

**BTS Services informatiques aux organisations - SISR****Session 2022****E4 – Support et mise à disposition de services informatiques****Coefficient 4****DESCRIPTION DE LA REALISATION PROFESSIONNELLE****NOM et prénom du candidat : LEDUC Quentin****N° candidat : 02145639104****Contexte de la réalisation professionnelle**

La plupart des membres de Scani sont des personnes qui ne sont que très peu habilitées à la technique. Par conséquent, Scani veut améliorer le confort des actifs sur les logiciels essentiels au bon déroulement de la société. Les DNS de Scani sont actuellement sur BIND9, ce logiciel pourtant fonctionnel, peut nous faire peur via son interface en commande. Nous devons trouver une solution Web avec peu de commande pour faire fonctionner des DNS.

**Intitulé de la réalisation professionnelle**

Installation de PowerDNS et de l'interface web Admin

**Période de réalisation : DU 28/05/21 AU 03/06/21****Lieu : Joigny****Modalité : Individuelle****Principale(s) activité(s) concernée(s) :**

- Déployer un service
- Analyser les objectifs et les modalités d'organisation d'un projet

**Conditions de réalisation**

- **Ressources présentes**                      **Dns créer et gérer avec Bind9**
- **Résultats attendus**                      **Migrer vers PowerDNS avec une interface WEB**
- **Durée de réalisation**                      **4H**

**Modalités d'accès à cette réalisation professionnelle.**Site internet : [www.netwaze.fr](http://www.netwaze.fr)

Aller dans « Réalisations Professionnelles » Mot de passe : Mr.Robot

**Partie 1 – Procédure de mise en œuvre.****PRÉREQUIS MATÉRIEL**

- Un serveur Ubuntu
- MariaDB ou Mysql
- Nginx
- NodeJS
- Yarn

# PRÉREQUIS

## INSTALLATION DE MARIADB + CRÉATION BDD

Pour commencer nous devons mettre à jour notre distribution Ubuntu :  
`apt update && apt upgrade -y`

```
kiu — root@powerdns: /home/kiu — ssh kiu@192.168.131.2 — 80x24
root@powerdns:/home/kiu# apt update && apt upgrade -y
Hit:1 http://fr.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://fr.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://fr.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://fr.archive.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@powerdns:/home/kiu#
```

```
kiu — root@powerdns: /home/kiu — ssh kiu@192.168.131.2 — 80x56
root@powerdns:/home/kiu# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

[Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

[Set root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

[Remove anonymous users? [Y/n] y
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

[Disallow root login remotely? [Y/n] y
... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

[Remove test database and access to it? [Y/n] y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

[Reload privilege tables now? [Y/n] y
... Success!

Cleaning up...
```

On installe MariaDB : `$ apt install mariadb-server`  
Une fois installer nous allons passer a la post-  
installation qui va définir la sécurité de MariaDB.  
`$ mysql_secure_installation`

Puis nous pouvons créer notre utilisateur et notre  
base de données.  
`$ mysql -u root -p`  
`$ create database pwndns;`  
`$ grant all on pwndns.* to pdnsadmin@localhost  
identified by 'password';`  
on met à jour les privilèges et on quitte MariaDB  
`$ flush privileges;`  
`$ quit`

# INSTALLATION DE POWERDNS

Pour n'avoir aucun problème nous devons désactiver Systemd puis nous le réactiveront un peu plus tard.

```
$ systemctl disable --now systemd-resolved
```

```
kiu — root@powerdns: /home/kiu — ssh kiu@192.168.131.2 — 80x20 On
root@powerdns:/home/kiu# systemctl disable --now systemd-resolved
Removed /etc/systemd/system/dbus-org.freedesktop.resolve1.service.
Removed /etc/systemd/system/multi-user.target.wants/systemd-resolved.service.
root@powerdns:/home/kiu# █
```

