

ONAP-Automated OPNFV (Auto)

Project Name:

- Proposed name for the project: ONAP-Automated OPNFV (Auto)
- Proposed name for the repository: auto

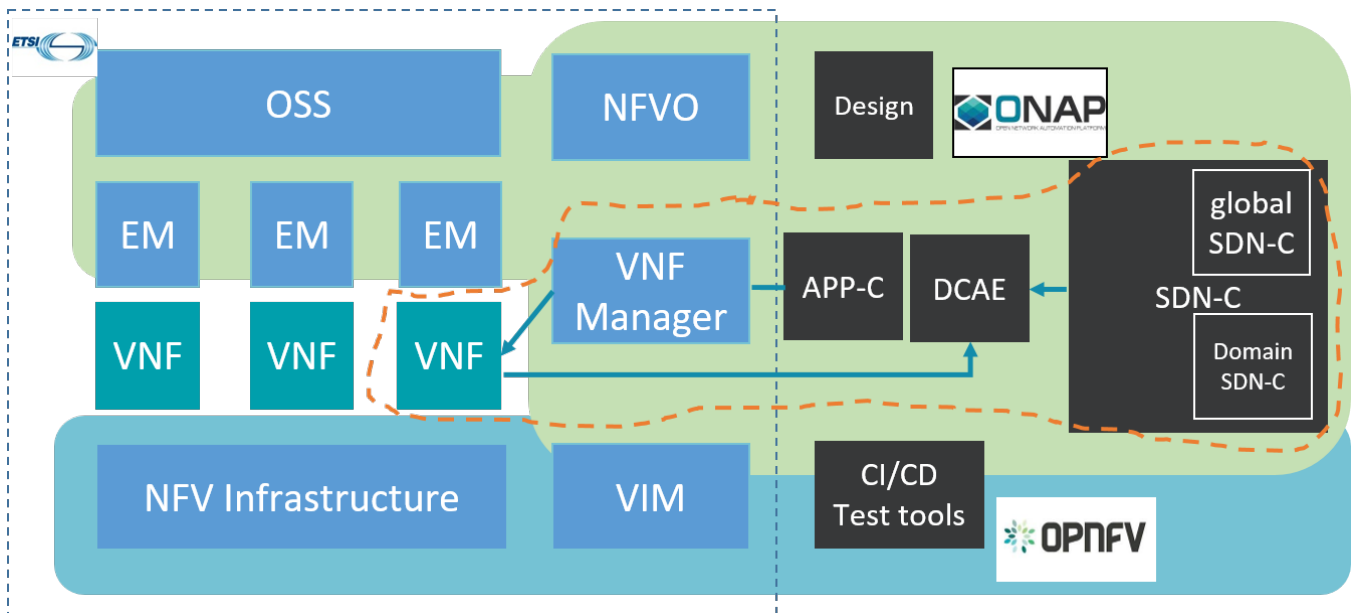
Project description:

- OPNFV is a SDNFV system integration project for open source components, which so far have been mostly limited to the NFVI+VIM as generally described by ETSI.
- In particular, OPNFV has yet to integrate higher-level automation features.
- This project will focus on ONAP component integration and verification with OPNFV reference platforms/scenarios, through primarily a post-install process in order to avoid impact to OPNFV installer projects. As much as possible, this will use a generic installation/integration process (not specific to any OPNFV installer's technology).
- While all of ONAP is in scope, as it proceeds, the project will focus on specific aspects of this integration and verification in each release. Some example topics and work items in scope **for the first release** include:
 - How ONAP meets VNFM standards, interact with different vendors' VNF
 - How ONAP SDN-C uses OPNFV existing features, e.g. NetReady, in a two layer controller architecture in which the upper layer (global controller) is replaceable, and the lower layer can use different vendor's local controller to interact with SDN-C
 - What data collection interface VNF and controllers provide to ONAP DCAE, and through DCAE, to closed-loop control functions such as Policy
 - Tests which verify interop of ONAP automation/lifecycle features with specific NFVI and VIM features, as prioritized by the project with technical community and EUAG input. Examples include
 - Abstraction of networking tech/features e.g. through NetReady/Gluon
 - Blueprint-based VNF deployment (HOT, TOSCA, YANG)
 - Application level configuration and lifecycle thru YANG (for any aspects depending upon OPNFV NFVI+VIM components)
 - Policy (through DCAE)
 - Telemetry (through VES/DCAE)

First Release Scope:

NOTE: This diagram is only intended to be illustrative of the initial areas of focus for Auto. It's understood that:

- ONAP scope extends beyond the lines drawn below
- ONAP architecture does not necessarily align with the ETSI NFV inspired diagrams this is based upon (and typically used in OPNFV etc)



- Orange dotted lines = auto project scope for initial release
- Scope may be expanded for future releases.

Testability:

- Tests will be developed for use cases within the project scope.
- Tests will be added to Functest runs for supporting scenarios.

