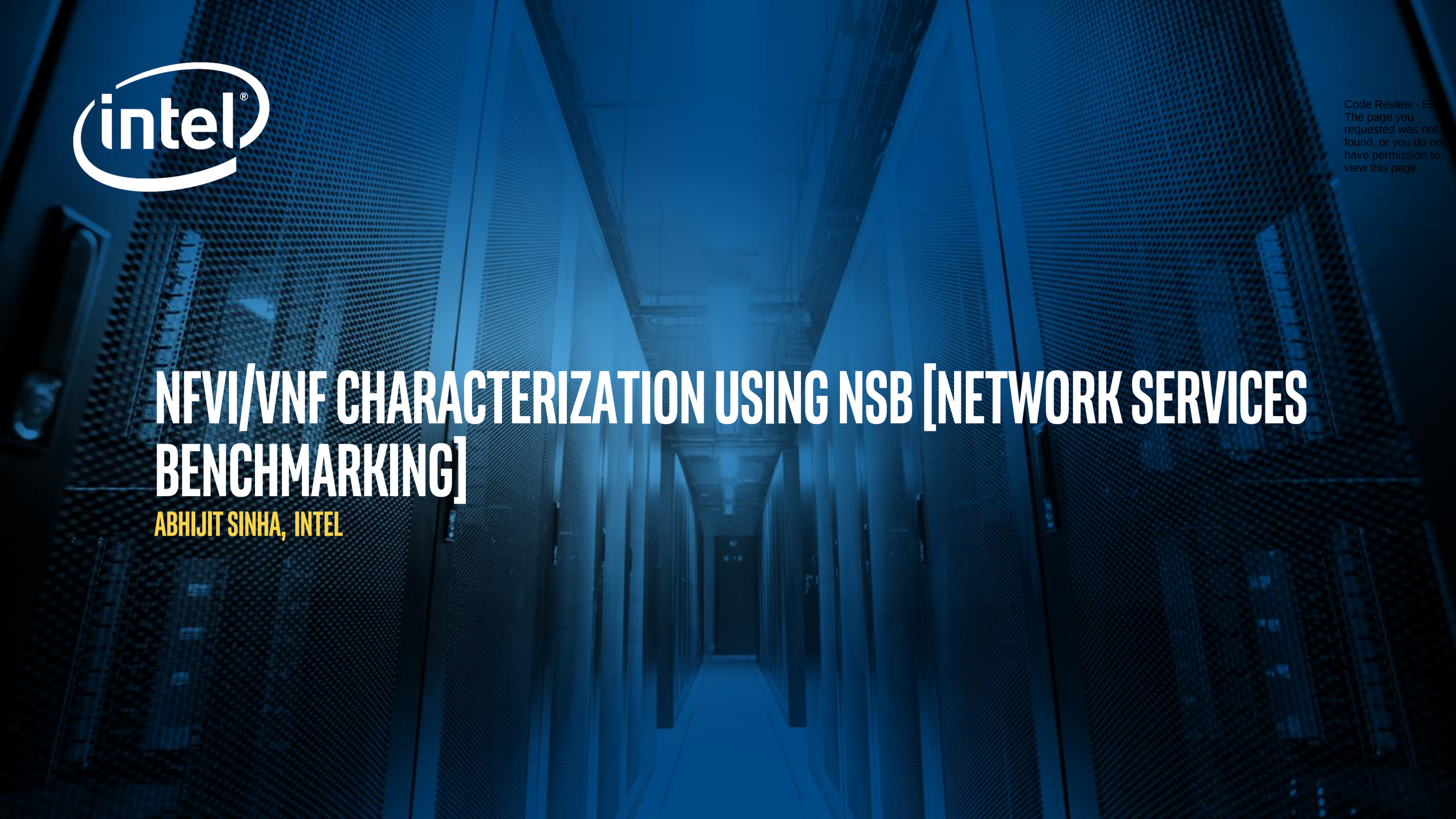


Code Review - Er
The page you requested was not found, or you do not have permission to view this page.

NFVI/VNF CHARACTERIZATION USING NSB [NETWORK SERVICES BENCHMARKING]

ABHIJIT SINHA, INTEL

LEGAL NOTICES AND DISCLAIMERS

- Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration **No product or component can be absolutely secure**. Check with your system manufacturer or retailer or learn more at intel.com.
- Intel, the Intel logo, [List the Intel trademarks in your document] are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.
- *Other names and brands may be claimed as the property of others.
-

Agenda

- Intro to NSB
- NSB Methodology
- NSB Architecture and OPNFV Yardstick
- Test case execution flow
- NFVi Characterization using NSB
- Workshop Session
- Questions

Why Network Services Benchmarking?

General Lack of Telco Grade Conformance/ Benchmarks

Unclear Network Workload Dimensions and Stress Vectors

Missing System Level Capacity Requirements

Network Workload Scalability/ Agility Implications on NFV

Operators **lack comprehensive information for TCO** models to plan, procure and deploy NFV

Many tools and benchmarks

Challenge of NFV characterization

Test should be realistic and repeatable.

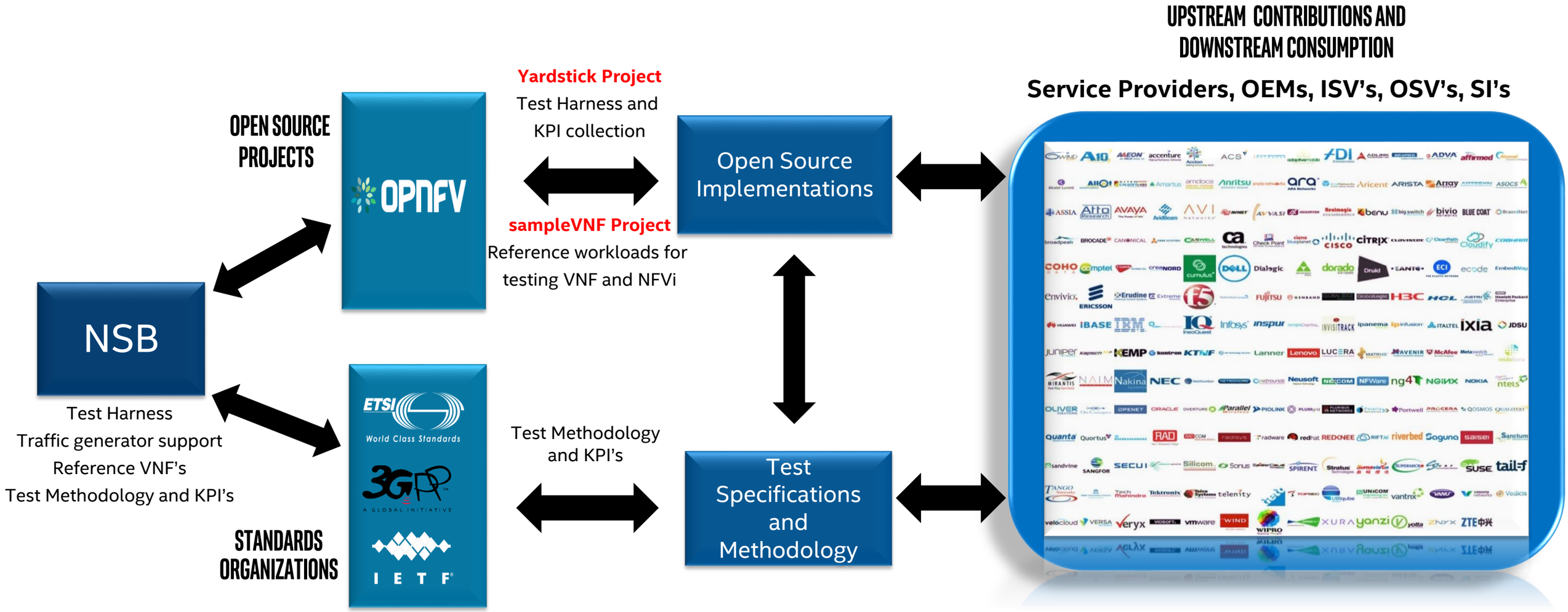
Main metrics and KPIs should be established

Results should be independent of the generators used.

