VTP Test case Result Translation to OVP format

Kanagaraj.Manickam@Huawei.com

ONAP VNFSDK VNF Test Platform (VTP)

Requirements

- Write python script to generate the result in OVP format from VTP format
- More details
    - VTP helps to run test cases and produce the result in following format
    https://wiki.onap.org/display/DW/OVP-VTP#OVP-VTP-Userguide
    - OVP defines the result format as defined here https://wiki.opnfv.org/display/dovetail/Results+Requirements+of+OVP+Portal

Python script

- **Vtp2ovp-result –execution-id <execution-id> -- vtp-home <vtp-home>**
  - **Usage: Helps to produce the VTP results in LFV OVP format. Run this command from the server machine, where VTP is running.**
  - **execution-id is mandatory, VTP test case execution id**
  - **vtp-home is option, if not given this command uses environment variable OPEN_CLI_HOME.**

NOTE:

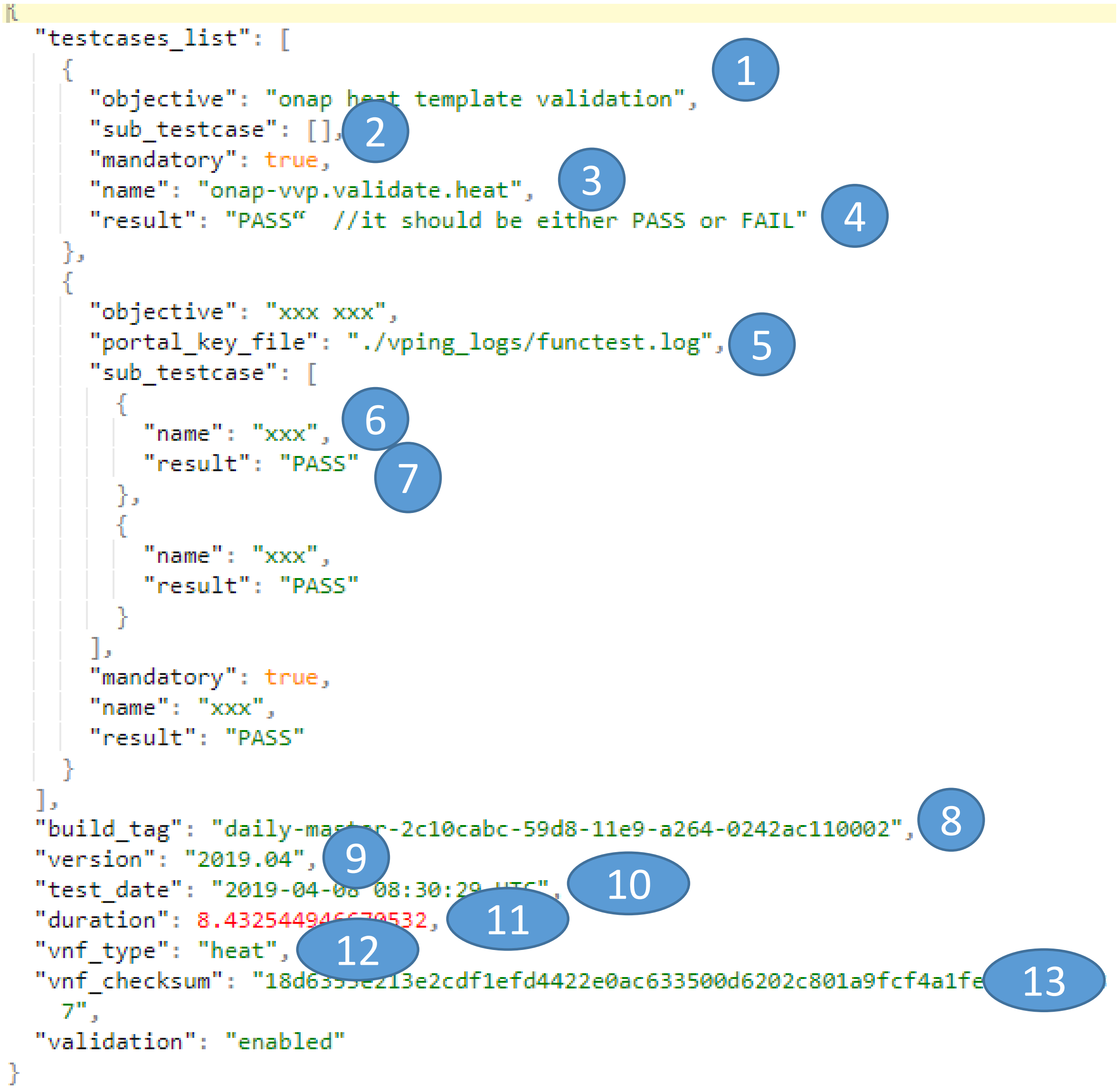
To implement use Use python argparse.ArgumentParser

