

# 2020-10-14 - OPNFV/CNTT - HA requirements and testing approaches

## Topic Leader(s)

- Georg Kunz
- Mark Beierl
- Pankaj Goyal

## Topic Overview

Discussions of current CNTT Release 1 HA requirements are approach to testing

## Slides & Recording

[https://zoom.us/rec/share/7Z-OKpnOQxP9Ts4qX-5JorXNa9QMMtwKcqRDe7\\_wKXuue1Ha\\_pMa7kM7KdmvHKiA.73iS3Ybl-FxTTKaz?startTime=1602689230000](https://zoom.us/rec/share/7Z-OKpnOQxP9Ts4qX-5JorXNa9QMMtwKcqRDe7_wKXuue1Ha_pMa7kM7KdmvHKiA.73iS3Ybl-FxTTKaz?startTime=1602689230000)

## Agenda

### CNTT Requirements

<https://wiki.opnfv.org/display/SWREL/Jerma+Requirements+Working+Group+Assessment>

- req.gen.rsl.01:  
The Architecture must support resilient OpenStack components that are required for the continued availability of running workloads.
- req.inf.ntw.07  
The Architecture must support network resiliency.

### Existing HA test cases in OPNFV - Yardstick

Example test cases

- [Control node restart](#): restart entire node
- [Neutron service restart](#): kill Neutron process and measure API response and recovery. Same concept for Nova, Glance, Cinder, Keystone, MySQL, RabbitMQ, HAProxy
- [CPU load](#)
- [Disk IO load](#)

Properties

- Framework for building resilience test scenarios
- Framework geared towards OpenStack: translation of Yardstick scenarios to Heat
- Majority of the tests white box testing which is not suitable

### High-level questions

- What kind of test cases can we actually design for?
- No white box testing - only black box testing
- how to define pass / fail criteria
- Node level
- Network resilience
  - Switch level, port level?
  - Availability of redundant fabric in OPNFV labs, Packet
  - API for configuring switches

### Existing resilience and robustness testing

Instead of building a new framework, integration of existing resilience testing frameworks.

Non-exhaustive list of tools - **extend with more suitable candidates you are aware of**

- Litmus (<https://github.com/litmuschaos/litmus>)
- PowerfulSeal (<https://github.com/powerfulseal/powerfulseal>)
- OpenShift Kraken (<https://github.com/openshift-scale/kraken>)
- [Chaos Toolkit](#)

- [Pumba](#)
- [Litmus](#)
- [Chaos Mesh](#)

## Minutes

- Cedric
  - RC-1/2 should be used in production environments and hence not execute destructive testing
  - the Yardstick framework is hard to maintain questionable if we want to re-active it
- key question: is resilience testing in the scope of RC-1/2
  - CNTT specifies requirements on resilience there is a need for validating such requirements via an automated test
  - we likely need such tests and then need to de-/select destructive tests depending on use case: workload onboarding (non-destructive) vs. OVP badging (destructive)
- Need to distinguish between HA and resiliency. A resilient system continues to function in case of a failure (we can limit to a single failure scenario)
- In a cloud environment one expects infrastructure failures and thus expect resiliency and HA from the software systems (OSTK, etc.) – # of deployments, etc.
- Recovery also needs to be taken into account. If the recovery impacts the workloads to the point where they are no longer functional, then it cannot be considered resilient
- RA1 Chapters [3](#) and [4](#) specify the services, # of minimum deployments, etc. to meet the requirements specified in Chapter 2; also review [Ch5](#) (Thanks, Cedric)
- Opened CNTT Issue [#2061](#) to make the network resiliency requirement more specific

## Action Items

