Installer, configurer et gérer le sousdomaine "db-cms-admin"

- **Notes** : ce domaine hébergera l'outil *Adminer* permettant d'administrer la base de données *Mariadb* de *Wordpress*. Il est embarqué dans un container *Docker* et fait partie de la stack *cms.silene.eu*.

Installer le domaine

- Créer un fichier de configuration : `vi /etc/nginx/sites-available/db-cms-admin.conf`
  - Y placer le contenu suivant :

```
server {
    listen 80;
    listen [::]:80;

    server_name db-cms-admin.<domaine-sinp>;

    location / {
        proxy_set_header Host $http_host;
        proxy_set_header X-Real-IP $realip_remote_addr;
        proxy_set_header X-Forwarded-Host $host:$server_port;
        proxy_set_header X-Forwarded-Server $host;
        proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;

        proxy_pass http://127.0.0.1:50081/;# ATTENTION : bien
mettre un slash final ! Sinon => erreur 404
    }
}
```

- Créer un lien depuis les sites actifs : `cd /etc/nginx/sites-enabled/ ; ln -s ../sites-available/db-cms-admin.conf db-cms-admin.conf`
  - Tester la config et relancer Nginx si OK : `nginx-reload` ou `nginx -t && nginx -s reload`
  - Tester l'URL http://db-cms-admin.<domaine-sinp>/ qui doit afficher une erreur 502 car nous n'avons pas encore lancé le container Docker.
- En local, sur votre machine, se placer dans le dépôt Github "*sinp-paca-srv*" récupéré précédemment et si nécessaire resynchroniser le dossier *web-srv/docker/* avec le serveur de destination en exécutant la commande *Rsync* indiquée dans le fichier README.md.
- Sur le serveur dans le dossier *docker* de l'utilisateur *admin* :
  - vérifier la présence du réseau Docker spécifique à notre utilisation de type *bridge* nommé *nginx-proxy* (voir fichier *.env*) : `docker network ls`
  - se placer dans le dossier *cms.silene.eu* : `cd ~/docker/cms.silene.eu`
  - exécuter la commande : `docker-compose up`

    - vérifier que tout fonctionne à l'adresse : http://db-cms-admin.<domaine-sinp> (se loguer avec le compte root Mysql)
    - arrêter le container : CTRL+C
    - relancer le container en tant que service : `docker-compose up -d`
        - si besoin de l'arrêter utiliser : `docker compose down`

Activer le SSL et HTTP2

- Installer un certificat SSL via *Certbot* (*Letsencrypt*) : `certbot --nginx -d db-cms-admin.<domaine-sinp>`
    - Répondre : 2
    - Tester ensuite la redirection auto de HTTP vers HTTPS : http://db-cms-admin.<domaine-sinp>/ → doit redirigé vers HTTPS automatiquement
- Tester la configuration SSL : https://www.ssllabs.com/ssltest/analyze.html?d=db-cms-admin.<domaine-sinp>
- Tester l'URL https://db-cms-admin.<domaine-sinp>/
- La config finale :

```
server {
    listen 443 ssl http2; # managed by Certbot
    listen [::]:443 ssl http2; # managed by Certbot

    server_name db-cms-admin.<domaine-sinp>;

    location / {
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-Host $host:$server_port;
        proxy_set_header X-Forwarded-Server $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;

        proxy_pass http://127.0.0.1:50081/;# ATTENTION : bien mettre un slash final ! Sinon => erreur 404
    }

    ssl_certificate /etc/letsencrypt/live/cms.<domaine-sinp>/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/cms.<domaine-sinp>/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
}

server {
    listen 80;
    listen [::]:80;
```

```
    server_name db-cms-admin.<domaine-sinp>;

    return 302 https://$host$request_uri;
}
```

From:
https://sinp-wiki.cbn-alpin.fr/ - **CBNA SINP**

Permanent link:
https://sinp-wiki.cbn-alpin.fr/serveurs/installation/web-srv/docker-adminer

Last update: **2020/11/17 15:02**