CNaaS Auth POC server installation

The CNaaS-NMS API uses JSON Web Tokens (JWT) for authenticating and authorizing users. If you have an existing JWT server you might be able to use that.

A minimalistic JWT auth server was developed as a proof-of-concept for supporting the CNaaS NMS project. This document describes how to set it up.

Setup using docker compose

Set up a new VM and install docker and docker-compose. Create two new persistent docker volumes:

```
docker volume create cnaas-authserver-jwtcert
docker volume create cnaas-authserver-userdb
```

Create a new `docker-compose.yaml` file:

```
---
version: '3.7'
services:
  cnaas_auth:
    image: docker.sunet.se/auth-server-poc:latest
    ports:
      - 443:1443
    volumes:
      - type: volume
        source: cnaas-authserver-jwtcert
        target: /opt/auth-server-poc/cert/
      - type: volume
        source: cnaas-authserver-userdb
        target: /opt/auth-server-poc/userdb/
volumes:
  cnaas-authserver-jwtcert:
    external: true
  cnaas-authserver-userdb:
    external: true
```

Run `docker-compose up -d` or similar to start the container.

Generating keys and certificates

Enter the docker container using `docker exec -it cnaas_auth bash` (find the correct name of the container by running `docker ps`).

Inside the docker, run the following to create a new JWT private and public key pair. The key pair will be used to sign JWT tokens:

```
cd /opt/auth-server-poc/cert/
openssl ecparam -genkey -name prime256v1 -noout -out private.pem
openssl ec -in private.pem -pubout -out public.pem
chgrp www-data private.pem
chmod g+r private.pem
```

Restart the docker container or run `killall uwsgi` inside the container to enable the newly generated certificate.

Creating user accounts

Now it's time to create some accounts for the users that will access the CNaaS NMS API. The user accounts are saved in an Apache style `.htpasswd` file.

When creating the first user the `.htpasswd` file itself has to be created. This is done by passing the `-c` parameter to the `htpasswd` command.

Run this inside the container to create two new users (and remember to replace the example usernames with your wanted account names):

```
htpasswd -c /opt/auth-server-poc/userdb/.htpasswd indy
htpasswd /opt/auth-server-poc/userdb/.htpasswd bob
```

Trying it out

To sum up: You restarted the container with the newly generated JWT cert and created two users.

Now you should be able to ask the authentication API for a new JWT token. Run this from the VM/outside the container:

```
curl -ks https://localhost/api/v1.0/auth -X POST -u indy -p
```

This will prompt for a password and, if entered correctly, should return a JSON reply with a JWT token.

Connecting the Auth POC server to CNaaS-NMS

To make sure that the CNaaS NMS will accept this JWT token, one last step is needed: The public key of the auth container has to be "installed" on the CNaaS NMS API container.

To achieve this, simply copy the public key file `/opt/auth-server-poc/cert/public.pem` (on the auth container) to `/opt/cnaas/jwtcert/public.pem` (on the API container).