Creating an installation cross component health test

This page will outline the steps to create a cross component health test that could be used to confirm an installation from the standpoint of multiple components that need to work together as part of a end to end test or larger use case test. It does not replace the need for per component health checks but rather augments those initial health checks with tests that confirm the multiple components that depend on each other actually work for a basic sunny day thread.

The heatldist tag in robot is one of those tests that confirms that model distribution works by onboarding the simplest vFW VNF model and confirming that model distribution works.

We will use healthdist as an example for the process to create new tests in the robot framework repository (gerrit repo: testsuite)

Healthdist – is a tag that exists to run an onboarding and distribution of the vFW VNF (a simple 3 VM VNF as you know).

It tests that SDC to SO, AAI, SDNC via DMaaP is working. It generally uses the asdc_interface.robot script which exists for other end to end tests (e.g. the 'distribute' tag) but for just one VNF.

Assuming a good ONAP installation where healthcheck passes and robot's demo.sh init has been run to populate customers (not necessarily needed but always a best practice).

To run it execute:

For OOM installs from the ~/oom/kubernetes/robot directory

```
./ete-k8.sh onap healthdist
```

or

For HEAT installs from the robot VM

```
./ete.sh healthdist
```

This will onboard the vFW HEAT template as a Vendor Software Product, Create the Service model and then distribute the service from SDC via DMaaP to SO, AAI, SDNC etc. Success is a DISTRIBUTION_COMPLETE_OK in the final status from SO back to SDC. It only tests distribution not that VID sees the new model nor that the model can be instantiated. That would be for tests like 'instantiate' or 'instantiateVFW' or using the VID GUI since the model should be visible on VID under Browse Models.

- Ete.sh is simply a script that calls a robot test with the provided tag “heatlhdist”
- ~/testsuite/robot/testsuites/health-check.robot is the top level file that has the tag “heatlthdist” which calls the robot test:”Model Distribution For Directory”
- “Model Distribution For Directory” is in a test template ~testsuites/model-distribution.robot where it is re-used by other tests and this is an alternate path when we tie model distribution into other test suites.
- ~/testsuite/robot/resources/test_templates/model_test_template.robot is the source for the keyword “Model Distribution For Directory”
- This particular test in robot creates the zip file of the VNF Vendor Software Product for distribution from the data in the demo/heat project directory and then calls the keyword: “Distribute Model From ASDC”
- “Distribute Model From ASDC” is in the ~/testsuite/robot/resources/asdc_interface.robot and is an function used for distribution for other tests. It has all the keywords for interacting with SDC both regular health but other GET/PUT actions to SDC. For healthdist there was an existing function to use and that will generally be the case. New interfaces can be added if there isn't an existing robot interface file for a project.

Adding a new health/sanity test to health-check.robot would consist first of creating any needed additions to the *_interface.robot script.

If we wanted to consider a 5G closed loop use case; there is a policy_interface.robot, dcae_interface.robot, clamp_interface.robot, mr_interface.robot (DMaaP Message Router) already available. Re-use existing keywords as much as you can. But new keywords can be added to the appropriate interface file.

Build the test case in a test_templates to combine functions across project_interface keywords like “Model Distribution For Directory”

Finally stitch the test case into a tag (healthdist) in the health-check.robot file.

Configuring ONAP in preparation for a sanity test

If you need to do work that configures ONAP we put that into 'demo.sh init' which has a bit more tag checking but works basically the same way as ete.sh.

Testing your new tag/test/keyword

For testing I usually create a tag but don't document it in ete.sh/demo.sh and simply make a copy of ete.sh/demo.sh with my tag added to a 'ete.test.sh' for instance.

The ete.sh/demo.s ete-k8.sh/demo-k8.sh are outside of the container either in /opt on the robot VM in HEAT or in ~/oom/kubernetes/robot on rancher in the OOM

Install so its easy to add test version of those scripts to your local ONAP environment.