# FUNCTEST IN DEPTH

Jose Lausuch (Ericsson)

OPNFV Summit 2015

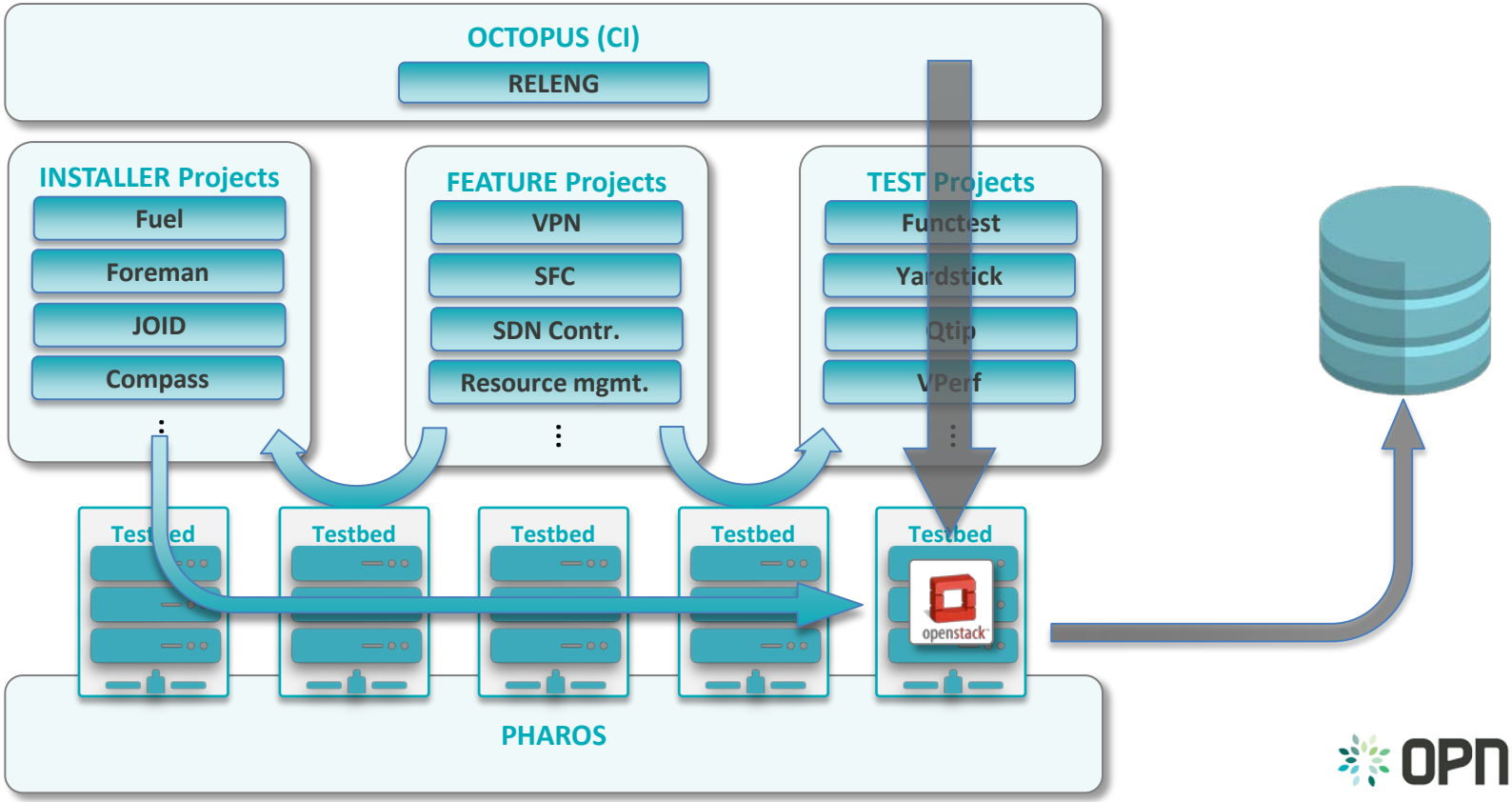
jose.lausuch@ericsson.com

LINUX FOUNDATION
COLLABORATIVE PROJECTS

# Agenda

- **What is Functest?**
- **Test cases**
- **Repository structure**
- **CI & Automation**
- **Brahmaputra Release**

# OPNFV Projects ecosystem



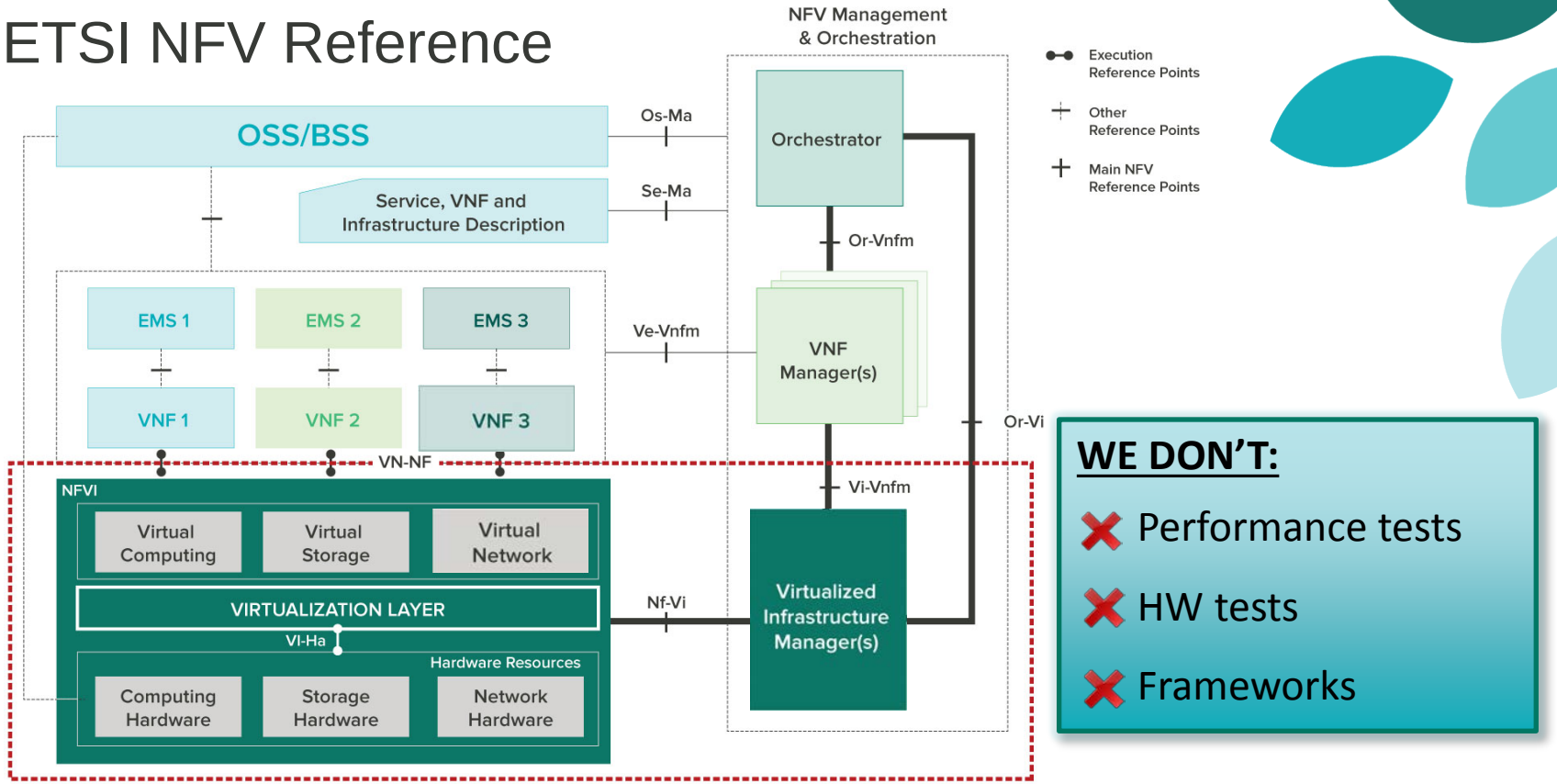
# What is Functest?