What is NSB?

- NSB is a benchmarking and characterisation tool that automates NFVi and VNF characterisation.
- Combination of many open source components to facilitate deterministic and repeatable benchmarking
- NSB characterisation provides simultaneous views of Network, NFVi and application metrics for a given test scenario.
- Presents the metrics in a unified fashion for the user to examine and do analysis

NSB Open Source and Standards



*Other names and brands may be claimed as the property of others

NSB provides

Test Framework

- Performance benchmarking of a Virtual Network Function (VNF) in a Network Functions Virtualization (NFV) environment
- Performance benchmarking of Network Function Virtualization infrastructure (NFVi).

Test cases

Test case examples are provided for running RFC2544, RFC3511 and 3GPP tests with different VNFs, traffic profiles and scaling factors (CPU, Memory, Ports, etc.)

Contexts

VNF and NFVi characterization and benchmarking can be executed in three different environments:

- Baremetal
- Standalone virtualization
- Managed virtualized environment e.g. Openstack*, Kubernetes*

Traffic generator Support

NSB supports **external software and hardware traffic generators**
e.g. T-Rex*, IxLoad*, IxNetwork*, Spirent Landslide*, PROX

KPIs

NSB provides **test automation** and **collection** of

- Network KPIs
- NFVi KPIs
- VNF KPIs

*Other names and brands may be claimed as the property of others

NSB Methodology

- Baremetal
- Standalone VIRT
- Managed VIRT

- Use case definition

- Packet format
- Flow distribution
- Stateless/stateful

- Throughtput
- Latency
- CPU, Cache
- Memory
- …

Define NFVi context

Define workload

Define traffic pattern

Define KPIs

Configuration

- System config.
- Traffic config.

