Authelia for NGINX Proxy Manager

- docker-compose.yml
- configuration.yml
- users_database.yml
- Authelia CONF
- Protected Domain CONF

docker-compose.yml

```
version: '3.3'

services:
  authelia:
    image: authelia/authelia
    container_name: authelia
    volumes:
      - /volume1/Configs/Authelia:/config #change this to a shared folder on your system. DO
NOT use a "volume"
    ports:
      - 9091:9091
    environment:
      - TZ=America/Denver
```

configuration.yml

After you've started your Authelia container, it will generate a configuration.yml file.

Once it's been generated, STOP the authelia container and replace the existing configuration.yml with the content below.

```
# yamllint disable rule:comments-indentation
---
###############################################################################
#                           Authelia Configuration                            #
###############################################################################

theme: dark #light/dark
jwt_secret: 1234567890abcdefghifjkl #any text or number you want to add here to create jwt
Token

default_redirection_url: https://google.com/ #where to redirect for a non-existent URL

server:
  host: 0.0.0.0
  port: 9091
  path: ""
  read_buffer_size: 4096
  write_buffer_size: 4096
  enable_pprof: false
  enable_expvars: false
  disable_healthcheck: false
  tls:
    key: ""
    certificate: ""

log:
  level: debug

totp:
  issuer: yourdomain.com #your authelia top-level domain
  period: 30
```

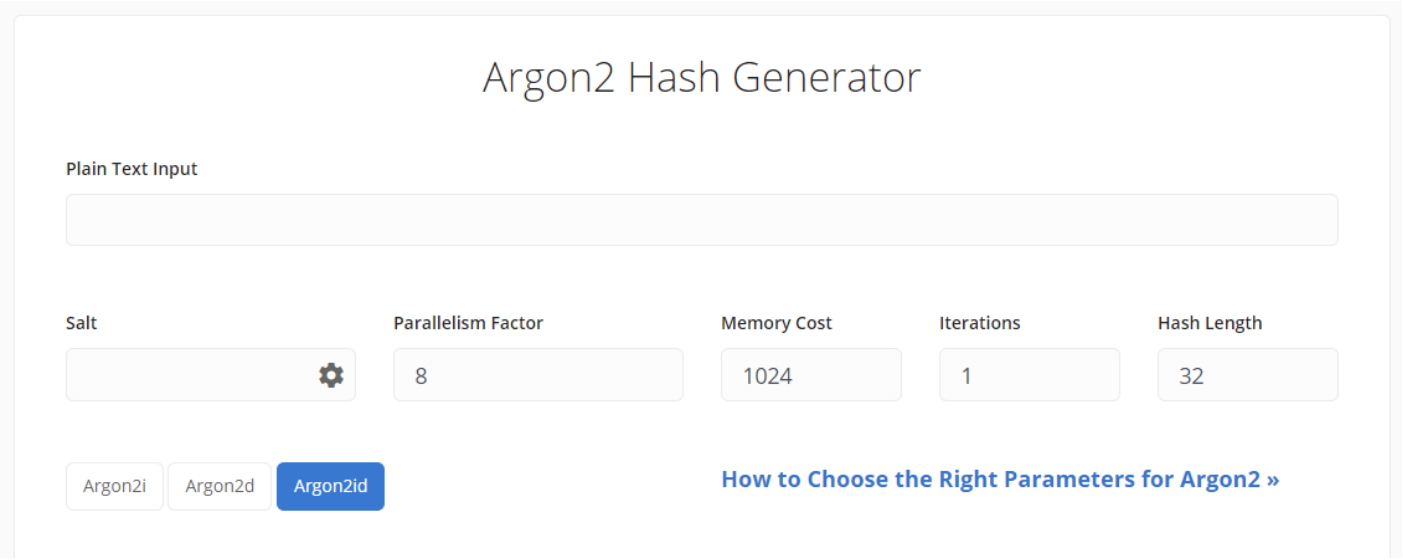
```
  skew: 1

authentication_backend:
  disable_reset_password: false
  refresh_interval: 5m
  file:
    path: /config/users_database.yml #this is where your authorized users are stored
    password:
      algorithm: argon2id
      iterations: 1
      key_length: 32
      salt_length: 16
      memory: 1024
      parallelism: 8

access_control:
  default_policy: deny
  rules:
    ## bypass rule
    - domain:
        - "auth.yourdomain.com" #This should be your authentication URL
      policy: bypass
    - domain: "yourdomain.com" #example domain to protect
      policy: one_factor
    - domain: "sub1.yourdomain.com" #example subdomain to protect
      policy: one_factor
    - domain: "sub2.yourdomain.com" #example subdomain to protect
      policy: one_factor
    - domain: "*.yourdomain.com" #example to protect all subdomains under top-level domain
      policy: one_factor
      #add or remove additional subdomains as necessary. currenlty only supports ONE top-level
domain
      #any time you add a new subdomain, you will need to restart the Authelia container to
recognize the new settings/rules

session:
  name: authelia_session
  secret: unsecure_session_secret #any text or number you want to add here to create jwt Token
  expiration: 3600  # 1 hour
  inactivity: 300  # 5 minutes
```

```
  domain: yourdomain.com  # Should match whatever your root protected domain is

regulation:
  max_retries: 3
  find_time: 10m
  ban_time: 12h

storage:
  local:
    path: /config/db.sqlite3 #this is your databse. You could use a mysql database if you
wanted, but we're going to use this one.
  encryption_key:
you_must_generate_a_random_string_of_more_than_twenty_chars_and_configure_this #added Dec 5
2021

notifier:
  disable_startup_check: true #true/false
  smtp:
    username: youremail@gmail.com #your email address
    password: Y0uRp@55W0rD! #your email password
    host: smtp.gmail.com #email smtp server
    port: 587 #email smtp port
    sender: youremail@gmail.com
    identifier: localhost
    subject: "[Authelia] {title}" #email subject
    startup_check_address: youremail@gmail.com
    disable_require_tls: false
    disable_html_emails: false
    tls:
      skip_verify: false
      minimum_version: TLS1.2
...
```

