

# Testing with Functest

- [OPNFV Release E \(Euphrates\)](#)
  - [Functest Test Case Extension Proposal](#)
  - [Test Scope:](#)
  - [High Level Testing Procedure:](#)
  - [Testing Assumptions \(Same assumptions for the D release\):](#)
- [OPNFV Release D \(Danube\)](#)

## OPNFV Release E (Euphrates)

The intention of the Barometer test in the OPNFV E release is simply to extend the test suite that was developed in the D release with new tests for the following features:

Theme: OPNFV Platform Equivalence	Test Jira
RDT Support	<a href="#">blocked URLBAROMETER-65 -Test collectd RDT Cache Plugin</a> <a href="#">OPEN</a>
Platform Legacy Support (IPMI)	<a href="#">blocked URLBAROMETER-66 -Test collectd IPMI Plugin</a> <a href="#">OPEN</a>
RAS Support	<a href="#">blocked URLBAROMETER-67 -Test collectd RAS Plugin</a> <a href="#">OPEN</a>
PMU Support	<a href="#">blocked URLBAROMETER-68 -Test collectd PMU Plugin</a> <a href="#">OPEN</a>
Libvirt Support	<a href="#">blocked URLBAROMETER-69 -Test collectd Libvirt Plugin</a> <a href="#">OPEN</a>
vSwitch Health	<a href="#">blocked URLBAROMETER-70 -Test collectd OVS Plugins</a> <a href="#">OPEN</a>
PCIe Error reporting	<a href="#">blocked URLBAROMETER-71 -Test collectd PCIe Plugin</a> <a href="#">OPEN</a>
SNMP write plugin	<a href="#">blocked URLBAROMETER-72 -Test collectd SNMP Plugin</a> <a href="#">OPEN</a>
Aodh plugin	<a href="#">blocked URLBAROMETER-73 -Test collectd Aodh Plugin</a> <a href="#">OPEN</a>
Gnocchi Support	<a href="#">blocked URLBAROMETER-74 -Test collectd Gnocchi Plugin</a> <a href="#">OPEN</a>
ETSI TST008 Alignment	<a href="#">blocked URLBAROMETER-75 -Test TST008 Alignment</a> <a href="#">OPEN</a>
clock speed and power state (CPU + platform)	<a href="#">blocked URLBAROMETER-75 -Test TST008 Alignment</a> <a href="#">OPEN</a>
Plugin rename to barometer as part of Apex Support	
Integration with Apex	Covered by all the other testing JIRAs
<b>Euphrates Stretch Goals</b>	
collectd agent extensions	<a href="#">blocked URLBAROMETER-76 -Test collectd agent extensions</a> <a href="#">OPEN</a>
Host Health Support	<a href="#">blocked URLBAROM</a>

## Functest Test Case Extension Proposal

Project: Barometer (formerly known as SFQM – Software Fastpath Service Quality Metrics)

Authors/Contributors: Calin Gherghe (Intel Corp)

Test Case Name: BarometerCollectd

Test Type: External (i.e. test code will be kept in the project repository)

Test Tier: features

Test Constrains:

Scenario(s) (in order of priority):

- os-nosdn-bar-noha
- os-nosdn-bar-ha
- os-nosdn-kvm\_ovs\_dpdk\_bar-noha
- os-nosdn-kvm\_ovs\_dpdk\_bar-ha

Installer(s): Apex

## Test Scope:

The test is intended to validate individually the collectd plugins which are part of the Barometer project, therefore is constituted as a test-suite.

The verifications are aimed to check the functionality according to the default (current) status and configuration of each plugin.

Testing is targeting the Compute Node(s) in the POD. No modifications are expected to be done on the target systems.

The tests' key focus is to confirm the corresponding metrics (and/or events) are properly dispatched by the read-plugins, respectively properly sent by the output plugins.

The list of covered plugins is subject to change as the project evolves.

Output (write-type) plugins: Ceilometer

Input (read-type) plugins: Intel RDT, Hugepages, Memory RAS, OVS Stats and Events, BIOS

Third-party Dependencies: The test may rely on the existing 'logfile', 'csv', and 'exec' collectd plugins

Test Code: Python and bash

Test Criteria: status == PASS

Test Declarations

The following files are proposed to be patched:

1. testcases.yaml (/home/opnfv/repos/funcstest/ci/testcases.yaml). Add the following block to declare the test and its constraints:

name: BarometerCollectd

criteria: 'status == "PASS"'

blocking: false

description: >-

Test suite for the Barometer project. Separate tests verify the proper configuration and functionality of the following collectd plugins: Ceilometer, Intel RDT, Hugepages, Memory RAS, OVS Stats and Events, and the plugins mentioned above

dependencies:

installer: 'Apex'

2. Updating the Test cases in the barometer repo to test the new plugins: [\[barometer.git\]](#) / [baro\\_tests](#) /

## High Level Testing Procedure:

**Note: this has not changed from the D release**

For Each Compute Node

Get configuration (Parse collectd.conf or plugin-specific conf file)

For Each Output Plugin

```

Verify plugin is enabled by default
Verify (sample) metrics (and/or events) are sent
Wait Interval
Verify metrics (and/or events) timestamps are updated
Log Result
For Each Input Plugin

    Use Output Plugin if PASS, otherwise use CSV (local files)
    Verify plugin is enabled by default
    Verify (if any) dependent modules are loaded
    Induce events (as applicable)
    Verify ALL corresponding metrics and/or events are dispatched
    Wait Interval
    Induce events (as applicable)
    Verify ALL metrics and/or events timestamps are updated
    Log Result

Done

Done

Done