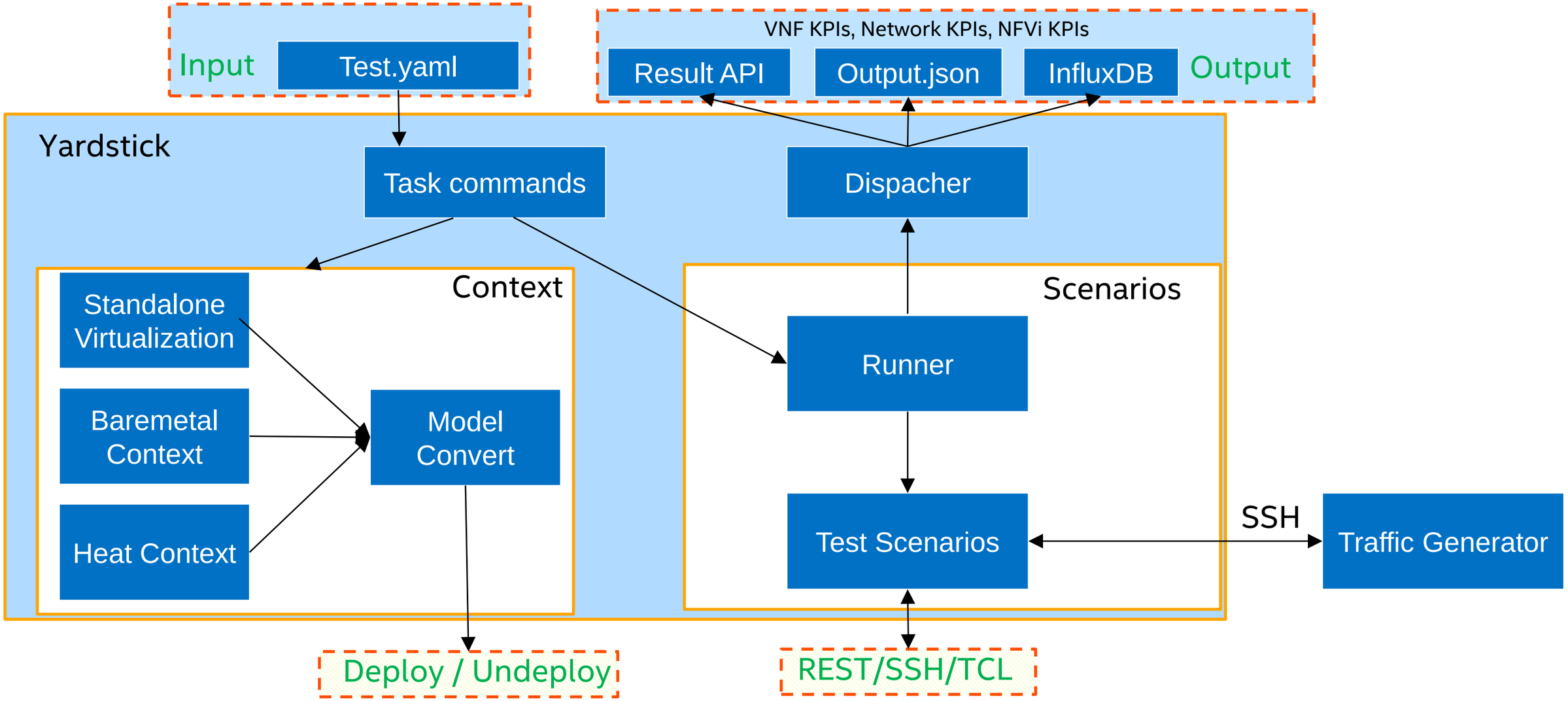
Execution & Data Collection

- Execute Tests
- Results summary

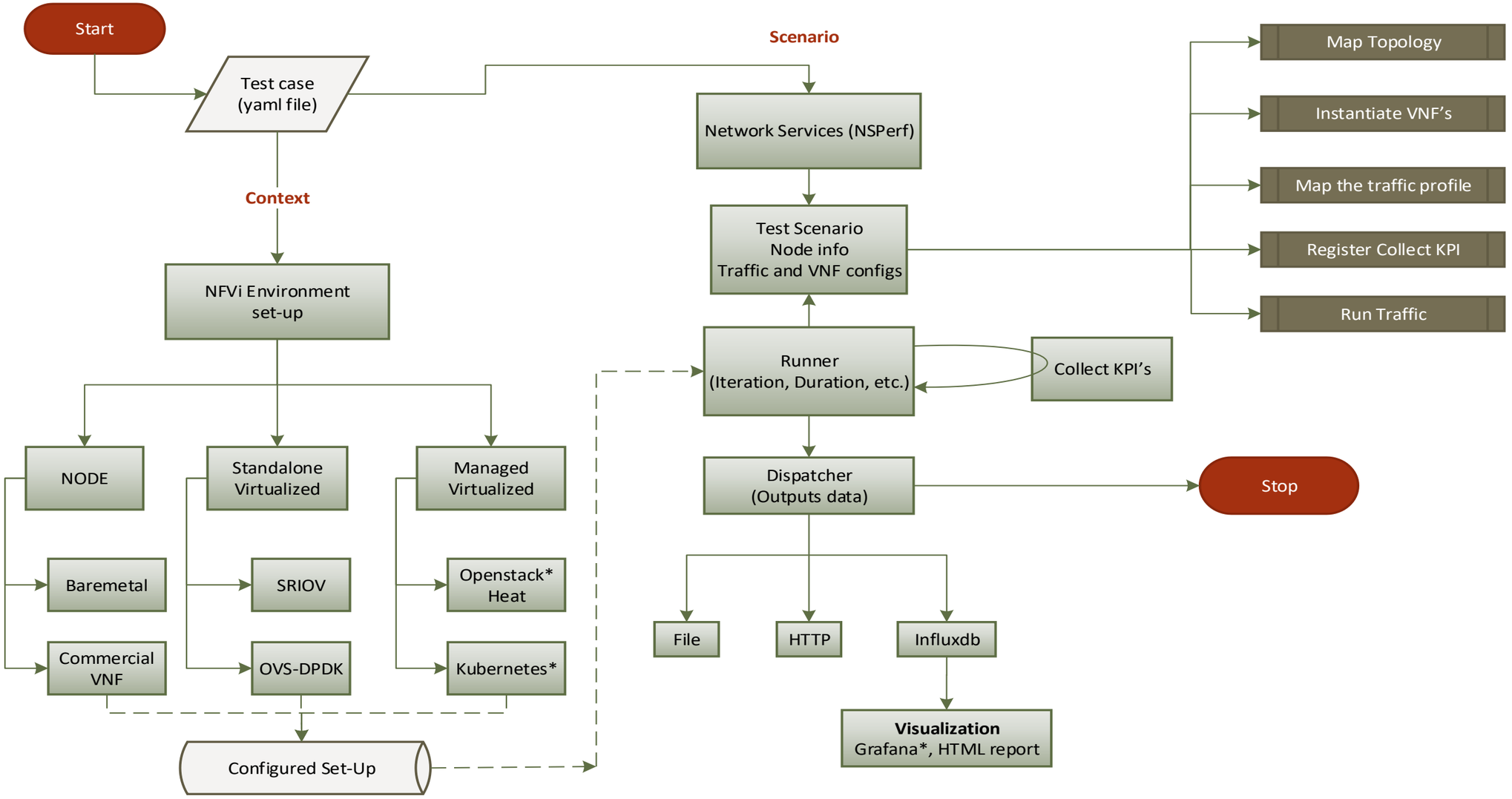
Benchmark Output

- Raw data
- Visualizations

NSB - Yardstick logical view

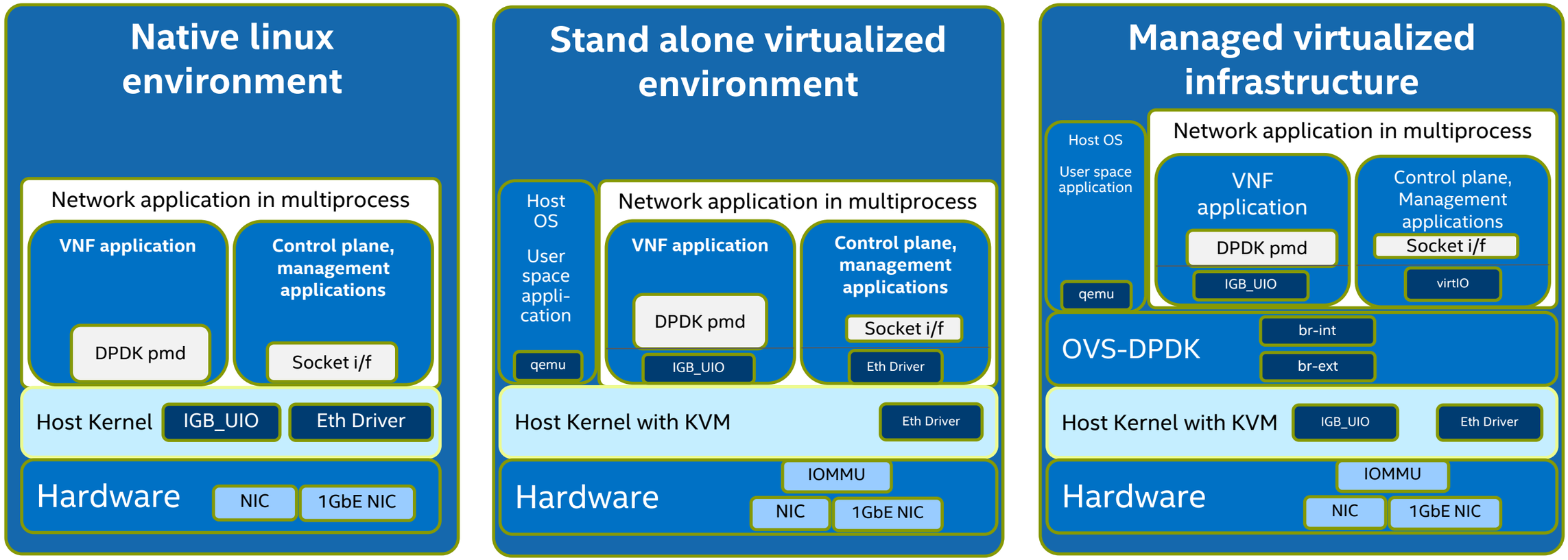


NSB – Yardstick Test Execution Flow



*Other names and brands may be claimed as the property of others

Benchmarking environments



*Other names and brands may be claimed as the property of others

NSB – Test case view

>>
samples/vnf_samples/nsut/vfw/**tc_baremetal_rfc2544_ipv4_1rule_1flow_64B_trex.yaml**

```
---
schema: yardstick:task:0.1
scenarios:
- type: NSPerf
  traffic_profile: ../../traffic_profiles/ipv4_throughput.yaml
  topology: vfw-tg-topology.yaml
  nodes:
    tg__0: trafficgen_1.yardstick
    vnf__0: vnf.yardstick
  options:
    framesize:
      uplink: {128B: 100}
      downlink: {128B: 100}
    flow:
      src_ip: [{'tg__0': 'xe0'}]
      dst_ip: [{'tg__0': 'xe1'}]
      count: 1
    traffic_type: 4
    rfc2544:
      allowed_drop_rate: 0.0001 - 0.0001
    vnf__0:
      rules: acl_1rule.yaml
      vnf_config: {lb_config: 'SW', lb_count: 1, worker_config: '1C/1T', worker_threads: 1}
runner:
    type: Iteration
    iterations: 10
    interval: 35
```

Baremetal/Node

```
context:
  type: Node
  name: yardstick
  nfvi_type: baremetal
  file: /etc/yardstick/nodes/pod.yaml
```

NSB – Test case view …

Standalone: SRIOV

```
contexts:
  - name: yardstick
    type: Node
    file: /etc/yardstick/nodes/standalone/trex_bm.yaml

  - type: StandaloneSriov
    file: /etc/yardstick/nodes/standalone/host_sriov.yaml
    name: yardstick
    vm_deploy: True

    flavor:
      images: "/var/lib/libvirt/images/yardstick-nsb-image.img"
      ram: 16384
      extra_specs:
        hw:cpu_sockets: 1
        hw:cpu_cores: 6
        hw:cpu_threads: 2
      user: ""
      password: ""
```

```
servers:
    vnf__0:
      network_ports:
        mgmt:
          cidr: '1.1.1.61/24'
        xe0:
          - uplink_0
        xe1:
          - downlink_0

networks:
    uplink_0:
      phy_port: "0000:05:00.0"
      vpci: "0000:00:07.0"
      cidr: '152.16.100.10/24'
      gateway_ip: '152.16.100.20'
    downlink_0:
      phy_port: "0000:05:00.1"
      vpci: "0000:00:08.0"
      cidr: '152.16.40.10/24'
      gateway_ip: '152.16.100.20'
```

NSB – Test case view …

Standalone: OVS_DPDK

```
contexts:
  - name: yardstick
    type: Node
    file: /etc/yardstick/nodes/standalone/trex_bm.yaml
  - type: StandaloneOvsDpdk
    name: yardstick
    file: /etc/yardstick/nodes/standalone/pod_ovs.yaml
    vm_deploy: True
    ovs_properties:
      version:
        ovs: 2.7.0
        dpdk: 16.11.1
      pmd_threads: 2
      ram:
        socket_0: 2048
        socket_1: 2048
      queues: 4
      vpath: "/usr/local"
    flavor:
      images: "/var/lib/libvirt/images/yardstick-nsb-image.img"
      ram: 16384
      extra_specs:
        hw:cpu_sockets: 1
        hw:cpu_cores: 6
        hw:cpu_threads: 2
```

```
servers:
    vnf__0:
      network_ports:
        mgmt:
          cidr: '1.1.1.7/24'
        xe0:
          - uplink_0
        xe1:
          - downlink_0
  networks:
    uplink_0:
      port_num: 0
      phy_port: "0000:05:00.0"
      vpci: "0000:00:07.0"
      cidr: '152.16.100.10/24'
      gateway_ip: '152.16.100.20'
    downlink_0:
      port_num: 1
      phy_port: "0000:05:00.1"
      vpci: "0000:00:08.0"
      cidr: '152.16.40.10/24'
      gateway_ip: '152.16.100.20'
```