# What we do…

*Base System Functionality Testing*

- Focus on verifying the OpenStack deployment and the SDN Controllers
- Have a full integration and automation mechanism
- Provide comprehensive testing methodology

# ETSI NFV Reference




**WE DON'T:**

- ✖ Performance tests
- ✖ HW tests
- ✖ Frameworks

# The gang…

Peter Bandzi

Jose Lausuch

Morgan (PTL) Richomme

Qinglong Lan

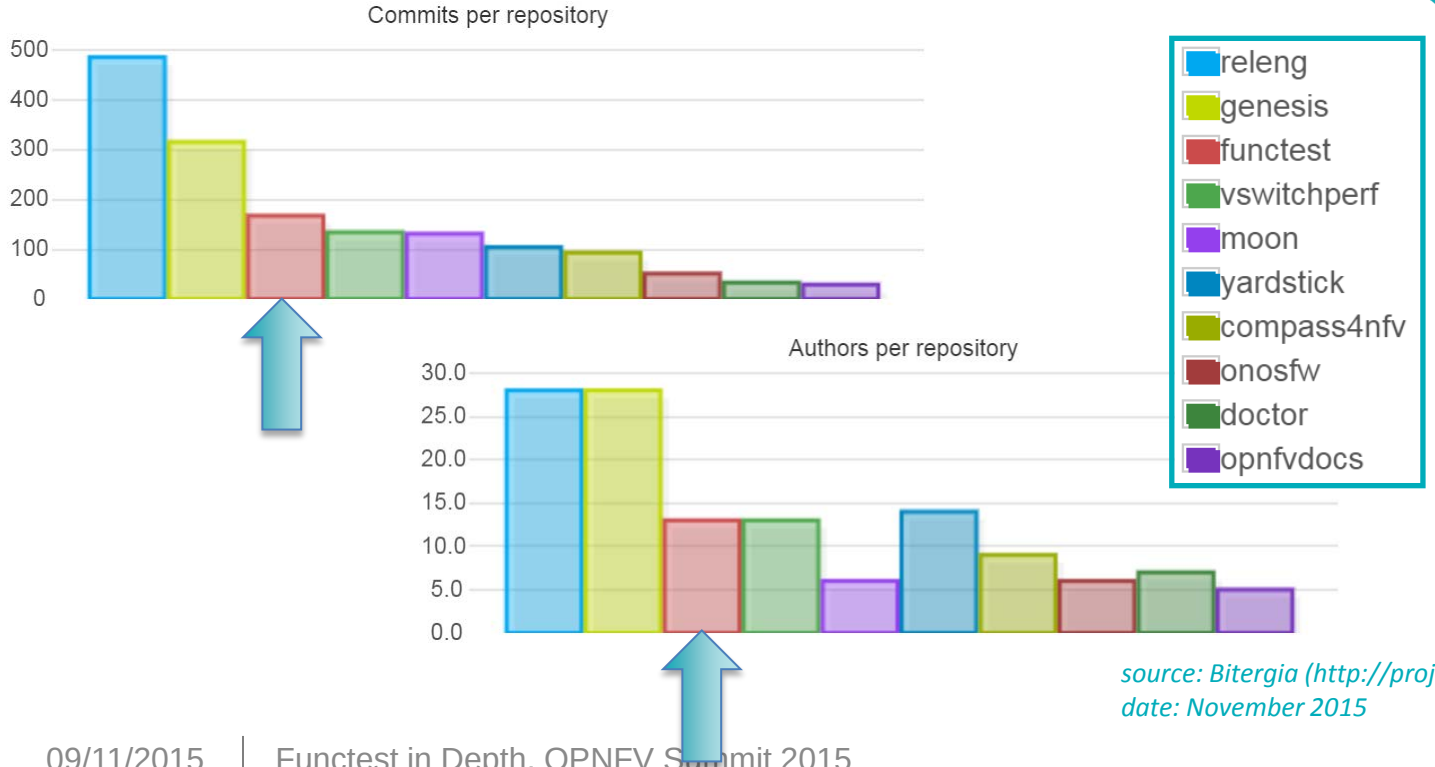
Juha Kosonen

Guy Rodrigue

Viktor Tikkanen

Valentin Boucher

# The project…




*source: Bitergia (http://projects.bitergia.com/opnfv)*
*date: November 2015*

# Functest Testcases

Arno

# Functest Testcases

**vPing test case**

Create machines and verify connectivity

**ODL test case**

Robot framework, ODL functional testing


**Rally bench tests**

Benchmark the OpenStack deployment


**Tempest test**

OpenStack native tests (100+ smoke-tests)

