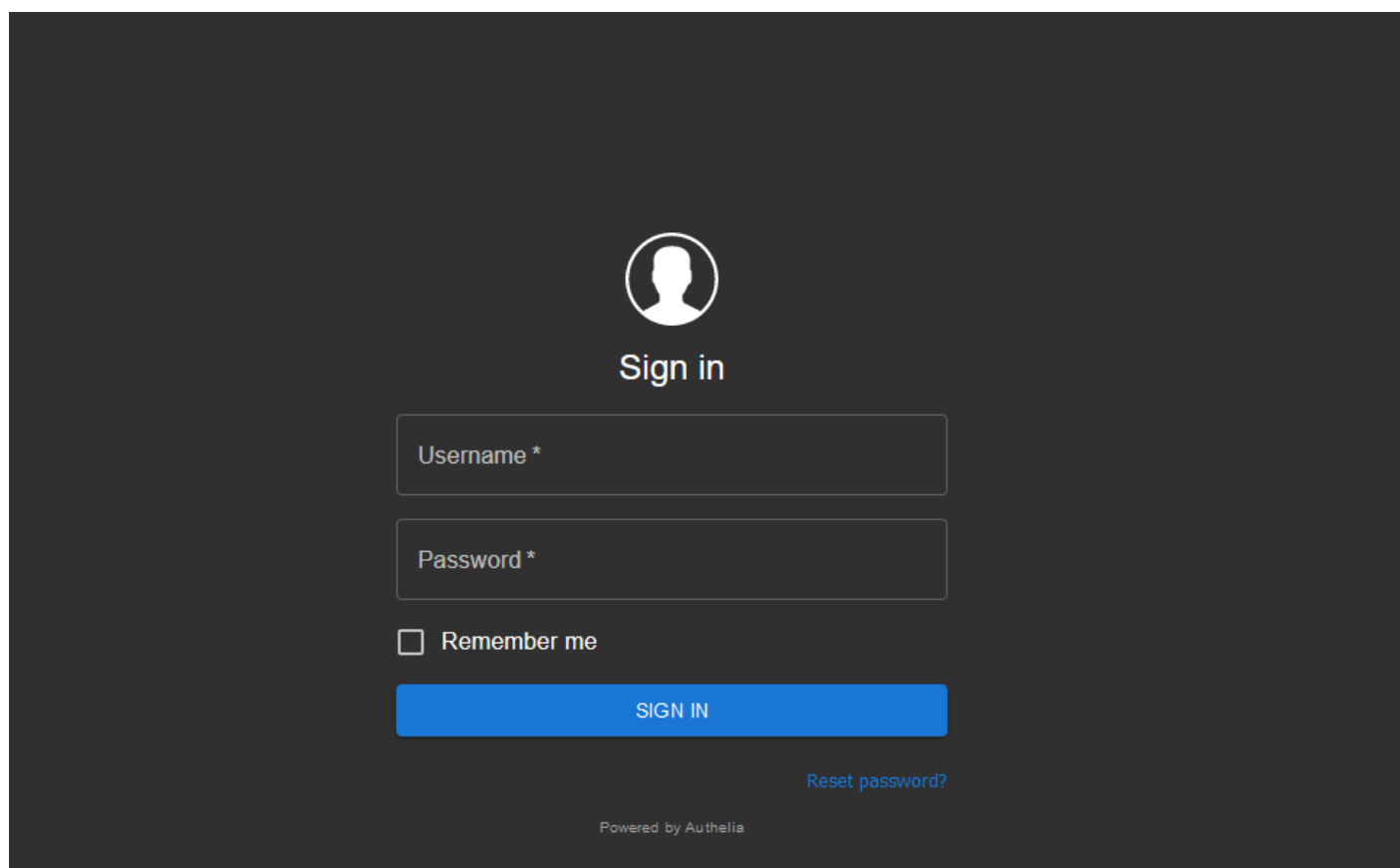


# Mise en place d'Authelia : OpenSource SSO

Authelia est une solution OpenSource qui agit comme un portail d'accès avec authentification, ou SSO. Il permet de centraliser l'authentification des utilisateurs et leur permet l'accès à des ressources protégées. L'authentification peut passer par une simple connexion user / password mais des fonctionnalités avancées sont disponibles : authentification à deux facteurs, utilisation d'une notification push Duo ou activation d'une clé de sécurité Yubikey.