supprime et on recréer le fichier resolv.conf

```
$ rm -rf /etc/resolv.conf
```

```
$ echo "nameserver 1.1.1.1" > /etc/resolv.conf
```

```
kiu — root@powerdns: /home/kiu — ssh kiu@192.168.131.2 — 80x20 On
root@powerdns:/home/kiu# rm -rf /etc/resolv.conf
root@powerdns:/home/kiu# echo "nameserver 1.1.1.1" > /etc/resolv.conf
root@powerdns:/home/kiu# cat /etc/resolv.conf
nameserver 1.1.1.1
root@powerdns:/home/kiu# █
```

installe PowerDNS : 

```
$ apt install pdns-server pdns-backend-mysql -y
```

puis nous importons le schéma de la base de données dans MariaDB

```
$ mysql -u pdnsadmin -p pdns < /usr/share/pdns-backend-mysql/schema/schema.mysql.sql
```

```
root@powerdns:/home/kiu# mysql -u pdnsadmin -p pdns < /usr/share/pdns-backend-
mysql/schema/schema.mysql.sql
[Enter password: ]
root@powerdns:/home/kiu# █
```

On créer un fichier de configuration de connexion a la base de données

```
$ nano /etc/powerdns/pdns.d/pdns.local.gmysql.conf
```

et on y rentre :

```
>
```

```
# MySQL Configuration
```

```
#
```

```
# Launch gmysql backend
```

```
launch+=gmysql
```

```
# gmysql parameters
```

```
<gmysql-host=127.0.0.1
```

```
gmysql-port=3306
```

```
gmysql-dbname=pwdns
```

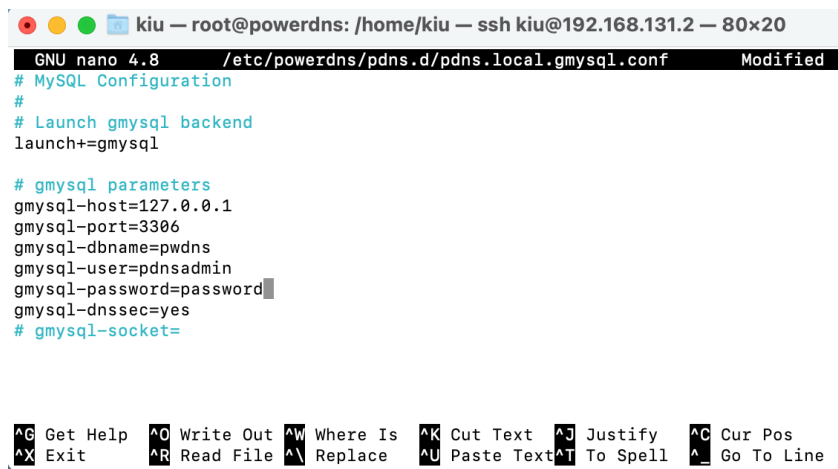
```
gmysql-user=pdnsadmin
```

```
gmysql-
```

```
password=password
```

```
gmysql-dnssec=yes
```

```
# gmysql-socket=
```



```
GNU nano 4.8 /etc/powerdns/pdns.d/pdns.local.gmysql.conf Modified
# MySQL Configuration
#
# Launch gmysql backend
launch+=gmysql

# gmysql parameters
gmysql-host=127.0.0.1
gmysql-port=3306
gmysql-dbname=pwdns
gmysql-user=pdnsadmin
gmysql-password=password
gmysql-dnssec=yes
# gmysql-socket=

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```

Maintenant nous devons donner les autorisations nécessaires au bon fonctionnement de PowerDNS.

```
$ chmod 640 /etc/powerdns/pdns.d/pdns.local.gmysql.conf
```

```
et on arrête PowerDNS : $ systemctl stop pdns
```

```
puis on lance la connexion a la base de données
```

```
$ pdns_server --daemon=no --guardian=no --loglevel=9
```

Si tout fonctionne nous devront avoir cette sortie.

