

Quick Reference Guide

Deploy Ghost CMS with Docker

Complete setup in 15 minutes — self-hosted

Mars 2026

Table of Contents

1. Prerequisites
2. Install Docker
3. Directory Structure
4. docker-compose.yml
5. config.production.json
6. Start and Verify
7. Cloudflare DNS Setup
8. SSL Reverse Proxy (Zoraxy / Nginx)
9. Maintenance Commands

1. Prerequisites

Before you start, make sure your server has:

- Ubuntu 20.04 LTS or later (Debian/Red Hat compatible)
- Root or sudo access on the target machine
- Docker Engine 24+ (see Section 2 for installation)
- Docker Compose v2 (via docker-compose-plugin)
- A domain name pointing to your server IP
- Ports 80 and 443 open in your firewall
- A Cloudflare account (or any DNS provider) for SSL

2. Install Docker

If Docker is not yet installed on your Ubuntu server:

```
sudo apt update && sudo apt upgrade -y
sudo apt install -y ca-certificates curl gnupg

# Add Docker's official GPG key
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg \
  | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

# Add Docker repo and install
echo "deb [arch=$(dpkg --print-architecture) \
  signed-by=/etc/apt/keyrings/docker.gpg] \
  https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo $VERSION_CODENAME) stable" \
  | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt update
sudo apt install -y docker-ce docker-ce-cli containerd.io \
  docker-buildx-plugin docker-compose-plugin

docker --version && docker compose version
```

i Official docs: <https://docs.docker.com/engine/install/ubuntu/>

3. Directory Structure

```
sudo mkdir -p /mnt/docker-data/ghost-blog/content

# Layout:
# /mnt/docker-data/ghost-blog/
#   +-- content/          <- Ghost data (images, themes, DB)
#   +-- docker-compose.yml
#   +-- config.production.json
```

```

version: '3.8'
services:
  ghost:
    image: ghost:5
    container_name: ghost-blog
    restart: unless-stopped
    ports:
      - "2368:2368"
    environment:
      NODE_ENV: production
    volumes:
      - ./content:/var/lib/ghost/content
      - ./config.production.json:/var/lib/ghost/config.production.json:ro

```

5. config.production.json

```

{
  "url": "https://blog.yourdomain.com",
  "server": { "port": 2368, "host": "0.0.0.0" },
  "database": {
    "client": "sqlite3",
    "connection": {
      "filename": "/var/lib/ghost/content/data/ghost.db"
    }
  },
  "mail": { "transport": "Direct" },
  "logging": { "transports": ["stdout"] },
  "process": "local",
  "env": "production"
}

```

i Replace blog.yourdomain.com with your actual domain.

6. Start and Verify

```

cd /mnt/docker-data/ghost-blog
docker compose up -d

docker ps | grep ghost-blog
docker logs -f ghost-blog
curl -s -o /dev/null -w '%{http_code}' http://localhost:2368

```

7. Cloudflare DNS Setup

1. Log in to dash.cloudflare.com and select your domain
2. DNS > Records > Add record
3. Type: A | Name: blog | IPv4: your-server-IP
4. Proxy status: Proxied (orange cloud) | TTL: Auto | Save

5. DNS propagation: 1-5 minutes with Cloudflare

8. SSL Reverse Proxy

Option A — Zoraxy (web UI)

1. Access your Zoraxy interface
2. Create rule: blog.yourdomain.com -> localhost:2368
3. Enable HTTPS with built-in Let's Encrypt

Option B — Nginx

```
server {
    listen 443 ssl;
    server_name blog.yourdomain.com;
    ssl_certificate      /etc/letsencrypt/.../fullchain.pem;
    ssl_certificate_key  /etc/letsencrypt/.../privkey.pem;
    location / {
        proxy_pass http://localhost:2368;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

9. Maintenance Commands

```
docker restart ghost-blog
docker logs --tail 50 ghost-blog
docker pull ghost:5 && docker compose up -d

# Backup the database
cp /mnt/docker-data/ghost-blog/content/data/ghost.db \
~/backups/ghost-$(date +%Y%m%d).db

docker compose down
```

Full article: blog.botum.ca/deploy-ghost-cms-docker

Website: www.botum.ca • contact@botum.ca •

— Canada