CNaaS-NMS Workshop - 2: Concepts

Here are some tasks that will hopefully make you understand the concepts in CNaaS-NMS a bit better.

In the minimal setup workshop we used the API to add a device to the database. Adding new devices to the database is normally handled automatically via the ZTP process (DHCP trigger), but in some scenarios you might need to add devices manually such as the first device in a new network.

The ZTP process also has specific requirements for how devices should be connected so that the network is not provisioned in a way that will be hard to operate. For example access switches needs to be connected to two separate distribution switches for redundancy, so that you can upgrade distribution switches without impacting too many user services at the same time.

It is however possible to sidestep most of these requirements if you add devices manually and configure custom interfaces.

Creating a linknet between distribution switches

In the next step will add a second distribution switch to our database, and then create a linknet between the two distribution switches. Normally you would connect the distribution switches to core switches so this is a slight sidestep of the normal design but will make sense in a small network.

See if you can use the API reference documentation to **1) add a second distribution switch** and **2) add a link between the two devices using interfaces called Ethernet4 on both sides**: https://cnaas-nms.readthedocs.io/en/latest/apiref/index.html

Once you have created the second device and a linknet, try generating the configuration again and see if configuration for the new link is added:

```
curl -ks -H "Authorization: Bearer $JWT_AUTH_TOKEN" https://localhost/api/v1.0/device/eosdist1/generate_config -H
"Content-Type: application/json" | jq .data.config.generated_config | sed 's/\\n/\n/g'
```

You will probably find that Ethernet4 configuration is not generated. This is because you also have to define how the interfaces in the distribution switch can be used. Linknet configuration will only be generated if the selected port has been defined as "fabric" interface class (ifclass). You can define interface classes either per device or per switch model. For example maybe the first interfaces of all your distribution switches will be used to connect access switches, and the last interfaces will be used for uplinks ("fabric" ifclass).

We can configure this by going in to the cnaas/git/cnaas-settings directory and then editing the file devices/eosdist1/interfaces.yml . Add a new Ethernet4 interface in the yaml code:

```
- name: Ethernet4
  ifclass: fabric
```

You will then need to git commit, git push, and then ask the API to fetch new settings:

```
curl -ks -H "Authorization: Bearer $JWT_AUTH_TOKEN" https://localhost/api/v1.0/repository/settings -d '{"action":
"refresh"}' -X PUT -H "Content-Type: application/json"
```

And if you then generate the config for eosdist1 you should get a new Ethernet4 interface configuration.

Creating a management domain

Core and distribution switches are managed via loopback interfaces routed in the overlay network. Access switches are however managed via a vlan interface on a subnet where all switches connected to the same pair of distribution switches are in one subnet. This is referred to as a management domain (mgmtdomain). See if you can use the API reference documentation to create a management domain for all access switches connected to the two distribution switches you added to the database earlier.

If you want to have a global management domain for all access switches in the entire network, you can also set up the management domain to have two core switches instead of distribution switches.