## Environnement

Dans cette documentation, notre architecture est la suivante, adaptez les valeurs pour "**copcol**" comme un enfant :

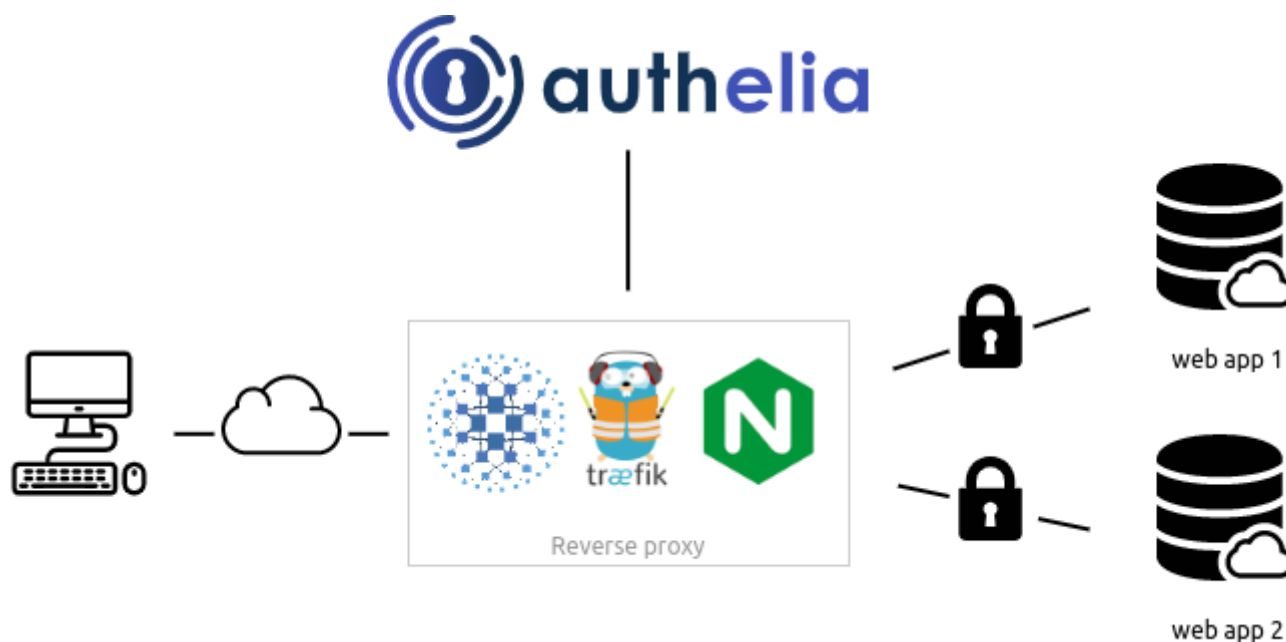
```
# Emplacement de la configuration des Dockers
export AUTHELIA_DOCKER_DIR="/data/Dockers/authelia"
export AUTHELIA_DOCKER_DIR_CONF="/data/Dockers/authelia/config"
export AUTHELIA_DOCKER_DIR_REDIS="/data/Dockers/authelia/redis"
```

```
# Emplacement de la stack Docker-compose
export AUTHELIA_DOCKERCOMPOSE_DIR="/opt/docker-compose"

# Domaines
export AUTHELIA_ROOT_DOMAIN="domain.com"
export AUTHELIA_AUTH_DOMAIN="auth.${AUTHELIA_ROOT_DOMAIN}"
```