```
-----
Jul 18 13:05:10 About to create 3 backend threads for UDP
Jul 18 13:05:10 gmysql Connection successful. Connected to database 'pwdns' on '
127.0.0.1'.
Jul 18 13:05:10 gmysql Connection successful. Connected to database 'pwdns' on '
127.0.0.1'.
Jul 18 13:05:10 gmysql Connection successful. Connected to database 'pwdns' on '
127.0.0.1'.
Jul 18 13:05:10 Done launching threads, ready to distribute questions
```

On redémarre pdns et nous regardons si PowerDNS écoute bien sur le port 53

```
$ systemctl start pdns
```

```
$ ss -alnp4 | grep pdns
```

```
root@powerdns:/etc/powerdns# systemctl start pdns
root@powerdns:/etc/powerdns# ss -alnp4 | grep pdns
udp UNCONN 0 0 0.0.0.0:53 0.0.0.0:*
  users:(( "pdns_server",pid=22130,fd=5))
tcp LISTEN 0 128 0.0.0.0:53 0.0.0.0:*
  users:(( "pdns_server",pid=22130,fd=7))
root@powerdns:/etc/powerdns#
```

# INSTALLATION DE L'INTERFACE WEB POWERDNS-ADMIN

Pour installer l'interface WEB nous avons besoin d'un serveur web. On va alors choisir NGINX. Nous allons aussi installer les dépendances de Python, de MariaDB et d'un tas d'autre ...

```
$ apt install nginx python3-dev libsasl2-dev libldap2-dev libssl-dev libxml2-dev libxslt1-dev libxmlsec1-dev libffi-dev pkg-config apt-transport-https virtualenv build-essential libmariadb-dev git python3-flask php-fpm php-y
```

On installe NodeJS et son référentiel

```
$ curl -sL https://deb.nodesource.com/setup_14.x | bash -
$ apt install nodejs -y
$ curl -sS https://dl.yarnpkg.com/debian/pubkey.gpg | apt-key add -
$ echo "deb https://dl.yarnpkg.com/debian/ stable main" | tee /etc/apt/sources.list.d/yarn.list
```

On remet à jour les dépôts et on installe Yarn

```
$ apt update -y
$ apt install yarn -y
```

On télécharge PowerDNS-Admin et on le déplace dans notre dossier NGINX

```
$ git clone https://github.com/ngoduykhanh/PowerDNS-Admin.git /var/www/html/pdns
```

```
[root@powerdns:/etc/powerdns# git clone https://github.com/ngoduykhanh/PowerDNS-Admin.git /var/www/html/pdns
Cloning into '/var/www/html/pdns'...
remote: Enumerating objects: 9425, done.
remote: Counting objects: 100% (34/34), done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 9425 (delta 11), reused 18 (delta 6), pack-reused 9391
Receiving objects: 100% (9425/9425), 33.29 MiB | 3.06 MiB/s, done.
Resolving deltas: 100% (4415/4415), done.
root@powerdns:/etc/powerdns# █
```

Maintenant nous devons créer un environnement python virtuel.

```
$ cd /var/www/html/pdns/
```

```
$ virtualenv -p python3 flask
```

Nous activons notre environnement virtuel

```
$ source ./flask/bin/activate
```

On installe toutes les autres dépendances de PowerDNS-Admin

```
$ pip install -r requirements.txt
```

```
[(flask) root@powerdns:/var/www/html/pdns# pip install -r requirements.txt
Collecting Flask==1.1.2
  Downloading Flask-1.1.2-py2.py3-none-any.whl (94 kB)
    |████████████████████████████████████████| 94 kB 673 kB/s
Collecting Flask-Assets==2.0
  Downloading Flask_Assets-2.0-py3-none-any.whl (8.5 kB)
Collecting Flask-Login==0.5.0
  Downloading Flask_Login-0.5.0-py2.py3-none-any.whl (16 kB)
Collecting Flask-SQLAlchemy==2.4.4
  Downloading Flask_SQLAlchemy-2.4.4-py2.py3-none-any.whl (17 kB)
Collecting Flask-Migrate==2.5.3
  Downloading Flask_Migrate-2.5.3-py2.py3-none-any.whl (13 kB)
Collecting SQLAlchemy==1.3.19
  Downloading SQLAlchemy-1.3.19-cp38-cp38-manylinux2010_x86_64.whl (1.3 MB)
    |████████████████████████████████████████| 1.3 MB 1.1 MB/s
..
```

On peut déconnecter notre environnement virtuel