# Functest Testcases

**vPing test case**

Create machines and verify connectivity

**ODL test case**

Robot framework, ODL functional testing

**Rally bench tests**

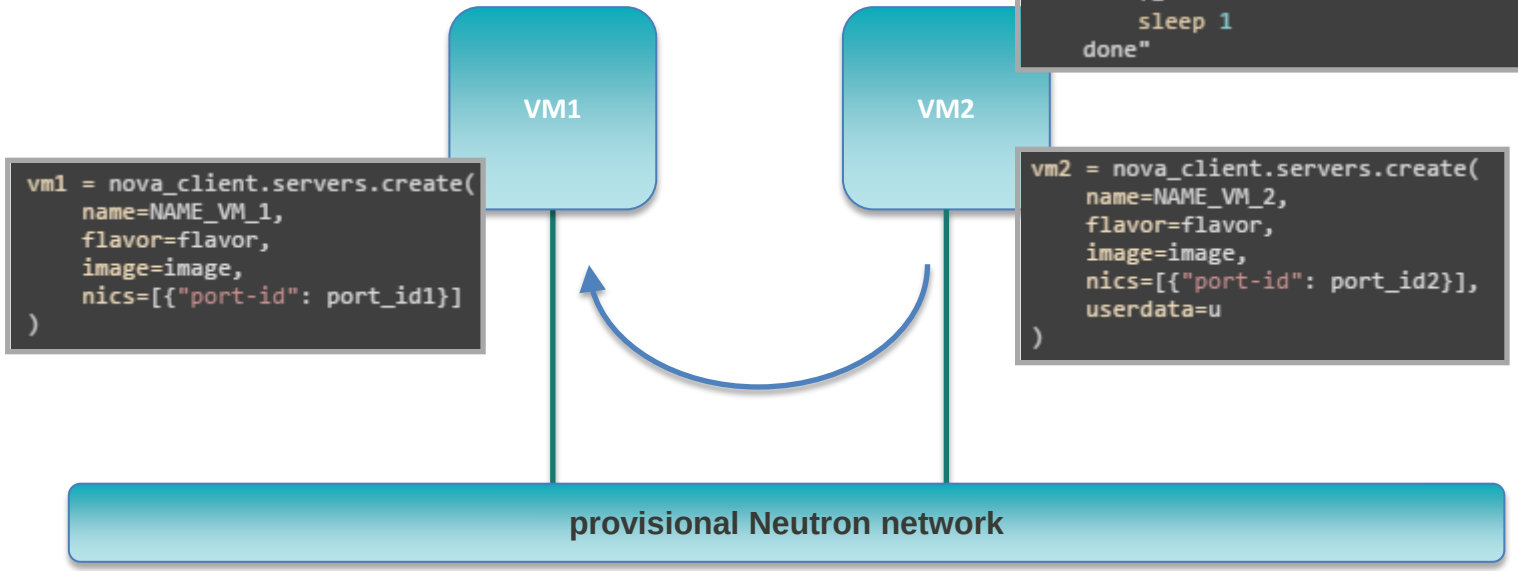
Benchmark the OpenStack deployment

**Tempest test**

OpenStack native tests (100+ smoke-tests)

# vPing test case

```
u = "#!/bin/sh
    while true; do
        ping -c 1 $test_ip 2>&1 >/dev/null
        RES=$?
        if [ '$RES' = 'Z0' ] ; then
            echo 'vPing OK'
            break
        else
            echo 'vPing KO'
        fi
        sleep 1
    done"
```



# Functest Testcases

**vPing test case**

Create machines and verify connectivity

**ODL test case**

Robot framework, ODL functional testing

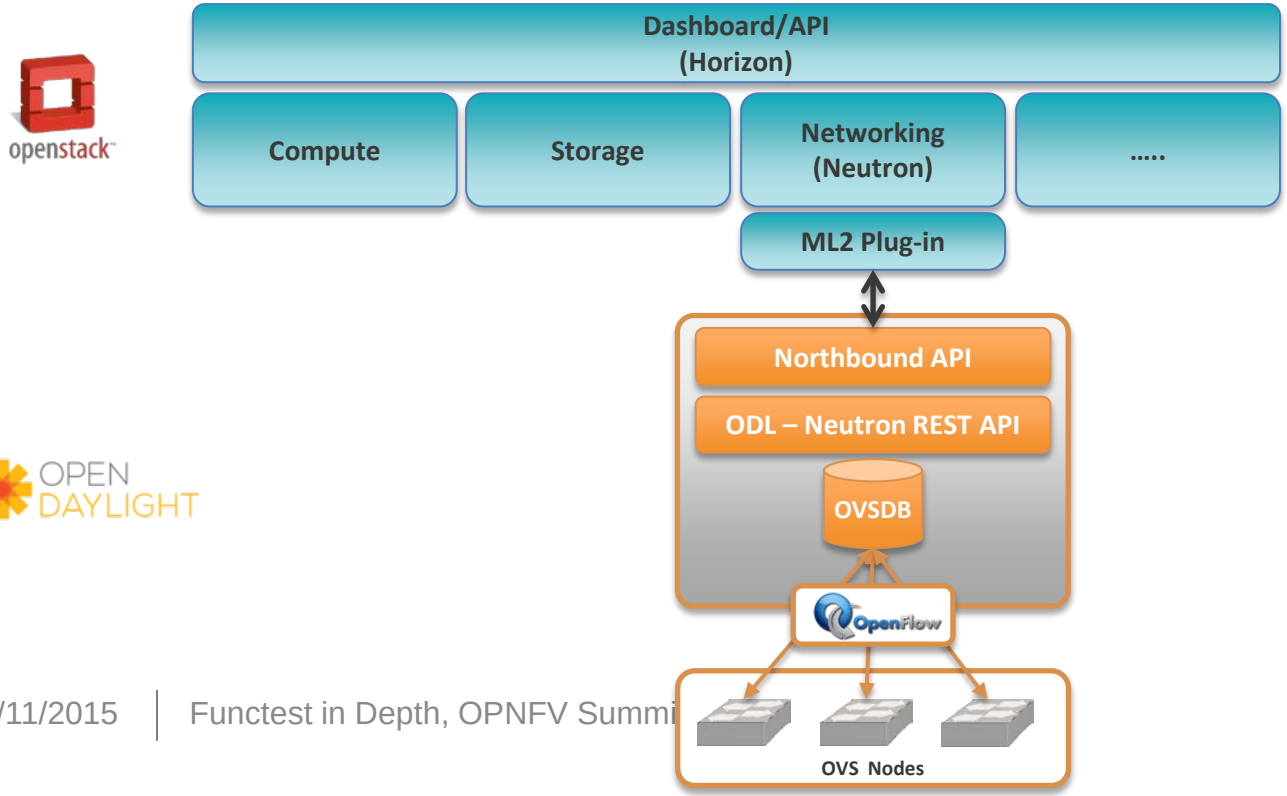
**Rally bench tests**

Benchmark the OpenStack deployment

**Tempest test**

OpenStack native tests (100+ smoke-tests)

# ODL test case



# ODL test cases

Borrowed from ODL repository

```
https://github.com/opendaylight/integration.git
```

4 basic test cases:

- Check ODL Restconf
- Create/Delete/Check Network
- Create/Delete/Check Subnet
- Create/Delete/Check Port

```
==============================================================================
Neutron :: Test suite for Neutron Plugin
==============================================================================
Neutron.Networks :: Checking Network created in OpenStack are pushed to Ope...
==============================================================================
Check OpenStack Networks :: Checking OpenStack Neutron for known n...  | PASS |
------------------------------------------------------------------------------
Check OpenDaylight Networks :: Checking OpenDaylight Neutron API f...  | PASS |
------------------------------------------------------------------------------
Create Network :: Create new network in OpenStack                      | PASS |
------------------------------------------------------------------------------
Check Network :: Check Network created in OpenDaylight                 | PASS |
------------------------------------------------------------------------------
Neutron.Networks :: Checking Network created in OpenStack are push...  | PASS |
4 critical tests, 4 passed, 0 failed
4 tests total, 4 passed, 0 failed
```

