Function Test

Introduction

Function test is conducted to verify the functionality of the Non-RT RIC as well as the individual components.

A test engine and a number of test script is used as a foundation for the test environment.

The test environment uses docker images both the Non-RT RIC components as well as for the the needed simulators.

Function test is normally started in "docker-mode" meaning that images will be started as container in a private docker network. As an option, the same test case also executed in a minikube environment. Running in minikube is still in an experimental state so docker should be used as a the standard execution environment.

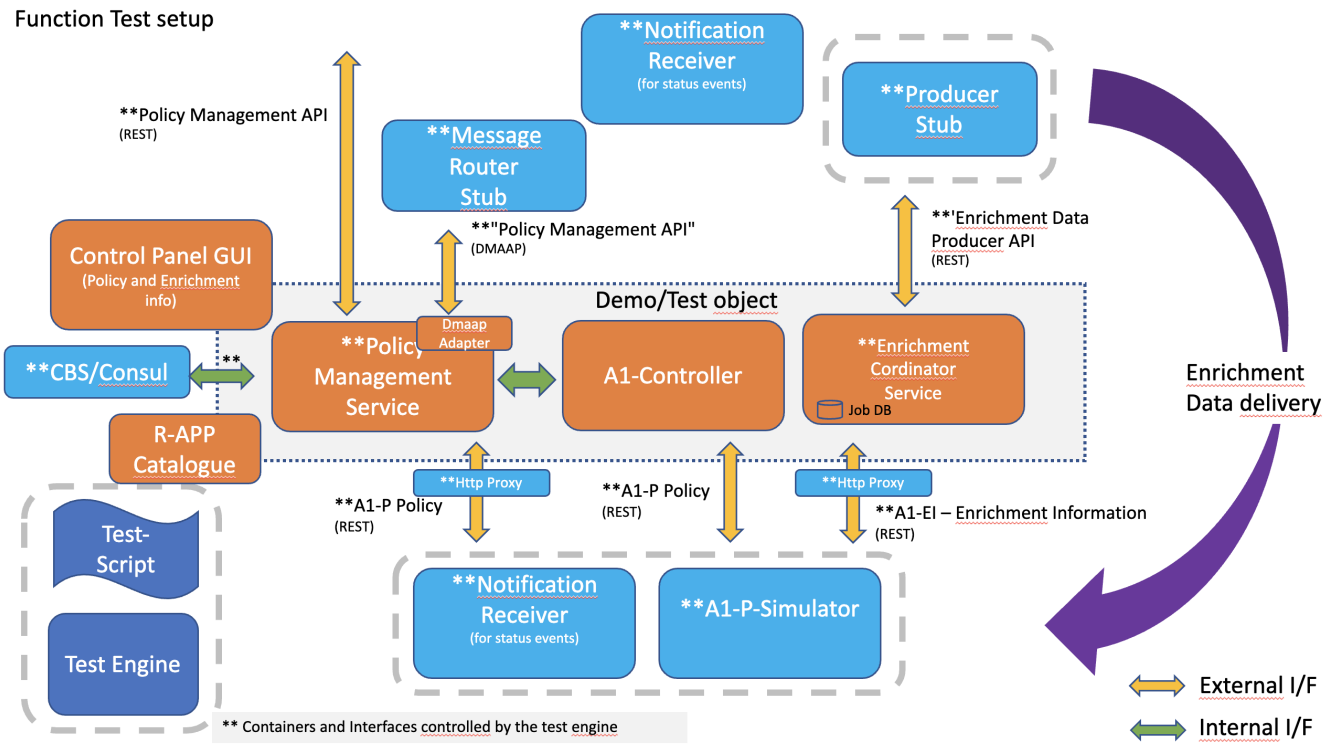
The basic idea of function test is to run the Non-RT RIC components and test the exposed interfaces using simulators. The test engine and the test scripts controls the behaviour of the simulators but also use the exposed interfaces for test and monitoring.

The test scripts are designed in a way so that the same script can be used for testing of the functionality in the master branch as well as previous release (s). The supported branches/releases are indicated by the existence of a environment variable file in the test/common dir with the name of the branch. The environment for master has the name of coming release.

The majority of the test scripts test one or several aspects of the functionality but there is also test scripts for stability tests over longer period of time (currently 3 days max intensity test).

Test setup

The picture below shows the setup used for function tests. Non-RT RIC components in orange, simulators in light blue and the test environment in dark blue. The purple arrow is outside the scope of the test but is used for demo test scripts.



Note: The Control Panel GUI consist of two parts, control panel and the gateway for accessing the nonrtric components.

Note: When running in kubernetes, the test engine uses a proxy (not shown) for accessing the nonrtric components.

Note: The helm manager, Dmaap Adaptor and Dmaap Mediator components are not yet included in the function test setup.

Note: Enrichement Coordinator has changed name to Information Coordinator

Test scripts/Test cases

The below table is a list of test cases. All scripts beginning with the letters "FTC" (Function Test Case) are considered test scripts to test functionality. The remaining test scripts are primarily used for demo purposes.
Note: Control Panel and its gateway are started in most test cases but not tested.