## Fonctionnement

Voici un exemple de mise en place d'Authelia avec Traefic / Nginx :



## Mise en place d'authelia

### Préparation des dossiers de configuration

Nous allons créer les dossiers de configuration et y créer les fichiers de base **configuration.yml** et **users\_database.yml** :

```
mkdir -p ${AUTHELIA_DOCKER_DIR}/config
touch ${AUTHELIA_DOCKER_DIR}/config/configuration.yml
touch ${AUTHELIA_DOCKER_DIR}/config/users_database.yml
```

### configuration.yml - Configuration d'authelia

Le fichier de configuration **configuration.yml** (Pour générer les tokens [suivez le guide](#))

```
#####
#                               Authelia configuration                               #
#####

host: 0.0.0.0
port: 9091
log_level: info
jwt_secret: A4gYb7QFpbfKaNWAX7P7FX5y
default_redirection_url: https://auth.domain.com
totp:
  issuer: domain.com
  period: 30
  skew: 1

#duo_api:
#  hostname: api-123456789.example.com
#  integration_key: ABCDEF
#  secret_key: yet-another-long-string-of-characters-and-numbers-and-symbols

authentication_backend:
  disable_reset_password: false
  file:
    path: /config/users_database.yml
    password:
      algorithm: argon2id
      iterations: 1
      salt_length: 16
      parallelism: 8
      memory: 64

access_control:
  default_policy: deny
  rules:
    - domain:
        - "radarr.domain.com"
        - "sonarr.domain.com"
        - "radarr.domain.com"
  policy: bypass
  resources:
    - "^/api.*"
```

```
- domain:
  - "auth.domain.com"
  - "www.domain.com"
policy: bypass
- domain:
  - "radarr.domain.com"
  - "sonarr.domain.com"
  - "deluge.domain.com"
policy: one_factor
subject:
  - ["group:admins", "group:users"]
```

session:

```
name: authelia_session
secret: quaeS9MaixieLlaelee0vov3J
expiration: 3600 # 1 hour
inactivity: 7200 # 2 hours
domain: domain.com # Root domain
```

redis:

```
host: redis
port: 6379
```

regulation:

```
max_retries: 5
find_time: 2m
ban_time: 10m
```

theme: dark # options: dark, light, grey

storage:

```
local:
  path: /config/db.sqlite3
```

notifier: # Permet la validation d'un compte si 2FA

# filesystem:

# filename: /config/notification.txt

smtp:

```
username: contact@domain.com
password: Belzah2iek7pheNgeileosaev
```

```
host: mail.domain.com
port: 587 # 25 non-ssl, 443 ssl, 587 tls
sender: contact@domain.com
subject: "[Authelia] {title}"
disable_require_tls: false # set to true if your domain uses no tls or ssl only
disable_html_emails: false # set to true if you don't want html in your emails
tls:
  server_name: mail.domain.com
  skip_verify: false
  minimum_version: TLS1.2
```

## users\_database.yml - Base utilisateurs Authelia

Nous allons générer un fichier pour stocker les utilisateurs et groupes pour Authelia :

```
#####
#                               Users Database                               #
#####

# This file can be used if you do not have an LDAP set up.

# List of users
users:
  johndoe:
    displayname: "John Doe"
    password: "$argon2id$v=19$m=1048576,t=1,p=8$MFJSeXh0V2VKVWZEFJiZg$E0Sz20gjIIV//MWf8"
    email: johndoe@domain.com
    groups:
      - admins
      - users
```

Pour générer le Hash du password, exécutez la commande suivante :

```
docker run --rm authelia/authelia:latest authelia hash-password 'votre-mot-de-passe'
```