# Functest Testcases

**vPing test case**

Create machines and verify connectivity

**ODL test case**

Robot framework, ODL functional testing

**Rally bench tests**

Benchmark the OpenStack deployment

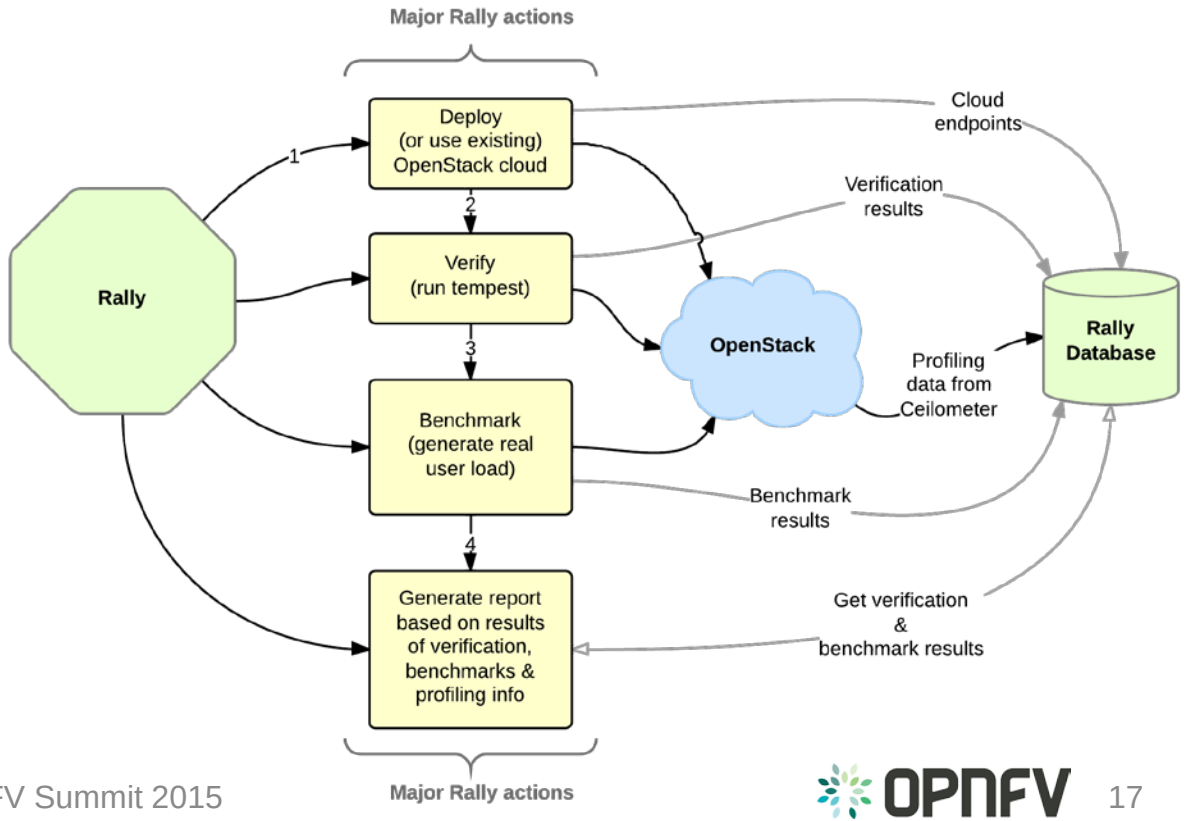

**Tempest test**

OpenStack native tests (100+ smoke-tests)

# Rally

Rally is a benchmarking tool that answers the question:

“How does OpenStack work at scale?”.



# Rally - scenario example

```
"GlanceImages.create_and_delete_image": [
    {
        "args": {
            "image_location": "http://download.cirros-cloud.net/0.3.1/cirros-0.3.1-x86_64-disk.img",
            "container_format": "bare",
            "disk_format": "qcow2"
        },
        "runner": {
            "type": "constant",
            "times": 10,
            "concurrency": 2
        },
        "context": {
            "users": {
                "tenants": 2,
                "users_per_tenant": 3
            }
        },
        "sla": {
            "max_seconds_per_iteration": 10,
            "failure_rate": {
                "max": 25
            }
        }
    }
],
```

**Runners:**

- **Serial**: launch scenario N times in one thread
- **Constant**: launch scenario N times in M parallel threads
- **RPS**: generates N threads per second M times

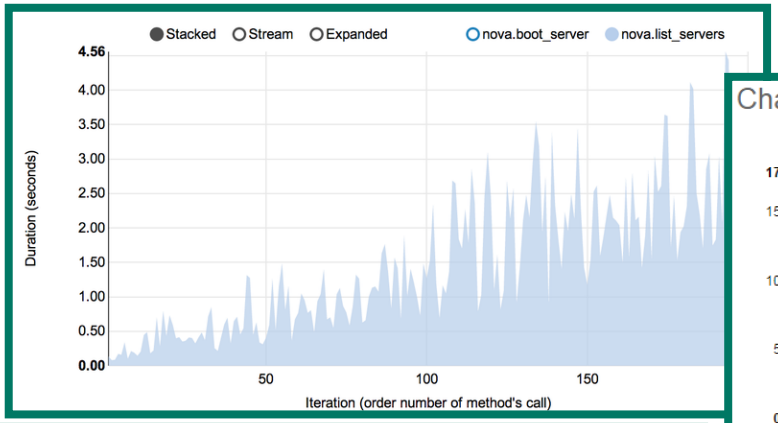
**Context:**

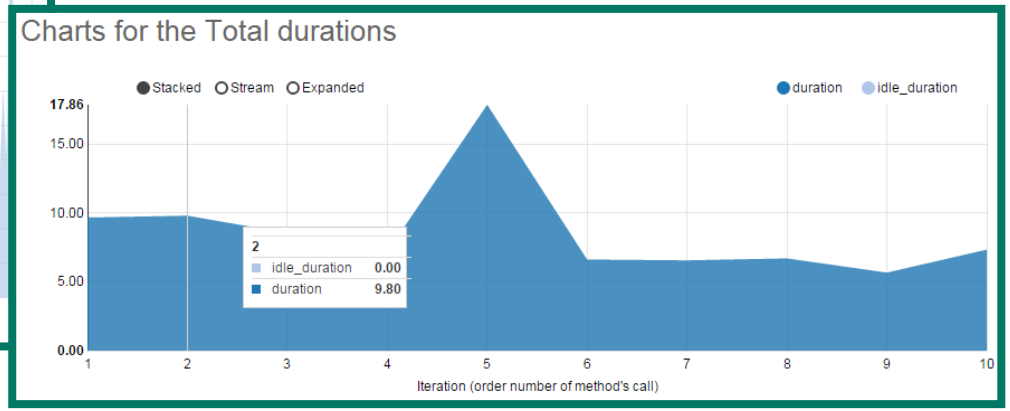
- Contexts needed for this scenario (tenants/users)

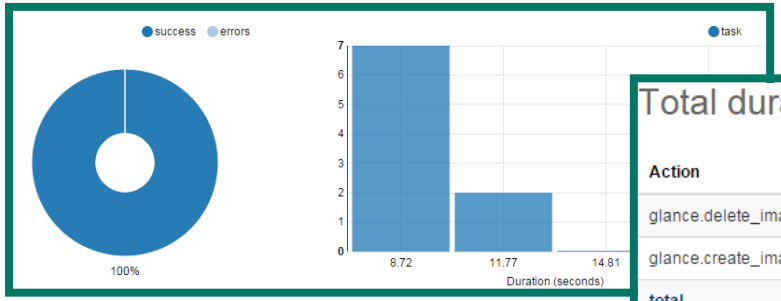
**Success criteria (SLA)**

- You can define a maximum time per iteration
- It will skip the test if 1 iteration fails

# Rally - output




## Total durations

| Action | Min (sec) | Median (sec) | 90%ile (sec) | 95%ile (sec) | Max (sec) | Avg (sec) | Success | Count |
|---|---|---|---|---|---|---|---|---|
| glance.delete_image | 0.873 | 1.16 | 1.503 | 1.56 | 1.617 | 1.232 | 100.0% | 10 |
| glance.create_image | 4.196 | 5.903 | 9.41 | 13.059 | 16.708 | 7.36 | 100.0% | 10 |
| **total** | **5.666** | **7.258** | **10.605** | **14.235** | **17.865** | **8.592** | **100.0%** | **10** |

