

How do I know the Functest status of a scenario for Colorado?

For Colorado we created an automatic reporting page. This page is dynamically created through a cron job (update every hour) and is based on the results stored in the Test DB.

You can access this reporting page: <http://testresults.opnfv.org/reporting/functest/release/master/index-status-apex.html>

Functest status page (master)

Version
Installer selection
Installer selected

compass

List of last scenarios (master) run over the last 50 days

Scenario	Status	Score	Iteration
os-onos-sfc-ha	✖	14/15	24
os-odl_13-nofeature-ha	✖	6/9	25
os-ocl-nofeature-ha	✖	6/12	12
os-onos-nofeature-ha	✖	13/15	43
os-odl_12-nofeature-ha	✖	10/12	37
os-odl_12-moon-ha	✖	12/15	7
os-nosdn-nofeature-ha	✖	11/12	94

Scenario summary

Scenario details

os-onos-sfc-ha

vPing (ssh)	vPing (userdata)	Tempest (smoke)	Rally (smoke)	ONOS	SFC
☀	☀	☁	☀	☀	☁

os-odl_13-nofeature-ha

vPing (userdata)	Tempest (smoke)	Rally (smoke)
☀	☁	☀

os-ocl-nofeature-ha

vPing (ssh)	vPing (userdata)	Tempest (smoke)	Rally (smoke)
☀	☀	☁	☁

os-onos-nofeature-ha

There is one page per installer. For each installer we list the scenarios run from CI since 10 days.

The page can be divided in 2 parts:

- the scenario summary including the scenario name, its status, a scoring and the number of iterations
- the scenario details with indication case by case

The scenario is detected through one test case that is supposed to be run all the time (tempest_smoke_serial). If for any reason, a scenario crashed before running this test (healthcheck or vPings) then the scenario will not be referenced.

The number of cases differ from one scenario to another (and is sometimes also different from one installer to another). It is due to the dependencies declared by the case owner (e.g. onos case has no sense on no onos scenarios). The test criteria have been discussed in [Test Release Criteria](#)





For colorado we consider the cases from the first tiers, i.e. healthcheck, smoke, sdn_suites and features. It means that tempest_full_parallel, rally_full of vims are not considered for the scoring.

The results are all coming from the database and may come from different pods (bm or virtual). The results are be pushed from CI to the database if the pod id declared in the test DB: <http://testresults.opnfv.org/test/api/v1/pods>

The scoring is a basic calculation: we consider the results for each test case and we sum the scoring of each case to give a scenario scoring.

Points are attributed as follow:

Comment	Points
---------	--------

	test never reached the success criteria since 50 days	0
	test success criteria reached at least once within the last 50 days	1
	test success criteria reached more than once but not the last 4 iterations	2
	test success criteria reached on the last 4 iterations	3

As an illustration if we consider the scenario os-nosdn-nofeature-ha for onos (source 17/08/2016), we can see that the scoring is 11/12.

os-nosdn-nofeature-ha



11/12





94

So the status is red (need 12/12 to be green).

We may also notice that this scenario has been run 94 times since 50 days, which is also a good indicator of the scenario activity.

If we consider the the details of the scenario, we can see that the last issue is due to the rally_sanity (got a 2/3).

> os-nosdn-nofeature-ha

vPing (ssh)	vPing (userdata)	Tempest (smoke)	Rally (smoke)
			

The scoring can be detailed as follow:

$$\text{score}(\text{compass/os-nosdn-nofeature-ha}, 17/08\ 9\text{h}30\ \text{UTC}) = 3 (\text{vping_ssh}) + 3 (\text{vping_userdata}) + 3 (\text{tempest_smoke_serial}) + 2 (\text{rally_sanity}) = 11$$

the name of the test cases have been modified for a better display, the corrspondance case name, case display name and case DB name is available in <https://git.opnfv.org/cgit/relog/tree/utills/test/reporting/funcstest/testCase.py>

regarding rally_sanity case, on the 17/08 9:30 UTC, if we consider the last 4 results stored in the database, the 4th one was FAIL.

```

{
  "results": [
    {
      "project_name": "functest",
      "stop_date": "2016-08-17 08:52:16",
      "case_name": "rally_sanity",
      "build_tag": "jenkins-functest-compass-baremetal-daily-master-224",
      "pod_name": "huawei-pod1",
      "installer": "compass",
      "scenario": "os-nosdn-nofeature-ha",
      "trust_indicator": 0,
      "version": "master",
      "details": [...],
      "criteria": "PASS",
      "_id": "57b427999377c54b278bdd2d",
      "start_date": "2016-08-17 08:43:42"
    },
    {
      "project_name": "functest",
      "stop_date": "2016-08-17 04:27:12",
      "case_name": "rally_sanity",
      "build_tag": "jenkins-functest-compass-huawei-pod2-daily-master-132",
      "pod_name": "huawei-pod2",
      "installer": "compass",
      "scenario": "os-nosdn-nofeature-ha",
      "trust_indicator": 0,
      "version": "master",
      "details": [...],
      "criteria": "PASS",
      "_id": "57b3a4ac9377c54b278bdcc3",
      "start_date": "2016-08-16 23:31:27"
    },
    {
      "project_name": "functest",
      "stop_date": "2016-08-16 23:40:18",
      "case_name": "rally_sanity",
      "build_tag": "jenkins-functest-compass-virtual-daily-master-341",
      "pod_name": "huawei-virtual2",
      "installer": "compass",
      "scenario": "os-nosdn-nofeature-ha",
      "trust_indicator": 0,
      "version": "master",
      "details": [...],
      "criteria": "PASS",
      "_id": "57b3a4ac9377c54b278bdcc3",
      "start_date": "2016-08-16 23:31:27"
    },
    {
      "project_name": "functest",
      "stop_date": "2016-08-16 04:28:32",
      "case_name": "rally_sanity",
      "build_tag": "jenkins-functest-compass-huawei-pod2-daily-master-130",
      "pod_name": "huawei-pod2",
      "installer": "compass",
      "scenario": "os-nosdn-nofeature-ha",
      "trust_indicator": 0,
      "version": "master",
      "details": [...],
      "criteria": "FAIL",
      "_id": "57b2272e9377c54b278bdb1a",
      "start_date": "2016-08-16 04:18:10"
    }
  ]
}









```

Note that if the scenario is validated (full scoring), we save the name, installer and date. So if scenario is not run anymore and/or regression occurred we keep the trace of success.

Some scenarios have lots of feature tests (e.g. fuel/os-odl_l2-ha below: 3 feature test suites (promise, SFC, parser)).

By default we enable them into the automatic reporting. It means that if a feature test is failing and if it is run on all the scenarios of an installers (no dependencies except the installer e.g. doctor on apex), all the status may remain red.

➤ **os-odl_l2-sfc-ha**

vPing (ssh)	vPing (userdata)	Tempest (smoke)	Rally (smoke)	ODL	Promise	SFC	parser
							

It is possible to exclude case in a black list for the reporting by adding the case name in blacklist array in <https://git.opnfv.org/cgit/releng/tree/utils/test/reporting/functest/reportingConf.py>