Kea Dhcp4 + Kea Control Agent (Kea API) Test Setup

> Note: This setup works under the assumption that the Docker `bridge` network driver creates a new broadcast domain for its containers, isolated from any of the Docker host's physical interfaces' broadcast domains. Otherwise, the Kea Dhcp4 server could cause ip-assignment issues in the physical network!

How To Use

- Get NAV development containers up and running:

  ```
  git clone "https://github.com/Uninett/nav.git"
  cd nav
  docker compose up -d
  ```

  Note: If your local NAV repository root directory is not called `nav` (that is, you do not use `cd nav` to get to the root like above) then change all occurrences of the string 'nav_default' in *this* (nav-kea-testing) repository's 'docker-compose.yaml' file to ’<dir>_default', where <dir> is the name of the NAV repository root directory. Otherwise, you can ignore this note.

- Get the containers for this test setup up(sic) and running:

  ```
  # Sets up 5 dhcp clients and a kea-dhcp4 server with kea-ctrl-agent on its own bridge n
  git clone "https://github.com/jorund1/nav-kea-testing.git"
  cd nav-kea-testing
  docker compose up -d
  ```

- Now NAV can reach the Kea API at `http://kea:8000/`. To test NAV's usage of the API, add the following entry to the `dhcpmetrics.conf` NAV configuration file[1]:

  ```
  [http://kea:8000/]
  dhcp_version=4
  service=kea-management-api
  ```

- Run `docker compose exec dhcpmetrics` to check if the Kea API server is found.

- Create a VLAN in 'seeddb' with subnet containing the subnet '172.31.255.0/24' and afterwards go to the VLAN's web page to see DHCP stats for that VLAN.

[1] for example by editing it in the source tree under `python/nav/etc/dhcpmetrics.conf`and then running `docker compose exec nav config install --overwrite /etc/nav`