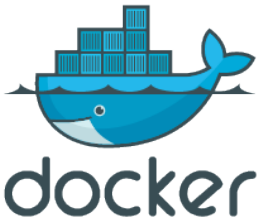
Step-by-Step Install Guide for Wallos with Traefik on Docker Swarm



Live-Demo

User: **demo** / Password: **demo**

Introduction

This guide provides step-by-step instructions to set up **Wallos**, a self-hosted subscription management application, on a Docker Swarm cluster with Traefik as a reverse proxy. Wallos can be accessed locally or through Traefik on the internet. This setup ensures persistent data using GlusterFS and reliable networking with Keepalived.

Prerequisites

- Docker Swarm is set up and running.
- Traefik is configured for reverse proxy and SSL management.
- Keepalived & GlusterFS are set up for distributed storage and high availability.

Step 1: Update System Packages

Before proceeding, ensure all Raspberry Pis and Docker installations are up to date. Run the following commands on each node:

```
sudo apt update && sudo apt upgrade -y
sudo apt install -y docker.io docker-compose
```

Step 2: Create Directories for Wallos Data

If you're running Wallos on Docker Swarm with GlusterFS, create the following directories on the shared GlusterFS mount:

```
mkdir -p /mnt/glustermount/data/wallos_data/db
mkdir -p /mnt/glustermount/data/wallos_data/logos
```

Step 3: Create the Docker Compose File

Define the Wallos service using the following `docker-compose.yml` file. You can create this file manually or configure it through Portainer.

```
version: "3.7"

services:
  wallos:
    container_name: wallos_subscription_tracker
    image: bellamy/wallos:latest
    ports:
      - "8282:80"
    environment:
      TZ: 'Europe/Zurich'
    volumes:
      - '/mnt/glustermount/data/wallos_data/db:/var/www/html/db'
      - '/mnt/glustermount/data/wallos_data/logos:/var/www/html/images/uploads/logos'
    restart: unless-stopped
    networks:
      - management_net
    deploy:
      mode: replicated
      replicas: 1
      labels:
```

```
      - "traefik.enable=true"
      - "traefik.http.services.wallos.loadbalancer.server.port=80"


networks:
  management_net:
    external: true
```

Explanation of Key Components:

- **Replicated Mode:** Ensures Wallos is deployed with multiple instances for high availability if replicas > 1.
- **Traefik Labels:** Enables Traefik routing and port mapping for this service. Modify the labels to match your domain and Traefik configuration.

Step 4: Deploy the Stack

Deploy the stack using Docker Swarm. Navigate to the directory containing the `docker-compose.yml` file and execute:

```
docker stack deploy -c docker-compose.yml wallos
```

> You can also deploy your Stack in Portainer

Step 5: Access Wallos

- Access Wallos locally via `http://IP_ADDRESS:8282`.
- If Traefik is configured, use `https://wallos.domain.com` with the websecure entry point OR
- `http://wallos.domain.com` with the web entry point.

Optional:

You could add a Local DNS Entry into PiHole to Access Wallos.

Additional Notes

- **Persistent Data:** All Wallos data is stored in GlusterFS directories, ensuring availability across all Swarm nodes.
- **Traefik Configuration:** Ensure Traefik is set up with SSL certificates for secure access.
- **Logs and Debugging:** Monitor logs using `docker logs wallos_subscription_tracker` for troubleshooting or access Portainer WebUI.

Revision #5
Created 5 January 2025 10:29:07 by aeoneros
Updated 5 January 2025 11:17:02 by aeoneros