CNaaS template development

Here you can find some tips for easier development of the Jinja2/CLI templates required for CNaaS NMS.

If you haven't looked at the product documentation/repository reference for templates yet start there: https://cnaas-nms.readthedocs.io/en/latest/reporef/index.html#templates

Available variables

Too see what variables are available (and their values) for the template rendering for a specific device you can use an API call like this:

```
curl -s -H "Authorization: Bearer $JWT_AUTH_TOKEN" "${CNAASURL}/api/v1.0/device/eosdist1/generate_config" | jq .
data.config.available_variables
```

Helper scripts

Normally you would need to go through a workflow like this to test changes to a template: make change git commit git push API refresh API syncto dry_run. This can be a lot of steps if you need to test small incremental changes, therefore you can use the script template_dry_run.py to help make this process faster.

Example usage (you will need python3 with modules: requests, yaml, jinja2):

```
wget -O ~/template_dry_run.py https://raw.githubusercontent.com/SUNET/cnaas-nms/develop/src/cnaas_nms/tools
/template_dry_run.py
wget -O ~/jinja_helpers.py https://raw.githubusercontent.com/SUNET/cnaas-nms/develop/src/cnaas_nms/tools
/jinja_helpers.py
export CNAASURL=https://localhost
export JWT_AUTH_TOKEN=mytoken
export TEMPLATE_SECRET_ADMIN_HASH=dummyvalue
cd git/cnaas-templates/
~/template_dry_run.py <hostname>
```

This will render the configuration from templates locally on your machine, and then send the finished configuration for a dry_run to the API where the diff will be generated.

If you're using variables like TEMPLATE_SECRET_ADMIN_HASH etc in your templates to you will need to export them in your local environment, but only export dummy values instead of the real hash since this might be logged.

You can use the WebUI/CLI etc to view the actual diff in the job that was scheduled by template_dry_run.py

Jinja2 specifics

You can use include blocks (or extend blocks) in Jinja to divide your templates up into multiple parts for easier maintenance. Remember to always end your templates with a newline since you might run in to issues where two parts might end up on the same line when you do includes otherwise.

Device / OS specific templates

You can use the variables "device_model" or "device_os_version" in an if-statement if there are some configurations that only should be applied to a specific device model etc.