

Add custom error pages

We can change the default error pages by adding the errorpages middleware to specific services or the whole entrypoint. Independent of if we want error pages for specific services or for a whole entrypoint we first need to setup a few things.

Setup

If we want to service custom error pages, we will need to host a webserver like nginx, since traefik itself can only route traffic and not serve additional pages.

We will add an nginx container to our traefik `docker-compose.yml`, since we want to use both in combination with each other. Simply add the following under `services`:

```
services:
  error_pages:
    image: nginx:latest
    container_name: error_pages
    volumes:
      - ./config/error_handling/error_pages:/usr/share/nginx/error_pages
      - ./config/error_handling/default.conf:/etc/nginx/conf.d/default.conf
      - ./logs:/var/log/nginx
    labels:
      traefik.enable: true
      traefik.http.routers.error-router.rule: HostRegexp(`{host:.+}`)
      traefik.http.routers.error-router.priority: 1
      traefik.http.routers.error-router.entrypoints: websecure
      traefik.http.routers.error-router.middlewares: error-pages-middleware
      traefik.http.middlewares.error-pages-middleware.errors.status: 400-599
      traefik.http.middlewares.error-pages-middleware.errors.service: error-pages-service
      traefik.http.middlewares.error-pages-middleware.errors.query: /{status}.html
      traefik.http.services.error-pages-service.loadbalancer.server.port: 80
```

We then need to create the folder `config/error_handling` as well as `config/error_handling/error_pages`. In addition we need to create the file `config/error_handling/default.conf`.

We will use the nginx image to create a webserver and attach `error_pages` and `default_conf` to it.

With the help of labels we create a new traefik router `error-router` with a low priority, that uses regex to catch all requests not otherwise routed. Next we create a new middleware `error-pages-middleware`, that takes all requests that couldn't be routed (status 400 to 599) and query the `error-pages-service` (nginx itself at port 80) at `/{{status}}.html`. Imagine we get a request for `this-does-not-exist.example.com`. Instead of traefik directly responding with a 404 status, we instead catch the request and query nginx for `/404.html`

Now add the following to `config/error_handling/default.conf`:

```
server {  
    listen      80;  
    server_name localhost;  
  
    error_page 404 /404.html;  
    error_page 403 /403.html;  
    # error_page <status> /<status>.html  
  
    location /{  
        internal;  
        root /usr/share/nginx/error_pages;  
    }  
}
```

Whenever nginx gets a request for `/404.html` for example, it will serve the file `config/error_handling/error_pages/404.html`. Simply add your custom error pages to that folder and they will be served when needed.

Using error pages for specific services

To use custom pages for a specific service, add the following to the labels of your service inside your `docker-compose.yml`:

```
services:  
    traefik:  
        labels:  
            traefik.http.routers.traefik.middlewares: "error-pages-middleware@docker"
```

Using error pages for every service

To avoid having to manually assign the new error pages middleware to every service, we can instead use it for every service. Add the following to the websecure entrypoint inside your `config/traefik.yml`

:

```
entryPoints:
  websecure:
    http:
      middlewares:
        - error-pages-middleware@docker
```

Combine ipwhitelist with custom error pages

To combine both middlewares, we can use the `chain` middleware. Add the following to the http tag inside your `config/dynamic_config.yml`:

```
http:
  middlewares:
    localonly:
      ipWhitelist:
        sourceRange:
          - "192.168.0.0/24"
          - "127.0.0.1/32"
    secured:
      chain:
        middlewares:
          - error-pages-middleware@docker
          - localonly
```

This will combine the middlewares in **reverse order**. Every request goes thorough the localonly middleware first and if the ip is not whitelisted, nginx will be queried for the appropriate error page.

NOTE: This should theoretically return the 403.html file. Yet somehow, although nginx gets queried for `/403.html`, nginx returns the `/404.html` page. I can't figure out why....

Now, if a service should only be accessible from inside the home network, we can instead add the following label:

```
traefik.http.routers.traefik.middlewares: "secured@file"
```

When we want to allow public access to a service, we simply omit this line and traefik will automatically only use the custom error pages middleware.

See also

- <https://doc.traefik.io/traefik/middlewares/http/errorpages/>
- <https://www.imandrea.me/blog/traefik-custom-404/>

Revision #2

Created 14 September 2023 16:36:21 by Levin

Updated 14 September 2023 17:51:13 by Levin