```
$ deactivate
```

Puis on peut connecter notre PowerDNS-Admin a notre base de données

```
$ nano /var/www/html/pdns/powerdnsadmin/default_config.py
```

```
GNU nano 4.8 /var/www/html/pdns/powerdnsadmin/default_config.py Modified
import os
basedir = os.path.abspath(os.path.abspath(os.path.dirname(__file__)))

### BASIC APP CONFIG
SALT = '$2b$12$yLUMTIfl21FKJQpTkRQXCu'
SECRET_KEY = 'e951e5a1f4b94151b360f47edf596dd2'
BIND_ADDRESS = '0.0.0.0'
PORT = 9191
HSTS_ENABLED = False
OFFLINE_MODE = False
FILESYSTEM_SESSIONS_ENABLED = False

### DATABASE CONFIG
SQLA_DB_USER = 'pdnsadmin'
SQLA_DB_PASSWORD = 'password'
SQLA_DB_HOST = '127.0.0.1'
SQLA_DB_NAME = 'pdns'
SQLALCHEMY_TRACK_MODIFICATIONS = True
```

On peut réactiver notre environnement virtuel

```
$ cd /var/www/html/pdns/
```

```
$ source ./flask/bin/activate
```

puis on met a jour notre base de données

```
$ export FLASK_APP=powerdnsadmin/__init__.py
```

```
$ flask db upgrade
```

```
INFO [alembic.runtime.migration] Running upgrade 0fb6d23a4863 -> 856bb94b7040,
Add comment column in domain template record table
INFO [alembic.runtime.migration] Running upgrade 856bb94b7040 -> b0fea72a3f20,
Update domain serial columns type
INFO [alembic.runtime.migration] Running upgrade b0fea72a3f20 -> 3f76448bb6de,
Add user.confirmed column
INFO [alembic.runtime.migration] Running upgrade 3f76448bb6de -> 0d3d93f1c2e0,
Add domain_id to history table
(flask) root@powerdns:/var/www/html/pdns# █
```

```
$ yarn install --pure-lockfile
```

```
[(flask) root@powerdns:/var/www/html/pdns# yarn install --pure-lockfile
yarn install v1.22.5
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...
Done in 22.17s.
(flask) root@powerdns:/var/www/html/pdns# █
```

Et on rebuild le tout.

```
$ flask assets build
```

une fois fini nous pouvons désactiver notre environnement

```
$ deactivate
```

Maintenant nous devons activer l'API de PowerDNS pour lire les statistiques, modifier le contenu de la zone et les métadonnées.

```
$ nano /etc/powerdns/pdns.conf
```

Nous décommandons `api=no` et `api-key=` et nous les modifiant en :

```
api=yes
```

```
api-key=e951e5a1f4b94151b360f47edf596dd2
```

Puis nous pouvons redémarrer PowerDNS pour qu'il prennent en compte les changements.