## Mise en place de la stack Docker-compose

Voici un exemple de docker-compose pour Authelia et son gestionnaire de session Redis :

```
version: '3.3'
services:
```

```
authelia:
  container_name: authelia
  image: authelia/authelia
  restart: always
  volumes:
    - /data/Dockers/authelia/config:/config
  ports:
    - 9091:9091
  healthcheck:
    disable: true
  environment:
    - TZ=Europe/Paris
  depends_on:
    - redis
redis:
  container_name: redis
  image: redis:alpine
  restart: always
  volumes:
    - /data/Dockers/authelia/redis:/data
  expose:
    - 6379
  environment:
    - TZ=Europe/Paris
```

Démarrez la stack :

```
docker-compose -p authelia -f ${AUTHELIA_DOCKERCOMPOSE_DIR}/authelia.yml up -d redis
docker-compose -p authelia -f ${AUTHELIA_DOCKERCOMPOSE_DIR}/authelia.yml up -d authelia
```

## Configuration d'Authelia sur Nginx Proxy Manager

Pour qu'Authelia puisse être fonctionnel sur le sous-domaine / domaine choisi, il doit être listé dans la liste des access control, et une configuration Nginx doit être ajoutée.

### Création de la configuration Nginx pour auth.domain.com

Créer une configuration reverse proxy pour le domaine **auth.domain.com** en upstream sur notre docker **authelia** port **9091** puis ajoutez la configuration suivante :

```

location / {
    set $upstream_authelia http://authelia:9091; # Adapter l'upstream authelia
    proxy_pass $upstream_authelia;
    client_body_buffer_size 128k;

    #Timeout if the real server is dead
    proxy_next_upstream error timeout invalid_header http_500 http_502 http_503;

    # Advanced Proxy Config
    send_timeout 5m;
    proxy_read_timeout 360;
    proxy_send_timeout 360;
    proxy_connect_timeout 360;

    # Basic Proxy Config
    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_set_header X-Forwarded-Proto $scheme;
    proxy_set_header X-Forwarded-Host $http_host;
    proxy_set_header X-Forwarded-Uri $request_uri;
    proxy_set_header X-Forwarded-Ssl on;
    proxy_redirect http:// $scheme://;
    proxy_http_version 1.1;
    proxy_set_header Connection "";
    proxy_cache_bypass $cookie_session;
    proxy_no_cache $cookie_session;
    proxy_buffers 64 256k;

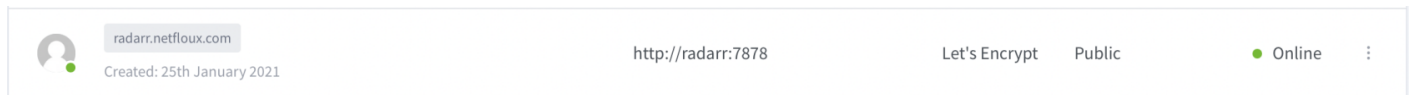
    # If behind reverse proxy, forwards the correct IP
    set_real_ip_from 10.0.0.0/8;
    set_real_ip_from 172.0.0.0/8;
    set_real_ip_from 192.168.0.0/16;
    set_real_ip_from fc00::/7;
    real_ip_header X-Forwarded-For;
    real_ip_recursive on;
}

```

Cette configuration permet la mise en place de l'authentification via Authelia sur Nginx pour le domaine d'auth.

# Sous domaine : radarr.domain.com

Admettons que votre Proxy Host est déjà configuré sur [Nginx Proxy Manager](#) :