```

### Testing Assumptions (Same assumptions for the D release):

- The metrics' actual values are NOT validated for correctness (at least in this release). Only the metrics list and time stamps are verified.
- If the Output plugin(s) verification fails, the CSV plugin will be enabled and (collectd service restarted) in order to be able to proceed with the Input plugins testing
- Log messages will be generated at every step
- Failure of ANY plugin test would generate an Overall FAIL Result (unless a success\_rate criteria is adopted)
- Any change in configuration (such as enabling the CSV plugin if necessary) would be restored upon the tests' exit.
- Testing will exit when all loops are completed once. No additional runtime is allotted for checking potential stress failures (such as monitoring over longer time for: memory leaks, traffic congestion, latency issues, disk space, database chocking, etc)

## OPNFV Release D (Danube)

### MS2: Detailed test case descriptions communicated to test project teams

Due: 11/22/2016

Detailed Milestone Description (per OPNFV Danube Wiki): Each feature project team will complete a test plan, describing the test cases that they intend to use to validate their project, including pass/fail criteria, and will share this plan with the test working group and other team members and solicit feedback. The test plan will also describe dependencies on any new test capability that is not currently available through existing test frameworks and how these dependencies will be satisfied. Project teams are encouraged to attend the Test Working Group meeting, or to setup separate meetings to discuss requirements

#### New Functest Test Case Proposal

Project: Barometer (formerly known as SFQM – Software Fastpath Service Quality Metrics)

Authors/Contributors: Calin Gherghe (Intel Corp)

Test Case Name: BarometerCollectd

Test Type: External (i.e. test code will be kept in the project repository)

Test Tier: features

Test Constrains:

Scenario(s) (in order of priority):

- os-nosdn-kvm\_ovs-noha
- os-nosdn-kvm\_ovs\_dpdk-noha
- os-nosdn-kvm\_ovs-ha
- os-nosdn-kvm\_ovs\_dpdk-ha

Installer(s): Fuel

Test Scope: The test is intended to validate individually the collectd plugins which are part of the Barometer project, therefore is constituted as a test-suite. The verifications are aimed to check the functionality according to the default (current) status and configuration of each plugin. Testing is targeting the Compute Node(s) in the POD. No modifications are expected to be done on the target systems. The tests' key focus is to confirm the corresponding metrics (and/or events) are properly dispatched by the read-plugins, respectively properly sent by the output plugins. The list of covered plugins is subject to change as the project evolves.

Output (write-type) plugins: Ceilometer

Input (read-type) plugins: Intel RDT, Hugepages, Memory RAS, OVS Stats and Events, BIOS

Third-party Dependencies: The test may rely on the existing 'logfile', 'csv', and 'exec' collectd plugins

Test Code: Python and bash

Test Criteria: status == PASS

Test Declarations

The following files are proposed to be patched:

1. Dockerfile (/home/opnfv/repos/functest/docker/Dockerfile). Add the following line in order to clone the test code from the external repository into the Functest container:

```
RUN git clone https://gerrit.opnfv.org/gerrit/BarometerCollectd
```

```
${repos_dir}/BarometerCollectd
```

2. exec\_test.sh (/home/opnfv/repos/functest/ci/exec\_test.sh). Add the following to enable running the test in CI or CLI:

```
"BarometerCollectd")
```

```
python ${FUNCTEST_REPO_DIR}/testcases/features/BarometerCollectd.py
```

```
;;
```

3. testcases.yaml (/home/opnfv/repos/functest/ci/testcases.yaml). Add the following block to declare the test and its constraints:

```
name: BarometerCollectd
```

```
criteria: 'status == "PASS"'
```

```
blocking: false
```

```
description: >-
```

```
Test suite for the Barometer project. Separate tests verify the proper configuration and functionality of the following collectd plugins: Ceilometer, Intel RDT, Hugepages, Memory RAS, OVS Stats and Events, BIOS
```

```
dependencies:
```

```
installer: 'fuel'
```

```
scenario: 'os-nosdn-kvm_ovs-noha, os-nosdn-kvm_ovs_dpdk-noha, os-nosdn-kvm_ovsha, os-nosdn-kvm_ovs_dpdk-ha'
```

4. config\_functest.yaml (/home/opnfv/repos/functest/ci/config\_functest.yaml). Add the following line to define the repository absolute path:

```
dir_repo_BarometerCollectd: /home/opnfv/repos/BarometerCollectd
```

High Level Testing Procedure:

For Each Compute Node

Get configuration (Parse collectd.conf or plugin-specific conf file)

For Each Output Plugin

Verify plugin is enabled by default

Verify (sample) metrics (and/or events) are sent

Wait Interval

Verify metrics (and/or events) timestamps are updated

Log Result

For Each Input Plugin

Use Output Plugin if PASS, otherwise use CSV (local files)

Verify plugin is enabled by default

Verify (if any) dependent modules are loaded

Induce events (as applicable)

Verify ALL corresponding metrics and/or events are dispatched

Wait Interval

Induce events (as applicable)

Verify ALL metrics and/or events timestamps are updated

Log Result

Done

Done

Done

Testing Assumptions:

? The metrics' actual values are NOT validated for correctness (at least in this release). Only the metrics list and timestamps are verified.

? If the Output plugin(s) verification fails, the CSV plugin will be enabled and (collectd service restarted) in order to be able to proceed with the Input plugins testing

? Log messages will be generated at every step

? Failure of ANY plugin test would generate an Overall FAIL Result (unless a success\_rate criteria is adopted)

? Any change in configuration (such as enabling the CSV plugin if necessary) would be restored upon the tests' exit.

? Testing will exit when all loops are completed once. No additional runtime is allotted for checking potential stress failures (such as monitoring over longer time for: memory leaks, traffic congestion, latency issues, disk space, database chocking, etc)