# Functest Testcases

**vPing test case**

Create machines and verify connectivity

**ODL test case**

Robot framework, ODL functional testing

**Rally bench tests**

Benchmark the OpenStack deployment

**Tempest test**

OpenStack native tests (100+ smoke-tests)

# Tempest test

Tempest is a set of integration tests to be run against a live OpenStack cluster

Rally performs the tempest installation and generation of tempest.conf

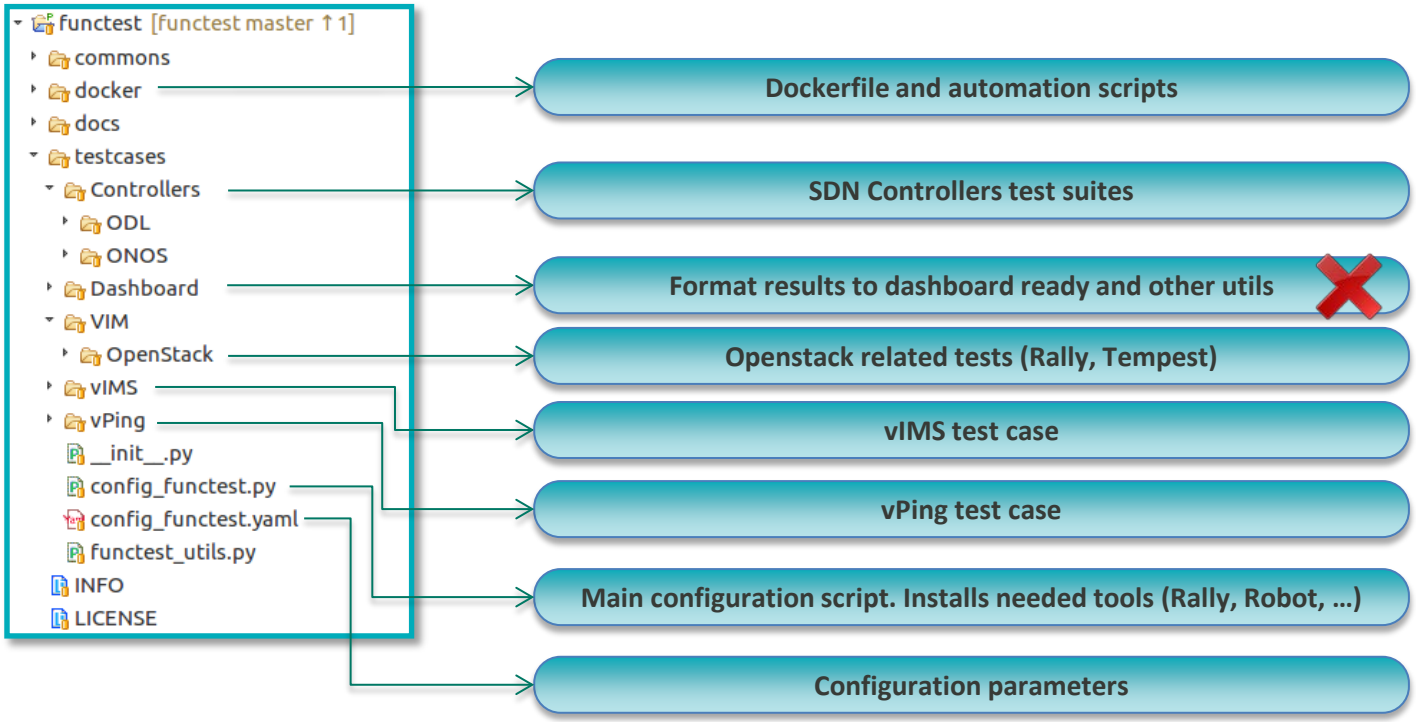
```
$ rally verify install
$ rally verify genconfig
$ rally verify start smoke
```

```
root@49227ece4491:/# ls ~/.rally/tempest/for-deployment-f807dba1-2a7c-4ebd-bb8a-6a918f5a7957/
HACKING.rst  README.rst     doc  openstack-common.conf  run_tempest.sh  setup.cfg  subunit.stream  tempest.conf      tempest.log           tools
LICENSE      REVIEWING.rst  etc  requirements.txt       run_tests.sh    setup.py   tempest         tempest.egg-info  test-requirements.txt  tox.ini
```

# Functest Structure

today

# Directory structure



# Configuration file

## Configuration yaml file

- Predefined tested values
- Can be given as an input to Functest
- Possible to specify commit IDs of the used repositories
  - Versioning control

```yaml
general:
    directories:
        # Relative to the path where the repo is cloned:
        dir_vping:      testcases/vPing/CI/libraries/
        dir_odl:        testcases/Controllers/ODL/CI/
        dir_rally:      testcases/VIM/OpenStack/CI/libraries/
        dir_rally_scn:  testcases/VIM/OpenStack/CI/suites/
        dir_vIMS:       testcases/vIMS/CI/
        dir_onos:       testcases/Controllers/ONOS/Teston/CI/

        # Absolute path
        dir_repos:          /home/opnfv/repos
        dir_repo_functest:  /home/opnfv/repos/functest
        dir_repo_rally:     /home/opnfv/repos/rally
        dir_repo_releng:    /home/opnfv/repos/releng
        dir_repo_vims_test: /home/opnfv/repos/vims-test
        dir_functest:       /home/opnfv/functest
        dir_results:        /home/opnfv/functest/results
        dir_functest_conf:  /home/opnfv/functest/conf
        dir_rally_res:      /home/opnfv/functest/results/rally
        dir_functest_data:  /home/opnfv/functest/data
        dir_vIMS_data:      /home/opnfv/functest/data/vIMS
        dir_rally_inst:     ~/.rally

    repositories:
        # branch and commit ID to which the repos will be reset (HEAD)
        functest_branch:    master
        functest_commit:    latest
        releng_branch:  master
        releng_commit:  latest
        rally_branch:   master
        rally_commit:   9a17d8490e7fb4afee2f5629e6db41eabe4bc4d4
        vims_test_branch:   stable
        vims_test_commit:   latest
```

CI & Automation

# CI & Automation

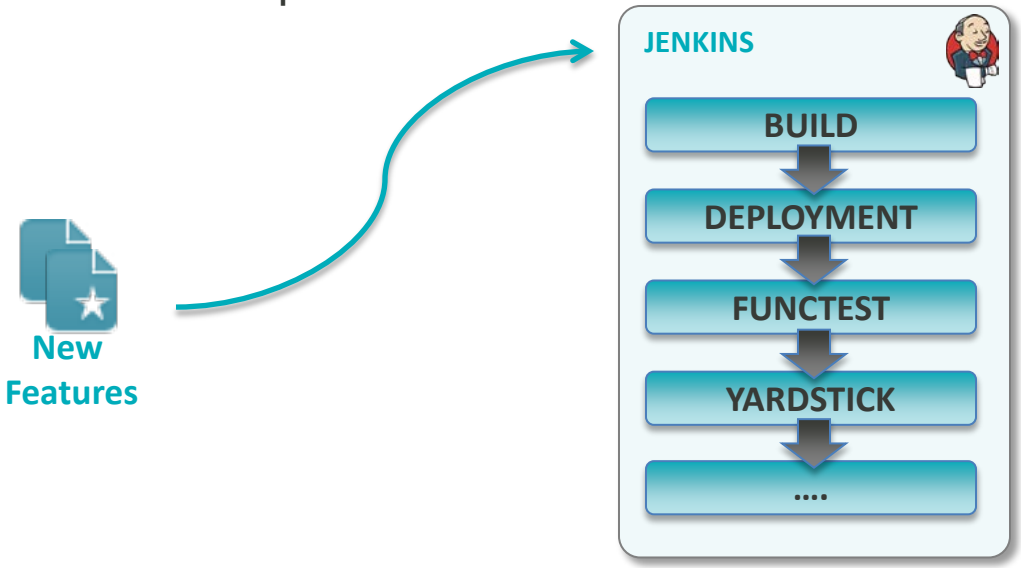
## Functest Jenkins job

- Triggered automatically after a successful deployment
- One macro per test case
  - You can execute a single test From Jenkins