Documentation:

- Documents will be created to describe the project scope, priorities, release deliverables, and user guide to the feature deployment and use case tests.

Relationship to other OPNFV projects:

This project is closely related to, and will work closely with following projects:

- [Pharos Lab as a Service](#), uses Pharos in a cloud environment to provide individual developers with an easy way of getting started ... i.e. to "try-out" an OPNFV deployment and begin to develop and test features with minimal overhead or knowledge about configuring and deploying an OPNFV instance.
- [Opera](#), focused on Open-O platform integration as a whole, potentially evolving to ONAP integration
 - Auto will focus on using the integrated ONAP platform for specific development/test objectives
 - The potential for inclusion of Auto under the Opera project has been discussed in the MANO WG and Tech Community calls: key questions that remain open in the decision process
 - For any dependency of Auto on Opera: **what Opera project resources (developers) will be available in F to complete the ONAP Nov'17 release integration?**
 - In Auto we had not planned to deploy ONAP as a complete platform, but rather if possible to deploy the components that are needed to test the interop use cases in Auto's scope. The Auto project supporters have not intended to work on complete deployment of ONAP e.g. using some OPNFV installer-specific approach or as an OPNFV scenario component.
 - If we find that approach (partial ONAP install) is infeasible/unable to meet our goals, we will depend upon a complete ONAP platform install instead, either through Opera or as described for the [LaaS proposal integrating ONAP and OPNFV into a multi-lab testbed environment](#). In this case, who from the Opera team would be available to work with us on one of those approaches?
 - **China Mobile and Huawei have committed to carrying forward the Opera project at least.**
 - The overall goal should be to keep project overhead low, progress agile, and the approach flexible. Given ability to meet those goals, it shouldn't matter in which project the work is done. Here are some questions that will help us in the decision per those goals:
 - **Do we think it will be effective to organize a combined project as multiple, potentially loosely connected tracks?**
 - We need regular project meetings in order to make rapid progress. Would the ONAP integration team prefer to have their own meetings? Although we know there was a lot of work done on Open-O integration, we don't see a lot of record of those meetings on the wiki etc. That's OK, if the Opera team can deliver ONAP integration without a lot of involvement from the Auto project team (mostly US based). We want people to work the way they need to, to be successful. But given that we would still need calls for the progress on Auto scope, would there be any concerns that we have at least two tracks in the combined project?
 - Here is a view of three (at least) distinct focuses which would be represented in a combined Opera/Auto project. These will likely require different developers with different skill sets. Who will be available from the Opera project to support the first two?
 - ONAP integration per current Opera: OPNFV installer support (if needed), or post-deploy install process (preferred)
 - VNF use case integration into Functest etc, ala vIMS, vCPE, etc
 - detailed functional interop test cases per the Auto proposal, using the Opera-installed ONAP, or discrete ONAP component install tools, or the cross-lab LaaS as above
 - There will be multiple (in detail) ways to (1) deploy ONAP; (2) manage VNFs using it. **While a single project can help us maximize common ground and synergy between different approaches, would there be a problem if in a single project we want/need to approach those aspects using a diversity of technologies/processes?** Some examples:
 - HOT vs TOSCA, e.g. to deploy VNF use cases
 - Abstract vs VIM-specific APIs (a near-term VIM-diversity accommodation IMO)
 - OPNFV installer-dependent, or independent
 - OpenStack vs Cloud-Native ONAP deployment
- [Yardstick](#), responsible for a test framework, including test cases and test stimuli, to verify the infrastructure compliance when running VNF applications.
- [Netready](#), focusing on evolving OpenStack networking step-by-step to find the most efficient way to fulfill the requirements of the identified NFV use cases, taking into account the NFV mindset and the capabilities of SDN controllers.
- [VES](#), focusing on integrating modeled telemetry as developed by the ONAP DCAE project, into the OPNFV platform and sample VNFs, for managing VNF/NFV/VIM health and lifecycle.
- [Models](#), focusing on developing model-driven NFV samples/tests for overall model-based automation and modeled specific features

Dependencies:

- [Integration \(Nov 2017\)](#) release of ONAP, responsible for ONAP cross-project system integration, CI/CD, and all related end-to-end release use cases testing with VNFs necessary for the successful delivery and industry adaption of the ONAP project as a whole.
- [Open Lab](#) of ONAP, responsible for defining and maintaining requirements, monitor availability and coordinate the usage of the community labs. The subcommittee will coordinate community lab resources to support release use case testing, community demo for marketing events if needed, and the interoperation testing with 3rd part components, or others.
- The goal is to be able to test in an existing OPNFV POD environment, but any additional requirements will be considered as part of the project. This may include cross-lab environments such as described in [Lab as a Service](#).

