Adding HTTPS support

For now, traefik only accepts insecure http traffic. To allow secure access over https, we need to get a valid ssl certificate. For this we will use traefiks automatic certificate generation. Traefik supports multiple acme challenges, but we will focus on using the DNS challenge.

Getting a letsencrypt certificate

First, we need to make some changes to our static configuration file `config/traefik.yml`.

1. Define an additional entrypoint `websecure`:

```
entryPoints:
  websecure:
    address: :443
    http:
      tls:
        certResolver: netcup
```

2. Add a certificate resolver. We will use netcup for this example. The settings will need to set according to your provider (See here).

```
certificatesResolvers:
  netcup:
    acme:
      email: <email@example.com>
      storage: /etc/traefik/acme.json
      caServer: https://acme-staging-v02.api.letsencrypt.org/directory
      dnsChallenge:
        provider: netcup
        delayBeforeCheck: 900
        resolvers:
          - "root-dns.netcup.net:53"
          - "second-dns.netcup.net:53"
```

   We will start to use the letsencrypt staging server to check if everything is working. If everythin works at the end, the `caServer` entry can simply be deleted.
3. Remove the `insecure: true` line from the api tag.

4. Add the dynamic config file provider

```
providers:
  file:
    filename: /etc/traefik/dynamic_config.yml
    watch: true
```

Next, we need to add the dynamic configuration by creating the file `config/dynamic_config.yml`.

```
tls:
  stores:
    default:
      defaultGeneratedCert:
        resolver: netcup
        domain:
          # main: "example.com"
          sans: "*.example.com"
```

Here we our ssl certificate, that we will soon have, as our default one. This way we don't need to manually set it for every single service, that we want to access over https. We will simply acquire a wildcard certificate. If you also want to get a certificate for the base domain, you can uncomment the `main: "example.com"` line.

Now we only need to make some changes to our `docker-compose` file as follows:

1. Add `- "443:443"` to the ports of the container.
2. Bind the dynamic config and `acme.json` to the container

```
services:
  traefik:
    volumes:
      - ./config/dynamic_config.yml:/etc/traefik/dynamic_config.yml
      - ./config/acme.json:/etc/traefik/acme.json
```

3. Add the netcup customer information to the environment tag

```
services:
  traefik:
    environment:
      - NETCUP_CUSTOMER_NUMBER=${NETCUP_CUSTOMER_NUMBER}
      - NETCUP_API_KEY=${NETCUP_API_KEY}
      - NETCUP_API_PASSWORD=${NETCUP_API_PASSWORD}
```

4. Allow access to the dashboard over the websecure entdpoint

```
services:
  traefik:
    labels:
      traefik.enable: true
      traefik.http.routers.traefik.service: api@internal
      traefik.http.routers.traefik.entrypoints: websecure
      traefik.http.routers.traefik.rule: Host(`traefik.${SITE}`)
```

When we start up the container now, traefik will try to get the ssl certificate from the staging server. After waiting some time, when you go to `https://traefik.example.com` you will hopefully see the letsencrypt certificate from the staging server. If yes, then everything works and you can switch to the non-staging caServer (see above). If not, make sure that everything is entered correctly.

After a restart of the container, traefik should have gotten a valid letsencrypt certificate and you should be able to access the dashboard correctly.

See also

- https://doc.traefik.io/traefik/https/overview/

Revision #4
Created 14 September 2023 15:58:02 by Levin
Updated 14 September 2023 17:51:13 by Levin