Test case	Tested apps	Slogan
FTC1	PMS	Sanity test, create service and then create, update and delete a policy using http/https and Agent REST/DMAAP with/with out SDNC controller
FTC10	PMS	Basic use case, register service, create/update policy, delete policy, de-register service using both STD and OSC interface while mixing REST and Dmaap
FTC100	PMS	Full agent API walkthrough using agent REST/DMAAP and with/without SDNC A1 Controller
FTC110	PMS	Testing of service registration timeouts and keepalive
FTC150	SDNC	Sample tests of the SDNC A1 controller restconf API using http/https (no PMS)
FTC300	PMS	Resync 10000 policies using OSC and STD interface
FTC310	PMS	Resync of RIC via changes in the consul config or pushed config
FTC350	PMS	Change supported policy types and reconfigure rics
FTC800	PMS	Create 10000 policies in sequence using http/https and Agent REST/DMAAP with/without SDNC controller
FTC810	PMS	Repeatedly create and delete policies in each RICs for 24h (or configured number of days). Via agent REST/DMAAP /DMAAP_BATCH and SDNC using http or https
FTC850	PMS	Create/delete policies in parallel over a number of ric using a number of child process
FTC900	PMS	Preparation for test of the Control Panel and the Health Check app - populating a number of rics simulators with types and instances"
FTC1100	ICS	ICS full interface walkthrough
FTC1800	ICS	ICS Create 10000 jobs and restart, test job persistency
FTC2001	PMS ICS	Testing southbound proxy for PMS and ICS
FTC2002	SDNC	Testing southbound proxy for SDNC - docker only
FTC2003	DMAAP-ADP	Testing southbound proxy for Dmaap Adaptor
FTC3000	DMAAP-ADP DMAAP-MED	App test DMAAP Meditor and DMAAP Adapter
FTC3001	DMAAP-ADP DMAAP-MED	App test DMAAP Meditor and DMAAP Adapter with 100 jobs,types and topics
FTC4000	HELM-MANAGER	Test of Helm Manager
PM_DEMO	PMS	Preparation demo setup - PMS - populating a number of ric simulators with types and instances
PM_EI_DEMO	PMS ICS RAPP-CATALOGUE	Preparation demo setup - PMS and ICS - policy management and enrichment information

Test execution

All function test scripts can be executed in your local environment. It is proven to work on MacOS and Ubuntu (brief tests has been made on Windows using git bash).

Required environment:

- Bash shell
- docker (latest)
- docker-compose (latest)
- python3 (latest)
- minikube (latest) or a local kubernetes cluster - only needed when running in kubernetes mode

The test engine as well as all available test scripts is available in the nonrtric repo. Clone the repo and go to the auto-test directory:

```
$ git clone "https://gerrit.o-ran-sc.org/r/nonrtric"
$ cd nonrtric/test/autotest
```

Start the appropriate test script with the command below. No further action is required. The tests are fully automated. Once the execution is completed a test report is printed.

Example, run images from the staging repo - which is the default. To run only release images, add the parameter 'release' after the 'docker' parameter.

When running with images from the staging or release repo, it is still possible to point to images in the local repo or in the release, staging or snapshot repo for individual applications.

```
$ ./FTC100.sh remote docker --env-file ../common/test_env-oran-cherry.sh
```

For further information and details, see README in the dirs 'auto-test' and 'common'.

Tested image versions

Component	image (release repo)	tag for d-release profile	tag for d-release profile	tag for cherry profile
Policy Management Service (new image name)	nexus3.o-ran-sc.org:10002/nonrtric-a1-policy-management-service	2.3.1	-	-
Policy Management Service (old image name)	nexus3.o-ran-sc.org:10002/o-ran-sc/nonrtric-policy-agent	-	2.2.0	2.1.1
A1-controller	nexus3.o-ran-sc.org:10002/o-ran-sc/nonrtric-a1-controller	-	-	2.0.1
SDNC A1 Controller (Replacement for the above A1-Controller)	nexus3.onap.org:10002/onap/sdnc-image	2.1.6	2.1.2	-
Information Coordinator Service (Replaces Enrichment Coordinator Service)	nexus3.o-ran-sc.org:10002/o-ran-sc/nonrtric-information-coordinator-service	1.2.1	-	-
Enrichment Coordinator Service	nexus3.o-ran-sc.org:10002/o-ran-sc/nonrtric-enrichment-coordinator-service	-	1.1.0	1.0.1
Control Panel	nexus3.o-ran-sc.org:10002/o-ran-sc/nonrtric-controlpanel	2.3.0	2.2.0	2.1.0
Gateway	nexus3.o-ran-sc.org:10002/o-ran-sc/nonrtric-gateway	1.1.0	1.1.0	-
A1-Simulator	nexus3.o-ran-sc.org:10002/o-ran-sc/a1-simulator	2.2.0	2.1.0	2.1.0
R-APP Catalogue	nexus3.o-ran-sc.org:10002/o-ran-sc/nonrtric-r-app-catalogue	1.0.2	1.0.1	1.0.1
Dmaap Adaptor	nexus3.o-ran-sc.org:10002/o-ran-sc/nonrtric-dmaap-adaptor	1.0.1	-	-
Dmaap Mediator	nexus3.o-ran-sc.org:10002/o-ran-sc/nonrtric-dmaap-mediator-producer	1.0.1	-	-

"profile" refers to the name of the environment variable profile used when executed a test case. The profiles has the pattern 'test_env-oran-<profile-name>.sh in the test/common dir.