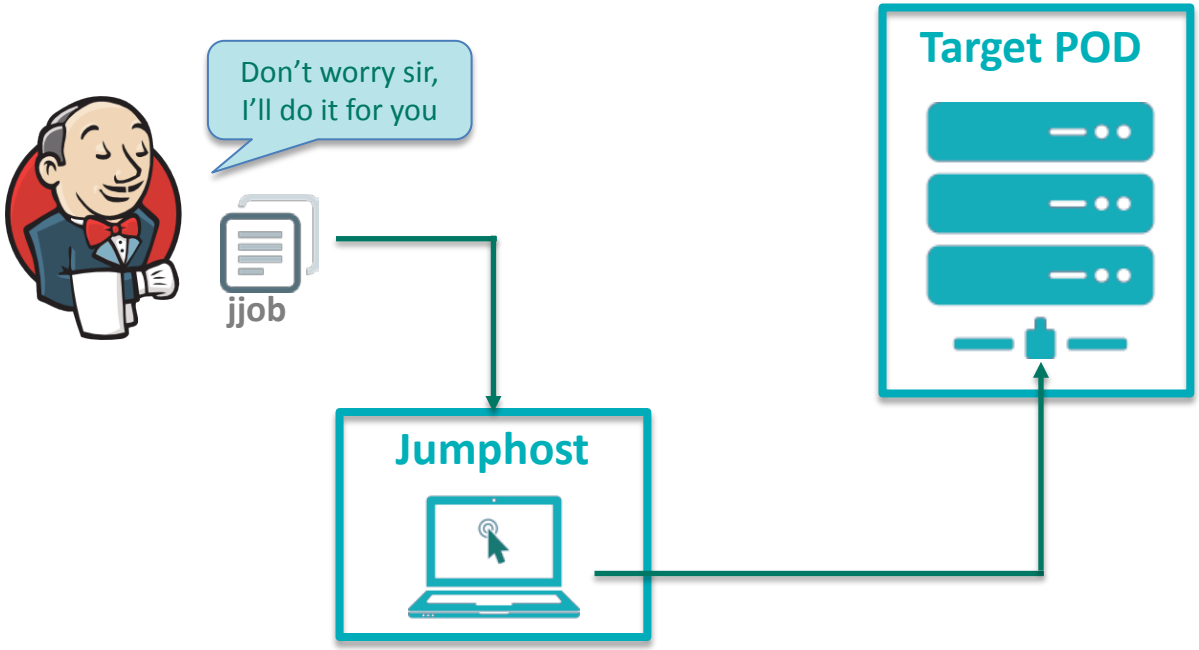
| S | W | Name ↓ | Last Success | Last Failure | Last Duration |
|---|---|---|---|---|---|
| | | functest-daily-master | 5 hr 25 min - #154 | N/A | 44 sec |
| | | functest-daily-stable-arno | N/A | 4 mo 14 days - #1 | 4 sec |
| | | functest-foreman-master | 1 mo 6 days - #19 | 1 mo 6 days - #18 | 1 hr 24 min |
| | | functest-foreman-stable-arno | 28 days - #25 | N/A | 1 hr 7 min |
| | | functest-fuel-master | 9 hr 31 min - #72 | 15 hr - #71 | 45 min |
| | | functest-fuel-stable-arno | 29 days - #16 | 1 mo 5 days - #10 | 1 hr 16 min |
| | | functest-merge-master | 4 hr 39 min - #69 | N/A | 45 sec |
| | | functest-merge-stable-arno | 29 days - #2 | N/A | 51 sec |
| | | functest-odl-test-orange-build | N/A | N/A | N/A |
| | | functest-openstack-bench-test-orange-build | N/A | N/A | N/A |
| | | functest-openstack-tempest-smoke-test-orange-build | N/A | N/A | N/A |
| | | functest-orange-build | N/A | 3 mo 22 days - #17 | 1 hr 15 min |
| | | functest-verify-master | 4 hr 44 min - #183 | N/A | 43 sec |
| | | functest-verify-stable-arno | 29 days - #6 | N/A | 47 sec |
| | | functest-vims-foreman-master | N/A | N/A | N/A |
| | | functest-vims-foreman-stable-arno | N/A | N/A | N/A |
| | | functest-vims-fuel-master | N/A | 4 hr 6 min - #12 | 1 hr 3 min |
| | | functest-vims-fuel-stable-arno | N/A | N/A | N/A |
| | | functest-vping-test-orange-build | N/A | N/A | N/A |

# CI & Automation

Daily verification of OPNFV platform



# Execution of tests



# Brahmaputra

release

# Functest in Brahmaputra release

- **Dockerization**
- **Test Result collection**
- **Dashboard**
- **Virtual IMS test case**
- **+ SDN Controllers tests**
- **Feature projects**

Docker -ization

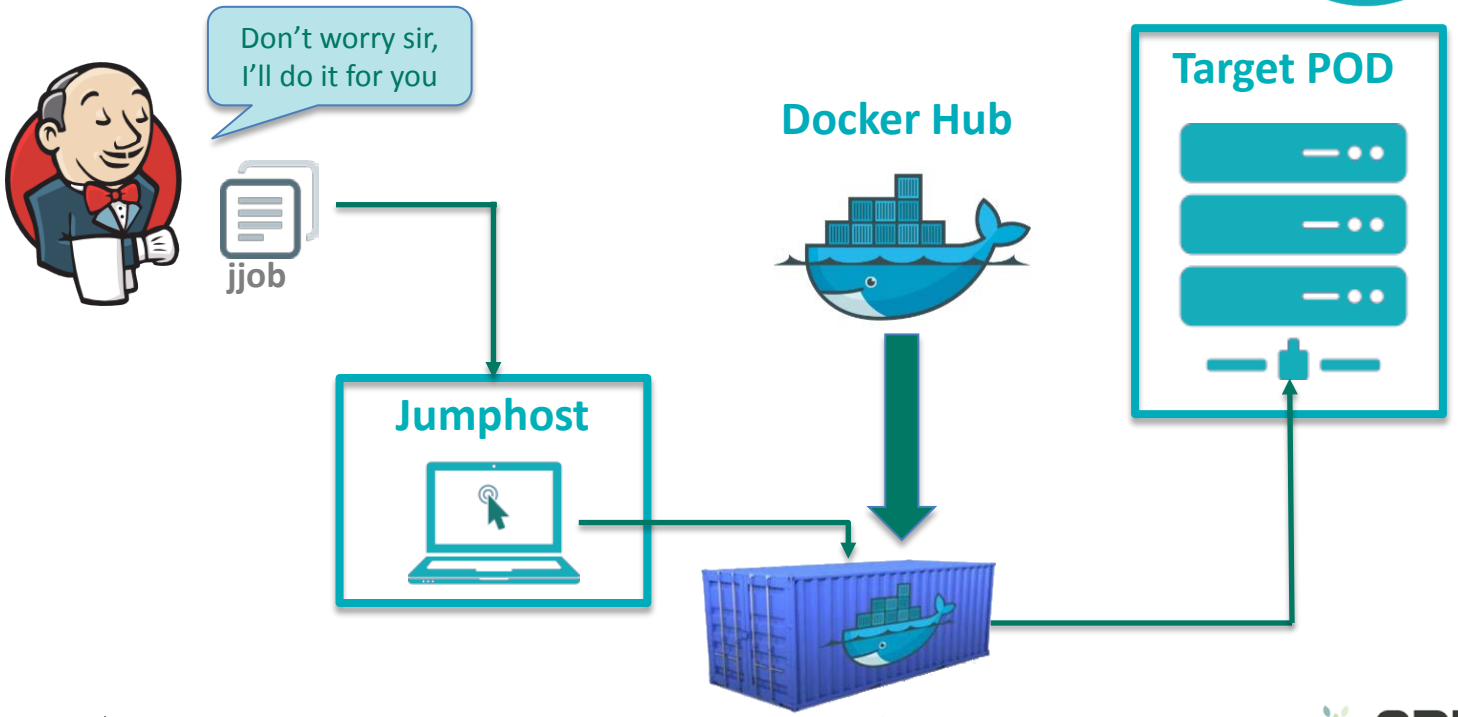
# Functest Docker

```
$docker run -t opnfv/functest $dir/run_test.sh
```

