

# Restoring Nginx Proxy Manager from Backup

This guide walks you through restoring your **Nginx Proxy Manager (NPM)** Docker container from a backup on an OpenMediaVault system.

---

## ? Prerequisites

- OpenMediaVault system with Docker and Docker Compose installed
  - Backups of the following folders:
    - `/srv/dev-disk-by-uuid-2cae64a3-155c-4238-8a9d-99515c6f200f/Muninn/2025backup/nginx-proxy-manager-data`
    - `/srv/dev-disk-by-uuid-2cae64a3-155c-4238-8a9d-99515c6f200f/Muninn/2025backup/nginx-proxy-manager-letsencrypt`
  - Target restore directory:
    - `/srv/dev-disk-by-uuid-b588f7eb-f98f-4e15-b964-25574b92227e/HELCONFIG/nginx/`
- 

## ? Folder Structure

Ensure the target directory structure looks like this:

```
/srv/dev-disk-by-uuid-b588f7eb-f98f-4e15-b964-25574b92227e/HELCONFIG/  
├─ nginx/  
│   └─ nginx-proxy-manager-data/  
│       └─ nginx-proxy-manager-letsencrypt/
```

---

# ? Working Docker Compose File

Place the following `docker-compose.yml` in the `/srv/dev-disk-by-uuid-b588f7eb-f98f-4e15-b964-25574b92227e/HELCONFIG/nginx/` folder:

```
version: '3'
services:
  nginx-proxy-manager:
    image: jc21/nginx-proxy-manager:latest
    container_name: nginx-proxy-manager
    restart: unless-stopped
    ports:
      - '80:80'
      - '81:81'
      - '443:443'
    volumes:
      - ./nginx-proxy-manager-data:/data
      - ./nginx-proxy-manager-letsencrypt:/etc/letsencrypt
```

# ? Restore Backups

Use the following commands to copy your backup data into the live configuration folder:

```
# letsencrypt
sudo cp -a /srv/dev-disk-by-uuid-2cae64a3-155c-4238-8a9d-99515c6f200f/Muninn/2025backup/nginx-proxy-manager-letsencrypt/. \
  /srv/dev-disk-by-uuid-b588f7eb-f98f-4e15-b964-25574b92227e/HELCONFIG/nginx/nginx-proxy-manager-letsencrypt/

# data
sudo cp -a /srv/dev-disk-by-uuid-2cae64a3-155c-4238-8a9d-99515c6f200f/Muninn/2025backup/nginx-proxy-manager-data/. \
  /srv/dev-disk-by-uuid-b588f7eb-f98f-4e15-b964-25574b92227e/HELCONFIG/nginx/nginx-proxy-manager-data/
```

Ensure trailing `/.` is used to copy **contents** only.

---

## ? Start Container

Once restored:

```
cd /srv/dev-disk-by-uuid-b588f7eb-f98f-4e15-b964-25574b92227e/HELCONFIG/nginx/  
sudo docker compose up -d
```

---

## ? Troubleshooting

### ? Can't create network

```
Error: could not find an available, non-overlapping IPv4 address pool
```

**Fix:**

```
sudo docker network prune
```

### ? Container fails with alias error

```
Error: network-scoped alias is supported only for user-defined networks
```

**Fix:** Define and use a custom bridge network:

```
sudo docker network create nginx_network
```

Then add to `docker-compose.yml`:

```
networks:  
  default:  
    external:  
      name: nginx_network
```

### ? Can't access web interface

- Check container logs:

```
sudo docker logs nginx-proxy-manager
```

- Ensure ports are published correctly (`80`, `81`, `443`)
- Check firewall / router rules

## ? SSL Renewal Errors

```
expected ../cert.pem to be a symlink
```

These indicate broken or partial LetsEncrypt data. Copy the backup of the letsencrypt folder and ensure correct permissions.

---

## ? Login Defaults

Once online, access via:

```
http://[host-IP]:81
```

Default credentials:

```
Email:    admin@example.com
```

```
Password: changeme
```

## ? Notes

- Always stop the container before copying data
  - Use `docker compose down` if needed before restoring
  - Check `database.sqlite` timestamp to confirm restoration
- 

## ? Optional: Full Refresh

If issues persist, delete the container and volumes, then redeploy and restore the backups fresh:

```
sudo docker compose down
sudo rm -rf nginx-proxy-manager-data nginx-proxy-manager-letsencrypt
# Then restore backups and re-up
```

---

For any further help, ensure Docker is up-to-date and Portainer (if used) is refreshed to reflect changes.

---

Revision #4

Created 2025-05-28 04:08:57 UTC by Moose

Updated 2025-05-28 04:12:58 UTC by Moose