Rendez-vous dans la configuration du Proxy Host puis dans l'onglet **Advanced** et ajoutez les lignes suivantes :

```
location /authelia {
    internal;
    set $upstream_authelia http://authelia:9091/api/verify; # Adapter l'upstream en fonction
de sa configuration
    proxy_pass_request_body off;
    proxy_pass $upstream_authelia;
    proxy_set_header Content-Length "";

    proxy_next_upstream error timeout invalid_header http_500 http_502 http_503;
    client_body_buffer_size 128k;
    proxy_set_header Host $host;
    proxy_set_header X-Original-URL $scheme://$http_host$request_uri;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $remote_addr;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_set_header X-Forwarded-Host $http_host;
    proxy_set_header X-Forwarded-Uri $request_uri;
    proxy_set_header X-Forwarded-Ssl on;
    proxy_redirect http:// $scheme://;
    proxy_http_version 1.1;
    proxy_set_header Connection "";
    proxy_cache_bypass $cookie_session;
    proxy_no_cache $cookie_session;
    proxy_buffers 4 32k;
    send_timeout 5m;
    proxy_read_timeout 240;
    proxy_send_timeout 240;
    proxy_connect_timeout 240;
}

location / {
```



```
set $upstream_radarr http://radarr:7878; # Adapter l'upstream en fonction de sa
configuration
```

```
proxy_pass $upstream_radarr;
```

```
auth_request /authelia;
```

```
auth_request_set $target_url $scheme://$http_host$request_uri;
```

```
auth_request_set $user $upstream_http_remote_user;
```

```
auth_request_set $groups $upstream_http_remote_groups;
```

```
proxy_set_header Remote-User $user;
```

```
proxy_set_header Remote-Groups $groups;
```

```
error_page 401 =302 https://auth.netfloux.com/?rd=$target_url;
```

```
client_body_buffer_size 128k;
```

```
proxy_next_upstream error timeout invalid_header http_500 http_502 http_503;
```

```
send_timeout 5m;
```

```
proxy_read_timeout 360;
```

```
proxy_send_timeout 360;
```

```
proxy_connect_timeout 360;
```

```
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_set_header X-Forwarded-Proto $scheme;
```

```
proxy_set_header X-Forwarded-Host $http_host;
```

```
proxy_set_header X-Forwarded-Uri $request_uri;
```

```
proxy_set_header X-Forwarded-Ssl on;
```

```
proxy_redirect http:// $scheme://;
```

```
proxy_http_version 1.1;
```

```
proxy_set_header Connection "";
```

```
proxy_cache_bypass $cookie_session;
```

```
proxy_no_cache $cookie_session;
```

```
proxy_buffers 64 256k;
```

```
set_real_ip_from 192.168.1.0/16;
```

```
real_ip_header X-Forwarded-For;
```

```
real_ip_recursive on;
```

```
}
```

