

core-system

- NGINX Proxy Manager
 - jwilder/nginx-proxy
 - portainer
-
- [NGINX Proxy Manager \[0.0.0.0:80/443\]](#)

NGINX Proxy Manager

[0.0.0.0:80/443]

Nginx proxy Manager from <https://nginxproxymanager.com/>

- ☐ essential core service which should be configured first
- ☐ requires docker already installed
- ☐ exposes open ports on the main server public IP
 - ☐ Port 80 listens and if matched to a valid hostname or wildcard, routes the traffic; all http should simply redirect to https
 - ☐ Port 443 listens and routes traffic dependiong on hostnames
 - ☐ *.pknw1.co.uk domains are uysd for publicly accessible services
 - ☐ *.admin.pknw1.co.uk domains are used for restricted access services
 - ☐ routed to an appropriate proxy host configuration based on DNS name
 - ☐ the FQDN SSL connection is verified against the system wide wildcard cert for either *.pknw1.co.uk* or *.admin.pknw1.co.uk*

tcp	0	0	149.202.72.112:80	0.0.0.0:*	LISTEN	948505/docker-proxy
tcp	0	0	149.202.72.112:443	0.0.0.0:*	LISTEN	948532/docker-proxy

Named Proxy Hosts in NGINX	catch any *.pknw1.co.uk	catch *.admin.pknw1.co.uk
<p>The preferred method of routing is by matching the request wirh the appropriate wildcard domain configuration</p> <p>all wildcard matches are validated against any access rules and then passed through to the nginx proxy which directs the request to the configured container</p> <p>This setuop allows the domain config to be done with the service container environment variables in an automated way rather than manually setting a proxy host and reverese proxy config for each new service</p>	<p>passed to the backend and routed via internal proxy;</p> <p>a manual config can also be added and further restricted</p> <p>all public containers should be on the proxy docker network</p> <p>172.22.20.1</p>	<p>matched as a wildcard for the admin domain</p> <p>before passing to the internal_proxy for hostname based routing, the source of the session is determined</p> <p>as this is admin only - only over tailscale, we configure any source with a local docker address 172.22.0.0/16 or from the Tailscale network 100.100.69.0/24 it is allowed</p> <p>any non matches are deny;</p>

Example wildcard proxy configuration for *.pknw1.co.uk

Screenshot 2025-06-21 at 21.25.04.png	Screenshot 2025-06-21 at 21.25.04.png	Screenshot 2025-06-21 at 21.25.04.png	Screenshot 2025-06-21 at 21.25.04.png
---------------------------------------	---------------------------------------	---------------------------------------	---------------------------------------

Example wildcard proxy configuration for *.admin.pknw1.co.uk

Screenshot 2025-06-21 at 21.25.57.png	Screenshot 2025-06-21 at 21.25.57.png	Screenshot 2025-06-21 at 21.25.57.png	Screenshot 2025-06-21 at 21.25.57.png
---------------------------------------	---------------------------------------	---------------------------------------	---------------------------------------

nginx proxy manager proxy host config files on disk

```
cd /etc/pknw1/config/core-system/nginx_proxy_manager/data/nginx/proxy_host

for f in $(ls *conf); do DOM=$(head -n2 $f | tail -n1); FN=$(echo $f | awk -F. '{print $1}');
echo $FN $DOM | sed 's/#//'; done
```

```
1 *.pknw1.co.uk
10 dev.pknw1.co.uk, yt-dev.pknw1.co.uk
12 search.pknw1.co.uk
15 mobile.engineering.pknw1.co.uk
16 webmin.pknw1.co.uk
18 engineering.pknw1.co.uk
2 *.admin.pknw1.co.uk
23 admin.pknw1.co.uk
25 jackett.pknw1.co.uk
26 *.streaming-from.cloud, streaming-from.cloud
28 accounts.pknw1.co.uk, my.pknw1.co.uk
3 webmin.admin.pknw1.co.uk
31 check.pknw1.co.uk
32 notflix.pknw1.co.uk, test.pknw1.co.uk
33 requests.pknw1.co.uk
35 tv.pknw1.co.uk
36 jf.pknw1.co.uk
37 hub.pknw1.co.uk
38 ut.admin.pknw1.co.uk, utorrent.admin.pknw1.co.uk
41 404.pknw1.co.uk
42 reddit_webhook.pknw1.co.uk, webhook.pknw1.co.uk
43 vue.pknw1.co.uk
6 login.pknw1.co.uk
```

The following manual configurations are required for mapping non-docker services via the proxy

Service Name	External URL	Mapped Service
Webmin	https://webmin.admin.pknw1.co.uk	<div>http port 80 -> 443</div> <div>443 -> webmin proxy host config</div> <div>SSL cert validated</div> <div>*.admin.pknw1.co.uk</div> <div>source checked via advanced config</div> <div>if allowed, direct the connection</div> <div>NPM -> proxy docker network -> 10000</div>

pre-installation

- ☐ docker must be installed and running
- ☐ docker compose must be installed and available
- ☐ docker networks need to be configured
 - ☐ proxy 172.22.20.0/24
 - ☐ admin 172.22.22.0/24
- ☐ host networks need to be configured
 - ☐ public IP 149.202.72.112/32
 - ☐ tailscale IP 100.100.69.2
 - ☐ internal_proxy is required for DNS routed access to teh admin URL
- ☐ the folders for the core-system group of containers is required /etc/pknw1.services/core-services/ for the docker-compose.yml file
- ☐ the folders for the application persistence are required /etc/pknw1.config/core-system.nginx_proxy_manager
- ☐ any other included file locations muist exist before start

