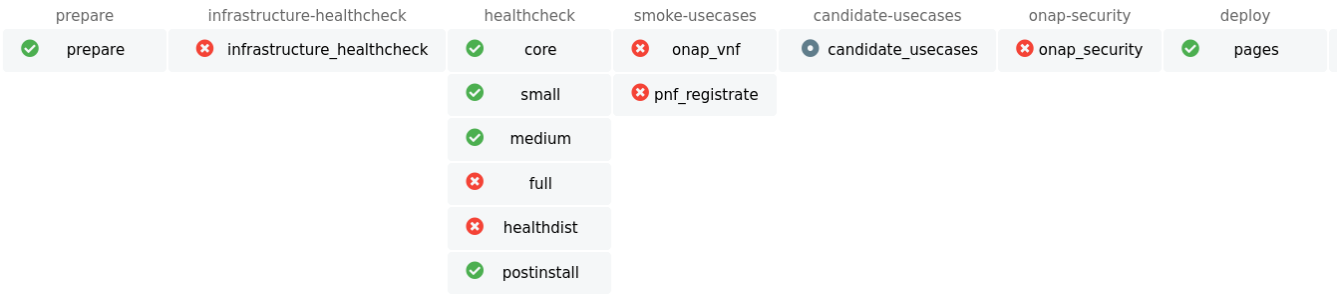
How can I include my test(s) in CI chains?

We would like to automate as many use cases as possible.
it means we would like to move from a declarative wiki status page corresponging to a use case run on one lab to a gate where the different use cases could be executed on any lab

today in the "gates" we have already some test suites
these test suites are systematically replayed every day on "Daily" chains and on gating chain (on patchset submission)

E.g. El Alto daily Gate



The testsuites usually include several tests. Tests can be run sequentially or in parallel.

To be integrated in CI, you must integrate your tests in of of the defined test categories (see Integration categories & testsuites discussions):

- infrastructure-healthcheck
- healthcheck
- smoke-usecases
- candidate-usecases
- security
- benchmark

The testing part

1) Your tests must be fully automated...

Your tests must be fully automated. It means that it shall take into account the env on which it is executed, setup the resources, run the tests, delete the resources created for the test
if you are using non open source VNF, proprietary third components/equipments you will not be able to fully automate.

Usually people start by automating partially the use cases, it means they develop scripts but do not always consider the setup/teardown phase (it should not be linked to the installation) and or require some manual steps preventing from a full automation.

You can develop your test using any language (python, bash, robotframework, go..) and any framework.

Ideally use case owner shall be able to provide the automatic procedure at this stage.

Please note that this part is the most important and represents about 95% of the work...writing good tests requires a good knowledge of the system under test.

2) ...and integrated in a docker based on Xtesting

As any framework is allowed, at the end we may have a huge diversity of tests, test frameworks, test artifacts. In order to simplify the integration of the tests in any CI chains, we decided to leverage the OPNFV xtesting framexork.

Whatever your tests, they will be launched always the same way and generate consistent artifacts. As they will be dockerized, they will be portable in any CI/CD system.

Your test suites must then be "xtestingized", it means they must be embedded in a docker file leveraging Xtesting.

This task can be done by the integration team - the most difficult is the test itself... If writing the test takes 90% of the the time, building a xtesting docker is about 4%.

Xtesting is a light framework aiming to harmonize test inputs and outputs, which is very helpful for integration.It has been developped in OPNFV and already proposed several infrastructure test dockers for Openstack and kubernetes, leveraging upstream tests (https://wiki.opnfv.org/pages/viewpage.action?pageId=13211751).

These tests have been selected for CNTT verification gates

It means that developers are free to do what they want, but at the end we add a light abstraction in order to

- launch the test the same way
- include a good management of the depenencies self content in a docker
- report the results the same way
- push optionnaly the results in a test Database

As a testsuite provider, you shall provide the entrypoint, the list of needed dependencies and dclare your tests.

You need to amend 3 files:

- a Dockerfile (to include your needed framework/test cade) - usually consists in installation/clon of repositories
- a yaml file testcases.yaml to declare your use case(s): in this file you will have to declare the test driver (robot, python, bash, behave,..*)
- a requirement.txt to indicate the dependencies

As an illustration, see https://gitlab.com/Orange-OpenSource/lfn/onap/integration/xtesting, you will see how the dockers are defined, built and shall be consumed by the CI chains.

And if you are able to run your test in 1 click, do not hesitate to contact the integration team to guide you on CI integration.