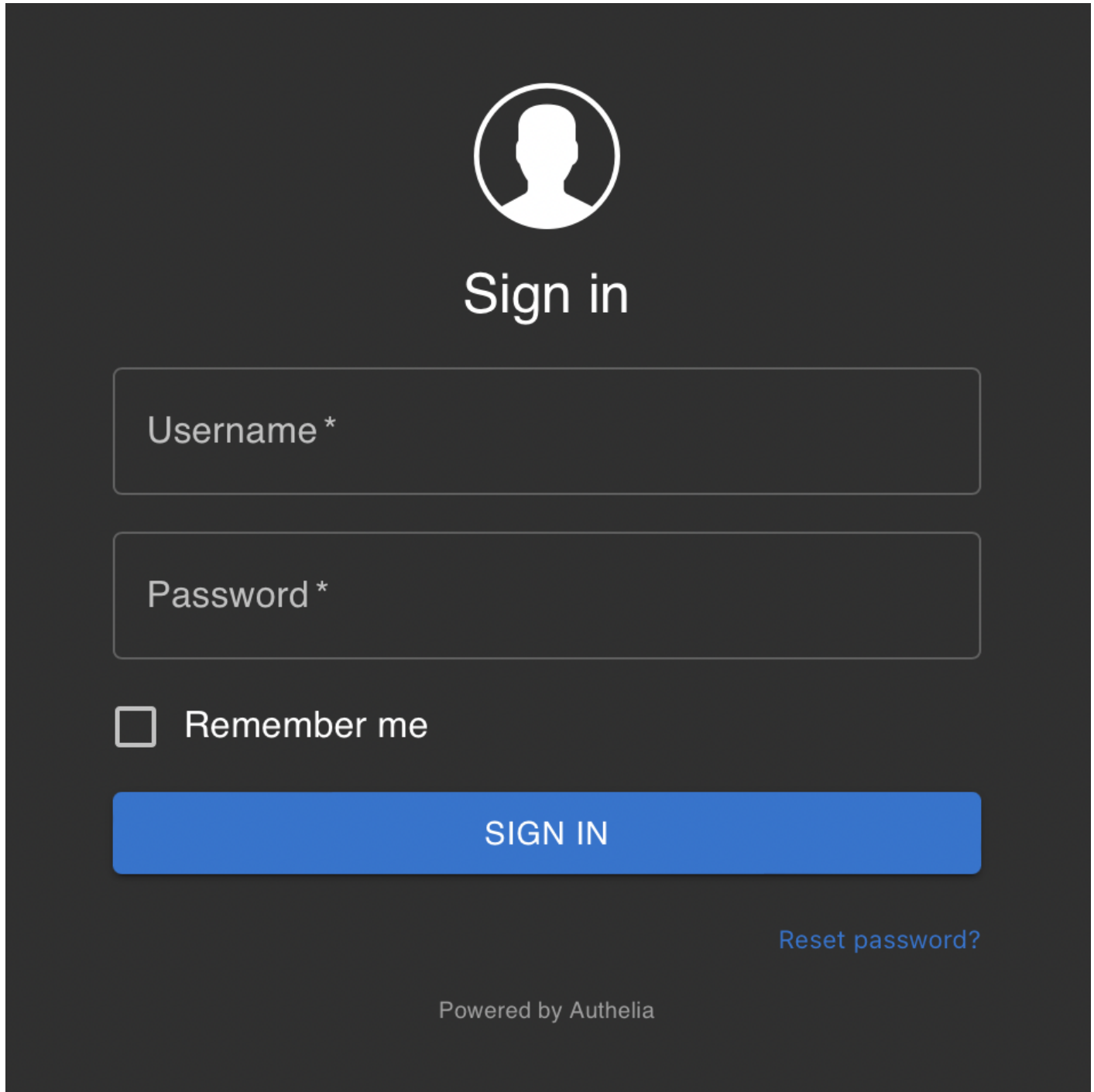
```
# Bypass de l'auth pour l'accès à l'API
```

```
location /api {
```

```
proxy_pass http://radarr:7878;
```

```
proxy_set_header Host $host;  
proxy_set_header X-Real-IP $remote_addr;  
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
}
```

Rendez-vous sur [radarr.domain.com](http://radarr.domain.com), vous devrez être redirigé sur une page de login Authelia :

The image shows a dark-themed login page for Authelia. At the top center is a white circular icon containing a silhouette of a person's head and shoulders. Below this icon, the text "Sign in" is displayed in a large, white, sans-serif font. Underneath the text are two stacked rectangular input fields with rounded corners and thin white borders. The first field is labeled "Username\*" and the second is labeled "Password\*" in a light gray font. Below the password field is a checkbox, which is currently unchecked, followed by the text "Remember me" in a light gray font. At the bottom of the form is a wide, solid blue button with the text "SIGN IN" in white, uppercase letters. To the right of the button, the text "Reset password?" is visible in a small, light blue font. At the very bottom center of the page, the text "Powered by Authelia" is displayed in a small, light gray font.

Location : [domain.com/radarr](http://domain.com/radarr)

## Configuration d'Authelia sur Nginx

# Tooling

Génération de tokens / passwords :

```
pip3 install pwgen  
pwgen -l 25  
quim5AhNgool9eimooceeseegh
```

---

Revision #9

Created 30 October 2021 06:36:32 by Martin Bouillaud

Updated 30 October 2021 07:56:42 by Martin Bouillaud