Install Get Healthy

https://pt.opensourceisawesome.com/videos/embed/x6SxzrvdRemA4gaZU7apdp

Setup a Server

Install Updates

Update and upgrade your server's packages using the following commands:

- For Ubuntu/Debian: `sudo apt update && sudo apt upgrade -y`
- For RedHat/CentOS/Fedora/Alma/Rocky: `sudo dnf update -y`

Create a Non-Root User

Create a non-root user with superuser (sudo) privileges:

1. Add a new user using `adduser <username>`
2. Set the password for this user.
3. Enter the relevant information (optional)
4. Enter 'Y', then press Enter.
5. Add the user to the "sudo" group:
   - For Ubuntu/Debian: `usermod -aG sudo <username>`
   - For RedHat/CentOS/Fedora/Alma/Rocky: `usermod -aG wheel <username>`

Ubuntu / Debian

`usermod -aG sudo <username>`

RedHat / CentOS / Fedora / Alma / Rocky

`usermod -aG wheel <username>`

Now, you can log out of the system, and log back in as your new non-root super user.

Install Docker and Docker Compose

Install Docker and Docker Compose on your server:

1. Install the `curl` utility:
   - For Ubuntu/Debian: `sudo apt install curl -y`
   - For RedHat/CentOS/Fedora/Alma/Rocky: `sudo dnf install curl -y`
2. Run the command to install Docker and Docker Compose:
   `curl https://get.docker.com | sh`

Add Your User to the Docker Group

Add your non-root user to the docker group so you can use Docker commands without sudo:

`sudo usermod -aG docker <username>`

Now log out and back into your system, and you are ready to setup your application.

Install and Configure Get Healthy

What you'll need:

- A machine to use as a server
- Docker and Docker Compose (see instructions above)
- (Optional) - Domain Name and Reverse Proxy
- About 20 minutes of your time

1. Let's begin by creating our folder structure for our docker apps in our server. If you've already done this, move to the next step.

`mkdir -p docker/get-healthy`

2. We'll move into the folder we just created.

`cd docker/get-healthy`

3. Create a new file called 'compose.yaml'

`nano compose.yaml`

4. Copy the code block below, and paste it into your new file.

```
---
services:
 db:
   image: postgres:16-alpine
   container_name: get_healthy_db
   restart: unless-stopped
   environment:
```

```
      POSTGRES_USER: ${POSTGRES_USER:-gethealthy}
      POSTGRES_PASSWORD: ${POSTGRES_PASSWORD:-changeme}
      POSTGRES_DB: ${POSTGRES_DB:-get_healthy}
    volumes:
      - ./postgres_data:/var/lib/postgresql/data
    ports:
      - "5432:5432"
    healthcheck:
      test: ["CMD-SHELL", "pg_isready -U ${POSTGRES_USER:-gethealthy} -d ${POSTGRES_DB:-get_>
      interval: 10s
      timeout: 5s
      retries: 5

  backend:
    image: bmcgonag/get_healthy_backend:latest
    container_name: get_healthy_backend
    restart: unless-stopped
    environment:
      DATABASE_URL: postgresql://${POSTGRES_USER:-gethealthy}:${POSTGRES_PASSWORD:-changeme}>
      JWT_SECRET: ${JWT_SECRET:-dev-secret-change-in-production}
      NODE_ENV: production
      PORT: 3000
      FRONTEND_URL: ${FRONTEND_URL:-http://localhost:8025}
    volumes:
      - ./backend/uploads:/app/uploads
    depends_on:
      db:
        condition: service_healthy

  frontend:
    image: bmcgonag/get_healthy_frontend:latest
    container_name: get_healthy_frontend
    restart: unless-stopped
    environment:
      VITE_ALLOWED_HOSTS: ${VITE_ALLOWED_HOSTS:-localhost,127.0.0.1}
      VITE_API_URL: ${VITE_API_URL:-/api}
    ports:
      - "80:80"  # <- you can change the left side to a different port number if needed (e.g.
8080 or any open port)
    depends_on:
```

```
      - backend
```

5. Save this file and exit the nano editor with CTRL + O, then press Enter, then CTRL + X

6. Now we'll create a file where we'll define the values for our compose file to use (environment variables).

`nano .env`

7. Copy the set of variable below, and fill in your values where there are placeholders surrounded by < and > symbols.

```
## Database Values
POSTGRES_USER="postgres"
POSTGRES_PASSWORD="<a-l0n6-sr0n6-passw0rd-shou1d-60-here>"  # <- make  along strong password here.
POSTGRES_DB="get_healthy"

## Run the command below in the terminal to get a strong secret
JWT_SECRET="<use-openssl-to-make-this>" # openssl rnad -base64 32   <- use that command in the command line to create a key, then copy and paste it in

## Options are 'dev' and 'production'
NODE_ENV="production"

## Port for backend to run on
PORT=3000

## Frontend setings
FRONTEND_URL=https://<workout.yourdomain.net>  # <- replace with your domain name
VITE_ALLOWED_HOSTS="<workout.yourdomain.net>,localhost,127.0.0.1" # <- replace with your domain name
VITE_API_URL="/api"
```

Once done, you should not have the < and > signs in the variables, and the values should be replaced with your values.

Save with CTRL + O, press Enter, then use CTRL + X to exit the nano editor.

Pull Images and Start the Service

1. From inside the folder we created run:

`docker compose pull`

2. Next start the containers with:

```
docker compose up -d
```

3. If you want to view the logs after the containers start, you can do

```
docker compose logs -f
```

Once the server is up and running, you can quit following the logs with CTRL + C.

4. Open your favorite browser, and visit your new service. Use the IP and / or domain you added in the .env as an allowed VITE URL.

For instance, if you used workout.yourgreatdomain.net, then you'd visit that URL.

You should be greeted with the login page. Register as a new user, and the first user is set as the system admin.

Now you can begin setting up the system in Admin Settings, your profile.

Give the system a bit of time to setup the predefined templates if you don't see them right away.

Now you can start your journey to better health.

Support this Channel and Content

Become a Patron at Patreon

Check me out on Patreon

Be me a Coffee or Beer at Paypal

Paypal Support for Awesome Open Source

Get the Awesome Open Source Merch

Great Merch for Open Source Advocates!

Revision #3
Created 2026-08-03 12:34:42 UTC by Brian McGonagill
Updated 2026-08-03 16:36:38 UTC by Brian McGonagill