# Functest Docker

| Without Docker | With Docker |
| --- | --- |
| • Have to install needed libraries in your jumphost | • Needed libraries preinstalled in the docker image |
| • Maintain in the documentation the needed libraries and dependencies | • Update image when there is a new software dependency and push to the OPNFV Docker Hub |
| • Possible OS dependencies | • No OS dependencies |
| • Not portable | • Portable |

# Execution of tests



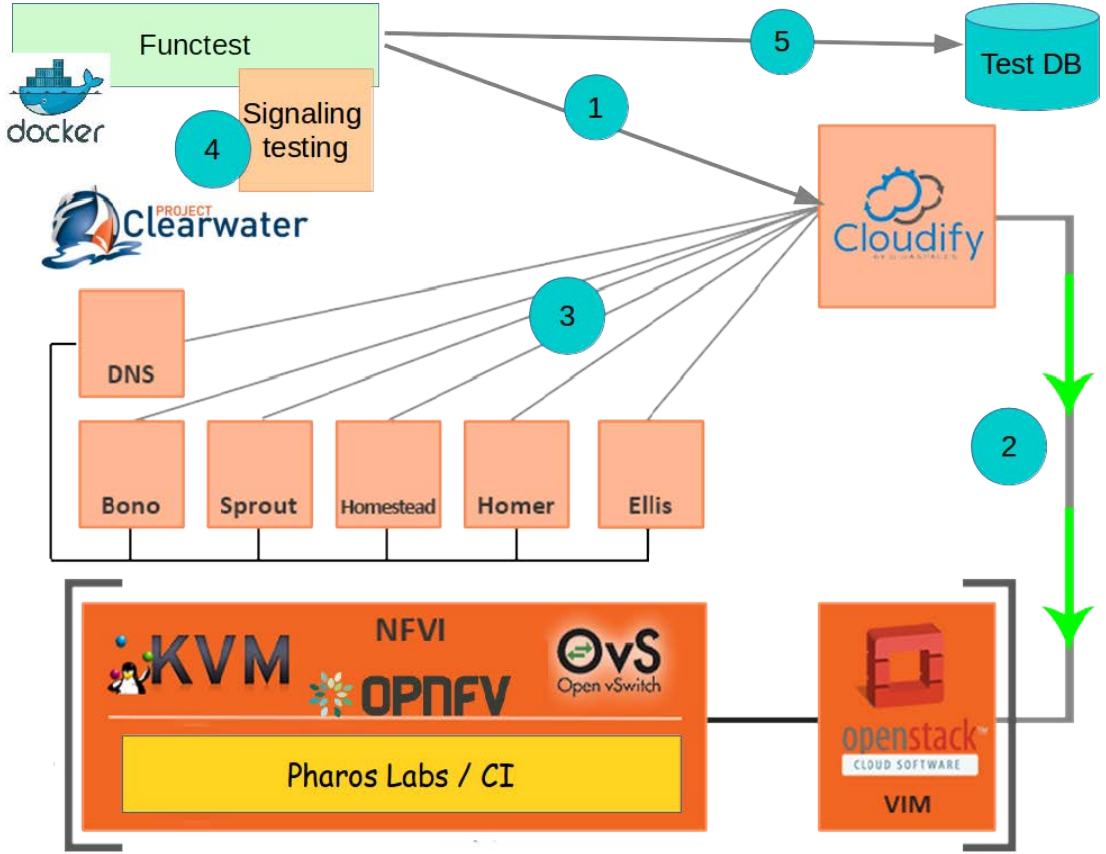

# vIMS test case

# vIMS test case

- Full automation of a core Open Source vIMS solution on OPNFV from Clearwater
- Deployment of an orchestrator (Cloudify) to manage the VNF
- Run suite of signaling tests on vIMS
- Successfully tested from Jenkins on LinuxFoundation POD2

# vIMS test case

1) Deploy the orchestrator
2) Create the tenant
3) Create the VNF (10 VMs)
4) Launch test suite (more than 100 signaling tests)
5) Collect results
6) Clean-up