NSB – Test case view …

Openstack Heat with OVS/OVS-DPDK

```
context:
name: yardstick
  image: yardstick-samplevnfs
  flavor:
    vcpus: 10
    ram: 20480
    disk: 6
    extra_specs: [Add EPA/HPA features here]
      hw:cpu_sockets: 1
      hw:cpu_cores: 10
      hw:cpu_threads: 1
      hw:mem_page_size: large [enable for ovs-dpdk]
  user: ubuntu
  placement_groups:
    pgrp1:
      policy: "availability"
  servers:
    vnf:
      floating_ip: true
      placement: "pgrp1"
    trafficgen_1:
      floating_ip: true
      placement: "pgrp1“
```

```
networks:
  mgmt:
    cidr: '10.0.1.0/24'
  uplink_0:
    cidr: '10.0.2.0/24'
    gateway_ip: 'null'
    port_security_enabled: False
    enable_dhcp: 'false'
  downlink_0:
    cidr: '10.0.3.0/24'
    gateway_ip: 'null'
    port_security_enabled: False
    enable_dhcp: 'false'
  uplink_1:
    cidr: '10.0.4.0/24'
    gateway_ip: 'null'
    port_security_enabled: False
    enable_dhcp: 'false'
  downlink_1:
    cidr: '10.0.5.0/24'
    gateway_ip: 'null'
    port_security_enabled: False
    enable_dhcp: 'false‘
```