Committers and Contributors:

- Names and affiliations of the committers
 - [Tina Tsou](#), Arm
 - [Harry Huang](#), Huawei
 - [Song Zhu](#), Arm

- [Prasad Gorja](#), NXP
- Liang Ou, China Telecom, (oul.gd@chinatelecom.cn)
- [lei chen](#), CTTL of CAICT
- [xiaoyu wang](#), CTTL of CAICT
- [Xu Lu](#),CTTL of CAICT
- [Mohankumar Navaneethan](#), Cavium
- Names and affiliations of any other contributors
 - [Bryan Sullivan](#), AT&T
 - [Fatih Degirmenci](#), Ericsson
 - [Aimee Ukasick](#), AT&T
 - [Timo Perala](#), Nokia
 - [Bob Monkman](#), Arm

Planned deliverables:

- Installer support for ONAP components, in CI/CD or post-install scripts.
- Use case tests within the project scope.

Proposed Release Schedule:

- When is the first release planned?
 - F-release (Q1/Q2 2018)
- Will this align with the current release cadence
 - Yes

Key Project Facts

Lifecycle State:

Primary Contact:

Project Lead: [Tina Tsou](#)

Jira Project Name: auto

Jira Project Prefix: [10 Characters max [A-Z]]

mailing list tag [Should match Jira Project Prefix]

Committers:

- [Tina Tsou](#), Arm
- [Harry Huang](#), Huawei
- [Song Zhu](#), Arm
- [Prasad Gorja](#), NXP
- Liang Ou, China Telecom, (oul.gd@chinatelecom.cn)
- [lei chen](#), CTTL of CAICT
- [xiaoyu wang](#), CTTL of CAICT
- [Xu Lu](#),CTTL of CAICT
- [Mohankumar Navaneethan](#), Cavium

Link to TSC approval: <http://meetbot.opnfv.org/meetings/opnfv-meeting/2017/opnfv-meeting.2017-08-15-12.59.html>

Link to approval of additional submitters: Example <http://meetbot.opnfv.org/meetings/opnfv-meeting/2015/opnfv-meeting.2015-03-03-15.01.html>

Presentation at Technical Community Discussion meeting on 07/20/2017



ONAP-Automated OPNFV.pptx

We had a discussion today on the takeaways from the call yesterday, to try to come to a broader consensus on the plan for how to achieve the Auto project's proposed scope and the Opera project's scope for F (ONAP integration). Here are the key questions I recommended we discuss over email before the TSC meeting:

1. As noted by Uli there may be a dependency of Auto on Opera. What Opera project resources (developers) will be available in F to complete the ONAP integration (the "ONAP integration team" as I refer to them below)?

* In Auto we had not planned to necessarily deploy ONAP as a complete platform, but rather if possible to deploy the components that are needed to test the interop use cases in Auto's scope.

* If we find that approach (partial ONAP install) is infeasible/unable to meet our goals, we will depend upon a complete ONAP platform install instead, either through Opera or as described for the LaaS proposal integrating ONAP and OPNFV into a multi-lab testbed environment <<https://wiki.opnfv.org/display/INF/Lab+as+a+Service>>. In this case, who from the Opera team would be available to work with us on one of those approaches?

2. My overall recommendation is to keep the project overhead low, progress agile, and the approach flexible. Given ability to meet those goals, I don't care where the work is done. Here are some questions that will help us in the decision per those goals:

* Can a combined project be organized as multiple, potentially loosely connected tracks?

i. We need regular project meetings in order to make rapid progress. Would the ONAP integration team prefer to have their own meetings? Although we know there was a lot of work done on Open-O integration, we don't see a lot of record of those meetings on the wiki etc. That's OK, if the team (e.g. led by Harry) can deliver ONAP integration without a lot of involvement from the Auto project team (mostly US based). We want people to work the way they need to, to be successful. But given that we would still need calls for the progress on Auto scope, would you have any concerns that we have at least two tracks in the combined project?

* Here is how I view the three (at least) distinct focuses in the overall project. These will likely require different developers with different skill sets. Who will be available to support the first two?

i. ONAP integration per current Opera: OPNFV installer support (if needed), or post-deploy install process (preferred)

ii. VNF use case integration into Functest etc, ala vIMS, vCPE, etc

iii. detailed functional interop test cases per the Auto proposal, using the Opera-installed ONAP, or discrete ONAP component install tools, or the cross-lab LaaS as above

* There will be multiple (in detail) ways to (1) deploy ONAP; (2) manage VNFs using it. While a single project can help us maximize common ground and synergy between different approaches, would you agree that a single project in OPNFV could approach those aspects using different technologies/processes? Some examples:

i. HOT vs TOSCA (my preference), e.g. to deploy VNF use cases

ii. Abstract vs VIM-specific APIs (a near-term VIM-diversity accommodation IMO)

iii. OPNFV installer-dependent, or independent (my preference)

iv. OpenStack vs Cloud-Native (my preference)
ONAP deployment