```
$ systemctl restart pdns
```

# CONFIGURATION DE NGINX

Pour pouvoir avoir l'interface WEB nous devons configurer notre serveur Web NGINX.

On créer un nouveau host virtuel dans conf.d de NGINX

```
$ nano /etc/nginx/conf.d/pdns-admin.conf
```

Puis on y rentre >

```
server {
    listen    *:80;
    server_name    pdnsadmin.powerdns.lan;

    index    index.html index.htm index.php;
    root    /var/www/html/pdns;
    access_log    /var/log/nginx/pdnsadmin_access.log combined;
    error_log    /var/log/nginx/pdnsadmin_error.log;

    client_max_body_size    10m;
    client_body_buffer_size    128k;
    proxy_redirect    off;
    proxy_connect_timeout    90;
    proxy_send_timeout    90;
    proxy_read_timeout    90;
    proxy_buffers    32 4k;
    proxy_buffer_size    8k;
    proxy_set_header    Host $host;
    proxy_set_header    X-Real-IP $remote_addr;
    proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_headers_hash_bucket_size    64;

    location ~ ^/static/ {
        include /etc/nginx/mime.types;
        root /var/www/html/pdns/powerdnsadmin;

        location ~* \.(jpegljpeglpnglgif)$ {
            expires 365d;
        }

        location ~* ^.+.(cssljs)$ {
            expires 7d;
        }
    }

    location / {
        proxy_pass    http://unix:/run/pdnsadmin/socket;
        proxy_read_timeout    120;
        proxy_connect_timeout    120;
        proxy_redirect    off;
    }
}
```

On test si notre syntaxe est correcte avec la commande \$ nginx -t

```
root@powerdns:/var/www/html/pdns# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@powerdns:/var/www/html/pdns#
```



Maintenant nous changeons le propriétaire du dossier pdns en data qui est l'utilisateur de NGINX.

