



OPNFV SUMMIT

JUNE 2017 | BEIJING, CHINA

OPNFV Scenarios consolidation

OPNFV Scenarios

- OPNFV provides the NFV platform in different variants
 - Choice of networking solution (Neutron/OVS vs. various SDN controllers)
 - Choice of vSwitch (OVS, Fd.io/VPP, etc.)
 - Specific stack configuration, e.g. optimized KVM settings
 - Specific NFVi features (L3VPN, SFC, etc.)
- Each variant is called a “Scenario”

OPNFV Scenarios – Challenges

- Number of scenarios grows every release → release effort grows
 - Arno: 2, Brahmaputra: 18, Colorado: 28, Danube: 58, Euphates: 61
- Hard to identify what's supported and to what extent. Reference platform?
- Need a process to drive consolidation and manage scenario lifecycle, without limiting innovation and speed
- Blueprint:
<http://artifacts.opnfv.org/octopus/docs/scenario-lifecycle/index.html>

(Re)define OPNFV scenarios

OPNFV scenarios serve two purposes that require different handling:

- **Generic Scenarios:** represent a particular stack composition choice
 - There to stay, gating required, promotion process
- **Specific Scenarios:** isolate a particular path of development
 - Temporary, little to no gating, derive from & deliver into generic scenarios

Generic Scenarios

- Specific stack composition choice
 - Platform (OS, K8) + SDN solution
- Stable, mature and long-lived
- Support for mature features & their combinations
- Target is to be supported by all installers
- All generic scenarios in an OPNFV release should have same common major versions of the included upstream components

Specific Scenarios

- Provide isolation for a specific feature/topic – similar to topic branch in Git
- Typically derived from a Generic Scenario (child or sibling)
- Freely choose version of upstream components or apply midstream patches
- Limited life time and will merge into generic scenarios
- Easy creation by request to CI & release manager, one installer sufficient

Configuration Options

- Many scenarios can be deployed in different ways without changing scope:
 - HA / NOHA configuration
 - Virtual or bare metal deployment (see POD descriptor file)
 - Hardware types (Intel/ARM)
- The deployed platform will provide the same capabilities
- Applies to both generic and specific scenarios
- CI Pipeline needs to test all configuration options

Next Steps:

- Concept as such widely discussed & approved by TSC
 - ⊠ time to implement!
- Suggested approach: choose one obvious generic scenario candidate and test
 - The consolidation process
 - Implications on the release process (single scenario owner across installers, how to ensure feature maintenance in generic scenarios, etc.)
- Candidates → OS+no_sdn, configuration variables: ovs, ovs-dpdk, fdio, ha, bar
- Volunteers from the Installer projects?
- Pick one installer as PoC and do the integration work
- Define a process (CI, Release)
- Report back to the TSC

A large white rectangular box with the text "Thank You" in a bold, black, sans-serif font, centered within the box.

Thank You