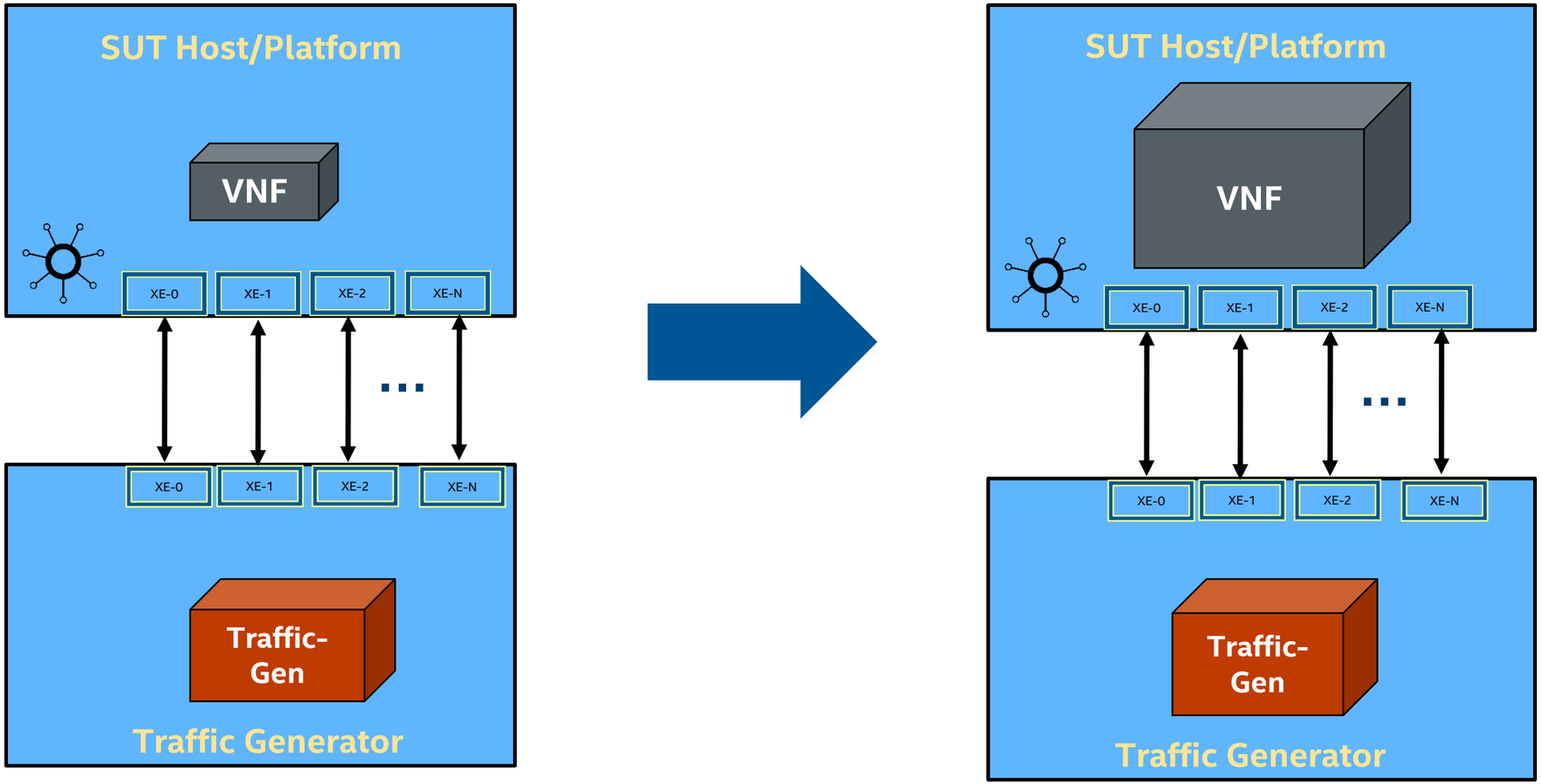
NSB – Test case view …

Openstack Heat with SRIOV

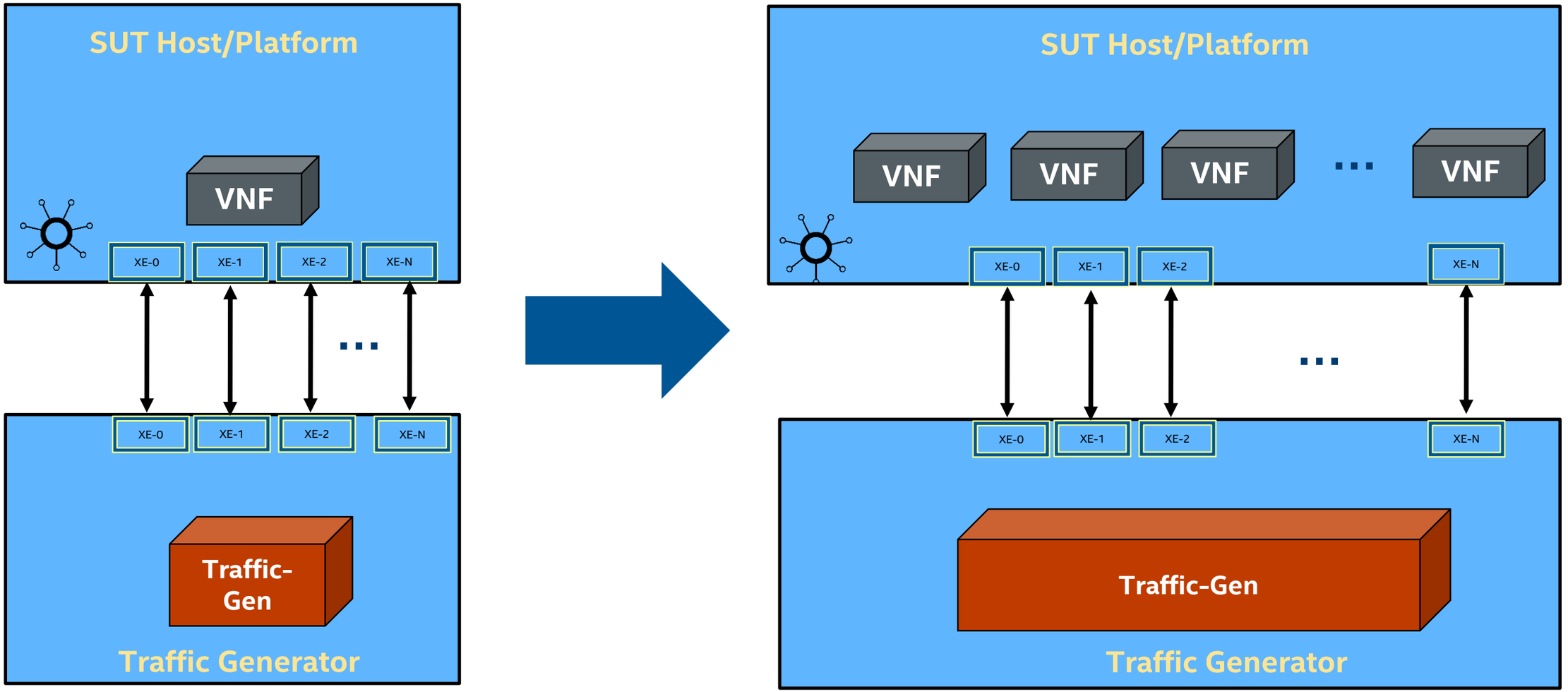
```
context:
name: yardstick
  image: yardstick-samplevnfs
  flavor:
    vcpus: 10
    ram: 20480
    disk: 6
    extra_specs: [Add EPA/HPA features here]
      hw:cpu_sockets: 1
      hw:cpu_cores: 10
      hw:cpu_threads: 1
user: ubuntu
  placement_groups:
    pgrp1:
      policy: "availability"
  servers:
    vnf:
      floating_ip: true
      placement: "pgrp1"
    trafficgen_1:
      floating_ip: true
      placement: "pgrp1“
 networks:
      mgmt:
        cidr: '10.0.1.0/24'
```

```
uplink_0:
        cidr: '10.1.0.0/24'
        gateway_ip: 'null'
        provider: true
        physical_network: phystenant1
        port_security_enabled: false
        enable_dhcp: 'false'
      downlink_0:
        cidr: '10.2.0.0/24'
        gateway_ip: 'null'
        provider: true
        physical_network: phystenant2
        port_security_enabled: false
        enable_dhcp: 'false'
      uplink_1:
        cidr: '10.3.0.0/24'
        gateway_ip: 'null'
        provider: true
        physical_network: phystenant3
        port_security_enabled: false
        enable_dhcp: 'false'
      downlink_2:
        cidr: '10.4.0.0/24'
        gateway_ip: 'null'
        provider: true
        physical_network: phystenant4
        port_security_enabled: false
        enable_dhcp: 'false'
```

Scale Up Tests – Vertical Scaling

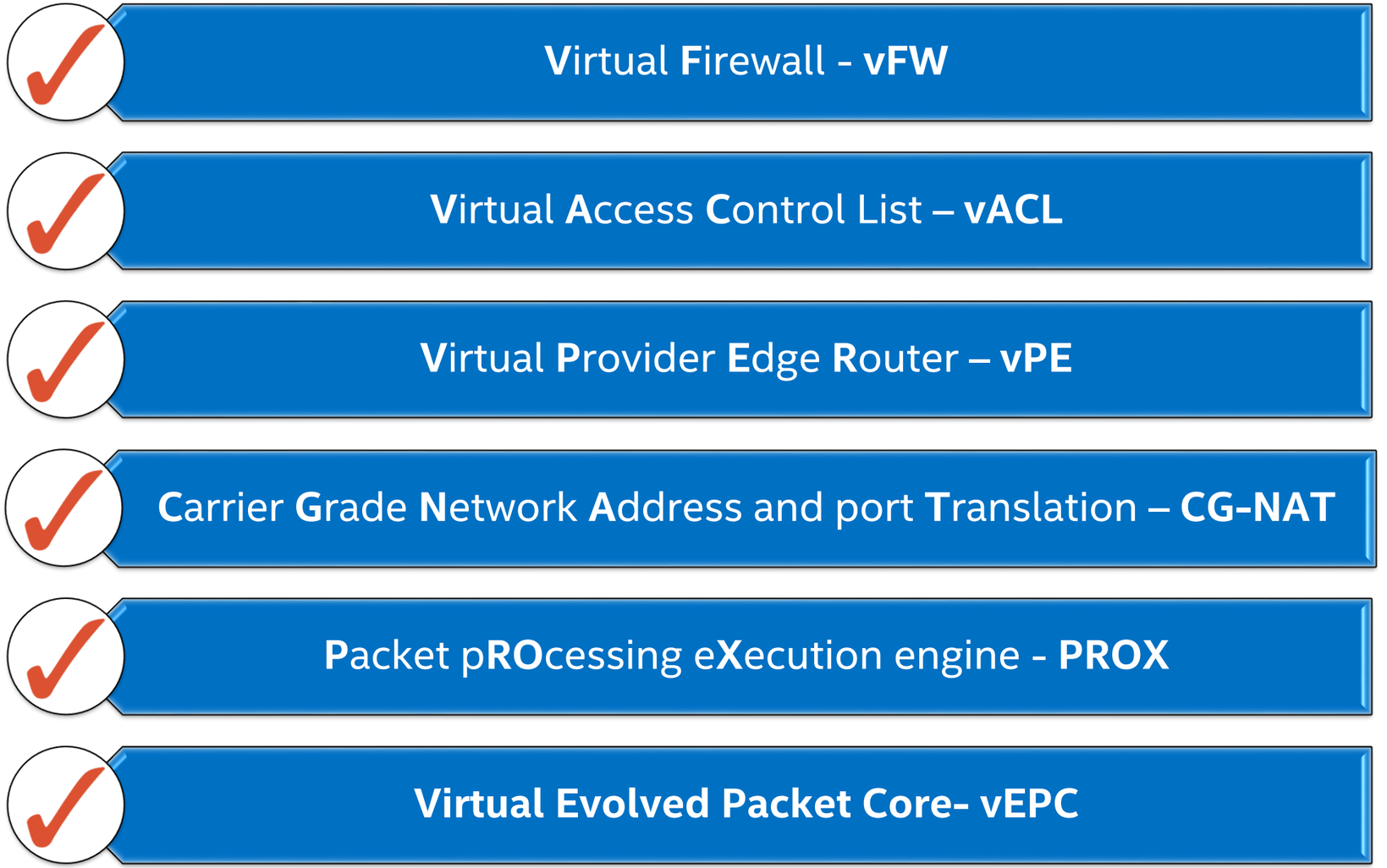


Scale Out Tests – Horizontal Scaling



Scale out test: https://github.com/opnfv/yardstick/blob/master/samples/vnf_samples/nsut/vfw/tc_heat_rfc2544_ipv4_1rule_1flow_64B_trex_scale_out.yaml

VNFs Supported (OPNFV Yardstick Gambia release)



NFVI CHARACTERIZATION

NFVi characterization methodology

Execute defined VNF's on SUT

Run test on
- Baremetal,
- Standalone virtualization
- Managed virtualized:
  - OpenStack, K8s

Test traffic

- Corresponding to VNF requirements

Configuration

- Validation of correct test harness configuration for NFVi testing

Test case definition

- Definition based on test-case VNF's

Data collection

- Benchmark run summarization, KPI's collections

Benchmarking

- Test suites under multiple scenarios and traffic intensities

Benchmark output

- Raw data
- Visualization

NFVi workloads

Workloads should be easy, simple to explain and be able to highlight specific aspect to NFVi (Configuration, OS, Hardware)

