Authentik Docker Compose Install

Authentik is an open-source Identity Provider (IdP) that helps you manage authentication and authorization across your apps and infrastructure. It supports:

- Single Sign-On (SSO) via OAuth2, OpenID Connect, SAML
- LDAP & SCIM integration
- Multi-factor authentication
- Reverse proxy for seamless app protection

Think of it as your self-hosted alternative to services like Okta or Auth0, but with full control and flexibility.

Prerequisites:

- Docker & Docker Compose

[Authentik Docker Compose Installation Guide]

Install Steps:

1. Open SSH and get to the device you want to run it on. (my case Overseer)
2. grab preconfigured yml

```
wget https://goauthentik.io/docker-compose.yml
```

If this is a fresh authentik installation, you need to generate a password and a secret key.

3. Run the following commands to generate a password and secret key and write them to your `.env` file:

```
echo "PG_PASS=$(openssl rand -base64 36 | tr -d '\n')" >> .env
echo "AUTHENTIK_SECRET_KEY=$(openssl rand -base64 60 | tr -d '\n')" >> .env
```

4. To enable error reporting, run the following command:

```
echo "AUTHENTIK_ERROR_REPORTING__ENABLED=true" >> .env
```

5. By default, authentik listens internally on port 9000 for HTTP and 9443 for HTTPS.

```
cd /docker/authentik/.env
```

6. To change the exposed ports to 80 and 443, you can set the following variables in `.env`:

```
COMPOSE_PORT_HTTP=80
COMPOSE_PORT_HTTPS=443
```

7. Startup docker compose

```
docker compose pull
docker compose up -d
```

To start the initial setup, navigate to **http://<your server's IP or hostname>:9000/if/flow/initial-setup/**

Alternative Install Steps:

1. Open SSH and get to the device you want to run it on. (my case Overseer)
2. Create Directory

```
mkdir /docker/authentik
cd /docker/authentik
```

3. Create docker-compose.yml and edit it

```
nano docker-compose.yml #might need to use sudo if it doesn't give you access
```

```
version: '3.8'

services:
  postgresql:
    image: postgres:15
    environment:
      POSTGRES_DB: authentik
```

```
    POSTGRES_USER: authentik
    POSTGRES_PASSWORD: authentik
  volumes:
    - postgresql_data:/var/lib/postgresql/data

redis:
  image: redis:7
  volumes:
    - redis_data:/data

server:
  image: ghcr.io/goauthentik/server:latest
  depends_on:
    - postgresql
    - redis
  environment:
    AUTHENTIK_SECRET_KEY: "supersecretkey"
    AUTHENTIK_POSTGRESQL__HOST: postgresql
    AUTHENTIK_POSTGRESQL__USER: authentik
    AUTHENTIK_POSTGRESQL__PASSWORD: authentik
    AUTHENTIK_POSTGRESQL__NAME: authentik
    AUTHENTIK_REDIS__HOST: redis
  ports:
    - "8080:8000"  # Web UI
    - "9444:9443"  # Proxy port
  volumes:
    - authentik_media:/media
    - authentik_static:/static

worker:
  image: ghcr.io/goauthentik/worker:latest
  depends_on:
    - server
  environment:
    AUTHENTIK_SECRET_KEY: "supersecretkey"
    AUTHENTIK_POSTGRESQL__HOST: postgresql
    AUTHENTIK_POSTGRESQL__USER: authentik
    AUTHENTIK_POSTGRESQL__PASSWORD: authentik
    AUTHENTIK_POSTGRESQL__NAME: authentik
    AUTHENTIK_REDIS__HOST: redis
```

```
    volumes:
      - authentik_media:/media
      - /var/run/docker.sock:/var/run/docker.sock

volumes:
  postgresql_data:
  redis_data:
  authentik_media:
  authentik_static:
```

4. Create the .env file

```
nano .env #might need to run it with sudo
```

```
# Database credentials
PG_USER=authentik
PG_PASS=supersecurepassword123
PG_DB=authentik

# Authentik image tag
AUTHENTIK_IMAGE=ghcr.io/goauthentik/server
AUTHENTIK_TAG=2025.6

# Optional: HTTP/HTTPS ports (not forwarded externally)
COMPOSE_PORT_HTTP=9000
COMPOSE_PORT_HTTPS=9444

# Secret Key
AUTHENTIK_SECRET_KEY=your-super-secret-key
```

5. Start the stack

```
docker-compose up -d
```

Once the stack is up, everything is finished installing you can check it with

```
docker-compose ps
```

To start the initial setup, navigate to http://<your server's IP or hostname>:9000/if/flow/initial-setup/.

Revision #5
Created 2025-08-06 01:18:27 UTC by overseer
Updated 2025-08-08 17:55:04 UTC by lumxux