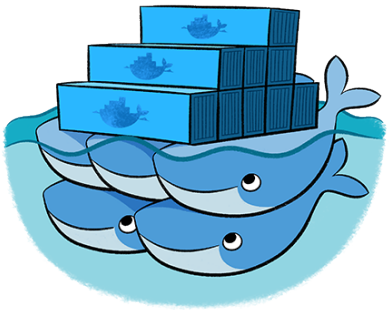
Step by Step Setup Guide for Authelia (+Traefik)



This article provides detailed instructions on integrating Authelia as a middleware with Traefik. Using Docker labels for configuration, this setup allows Traefik to query Authelia for authorization

on every web request. Authelia validates session cookies and access permissions for secure resource control. The information is partially sourced from Brynn Crowley, referencing his setup guide.

Prerequisites

- Docker Swarm
- GlusterFS & Keepalived
- Traefik Reverse Proxy

Important Notes

> Configuration uses Docker labels directly in `docker-compose.yaml` to configure the Traefik Middleware.
> Examples use a *whoami* application for demonstration.
>
> Advanced configurations (e.g., SMTP) are available in Authelia documentation.

Step-by-Step Guide

Step 1: Create Folders in the GlusterFS

```
mkdir -p /mnt/glustermount/data/authelia_data/
mkdir -p /mnt/glustermount/data/authelia_data/logs
mkdir -p /mnt/glustermount/data/authelia_data/config
mkdir -p /mnt/glustermount/data/authelia_data/secrets
```

Step 2: Create External Network for Traefik Proxy (if not already done)

Create the `management_net` network:

```
docker network create -d overlay management_net
```

Step 3: Configure User Database

Create a basic user database:

```
nano mkdir -p /mnt/glustermount/data/authelia_data/config/users.yml
```

Paste this Content:

```
users:
  authelia: ## Username
    displayname: 'Authelia User'
    ## WARNING: This is a default password for testing only!
    ## IMPORTANT: Change this password before deploying to production!
    ## Generate a new hash using the instructions at:
    ## https://www.authelia.com/reference/guides/passwords/#passwords
    ## Password is 'authelia'
    password:
'$6$rounds=50000$BpLnfgDsc2WD8F2q$Zis.ixdg9s/UOJYrs56b5QEZFiZECu0qZVNsIYxBaNJ7ucIL.nlxVCT5tqh8KH
G8X4tlwCFm5r6NTOZZ5qRFN/'
    email: 'authelia@authelia.com'
    groups:
      - 'admin'
      - 'dev'
```

> The current password listed is `authelia`. It is important you Generate a new password hash.

Step 3.1: Generate Password Hash (Optional)

```
docker run --rm -it authelia/authelia:latest authelia crypto hash generate argon2
```

Step 4: Create Secrets

First provide needed Rights:

```
chown 8000:8000 /mnt/glustermount/data/authelia_data/secrets/ && chmod 0700
/mnt/glustermount/data/authelia_data/secrets/
```

Then Create Secret Files:

```
docker run --rm -u 8000:8000 -v /mnt/glustermount/data/authelia_data/secrets/:/secrets
docker.io/authelia/authelia sh -c "cd /secrets && authelia crypto rand --length 64 session_secret.txt
storage_encryption_key.txt jwt_secret.txt"
```

Step 5: Create Basic Authelia Configuration

Create a File called "configuration.yaml" in your Config Folder (In this Example `/mnt/glustermount/data/authelia_data/config`)

```
nano /mnt/glustermount/data/authelia_data/config/configuration.yaml
```

> Make sure to Change the Tags "`YOURDOMAIN`" to your actual Domainname.

```
server:
  address: 'tcp4://:9091'

log:
  level: debug
  file_path: '/var/log/authelia/authelia.log'
  keep_stdout: true

identity_validation:
  elevated_session:
    require_second_factor: true
  reset_password:
    jwt_lifespan: '5 minutes'
    jwt_secret: {{ secret "/secrets/jwt_secret.txt" | mindent 0 "|" | msquote }}

totp:
  disable: false
  issuer: 'YOURDOMAIN.com'
  period: 30
```

```
  skew: 1

password_policy:
  zxcvbn:
    enabled: true
    min_score: 4

authentication_backend:
  file:
    path: '/config/users.yml'
    password:
      algorithm: 'argon2'
      argon2:
        variant: 'argon2id'
        iterations: 3
        memory: 65535
        parallelism: 4
        key_length: 32
        salt_length: 16

access_control:
  default_policy: 'deny'
  rules:
    - domain: 'traefik.YOURDOMAIN.com'
      policy: 'one_factor'
    - domain: 'whoami-secure.YOURDOMAIN.com'
      policy: 'two_factor'

session:
  name: 'authelia_session'
  secret: {{ secret "/secrets/session_secret.txt" | mindent 0 "|" | msquote }}
  cookies:
    - domain: 'YOURDOMAIN.com'
      authelia_url: 'https://auth.YOURDOMAIN.com'

regulation:
  max_retries: 4
  find_time: 120
  ban_time: 300
```

```
storage:
  encryption_key: {{ secret "/secrets/storage_encryption_key.txt" | mindent 0 "|" | msquote }}
  local:
    path: '/config/db.sqlite3'

notifier:
  disable_startup_check: false
  filesystem:
    filename: '/config/notification.txt'
```

Step 6: Create Docker Compose

> It is important to know that Traefik needs to wait for Authelia to startup. Thats what the depends Function is for.
> Otherwise Traefik will not notice the Authelia Middleware and maybe provide an Error.