Example			
L2 Forward	Forward packets after tagging/ untagging	ACL	Buffering

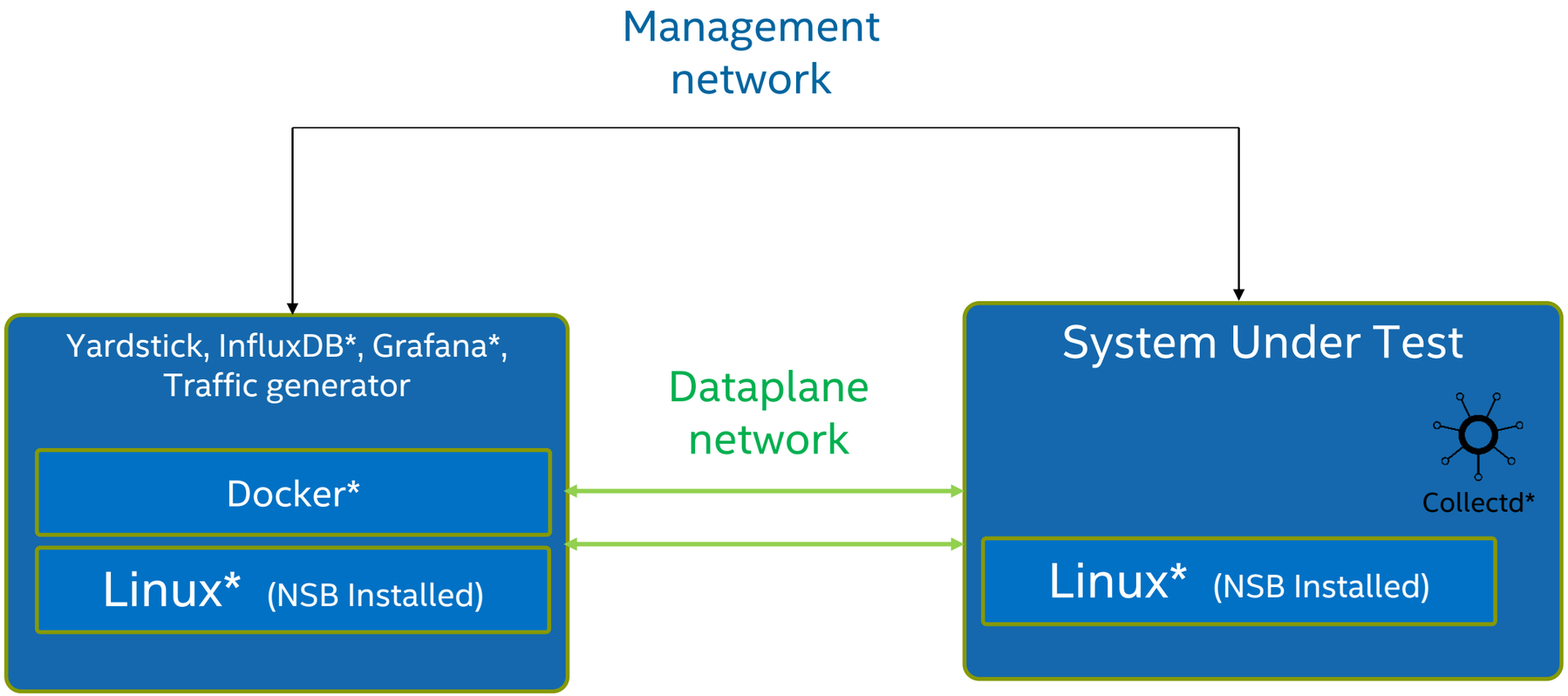
Test Cases in current NFVI Test Suite

- L2 Packet forwarding VNF [Baseline]
- Multi-flow L2 Forwarding – 200K flows
- L3 Forwarding - packet forwarding modifying MACs
- MPLS tagging – protocol conversion, adding/removing MPLS tag, packet length variation
- ACL – flow matching Access Control List, complex packet filtering
- LB / 5-tuple lookup – 5-tuple based flow matching table lookups for load balancing
- Buffering – packet flow buffering for at least 125ms, stresses cache and memory
- BNG: ARP, QinQ, LB, Routing, GRE, MPLS
- BNG + QoS
- vPE - ACL filtering, flow classification, routing (LPM lookup), metering, policing & marking
- lw-AFTR – lightweight Address Family Translation Router: IPv4 <-> IPv6

NFVI CHARACTERIZATION

Configuration files & Test run

Basic Test Setup



*Other names and brands may be claimed as the property of others

NFVi test setup

NSB:

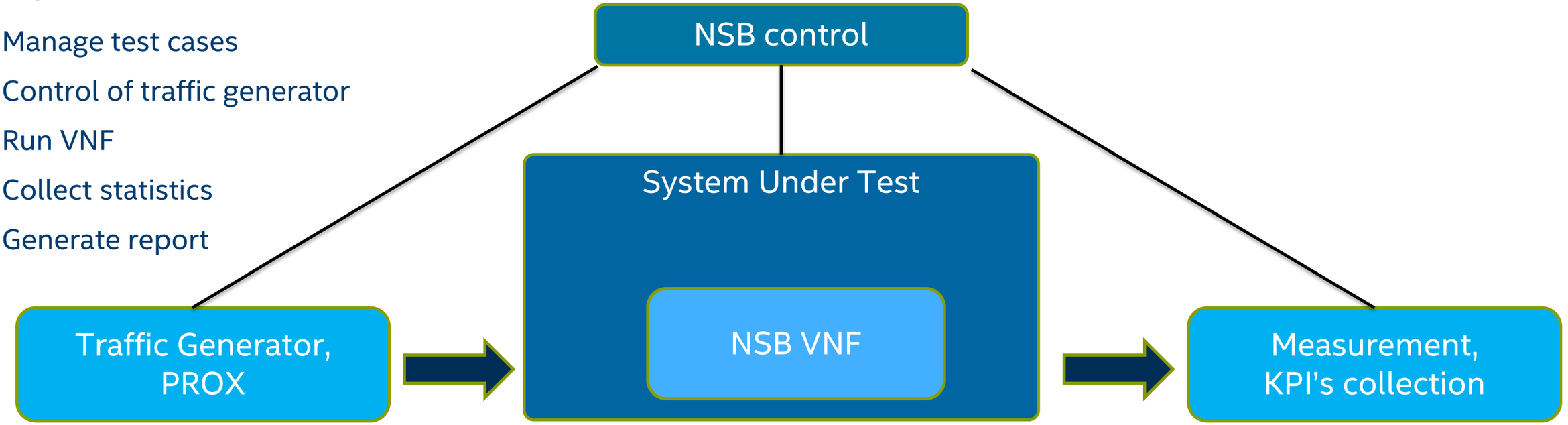
Manage test cases

Control of traffic generator

Run VNF

Collect statistics

Generate report



Configuration files

Configuration files need to be edited

yardstick.conf – contains the basic configuration of yardstick

prox-baremetal-2.yaml – contains info of SUT and TG

tc_prox_baremetal_l2fwd-2.yaml – contains test configuration

yardstick.conf

vim /etc/yardstick/yardstick.conf

[DEFAULT]

debug = False

dispatcher = influxdb

[dispatcher_http]

timeout = 5

target = http://127.0.0.1:8000/results

[dispatcher_file]

file_path = /tmp/yardstick.out

max_bytes = 0

backup_count = 0

[dispatcher_influxdb]

timeout = 5

target = http://192.168.0.101:8086

db_name = yardstick

username = root

password = rootpassword

[nsb]

trex_path = /opt/nsb_bin/trex/scripts

bin_path = /opt/nsb_bin

trex_client_lib = /opt/nsb_bin/trex_client/stl

Pod configuration file

/home/opnfv/repos/yardstick/samples/vnf_samples/nsut/prox/prox-baremetal-2.yaml

```
nodes:
-
   name: "tg_0"
   role: TrafficGen
   ip: 1.1.1.1
   user: "root"
   ssh_port: "22"
   password: "rootpassword"
   interfaces:
      xe0:
         vpci: "0000:86:00.0"
         local_mac: "3c:fd:fe:bb:cc:dd"
         driver: "i40e"
         local_ip: "152.16.100.19"
         netmask: "255.255.255.0"
         dpdk_port_num: 0
```