```
$ chown -R www-data:www-data /var/www/html/pdns
```

Puis on redémarre NGINX

```
$ systemctl restart nginx
```

Maintenant on crée un fichier systemd pour pouvoir lancer et arrêter les services si nous le souhaitons. \$ nano /etc/systemd/system/pdnsadmin.service

Puis on colle dedans >

[Unit]

Description=PowerDNS-Admin

Requires=pdnsadmin.socket

After=network.target

[Service]

PIDFile=/run/pdnsadmin/pid

User=pdns

Group=pdns

WorkingDirectory=/var/www/html/pdns

ExecStart=/var/www/html/pdns/flask/bin/gunicorn --pid /run/pdnsadmin/pid --bind unix:/run/pdnsadmin/socket 'powerdnsadmin:create\_app()'

ExecReload=/bin/kill -s HUP \$MAINPID

ExecStop=/bin/kill -s TERM \$MAINPID

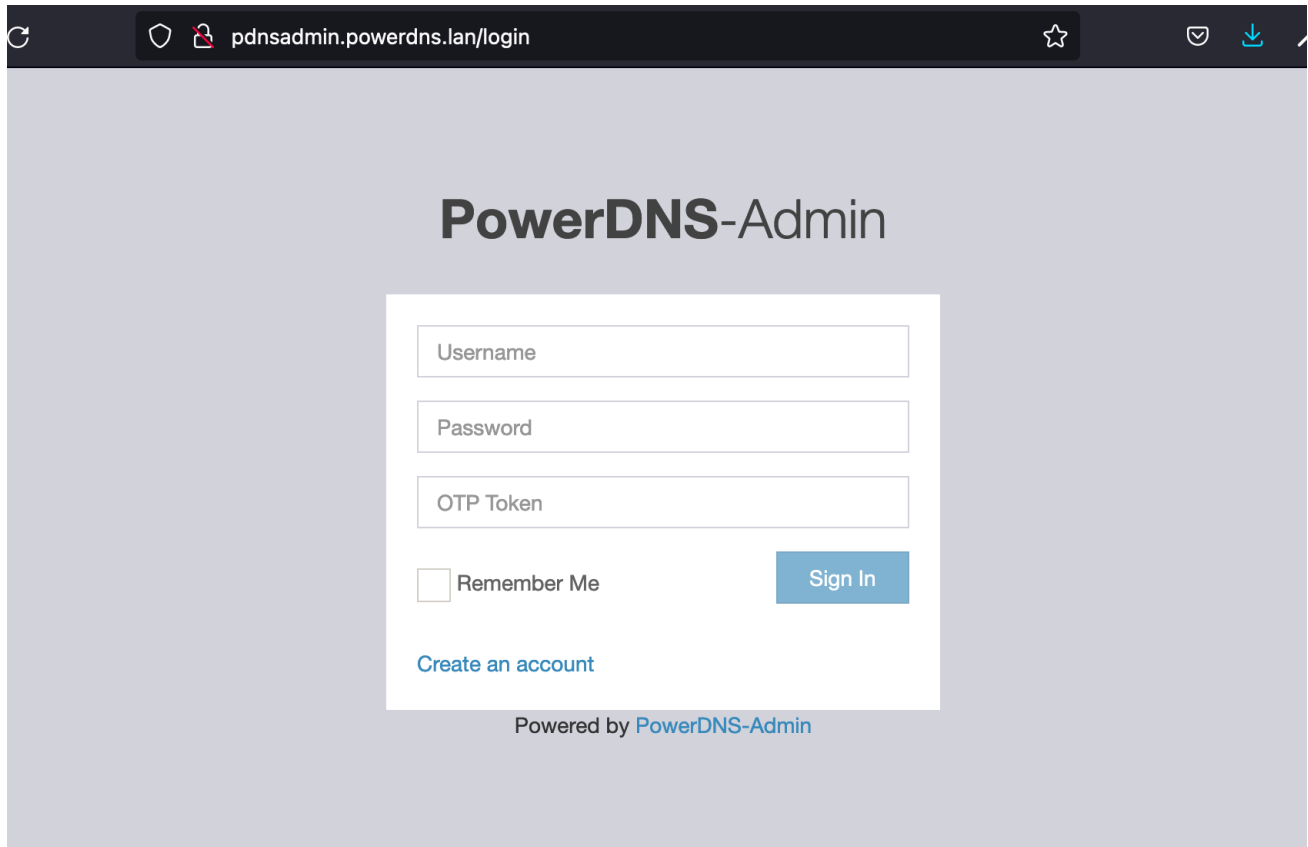
PrivateTmp=true

[Install]

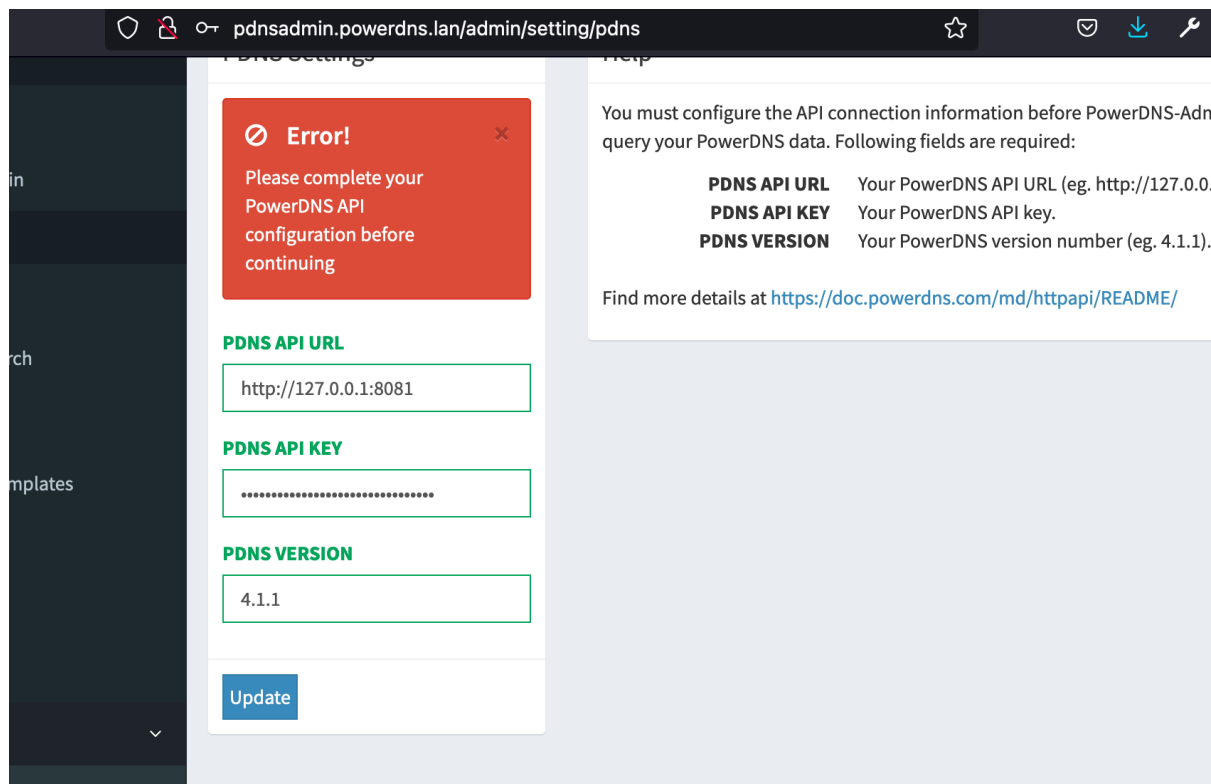
WantedBy=multi-user.target



On peut maintenant se connecter sur notre interface web avec notre nom NGINX : <http://pdnsadmin.powerdns.lan> et créer un compte.



Une fois notre compte créer nous devons mettre notre API dans l'interface :



## Partie 2 – Validation.

Nous pouvons maintenant créer un nouveau domaine {netwaze.xyz} que l'on redirige vers notre adresse ip (local) pour nos tests.



Manage domain: **netwaze.xyz**

15 records

Name	Type	Status	TTL	Data	Comment	Edit	Delete
@	SOA	Active	3600	pdns.Serveur1. kiu.netwaze.xyz. 2021071803 1 0800 3600 604800 3600		Edit	Delete
@	A	Active	60	192.168.1.1		Edit	Delete

Showing 1 to 2 of 2 entries

Avec notre ordinateur local on lance la commande dig pour connaître l'adresse ip qui se cache derrière notre nom de domaine netwaze.xyz

