Overview

This document is for collecting thoughts and tasks for the in-progress development of an initial OpenStack Neutron simulation for ovn-heater.

The document is shared with anyone who has the link, feel free to request comment or edit access if you would like to contribute.

Base test

To get us started let's see if we can mimic OpenStack Neutron with a very simple test

- Basic OpenStack test case
  - Assumptions
    - 1 network node, no DVR
    - 1 external network
    - 1 tenant network
    - 1 router attached to external network and tenant network
  - Input parameters
    - Number of hypervisor nodes
    - Number of VMs per hypervisor node
- I've added a skeleton for a OpenStack CMS plugin and test case that makes use of the in-flight architecture changes below here:
  - https://github.com/fnordahl/ovn-heater/commit/9e254e9471d52bb05d1cdf5706272305fbc514b0

Discovery of Neutron OVN API usage details

- One idea that multiple people have mentioned to get a quick overview
  - deploy an OpenStack and have Neutron populate the database
  - compare/record the state of the database between each action

OVN NB database changes during test scenario

I performed the above described test scenario while taking OVN NB database snapshots after each step. The snapshots are stripped of empty tables (and neutron liveness check records) for brevity. First line in each file shows the openstack command that resulted in the database state below. Convenient way for checking changes after each step is to use "git diff -u --color-words step_n step_n+1"

- initial step
- create tenant network
- create tenant subnet
- create external network
- create external network subnet

- create router
- add external network to the router
- add internal network to the router
- create guest VM
- create floating IP
- attach floating IP to the guest VM

OVN Heater Architecture changes

- The `ovn-tester` package contains most of the interesting logic including a `ovn-tester` module with the main entrypoint.
- At present, there are `ovn-kubernetes` specific items in its `create_nodes` and `run_base_cluster_bringup` functions and possibly other places
    - We need to figure out a good way to make this part modular, perhaps let the test case select its own base bringup function?
        - Initial proposal: https://github.com/ovn-org/ovn-heater/pull/173
- Similarly the `ovn_workload` module contains a combination of shared components such as `ClusterConfig`, `Node`, `CentralNode` and ovn-kubernetes specific components such as `WorkerNode`, `Cluster`, `Namespace`
    - ~~Note that the `Namespace` class has the potential to contain CMS specific code, so it should probably be split up, we will find out once we try to use it in an OpenStack test.~~ Actually the `Namespace` class appears to represent a Kubernetes namespace.
    - WIP/RFC proposal for extracting generic methods from `WorkerNode` and `Cluster` classes: https://github.com/ovn-org/ovn-heater/pull/175