Configuration test file

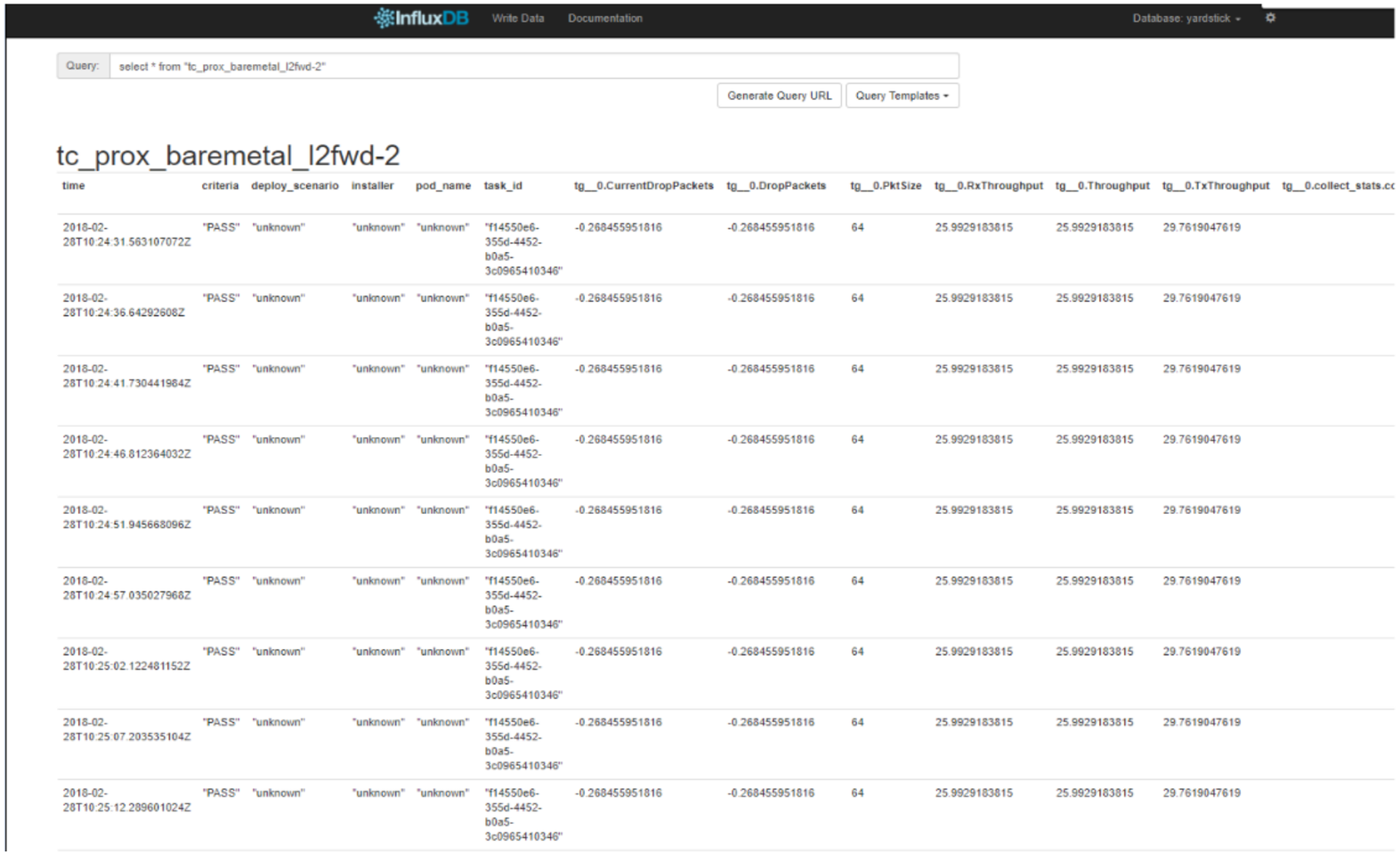
/home/opnfv/repos/yardstick/samples/vnf_samples/nsut
/prox# tc_prox_baremetal_l2fwd-2.yaml

```
traffic_profile: ../../traffic_profiles/prox_binsearch.yaml
 topology: prox-tg-topology-2.yaml

 nodes:
   tg__0: tg_0.yardstick
   vnf__0: vnf_0.yardstick

 options:
   vnf__0:
     prox_path: /opt/nsb_bin/prox
     prox_config: "configs/handle_l2fwd-2.cfg“
     prox_args:
       "-t": ""
```

```
tg__0:
prox_path: /opt/nsb_bin/prox
      prox_config: "configs/gen_l2fwd-2.cfg"
      prox_args:
        "-e": ""
        "-t": ""
  runner:
    type: Duration # we kill after duration, independent of test duration, so set this high
    duration: 300
context:
  type: Node
  name: yardstick
  nfvi_type: baremetal
  file: prox-baremetal-2.yaml
```