*: note that Xtesting is open source so if your favorite framework is missing you can add it

The CI part

This last part represent the missing 1%, it mainly deals with declaration of the case in the DB and in the CI system.

1) Your tests must be declared in the DB...

for that you need to have access to the common test DB used by OPNFV and ONAP: testresults.opnfv.org

An account onap-integration has been created. Public SSH keys of the contributors have been pushed to grant access to this machine.

So to log in use

```
$ ssh -i <key path> onap-integration@testresults.opnfv.org
```

Once log you can directly access to the database

```
$ mongo
MongoDB shell version: 3.2.16
connecting to: test
Server has startup warnings:
2020-10-01T07:54:55.852+0000 I CONTROL  [initandlisten]
2020-10-01T07:54:55.852+0000 I CONTROL  [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled
is 'always'.
2020-10-01T07:54:55.852+0000 I CONTROL  [initandlisten] **        We suggest setting it to 'never'
2020-10-01T07:54:55.852+0000 I CONTROL  [initandlisten]
2020-10-01T07:54:55.852+0000 I CONTROL  [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/defrag
is 'always'.
2020-10-01T07:54:55.852+0000 I CONTROL  [initandlisten] **        We suggest setting it to 'never'
2020-10-01T07:54:55.852+0000 I CONTROL  [initandlisten]
2020-10-01T07:54:55.852+0000 I CONTROL  [initandlisten] ** WARNING: soft rlimits too low. rlimits set to 4096
processes, 64000 files. Number of processes should be at least 32000 : 0.5 times number of files.
2020-10-01T07:54:55.852+0000 I CONTROL  [initandlisten]
singleNodeRepl:PRIMARY>
```

As DB admin, you may have to create/delete/update objects:

- pods (i.e. labs allowed to push results to the DB): http://testresults.opnfv.org/onap/api/v1/pods

- projects: http://testresults.opnfv.org/onap/api/v1/projects
- testcases (per projects): http://testresults.opnfv.org/onap/api/v1/projects/integration/cases

It is a basic Mongo DB, the different fields can be seen when accessing to the DB through the API.

Declare a new pod

```
singleNodeRepl:PRIMARY> use onap
switched to db onap
singleNodeRepl:PRIMARY> show collections
deployresults
pods
projects
results
scenarios
test
testcases
users
singleNodeRepl:PRIMARY> db.pods.insert({"name":"NEW_LAB_NAME","creator":"Cloud X","role":"gating","details":"
contact: The Admin of the lab","creation_date":"2020-10-15 8:00:00"})
```

Create a new use case

```
singleNodeRepl:PRIMARY> db.testcases.insert({'name':'basic_cnf', 'project_name': 'integration', 'tier':'smoke-
usecases', 'version': '>guilin', 'description':'Test the creation of simple cnf (nginx)', 'domains': 'onap',
'tags':'cnf,multicloud,so,k8splugin', 'catalog_description':'basic cnf'})
```

it is obviously possible to update, delete using usual mongo commands.

2) ... and in the CI

Once everything is declared in the DB, you can add the test in the CI chain.

This is done in the xtesting-onap project hosted in gitlab.com (as the CI is based on gitlab-ci)

You can create modify the gitlab-ci config and create a merge request

As the tests are embbeded in a xtesting dockers, in theory the integration in CI is simple because already done for other tests (robot, bash, python, ...)

Edit https://gitlab.com/Orange-OpenSource/lfn/onap/xtesting-onap/-/blob/master/gitlab-ci/base.yml

```
...
.MY_NEW_TEST: &MY_NEW_TEST
  variables:
    run_tiers: xtesting-smoke-usecases-robot
    run_type: my_new_test_declared in the xtesting entrypoint (shall be declared also in the DB if you want to
push the results to the DB)
    run_timeout: 1200
  stage: smoke-usecases
  allow_failure: true
  <<: *get_artifact
  <<: *runner_tags
  <<: *run_smoke_usecase_robot
  <<: *manage_artifacts

...

MY_NEW_TEST:
  <<: *MY_NEW_TEST
  only:
    refs:
      - triggers
```

You can assign the test to one of the declared stage/category (infrastructure-healthcheck, healthcheck, smoke-usecases, onap-security)

Xtesting references:

- the official xtesting documentation: https://xtesting.readthedocs.io/en/latest/
  Xtesting slide deck:http://testresults.opnfv.org/functest/xtesting/
  Xtesting 2020 slide deck update: http://testresults.opnfv.org/functest/xtesting2020/

Contacts

Morgan Richomme