```
kiu@MacBook-Pro-de-kiu ~ % dig netwaze.xyz

; <<>> DiG 9.10.6 <<>> netwaze.xyz
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17120
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;netwaze.xyz.                IN      A

;; ANSWER SECTION:
netwaze.xyz.                60      IN      A      192.168.1.1

;; Query time: 3 msec
;; SERVER: 192.168.43.168#53(192.168.43.168)
;; WHEN: Sun Jul 18 16:25:05 CEST 2021
;; MSG SIZE rcvd: 56

kiu@MacBook-Pro-de-kiu ~ % dig SOA netwaze.xyz

; <<>> DiG 9.10.6 <<>> SOA netwaze.xyz
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 56449
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;netwaze.xyz.                IN      SOA

;; ANSWER SECTION:
netwaze.xyz.                3600    IN      SOA    pdns.Serveur1. kiu.netwaze.xyz.
2021071804 10800 3600 604800 3600

;; Query time: 1 msec
;; SERVER: 192.168.43.168#53(192.168.43.168)
;; WHEN: Sun Jul 18 16:25:32 CEST 2021
;; MSG SIZE rcvd: 93

kiu@MacBook-Pro-de-kiu ~ %
```

Nous constatons que dig nous retourne les informations que l'on a mis dans notre PowerDNS.

### **Partie 3 – Veille technologique.**

Nous aurons pu installer un serveur DNS sous Windows Serveur et avoir l'interface de Windows pour faire notre configuration. Pour PowerDNS, une autre interface Web (plus rustique) est disponible : PowerAdmin

Link

Windows Serveur : <https://www.microsoft.com/fr-fr/evalcenter/evaluate-windows-server>

PowerAdmin : <https://www.poweradmin.org/>