# SDN Controllers

# SDN Controllers

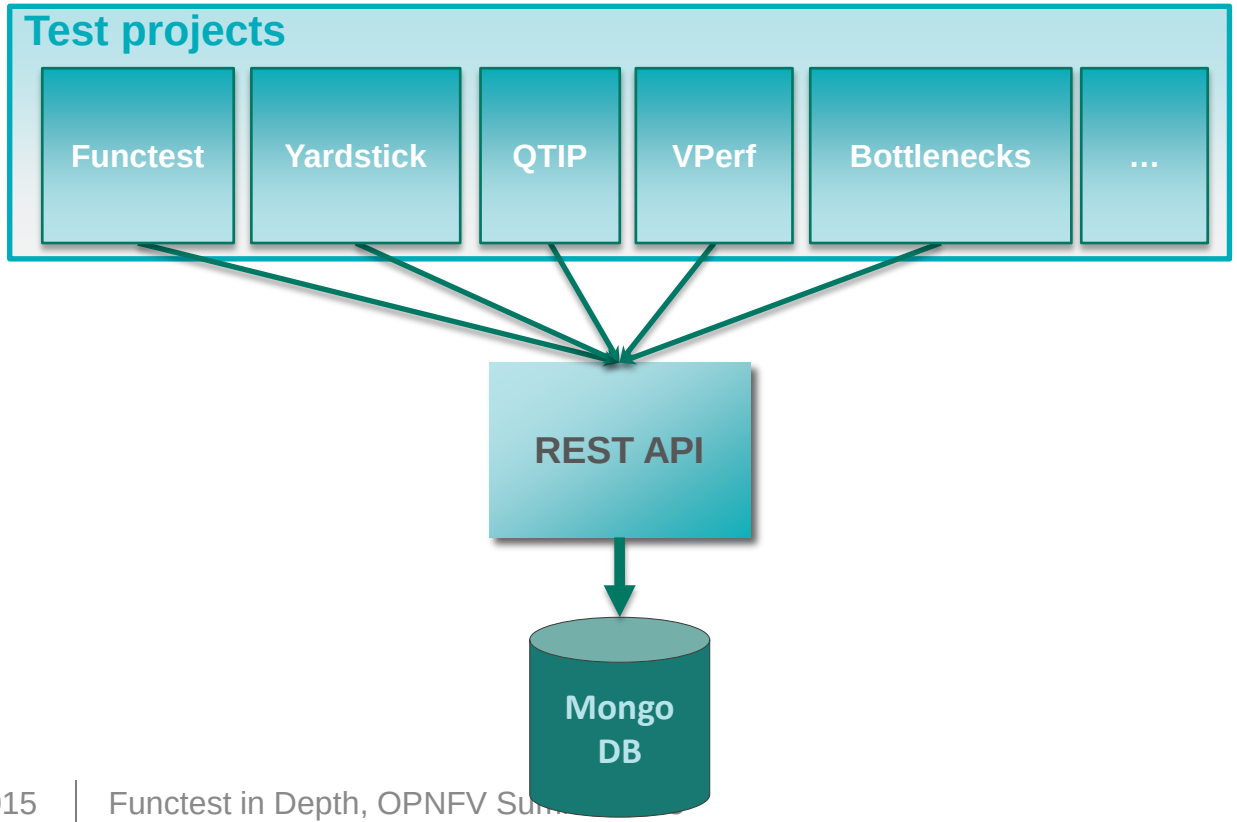
- Extension of OpenDaylight test cases


- Integration of ONOS test framework


- Open Contrail test

# Test results collection

# Test results collection



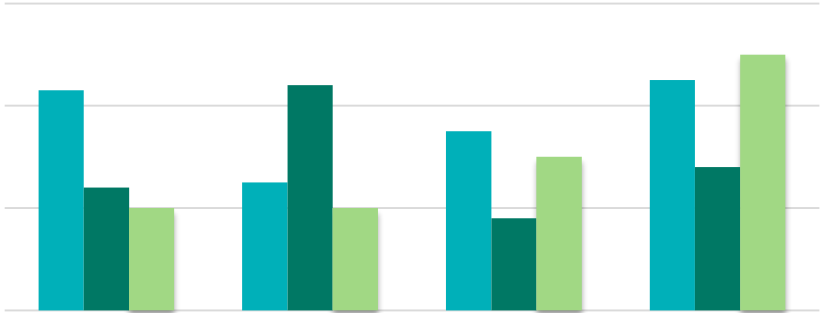
# Test results collection - API

- Filter per test case
  - http://<DB_Public_IP>/results?case=vPing
- Filter per pod
  - http://<DB_Public_IP>/results?pod=opnfv-jump-2
- Filter per project
  - http://<DB_Public_IP>/results?project=yardstick
- Combination of filters
  - http://<DB_Public_IP>/results?project=functest&case=Tempest&pod=opnfv-jump-2

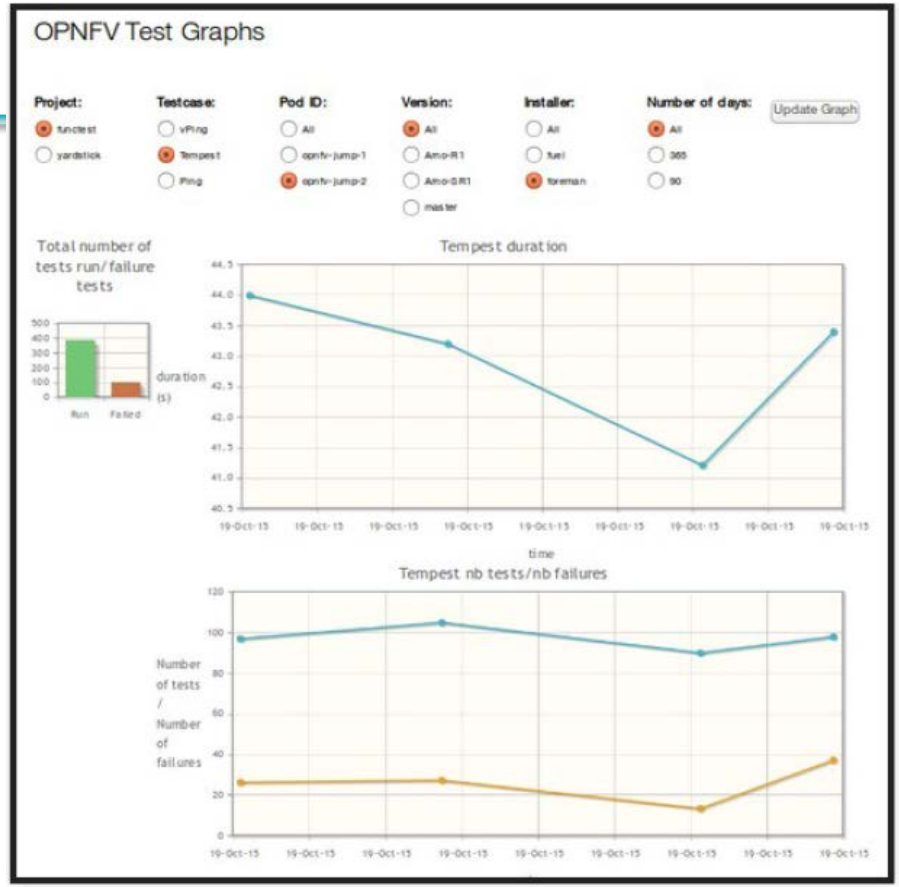
# Dashboard

# Dashboard

- Show all the results using simplistic **graphs**
- Possible to **filter** per project/test case/…
- **Comparison** of previous results with recent ones
- …



# Dashboard Draft



# Beyond Brahmaputra

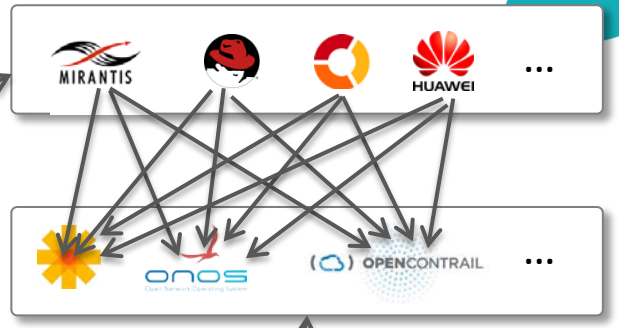
- New functional test (vCPE, vHGW, vCDN, …)
- Results analytics
- Better Rally usage
- More SDN controllers test integration
- Broaden test coverage (security, O&M, …)

# Lessons learned in Arno

- Need for stable labs some weeks before the release
- Need to run the tests on different PODs from CI
- Complex troubleshooting
- Functest cannot solve all problems

# Challenges

- Different OpenStack installers
- Different SDN controllers
- Many features




**Lots of combinations to be tested !!**

# Summary

- First test project in OPNFV
- Validate a fresh OPNFV deployment
- Umbrella for testing projects
- Umbrella for feature projects
- We don’t bite, come to us with questions ☺

# Thank you!