How to re-run a test on a daily platform?

Sometimes you need to troubleshoot the errors reported during a daily chain

You must first connect to the daily lab.

The controller can be reached over SSH master.onap.eu, istanbul.onap.eu, master-weekly.onap.eu. (you SSH key must obviously be referenced see How to add keys to access integration labs?).

Once connected, it is a "usual ONAP" lab, so you can run your test as in any ONAP platform.

All the configurations of the tests executed during the daily chains can be found under /tmp/xtesting

```
tests

debian@control01-onap-master:~$ cd /tmp/xtesting
debian@control01-onap-master:/tmp/xtesting$ tree -L 2
.
  config
  infrastructure-healthcheck
    k8s
    k8s-teardown
  security
    jdpw_ports
    kube_hunter
    nonssl_endpoints
    root_pods
    unlimitted_pods
  versions
  smoke-ucases
    basic_cds
    basic_clamp
    basic_cnf
    basic_cnf_macro
    basic_network
    basic_onboard
    basic_vm
    basic_vm_macro
    cds_resource_resolution
    pnf_macro
  xtesting-healthcheck
    core
    cps-healthcheck
    full
    healthdist
    postinstall
  xtesting-smoke-usecases-robot
    cmpv2
    dcaemod
    hv-ves
    pnf-registrate
    ves-collector

33 directories, 1 file
```

The tests are run using 2 main mechanisms:

- k8s job in the onap namespace
- docker executed from the controller node

How to replay a kubernetes job?

```
k8s test

debian@control01-onap-master:~$ kubectl get job -n onap | grep integration
integration-onap-cmpv2                   0/1           9d         9d
```
integration-onap-core                   1/1           2m6s       9d
integration-onap-cps-healthcheck        1/1           32s        9d
integration-onap-dcaemod                0/1           9d         9d
integration-onap-full                   1/1           2m50s      9d
integration-onap-healthdist             1/1           5m58s      9d
integration-onap-hv-ves                 1/1           4m18s      9d
integration-onap-internal-check-certs   0/1           9d         9d
integration-onap-pnf-registrate         1/1           7m36s      9d
integration-onap-postinstall            1/1           101s       9d
integration-onap-ves-collector          1/1           11m        9d

# Delete the job of the test you want to replay
debian@control01-onap-master:~$ kubectl delete job -n onap integration-onap-cps-healthcheck
job.batch "integration-onap-cps-healthcheck" deleted

# Recreate the job (it will automatically relaunch the test pod)
debian@control01-onap-master:~$ cd oom/master/
debian@control01-onap-master:~/oom/master$ ls
deployment.yaml         healthcheck-cps-healthcheck.yaml  healthcheck-healthcheck-healthdist.yaml            healthcheck-
                         pnf-registrate.yaml onap-overrides.yaml
healthcheck-cmpv2.yaml  healthcheck-dcaemod.yaml          healthcheck-hv-ves.yaml                healthcheck-
postinstall.yaml      xtesting-healthcheckhealthcheck-core.yaml   healthcheck-full.yaml             healthcheck-internal-check-certs.yaml  healthcheck-
                       ves-collector.yaml   xtesting-healthcheck
healthcheck-healthcheck-full.yaml         healthcheck-cps-healthcheck.yaml.yml --validate=false
kubectl apply -f healthcheck-cps-healthcheck.yaml

# Then look at the logs
kubectl logs -n onap integration-onap-cps-healthcheck--1-m7tbt -f
2022-03-07 13:27:41,030 - xtesting.ci.run_tests - INFO - Deployment description:

+-------------------------+----------------------------------------------------------+
|         ENV VAR         |                          VALUE                           |
|-------------------------+----------------------------------------------------------|
|         CI_LOOP         |                          daily                           |
|          DEBUG          |                          false                           |
|     DEPLOY_SCENARIO     |                   onap-nofeature-noha                    |
|      INSTALLER_TYPE     |                           oom                            |
|        BUILD_TAG        |      gitlab_ci-functest-kubespray-baremetal-daily-       |
|                         |                  master-479785435-onap                   |
|        NODE_NAME        |             onap_weekly_pod4_master-ONAP-oom             |
|       TEST_DB_URL       |     http://testresults.opnfv.org/onap/api/v1/results     |
|     TEST_DB_EXT_URL     |                                                          |
|     S3_ENDPOINT_URL     |                                                          |
|        S3_DST_URL       |                                                          |
|       HTTP_DST_URL      |                                                          |
+-------------------------+----------------------------------------------------------+

2022-03-07 13:27:41,321 - xtesting.core.robotframework - INFO -

Actuator :: CPS - Actuator endpoints
Test CPS Enhanced Healthcheck :: Runs CPS Health Check. It will ch... | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed

Output: /var/lib/xtesting/results/cps-healthcheck/output.xml

2022-03-07 13:27:41,326 - xtesting.core.robotframework - INFO - Results were successfully parsed
2022-03-07 13:27:41,395 - xtesting.core.robotframework - INFO - Results were successfully generated
2022-03-07 13:27:41,395 - xtesting.ci.run_tests - INFO - Test result:

<table>
<thead>
<tr>
<th>TEST CASE</th>
<th>PROJECT</th>
<th>DURATION</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>cps-healthcheck</td>
<td>integration</td>
<td>00:00</td>
<td>PASS</td>
</tr>
</tbody>
</table>

+-------------------------+---------------------+------------------+----------------+
|        TEST CASE        |       PROJECT       |     DURATION     |     RESULT     |
|-------------------------+---------------------+------------------+----------------+
|     cps-healthcheck     |     integration     |      00:00       |      PASS      |
|-------------------------+---------------------+------------------+----------------+
How to replay a docker based test?

