QuantumHive-Website Deployment Plan

This document outlines the setup of a single EC2-based deployment for the QuantumHive website using Docker, Traefik, and GitHub Actions CI/CD.

# 📁 Repository Structure

QuantumHive-Website/  
├── .github/  
│ └── workflows/  
│ └── deploy.yml # GitHub Actions CI/CD workflow  
├── frontend/  
│ ├── Dockerfile # Dockerfile for Next.js  
│ └── [Next.js app files]   
├── backend/  
│ ├── Dockerfile # Dockerfile for API  
│ └── [backend app files]  
├── traefik/  
│ ├── traefik.yml # Traefik config  
│ └── acme.json # Let's Encrypt cert cache (generated)  
├── docker-compose.yml  
└── README.md

# ⚙️ GitHub Actions Workflow (.github/workflows/deploy.yml)

name: Deploy to EC2  
  
on:  
 push:  
 branches: [main]  
  
jobs:  
 deploy:  
 runs-on: ubuntu-latest  
  
 steps:  
 - name: Checkout Code  
 uses: actions/checkout@v3  
  
 - name: Set up SSH  
 run: |  
 mkdir -p ~/.ssh  
 echo "${{ secrets.EC2\_SSH\_KEY }}" > ~/.ssh/id\_rsa  
 chmod 600 ~/.ssh/id\_rsa  
 ssh-keyscan -H ${{ secrets.EC2\_HOST }} >> ~/.ssh/known\_hosts  
  
 - name: Deploy via SSH  
 run: |  
 ssh ${{ secrets.EC2\_USER }}@${{ secrets.EC2\_HOST }} << 'EOF'  
 cd ~/QuantumHive-Website  
 git pull origin main  
 docker-compose down  
 docker-compose up -d --build  
 EOF

# 🐳 Docker Compose (docker-compose.yml)

version: '3.8'  
  
services:  
 traefik:  
 image: traefik:v2.10  
 command:  
 - --configFile=/etc/traefik/traefik.yml  
 ports:  
 - "80:80"  
 - "443:443"  
 - "8080:8080"  
 volumes:  
 - ./traefik/traefik.yml:/etc/traefik/traefik.yml  
 - ./traefik/acme.json:/letsencrypt/acme.json  
 - /var/run/docker.sock:/var/run/docker.sock  
 networks:  
 - web  
  
 frontend:  
 build: ./frontend  
 labels:  
 - "traefik.enable=true"  
 - "traefik.http.routers.frontend.rule=Host(`www.quantumhive.us`, `quantumhive.us`)"  
 - "traefik.http.routers.frontend.entrypoints=websecure"  
 - "traefik.http.routers.frontend.tls.certresolver=letsencrypt"  
 networks:  
 - web  
  
 backend:  
 build: ./backend  
 labels:  
 - "traefik.enable=true"  
 - "traefik.http.routers.backend.rule=Host(`api.quantumhive.us`)"  
 - "traefik.http.routers.backend.entrypoints=websecure"  
 - "traefik.http.routers.backend.tls.certresolver=letsencrypt"  
 networks:  
 - web  
  
networks:  
 web:  
 external: false

# 🔧 Traefik Config (traefik/traefik.yml)

entryPoints:  
 web:  
 address: ":80"  
 websecure:  
 address: ":443"  
  
api:  
 dashboard: true  
 insecure: true # Only for development  
  
providers:  
 docker:  
 exposedByDefault: false  
  
certificatesResolvers:  
 letsencrypt:  
 acme:  
 email: pkrhtdm@gmail.com  
 storage: /letsencrypt/acme.json  
 httpChallenge:  
 entryPoint: web

# 📦 Dockerfiles

## Frontend (frontend/Dockerfile)

FROM node:18-alpine  
WORKDIR /app  
COPY . .  
RUN yarn install && yarn build  
EXPOSE 3000  
CMD ["yarn", "start"]

## Backend (backend/Dockerfile)

FROM node:18-alpine  
WORKDIR /app  
COPY . .  
RUN yarn install  
EXPOSE 4000  
CMD ["yarn", "start"]

# 🔐 GitHub Secrets

Set the following secrets in GitHub (Repository → Settings → Secrets and Variables → Actions):  
  
- EC2\_SSH\_KEY: Private key content of your EC2 instance (PEM file)  
- EC2\_HOST: EC2 public IP or DNS  
- EC2\_USER: Usually 'ubuntu' (for Ubuntu AMIs)

# ✅ Final Notes

You may declare the following if needed:

- Your GitHub repository is connected to the EC2 instance with proper access.

- Ensure ports 80, 443, and 22 are open in EC2 security groups.

- Make sure domain DNS points to your EC2 Elastic IP.

- `acme.json` must have permission 600: `chmod 600 acme.json`