InfluxDB example results

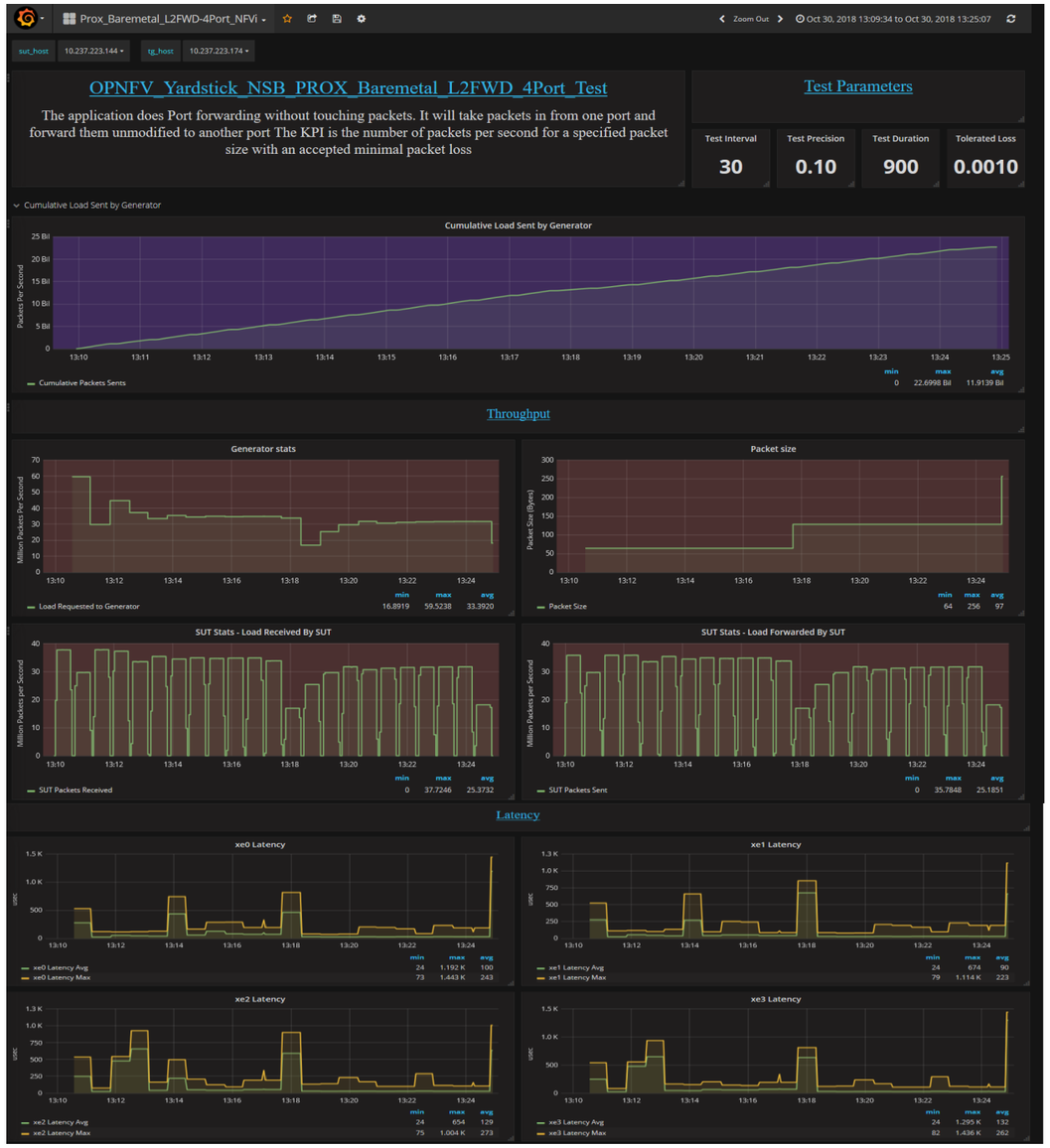
Prox L2 FWD

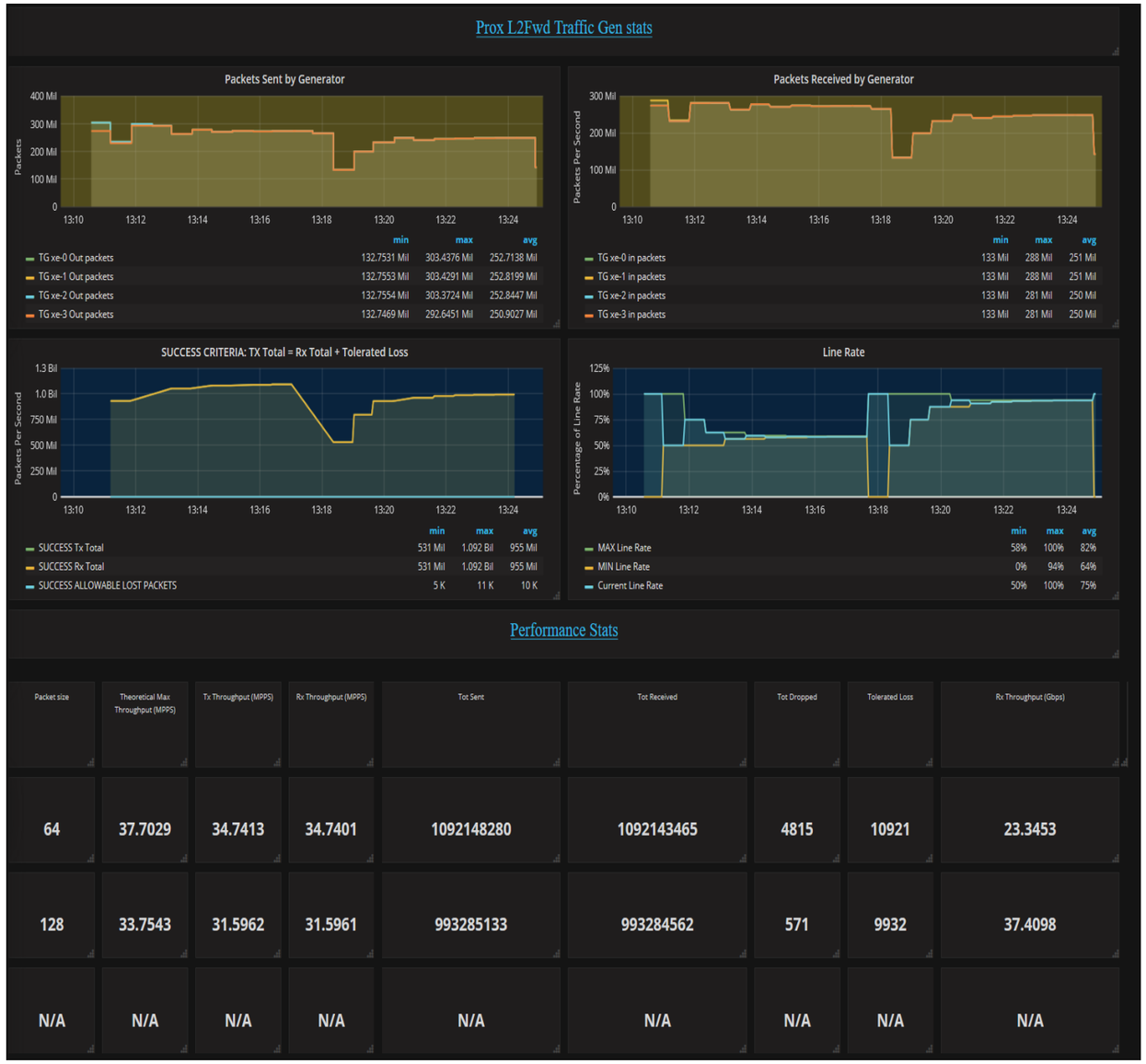


tc_prox_baremetal_l2fwd-2

time	criteria	deploy_scenario	installer	pod_name	task_id	tg__0.CurrentDropPackets	tg__0.DropPackets	tg__0.PktSize	tg__0.RxThroughput	tg__0.Throughput	tg__0.TxThroughput	tg__0.collect_stats.cc
2018-02-28T10:24:31.563107072Z	"PASS"	"unknown"	"unknown"	"unknown"	"f14550e6-355d-4452-b0a5-3c0965410346"	-0.268455951816	-0.268455951816	64	25.9929183815	25.9929183815	29.7619047619	
2018-02-28T10:24:36.64292608Z	"PASS"	"unknown"	"unknown"	"unknown"	"f14550e6-355d-4452-b0a5-3c0965410346"	-0.268455951816	-0.268455951816	64	25.9929183815	25.9929183815	29.7619047619	
2018-02-28T10:24:41.730441984Z	"PASS"	"unknown"	"unknown"	"unknown"	"f14550e6-355d-4452-b0a5-3c0965410346"	-0.268455951816	-0.268455951816	64	25.9929183815	25.9929183815	29.7619047619	
2018-02-28T10:24:46.812364032Z	"PASS"	"unknown"	"unknown"	"unknown"	"f14550e6-355d-4452-b0a5-3c0965410346"	-0.268455951816	-0.268455951816	64	25.9929183815	25.9929183815	29.7619047619	
2018-02-28T10:24:51.945668096Z	"PASS"	"unknown"	"unknown"	"unknown"	"f14550e6-355d-4452-b0a5-3c0965410346"	-0.268455951816	-0.268455951816	64	25.9929183815	25.9929183815	29.7619047619	
2018-02-28T10:24:57.035027968Z	"PASS"	"unknown"	"unknown"	"unknown"	"f14550e6-355d-4452-b0a5-3c0965410346"	-0.268455951816	-0.268455951816	64	25.9929183815	25.9929183815	29.7619047619	
2018-02-28T10:25:02.122481152Z	"PASS"	"unknown"	"unknown"	"unknown"	"f14550e6-355d-4452-b0a5-3c0965410346"	-0.268455951816	-0.268455951816	64	25.9929183815	25.9929183815	29.7619047619	
2018-02-28T10:25:07.203535104Z	"PASS"	"unknown"	"unknown"	"unknown"	"f14550e6-355d-4452-b0a5-3c0965410346"	-0.268455951816	-0.268455951816	64	25.9929183815	25.9929183815	29.7619047619	
2018-02-28T10:25:12.289601024Z	"PASS"	"unknown"	"unknown"	"unknown"	"f14550e6-355d-4452-b0a5-3c0965410346"	-0.268455951816	-0.268455951816	64	25.9929183815	25.9929183815	29.7619047619	

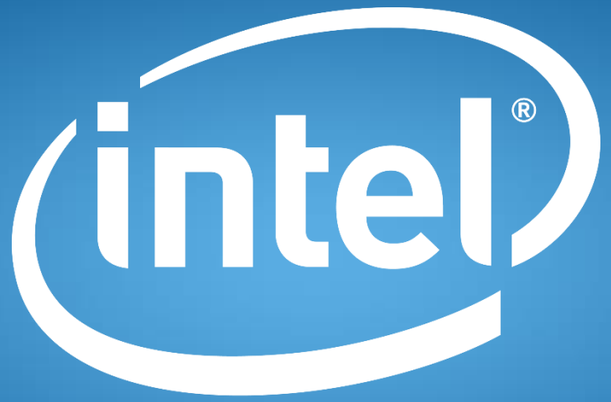
Grafana dashboard: Network Metrics





Grafana dashboard: NFVi Metrics*

* NFVi metrics comes from Collectd, covered by OPNFV Barometer project



experience
what's inside™