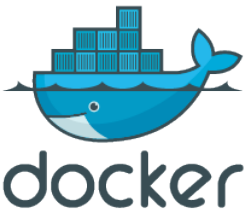
Step-by-Step Install Guide for Linkwarden with Traefik on Docker Swarm



Introduction

In this guide, we'll walk through the process of installing Linkwarden, a self-hosted bookmark manager, on Docker Swarm. This installation will include setting up Traefik as the reverse proxy and configuring persistent storage using GlusterFS. Additionally, we'll cover how to securely set up environment variables, such as the NEXTAUTH_SECRET, and ensure proper file permissions using a custom Linux user.

Step 1: Create a User for Linkwarden Folders

To ensure that only one Linux user has the necessary rights for the Linkwarden folders, we'll create a custom user and group called `linkwardenuser`. This user will not have a home directory, and the shell will be set to `/usr/sbin/nologin` for security purposes.

1. Create the custom user and group with the GID and UID set to 10002:

```
sudo groupadd -g 10002 linkwardenuser
sudo useradd -u 10002 -g linkwardenuser -s /usr/sbin/nologin -M linkwardenuser
```

- The `-s /usr/sbin/nologin` option ensures that this user cannot log in interactively, which is a security best practice for service users.
- The `-M` option prevents the creation of a home directory, as this user will only be managing Linkwarden's folder permissions and not need a home directory for other purposes.

Step 2: Create Folders for Linkwarden

Next, we'll create the necessary folders to store Linkwarden's data on the GlusterFS mount.

1. Create the folders for Linkwarden's data:

```
mkdir -p /mnt/glustermount/data/linkwarden_data
mkdir -p /mnt/glustermount/data/linkwarden_data/pgdata
mkdir -p /mnt/glustermount/data/linkwarden_data/lwdata
mkdir -p /mnt/glustermount/data/linkwarden_data/storage
```

2. Adjust ownership of these folders to the `linkwardenuser`, ensuring all files inside are accessible to this user:
   This ensures that only the `linkwardenuser` has access to these folders and files, maintaining data security.

```
sudo chown -R linkwardenuser:linkwardenuser /mnt/glustermount/data/linkwarden_data
```

3. **Permissions**: The following permissions ensure that the owner (user `10002`) has full access (read, write, execute) to the directories and files, and that no one else can modify the files.

```
sudo chmod -R 750 /mnt/glustermount/data/linkwarden_data
```

> **750** means:
> - 7: Owner (user 10002) has read, write, and execute permissions.
> - 5: Group has read and execute permissions (but not write).
> - 0: Others have no permissions.

Step 3: Create `docker-compose.yaml`

Now, we'll create the Docker Compose file that will define two services: one for Linkwarden and another for its PostgreSQL database. We will also configure Traefik to route traffic to Linkwarden.

> To Configure the Composefile check out the ENVIROMENT-VARIABLES Wiki Article.

```
version: "3.5"

services:
  linkwarden:
    image: ghcr.io/linkwarden/linkwarden:latest
    environment:
      - DATABASE_URL=postgresql://linkwarden:${POSTGRES_PASSWORD}@postgres:5432/linkwardendb
      - NEXTAUTH_URL=http://localhost:3000/api/v1/auth
      - NEXTAUTH_SECRET=${NEXTAUTH_SECRET}
      - STORAGE_FOLDER=/mnt/glustermount/data/linkwarden_data/storage
      - NEXT_PUBLIC_EMAIL_PROVIDER=${NEXT_PUBLIC_EMAIL_PROVIDER}
      - EMAIL_FROM=${EMAIL_FROM}
      - EMAIL_SERVER=${EMAIL_SERVER}
      - BASE_URL=${BASE_URL}
      - TZ=Europe/Zurich
      - GID=10002
      - UID=10002
    restart: always
    ports:
      - 3000:3000
    volumes:
      - /mnt/glustermount/data/linkwarden_data/lwdata:/data/data
    depends_on:
      - postgres
    networks:
     - management_net
    deploy:
      replicas: 1
      placement:
        constraints:
          - node.role == manager
      labels:
        - "traefik.enable=true"
        - "traefik.http.routers.linkwarden.rule=Host(`linkwarden.domain.tld`)"
```

```
      - "traefik.http.services.linkwarden.loadbalancer.server.port=3000"
      - "traefik.http.routers.linkwarden.entrypoints=websecure"
      - "traefik.http.routers.linkwarden.tls.certresolver=leresolver"

  postgres:
    image: postgres:16-alpine
    environment:
      POSTGRES_USER: linkwarden
      POSTGRES_PASSWORD: ${POSTGRES_PASSWORD}
      POSTGRES_DB: linkwardendb
      TZ: Europe/Zurich
      GID: 10002
      UID: 10002
    restart: always
    volumes:
      - /mnt/glustermount/data/linkwarden_data/pgdata:/var/lib/postgresql/data
    ports:
      - 5432:5432
    networks:
     - management_net
    deploy:
      replicas: 1
      placement:
        constraints:
          - node.role == manager
      labels:
        - "traefik.enable=true"
        - "traefik.http.services.postgres.loadbalancer.server.port=5432"

networks:
  management_net:
    external: true
```

SMTP Settings

- # The base URL of your Linkwarden installation (replace with your domain or local IP) BASE_URL=https://linkwarden.domain.tld
- # Email provider for sending notification emails (example: SMTP settings) NEXT_PUBLIC_EMAIL_PROVIDER=smtp

- # Email address that Linkwarden will use to send emails from (replace with your actual email)
  EMAIL_FROM=linkwarden@domain.tld
- # SMTP server details (example: for Gmail's SMTP server)
  EMAIL_SERVER=smtp://smtp.gmail.com:587

Step 4: Create the `NEXTAUTH_SECRET`

The `NEXTAUTH_SECRET` is used to sign authentication tokens securely. You need to generate a random string to be used as the `NEXTAUTH_SECRET`.

You can generate a secure `NEXTAUTH_SECRET` using the following command:

```
openssl rand -base64 32
```

This command will generate a 32-byte random string that you can add to your `.env` file as the `NEXTAUTH_SECRET`.

What Does `NEXTAUTH_SECRET` Do?

The `NEXTAUTH_SECRET` is critical for securing the authentication process in Linkwarden. It is used to sign and encrypt tokens, ensuring that user sessions are protected from tampering or unauthorized access.

Step 5: Start the Stack

Once everything is configured, you can start the Linkwarden stack either manually or through Portainer.

Start with Docker Swarm

To start the stack manually, run:

```
docker stack deploy -c docker-compose.yaml linkwarden
```

Start with Portainer

Alternatively, you can use Portainer’s graphical interface to import the `docker-compose.yaml` file and start the stack.

> Once the stack is deployed, Linkwarden will be available at `https://linkwarden.domain.tld` (or whatever domain you've configured).

Conclusion

This guide provides a detailed walkthrough for setting up Linkwarden with Traefik on Docker Swarm. From user and folder management to Docker Compose configuration, these steps ensure a secure and scalable deployment of your self-hosted bookmark manager.

Revision #7
Created 10 October 2024 12:54:03 by aeoneros
Updated 10 October 2024 21:12:53 by aeoneros