users_database.yml

```
users:
  user1: #username for user 1. change to whatever you'd like
    displayname: "User Name 1" #whatever you want the display name to be
    password:
"$argon2i$v=19$m=1024,t=1,p=8$eTQ3MXdqOGFiaDZoMUtMVw$OeHWQSg9zGKslOepe5t4D1T9BZJjHA1Z+doxZrZYD
gI" #generated at https://argon2.online/
    email: youremail@gmail.com #whatever your email address is
    groups: #enter the groups you want the user to be part of below
      - admins
      - dev
  user2: #username for user 2. change to whatever you'd like. Or delete this section if you
only have 1 user
    displayname: "User Name 2" #whatever you want the display name to be
    password:
"$argon2i$v=19$m=1024,t=1,p=8$eTQ3MXdqOGFiaDZoMUtMVw$OeHWQSg9zGKslOepe5t4D1T9BZJjHA1Z+doxZrZYD
gI" #generated at https://argon2.online/
    email: youremail2@gmail.com #whatever your email address is
    groups: #enter the groups you want the user to be part of below
      - dev

#any time you add a new user, you will need to restart the Authelia container to recognize the
new settings/rules
```

When you go to https://argon2.online/ to generate your passwords, the use the settings you see below:



Enter your password into the "Plain Text Input"

Click the gear in "Salt" to generate a random string of characters.

Other settings:

- Parallelism: 8
- Memory Cost: 1024
- Iterations: 1
- Hash Length: 32

Be sure to have "Argon2id" activated.

Click "Generate Hash"

Copy the string that starts with `$argon2id` into the associated user password in the users_database.yml

Authelia CONF

You will need to setup a domain/subdomain for your authentication. Setup your domain/subdomain in NGINX Proxy Manager as normal. Then add the following to the "Advanced" tab in the Proxy Host setup for the domain.

```
location / {
set $upstream_authelia http://192.168.1.25:9091; # This example assumes a Docker deployment.
Change the IP and Port to your setup
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, assumes you're using Cloudflare.  Adjust IP
```

```
for your Docker network.
set_real_ip_from 192.168.1.0/24; #make sure this IP range matches your netowrk setup
real_ip_header CF-Connecting-IP;
real_ip_recursive on;
}
```

Protected Domain CONF

This will be placed in the "Advanced" tab of the "Edit Proxy Host" in NGINX Proxy Manager for the domain you want to protect.

```
location /authelia {
internal;
set $upstream_authelia http://192.168.1.25:9091/api/verify; #change the IP and Port to match the IP and Port of your Authelia container
proxy_pass_request_body off;
proxy_pass $upstream_authelia;
proxy_set_header Content-Length "";

# Timeout if the real server is dead
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_uptime-kuma $forward_scheme://$server:$port; #change uptime-kuma to match your container name: $upstream_some-container-name or $upstream_somecontainername
proxy_pass $upstream_uptime-kuma; #change uptime-kuma to match your container name: $upstream_some-container-name or $upstream_somecontainername

auth_request /authelia;
auth_request_set $target_url https://$http_host$request_uri;
auth_request_set $user $upstream_http_remote_user;
auth_request_set $email $upstream_http_remote_email;
auth_request_set $groups $upstream_http_remote_groups;
proxy_set_header Remote-User $user;
proxy_set_header Remote-Email $email;
proxy_set_header Remote-Groups $groups;

error_page 401 =302 https://auth.yourdomain.com/?rd=$target_url; #change this to match your authentication domain/subdomain

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 Upgrade $http_upgrade;
proxy_set_header Connection upgrade;
proxy_set_header Accept-Encoding gzip;
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; #make sure this matches your network setup
real_ip_header CF-Connecting-IP;
real_ip_recursive on;

}
```