



# OPNFV

## PDF/IDF 2.0 Discussion

## TripleO (Installer)

Weekly Technical Discussion

February 10<sup>th</sup>, 2020, 14:00 UTC

Daniel Balsiger ([Daniel.Balsiger@swisscom.com](mailto:Daniel.Balsiger@swisscom.com))

# TripleO Summary

- Installing, upgrading and operating OpenStack clouds
- Using OpenStack's own cloud facilities as the foundation
- Building on Nova, Ironic, Neutron and Heat (in undercloud)
- Automate cloud management at datacenter scale.
- TripleO is supporting **only RHEL7/CentOS7 x86\_64** operating systems!
- Supports many features important for OPNFV (requires additional configuration)
  - OVS-DPDK / SR-IOV / ...
  - Multi Network Support / Bonding ...

Documentation:

- <https://docs.openstack.org/tripleo-docs/latest/>
- <https://docs.openstack.org/project-deploy-guide/tripleo-docs/latest/>



# Requirements Analysis

See: <https://wiki.opnfv.org/display/CIRV/Hardware+Delivery+Verification>

- 1) Heat templates (Yaml) (structured, not easy to edit/generate for OPNFV setups)
- 2) Node (H/W) data stored in Ironic DB (Introspection). Nodes can be enrolled automatically.
- 3) TripleO scales pretty well up to 100s of nodes. (not sure about 1000+ nodes), Ironic Node introspection detects most of the node H/W details (e.g. Nics/Disks).
- 4) Node description can be minimal, even automatic enrollment is possible.
- 5) TripleO supports different roles of nodes ctrl/compute/block-storage/object-storage. The CNTT specific roles could be implemented directly this way.
- 6) Downside of TripleO: only Centos/RHEL. Strictly bound to Openstack Ironic for H/W introspection (IPA). Heat templates are bound to OS heat. its only suited for OS.
- 7) This is already implemented in TripleO. Structured/Inherited Heat templates.
- 8) I think for RA2 the K8s installation is the main focus. TripleO does only OpenStack not K8s.
- 9) TripleO supports dynamic adding/removing of nodes and minimal lifecycle tasks.

# OPNFV Apex Installer (Deprecated)

- OPNFV installer based on TripleO
- Closes the gaps between OPNFV Pharos and TripleO upstream:
  - OPNFV specific OpenStack release
  - SDN Ctrl (ODL) support for TripleO
  - Multi Network Support (TripleO default is 2, Pharos defines min. 4)
  - SRIOV/OVS-DPDK Support
  - Simplification of configuration (node-inventory/network-config/scenario files)
  - OPNFV Pharos compliance
  -
- Unfortunately the Apex project is not active anymore



# Swisscom Lab Experiences

- 1) H/W Physical Installation/Cabling/ToR/Jump setup
  - 2) Node Inventory & Instrospection
  - 3) Network connectivity verification
  - 4) Operating System Installation
  - 5) Software Installation (RA1: OpenStack, RA2: K8s)
  - 6) Verification (e.g. OPNFV Functest)
- 2)-6) should be automated as much as possible.
  - 3) is the hardest part for OPNFV (Multi-Network / DPDK config / Bonding / SR-IOV / ...)
  - Should work for all Operating Systems: Ubuntu / CentOS / OpenSuse ...
  - Should work for particular OpenStack and K8s releases



# Facts regarding PDF/IDF/Installers

- PDF/IDF/SDF files were not consumed by every installer in the past
  - Why evolve on a format which was rarely consumed/used?
  - Many installer candidates already base on OpenStack Ironic
    - TripleO: also possible to get rid of nova/neutron → metalsmith
    - Bifrost: standalone Ironic, used by OpenStack Infra Group.
    - Airship2.0: Metal3.io is Ironic run in a cloud-native way.
- Introspection has a clear advantage compared to human editing of files.
  - With introspection data and the network/infra descriptor automated testing can be done (before or after Operating system deployment)
- Networking setup (ToR, DPDK, Multi-Network, SDN Controller)
  - Is usually different in each Lab and the primary source for deployment errors.
  - Network descriptor files (IDF) and Node properties must match for OPNFV scenarios (e.g. DPDK, NIC config, Numa topology, HugePage, Isolated CPUs,...)





# Questions?

Please direct any questions or  
comments to  
[opnfv-tech-discuss@lists.opnfv.org](mailto:opnfv-tech-discuss@lists.opnfv.org)  
for discussion