1. Mkdir temp folder and cd to that folder
2. Create result.json (and populate by referring slide 5)
3. Copy vtp-home/data folder here
4. Create tar with vtp-ovp-result-<execution-id>.tar of this folder
5. Display the result.json and path to the tar file in console.

VTP result vs OVP result

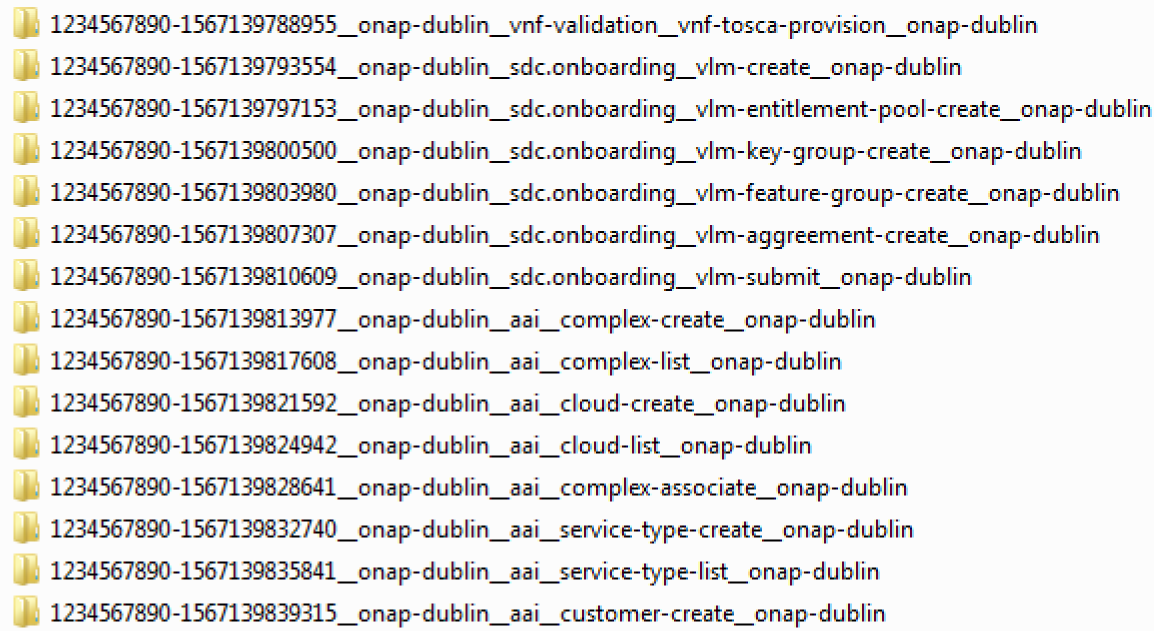
data.zip
Sample vtp test results

OVP test case results

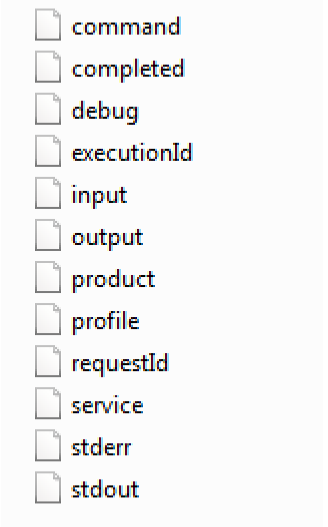


```
{
  "testcases_list": [
    {
      "objective": "onap heat template validation",
      "sub_testcase": [],
      "mandatory": true,
      "name": "onap-vvp.validate.heat",
      "result": "PASS“  //it should be either PASS or FAIL"
    },
    {
      "objective": "xxx xxx",
      "portal_key_file": "./vping_logs/functest.log",
      "sub_testcase": [
        {
          "name": "xxx",
          "result": "PASS"
        },
        {
          "name": "xxx",
          "result": "PASS"
        }
      ],
      "mandatory": true,
      "name": "xxx",
      "result": "PASS"
    }
  ],
  "build_tag": "daily-master-2c10cabc-59d8-11e9-a264-0242ac110002",
  "version": "2019.04",
  "test_date": "2019-04-08 08:30:29 UTC",
  "duration": 8.432544946670532,
  "vnf_type": "heat",
  "vnf_checksum": "18d6333e213e2cdf1efd4422e0ac633500d6202c801a9fcf4a1fe
    7",
  "validation": "enabled"
}
```

VTP test case results



```
1234567890-1567139788955__onap-dublin__vnf-validation__vnf-tosca-provision__onap-dublin
1234567890-1567139793554__onap-dublin__sdc.onboarding__vlm-create__onap-dublin
1234567890-1567139797153__onap-dublin__sdc.onboarding__vlm-entitlement-pool-create__onap-dublin
1234567890-1567139800500__onap-dublin__sdc.onboarding__vlm-key-group-create__onap-dublin
1234567890-1567139803980__onap-dublin__sdc.onboarding__vlm-feature-group-create__onap-dublin
