Agenda:

- VSPERF Demo (relevance for CNTT testing) - Sridhar Rao
  - Following up on the planned tests from last week, a full demo will be presented
  - Details on VSPERF integration with X-testing were presented.
    - X-testing works very well, adding a VSPERF test was very efficient
    - TestAPI is used to collect the X-testing results
      - The current template of objects needs to be expanded so that several CNTT-related fields can be added
        - For example, the CNTT test requirement name(s) need to be included, so that requirements compliance can be verified easily
        - Note that the the CNTT test requirement name(s) and test names mapping are deterministic, but one test may cover multiple requirements
        - For performance tests, a threshold is needed for input to the PASS/FAIL criteria field (already part of the API).
        - Ideally, new fields would be added without breaking existing tool communication/population of the test db, or the testdb itself!
        - ALL test configuration files would need to be uploaded to OPNFV Artifacts, and linked from the test results
        - There must be a general description of the CNTT-candidate's test environment, so the info is not repeated in every individual test configuration
      - This is an example where OPNFV requires more resources to support CNTT and OVP badging program.

- Additional Notes from Sridhar:
  - X-Testing Integration
    1. Things work great! - Just had to follow this link Deploy your own Xtesting CI/CD toolchains to come up with example integration.
    2. However, we need to Understand the jobs that get created in Jenkins once all starts running. For example, for 1 testcase, we say multiple jobs and some fail and some pass after few days. If there is a place where this is described, it would be very helpful to know that.
    3. X-Testing automatically (based on the testcases that is defined) sets up the TestAPI project, testcases and results database. VSPERF has to make changes to the way it publishes the results to testapi. In VSPERF, the testcase name used while running the test is different from the testcase name defined in the OPNFV Testapi - the vswitch name gets appended before publishing. This doesn't work in X-Testing, and it has to be the same.

- VSPERF action items: (a) Publish results to X-testing Testapi (b) Fix the Jenkins Failures (c) Add more testcases

Test_API Enhancement (in Details section)

```json
Id :
Build tag :
Case name :
Criteria :
Installer :
Pod name :
Project name :
Scenario :
Start date :
Stop date :
Version :
Details :
{
  standards_ref: (cntt requirement, etsi tst, nomenclature)
  performance: true/false
  threshold: (for pass/fail criteria)
  metric: (us, bps, ....)
  links to OPNFV Artifact: (configuration file, env-details, csvs, )
  ....
}
```

Further Discussion on Test API Test DB at the April 2020 Virtual Event

What project will own and maintain the code for this? Releng(no opinion, just the home for the code) - Dovetail - X-testing (a project that lives under Functest)

- Releng - testresults project volunteers - none so far? Trevor Bramwell's e-mail

Who consumes the Details field info, such as the test case requirement names/nomenclature from CNTT RA CH3 ??

- Test fails - Are the Requirements that were tested MUST or SHOULD? Some SHOULDs will be allowed exceptions, is this one of them? SHOULDs are intended to be conditional MUSTs.
- Conclusion: Dovetail web portal and the badging process are the consumers.

Other Topics/Options:
• Current Test API **Case name:** a string that we augment with the CNTT RX Ch3 Requirement?
• Issue: a **Pod name:** must exist in the POD Table of Test API for a test to report a test result.
• Issue: **Scenarios** need to be populated, and exist in an Scenario Table.
• Question: can a version 2.0 of the test API be defined? (with the enhancements as simple Key - Value pairs (not Details), and AVOID breaking all current inputs.
• Mark displayed the Web interface to the Test Results. Some Projects are not updating tests.

Summary: further details on approach discussed, but there is no success without an active project or team to take-up the needed development and help OPNFV projects to evolve to the enhanced/2.0 version!

Test_API Enhancement (in Details section)

Id :
Build tag :
Case name :
Criteria :
Installer :
Pod name :
Project name :
Scenario :
Start date :
Stop date :
Version :
Details :
{
  standards_ref: (cntt requirement, etsi tst, nomenclature)
  performance: true/false
  threshold: (for pass/fail criteria)
  metric: (us, bps, ....)
  links to OPNFV Artifact: (configuration file, env-details, csv files, )
  ....
}