Octopus Project Proposal

Project Name:

- Proposed name for the project: `Octopus`
- Proposed name for the repository: `CI`
- Project Category: `Integration & Testing`
- Project Status: `Formal Review`

Project description:

Problem Statement:

OPNFV will use many upstream open source projects to create the reference platform. All these projects are developed and tested independently and in many cases, have use cases of other projects in mind. Therefore it is to be expected that integration of these projects probably will unveil some gaps in functionality, since testing the OPNFV use cases needs the interworking of many upstream projects. Thus this integration work will bring major benefit to the community.

Therefore the goal of the CI project – Octopus – is to provide integration of a set of upstream projects. That is it will provide development environment with automated test and verification as a continuous integration environment, supporting both, the parallel evolutionary work in the upstream projects, and the improvement of NFV support in this reference platform.

Summary

The CI project provides the mechanism to build and verify OPNFV and upstream software as reference platforms based on the platform definition projects. The project will integrate stable versions of upstream projects and any necessary patches, and from there build and verify a reference platform for OPNFV.

The following list shows all necessary tasks for creation of the final environment and gives some idea how the work will be approached. More details need to be worked out during the project lifetime; the appendix provides some more details that are already available.

- The first step of initial integration of a first set of upstream projects will be done by a separate project `getstarted`
  Here released build artefacts, e.g. binaries, will be used for the intergration work to get fast results and a first base for sandbox and other work
- Create a **hierarchical build environment** for the same integrated upstream projects as "getstarted", that uses the build tools as defined by each of the upstream projects and combines them.
  This allows development and verification in OPNFV collaborative projects.
- Implement **automatic build process** on central servers
  Provide automation and periodic builds
- Execute the **continuous automated builds and basic verification**

Scope:

Octopus will create the complete basic development and build environment for OPNFV and start the periodic execution of the automatic build and basic Jenkins verification.

It will pull definition, configuration and packaging files from OPNFV or upstream projects' repositories and produce the OPNFV artifacts for deploying on bare metal datacenters as well as virtual environements.

Testability:

Octopus will invoke the basic Jenkins test and verification environment on every relevant change to the project source. It will also be able to invoke more involved testing suites form the integration and verification project

Documentation:

Documentation regarding the developer processes will be captured and maintained on wiki.opnfv.org.

Documentation of CI processes and functionality will be maintained in the CI project repository for CI developers and maintainers

Dependencies:

Octopus leverages the Bootstrap/Get-started project.

Octopus will pull from both upstream communities and OPNFV projects. These dependencies will be defined in detail by the projects providing the relevant information regarding the way those projects and sources will be built.

Octopus will also utilise test and verification artifacts from the test and verification projects to invoke the automated build and release test cases.

The first step of integration has no dependency on other OPNFV projects. However it depends on decisions that will be made outside the project, like for example the selection of Linux distributions, which may need TSC approval.

Test and verification in later project stages is dependent on the test objects provided by VNF deployment test cases project.

Octopus Continuous integration in later project stages is dependent on the Simultaneous Release project.

Committers and Contributors:

Committers:

- Daniel Radez (dradez@redhat.com)
- Daniel Ståhl (daniel.stahl@ericsson.com)
- Jiang Zhifeng(jiang.zhifeng@zte.com.cn)
- Palani Chinnakannan (pals@cisco.com)
- Qiqi Hu (huqiqi@huawei.com)
- Stefan Berg K (stefan.k.berg@ericsson.com)
- Wes Hayutin (whayutin@redhat.com)
- Yu Zhang (zhangyu11@huawei.com)
- M R, Chengappa HP (cm-r@hpe.com)
- Kumari, Sudha (sudha.kumari@hp.com)
- Ulrich Kleber (Ulrich.Kleber@huawei.com)

Contributors:

- Alan Pevec (apevec@redhat.com)
- Carlos Goncalves (carlos.goncalves@neclab.eu)
- Chengappa M R (cm-r@hpe.com)
- Chris Price (christopher.price@ericsson.com)
- Chris Wright (chrisw@redhat.com)
- Daniel Radez (dradez@redhat.com)
- Dave Lenrow (david.lenrow@hp.com)
- Dave Neary (dneary@redhat.com)
- Dirk Kutscher (dirk.kutscher@neclab.eu)
- Dongren Zhao (zhaodongren@huawei.com)
- Frank Brockners (fbrockne@cisco.com)
- Ian Wells (iawells@cisco.com)
- Jari Korpela (jari.korpela@nsn.com)
- Jia He (hejia1@huawei.com)
- Jiang Zhifeng(jiang.zhifeng@zte.com.cn)
- Jonas Bjurel (jonas.bjurel@ericsson.com)
- Jun Li (matthew.lijun@huawei.com)
- Jun Zhang (zhang.jun3g@zte.com.cn)
- Marika Rytkonen (marika.rytkonen@nsn.com)
- Markus Berglund (markus.berglund@nsn.com)
- Ashlee Young (ashlee@onosfw.com)
- Olivier Jacques (olivier.jacques@hp.com)
- Palani Chinnakannan (pals@cisco.com)
- Paul Quinn (paulq@cisco.com)
- Prakash Ramchandran (prakash.ramchandran@huawei.com)
- Qiqi Hu (huqiqi@huawei.com)
- Reinaldo Penno (rapenno@gmail.com)
- Sajeev Manikkoth (sajeevmanikkoth@gmail.com)
- Stefan Berg (stefan.k.berg@ericsson.com)
- Sudha Kumari (sudha.kumari@hp.com)
- Ulrich Kleber (ulrich.kleber@huawei.com)
- Vasile Radoaca (radoaca.vasile@huawei.com)
- Wes Hayutin (whayutin@redhat.com)
- Xiong Zhang (zhangxiong7@huawei.com)
- Yu Zhang (zhangyu11@huawei.com)
- Zhifeng Jiang (jiang.zhifeng@zte.com.cn)

Planned deliverables:

Octopus will develop and maintain the automated integration build and test environment. It will also execute the periodic builds of the defined OPNFV reference platforms.

Proposed Release Schedule:

- 1st Alfa release 6 months from start date