1234567890-1567139807307__onap-dublin__sdc.onboarding__vlm-aggreement-create__onap-dublin
1234567890-1567139810609__onap-dublin__sdc.onboarding__vlm-submit__onap-dublin
1234567890-1567139813977__onap-dublin__aai__complex-create__onap-dublin
1234567890-1567139817608__onap-dublin__aai__complex-list__onap-dublin
1234567890-1567139821592__onap-dublin__aai__cloud-create__onap-dublin
1234567890-1567139824942__onap-dublin__aai__cloud-list__onap-dublin
1234567890-1567139828641__onap-dublin__aai__complex-associate__onap-dublin
1234567890-1567139832740__onap-dublin__aai__service-type-create__onap-dublin
1234567890-1567139835841__onap-dublin__aai__service-type-list__onap-dublin
1234567890-1567139839315__onap-dublin__aai__customer-create__onap-dublin
```



```
command
completed
debug
executionId
input
output
product
profile
requestId
service
stderr
stdout
```

Map VTP result to OVP result

1. Find the folder name with given execution-id and add it as value after stripping off the execution id from the folder name
2. Split the execution-id by “-” and name first part as ‘request_id’.
    I. Search the vtp-home/data/executions folder with folder names starting with request_id. Each of these folder is sub test cases.
    II. For each sub test cases, populate 6 & 7
    III. If no sub testcases, make it empty array
3. Same as 1
4. If <folder name starts with execution-id>/completed file exist, then PASS. Otherwise FAIL
5. Path to the **./data/executions/<folder name starts with execution-id>/output** file
6. From Subtest case folder name, produce value same as 1
7. Same as 4
8. Execution-id
9. 2019.1
10. Updated date and time of **<folder name starts with execution-id>**
11. Time diff between <folder name starts with execution-id>/input and <folder name starts with execution-id>/output files creation time
12. TOSCA
13. Md5 checksum of executions folder zip.

Thank you