> Make sure to Change the Tags "`YOURDOMAIN`" to your actual Domainname.

```
version: '3.3'

services:
  traefik:
    user: 0:0 #Container being started with Root rights
    image: 'traefik:latest'
    security_opt:
      - 'no-new-privileges=true'
    restart: 'unless-stopped'
    depends_on:
      - authelia
    ports:
      # The Web UI (enabled by --api.insecure=true in traefik.toml)
      - '8080:8080'
      # The Available Ports (forward your router's incoming ports to the ports on the host)
      - '80:80'
      - '443:443'
    networks:
      management_net:
```

```
      aliases:
        - 'auth.domain.com'
    authelia: {}
  volumes:
    # So that Traefik can listen to the Docker events (read-only)
    - '/var/run/docker.sock:/var/run/docker.sock:ro'
    # LetsEncrypt ACME Configuration
    - '/mnt/glustermount/data/traefik_data/acme.json:/le/acme.json'
    # Mount for Traefik AccessLog
    - '/mnt/glustermount/data/traefik_data/access.log:/access.log'
    # (STATIC CONFIG)
    - './traefik/config/traefik.yml:/traefik.yml:ro'
    # (DYNAMIC CONFIG)
    - './traefik/config/dynamic.yml:/dynamic.yml:ro'

  environment:
    - TZ=Europe/Zurich
  deploy:
    mode: replicated
    replicas: 1
    labels:
      - 'traefik.enable=true'
      - 'traefik.http.routers.traefik.rule=Host(`traefik.YOURDOMAIN.com`)'
      - 'traefik.http.routers.traefik.service=api@internal'
      - 'traefik.http.services.traefik.loadbalancer.server.port=8080'
      - 'traefik.http.routers.traefik.tls.certresolver=leresolver'
      - 'traefik.http.routers.traefik.entrypoints=websecure'
      - 'traefik.http.routers.http-catchall.rule=hostregexp(`{host:.+}`)'
      - 'traefik.http.routers.http-catchall.entrypoints=web'
      - 'traefik.http.routers.http-catchall.middlewares=redirect-to-https'
      - 'traefik.http.middlewares.redirect-to-https.redirectscheme.scheme=https'
      #Authelia Integration
      - 'traefik.http.routers.dashboard.middlewares=authelia@docker'

authelia:
  image: 'authelia/authelia:4.38'
  container_name: 'authelia'
  volumes:
    - '/mnt/glustermount/data/authelia_data/secrets:/secrets:ro'
    - '/mnt/glustermount/data/authelia_data/config:/config'
```

```
      - '/mnt/glustermount/data/authelia_data/logs:/var/log/authelia/'
    networks:
      authelia: {}
      management_net: {}
    labels:
      ## Expose Authelia through Traefik
      traefik.enable: 'true'
      traefik.docker.network: 'authelia'
      traefik.http.routers.authelia.rule: 'Host(`auth.YOURDOMAIN.com`)'
      traefik.http.routers.authelia.entrypoints: 'websecure'
      traefik.http.routers.authelia.tls.certresolver: 'leresolver'
      traefik.http.services.authelia.loadbalancer.server.port: '9091'
      ## Setup Authelia ForwardAuth Middlewares
      traefik.http.middlewares.authelia.forwardAuth.address: 'http://traefik_authelia:9091/api/authz/forward-auth'
      traefik.http.middlewares.authelia.forwardAuth.trustForwardHeader: 'true'
      traefik.http.middlewares.authelia.forwardAuth.authResponseHeaders: 'Remote-User,Remote-
Groups,Remote-Name,Remote-Email'
    environment:
      TZ: 'Europe/Zurich'
      X_AUTHELIA_CONFIG_FILTERS: 'template'

  whoami-secure:
    image: 'traefik/whoami'
    restart: 'unless-stopped'
    container_name: 'whoami-secure'
    depends_on:
      - authelia
    labels:
      traefik.enable: 'true'
      traefik.http.routers.whoami-secure.rule: 'Host(`whoami-secure.YOURDOMAIN.com`)'
      traefik.http.routers.whoami-secure.entrypoints: 'websecure'
      traefik.http.routers.whoami-secure.middlewares: 'authelia@docker'
      traefik.http.services.whoami-secure.loadbalancer.server.port: '80'
      traefik.http.routers.whoami-secure.tls.certresolver: 'leresolver'
    networks:
      management_net: {}
networks:
  management_net:
    external: true  # Primary network for management
  authelia:
```

Step 7: Start the Stack

You can either do that with the provided Command or start the Stack with Portainer.

```
docker compose up -d
```

Step 8: Verify Setup

- Check container status: `docker compose ps`
- Access Traefik dashboard: https://traefik.domain.com
- Test authentication: https://whoami-secure.domain.com

> Now you are ready to Setup with further Configuration.
>
> **It is possible to add Authelia Middleware to your Custom Applications by adding a Traefiklabel to your Composefiles:**
> **traefik.http.routers.YOUR-APPLICATION.middlewares: 'authelia@docker'**

Troubleshooting

- Check logs: `docker logs authelia`
- Ensure secret files exist and have correct permissions.
- Check Official Docs: https://www.authelia.com/configuration/prologue/introduction/

Further Configuration

Check out other Posts in my Wiki about setting up SMTP for example.

https://wiki.aeoneros.com/books/authelia/chapter/configuration

Add OIDC

To be able to use Authelia for OIDC in 3rd Party Software make sure to Check out my OIDC Guide

https://wiki.aeoneros.com/books/authelia/chapter/openid-connect-10

Revision #19
Created 8 January 2025 20:52:44 by aeoneros
Updated 11 February 2025 08:35:53 by aeoneros