Several tests are executed by running a docker file on the controller. It is used for:

- infrastructure tests
- security tests
- pythonsdk-test tests

**docker tests**

```bash
# Select the test you want to replay (e.g. basic_cds)
debian@control01-onap-master:~$ cd /tmp/xtesting//smoke-usecases/basic_cds/

debian@control01-onap-master:~$ cd /tmp/xtesting//smoke-usecases/basic_cds/

# run the docker

debian@control01-onap-master:~$ cd /tmp/xtesting//smoke-usecases/basic_cds/

# you shall be in the docker
# for infrastructure and security tests you can directly run the test
# for pythonsdk you must copy/paste the content of the /etc/hosts (you could also use parameter in the docker exec command if you want), as onap url are not publicaly routable

# bash-5.1# vi /etc/hosts
# copy the content displayed by the command cat /etc/hosts |grep onap.org
bash-5.1# cat /etc/hosts

127.0.0.1 localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::1 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.17.0.2 265c7c83aaf0
10.253.0.217 portal.api.simpledemo.onap.org
10.253.0.217 vid.api.simpledemo.onap.org
10.253.0.217 sdc.api.fe.simpledemo.onap.org
10.253.0.217 sdc.api.be.simpledemo.onap.org
10.253.0.217 portal-sdk.simpledemo.onap.org
10.253.0.217 policy.api.sparky.simpledemo.onap.org
10.253.0.217 cli.api.simpledemo.onap.org
10.253.0.217 msb.api.simpledemo.onap.org
10.253.0.217 so.api.simpledemo.onap.org
10.253.0.217 appc.api.simpledemo.onap.org
10.253.0.217 sdnc.api.simpledemo.onap.org
10.253.0.217 nbi.api.simpledemo.onap.org
10.253.0.217 consul.api.simpledemo.onap.org
```
# execute the xtesting command

bash-5.1# run_tests -t basic_cds

2022-03-07 13:46:13,968 - xtesting.ci.run_tests - INFO - Deployment description:

<table>
<thead>
<tr>
<th>ENV VAR</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI_LOOP</td>
<td>daily</td>
</tr>
<tr>
<td>DEBUG</td>
<td>False</td>
</tr>
<tr>
<td>DEPLOY_SCENARIO</td>
<td>os-nosdn-nofeature-noha</td>
</tr>
<tr>
<td>INSTALLER_TYPE</td>
<td>oom</td>
</tr>
<tr>
<td>BUILD_TAG</td>
<td>gitlab_ci-functest-kubespray-baremetal-daily-master-479785435-onap</td>
</tr>
<tr>
<td>NODE_NAME</td>
<td>onap_weekly_pod4_master-ONAP-oom</td>
</tr>
<tr>
<td>TEST_DB_URL</td>
<td><a href="http://testresults.opnfv.org/onap/api/v1/results">http://testresults.opnfv.org/onap/api/v1/results</a></td>
</tr>
<tr>
<td>TEST_DB_EXT_URL</td>
<td></td>
</tr>
<tr>
<td>S3_ENDPOINT_URL</td>
<td></td>
</tr>
<tr>
<td>S3_DST_URL</td>
<td></td>
</tr>
<tr>
<td>HTTP_DST_URL</td>
<td></td>
</tr>
</tbody>
</table>

2022-03-07 13:46:14,013 - xtesting.ci.run_tests - INFO - Loading test case 'basic_cds'...

2022-03-07 13:46:14,022 DEBUG 25:cds_blueprint_enrichment.py(13) - CDS blueprint enrichment initialization
2022-03-07 13:46:14,822 DEBUG 219:extension.py(13) - found extension EntryPoint(name='basic_clamp', value='onaptests.scenario.basic_clamp:BasicClamp', group='xtesting.testcase')

2022-03-07 13:46:33,393 INFO 176:run_tests.py(13) - Test result:

<table>
<thead>
<tr>
<th>TEST CASE</th>
<th>PROJECT</th>
<th>DURATION</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>basic_cds</td>
<td>integration</td>
<td>00:18</td>
<td>PASS</td>
</tr>
</tbody>
</table>

2022-03-07 13:46:33,397 INFO 51:cds_blueprint_enrichment.py(13) - Generate Test report
2022-03-07 13:46:33,477 INFO 51:cds_blueprint_enrichment.py(13) - Generate Test report
2022-03-07 13:46:33,477 INFO 275:run_tests.py(13) - Execution exit value: Result.EX_OK

# note 1: the results are in /tmp/reporting.html and the logs in /tmp/pythonsdk.debug.log
# note 2: the env var set in the env file shall corresponds to the test you run e.g for basic_cds => ONAP_PYTHON_SDK_SETTINGS=onaptests.configuration.cba_enrichment_settings
# note 3 the name of the test is in the test DB and defined as the entry point of xtesting https://github.com/onap/testsuite-pythonsdk-tests/blob/master/setup.cfg
# note 4, if you want to play directly with the python test or sdk code, it is respectively under /usr/lib/python3.8/site-packages/onaptests and /usr/lib/python3.8/site-packages/onapsdk

bash-5.1# ls /usr/lib/python3.8/site-packages/onaptests
__init__.py  __pycache__  configuration  scenario  steps  templates  utils
bash-5.1# ls /usr/lib/python3.8/site-packages/onapsdk
__init__.py  aai  clamp  constants.py  exceptions.py  nbi
sdc  so  version.py  vid
__pycache__  cds  configuration  dmaap  msb  onap_service.py
sdnc  utils  ves