Installation

as nginx proxy manager runs in a container under docker, "installation" requires

1. the docker compose file modiofied for this server
2. the contaner config folders and eother a blank start or restored config files from backup

npm_preinstall_checks.sh

```
#!/bin/bash -e

# pre-install script for nginx_proxy_manager deployment under docker

# [X] Checks docker and docker compose available

# [X] Checks required networks are available

# [X] Checks compose and config folders are available

#

which docker && echo "docker installed" || ( echo "docker is required - please install" &&
exit 255 )

docker compose version || ( echo "docker compose is not available - please install" && exit
255 )

docker network inspect proxy || ( echo "docker network: proxy missing" && exit 255 )

docker network inspect admin || ( echo "docker network: admin missing" && exit 255 )

ip -4 a | grep "149.202.72.112" || ( echo "public IP not available" && exit 255 )

ip -4 a | grep "100.100.69.2" || (echo "tailscale IP not available" && exit 255 )


[ -d /etc/pknw1/services/core-system ] && echo "compose folder exists" || (echo
"creating compose folder" && mkdir -p /etc/pknw1/services/core-system )

[ -f /etc/pknw1/services/core-system/docker-compose.yml ] && grep jc21/nginx-proxy-manager
/etc/pknw1/services/core-system/docker-compose.yml && echo "configured" || echo
"need to config"


[ -d /etc/pknw1/config/core-system/nginx_proxy_manager ] && echo "config folder exists" || (
echo "creating config folder" && mkdir -p /etc/pknw1/config/core-system/nginx_proxy_manager )

[ -d /etc/pknw1/config/core-system/nginx_proxy_manager/data ] && echo
"configuration data exists" || echo "no configuration detected - setting up fresh"
```

/etc/pknw1/services/core-system/docker-compose.yml

```
services:

  nginx_proxy_manager:

    image: jc21/nginx-proxy-manager:latest

    restart: unless-stopped
```

```
ports:

- 149.202.72.112:80:80

- 149.202.72.112:443:443

- 100.100.69.2:80:80

- 100.100.69.2:443:443

- 100.100.69.2:81:81

- 100.100.69.2:3389:3389

- 100.100.69.2:3128:3128

- 100.100.69.2:53:53

- 100.100.69.2:3129:3129

- 172.22.20.1:80:80

privileged: true

volumes:

-
/etc/pknwl/config/core-system/nginx_proxy_manager/98-themepark:/etc/cont-init.d/99-themepark

- /etc/pknwl/config/core-system/nginx_proxy_manager/data:/data

- /etc/pknwl/config/core-system/nginx_proxy_manager/data/override/conf.d:/etc/nginx/conf.d

- /etc/pknwl/config/core-system/nginx_proxy_manager/data/override/templates:/app/templates

- /etc/pknwl/config/core-system/nginx_proxy_manager/letsencrypt:/etc/letsencrypt

labels:

- "com.centurylinklabs.watchtower.enable=true"

container_name: nginx_proxy_manager

dns:

- 8.8.8.8

- 172.22.20.1

hostname: proxymanager

networks:

- proxy

- admin

environment:

- PUID=0

- PGID=0
```

```
- VIRTUAL_HOST=proxymanager.admin.pknw1.co.uk  
- VIRTUAL_PORT=81  
- VIRTUAL_PROTO=http
```

```
healthcheck:
```

```
test: ["CMD", "/usr/bin/check-health"]
```

```
interval: 60s
```

```
timeout: 30s
```

```
networks:
```

```
admin:
```

```
external: true
```

```
proxy:
```

```
external: true
```

/etc/pknw1/config/core-system/nginx_proxy_manager

to be completed

backup and restore of config

```
cd /tmp  
vi pre-install  
chmod +x pre-install  
./pre-install
```

```
cd /etc/pknw1/service/core-system  
vi docker-cmopose.yml  
docker compose config  
docker compose pull  
docker compose up -d && docker compose logs -f
```

nginx proxy manager typical startup logs (FULL)

```
-----  
| Nginx Proxy Manager theme.park Mod |  
useradd warning: npm's uid 0 outside of the UID_MIN 1000 and UID_MAX 60000 range.  
-----  
  
Variables set:  
'APP_FILEPATH'=/app/frontend/  
'TP_DOMAIN'=  
'TP_COMMUNITY_THEME'=  
'TP_SCHEME'=  
'TP_THEME'=  
  
No domain set, defaulting to theme-park.dev  
No scheme set, defaulting to https  
No theme set, defaulting to organizr  
-----  
| Adding the stylesheet to html files |  
-----  
  
Stylesheet set to organizr on /app/frontend/index.html  
  
Stylesheet set to organizr on /app/frontend/login.html  
  
› Configuring npm user ...  
› Configuring npm group ...  
› Checking paths ...  
› Setting ownership ...  
› Dynamic resolvers ...  
› IPv6 ...  
Enabling IPV6 in hosts in: /etc/nginx/conf.d  
- /etc/nginx/conf.d/include/resolvers.conf  
- /etc/nginx/conf.d/include/ssl-cache-stream.conf  
- /etc/nginx/conf.d/include/ssl-ciphers.conf  
- /etc/nginx/conf.d/include/ssl-cache.conf  
- /etc/nginx/conf.d/include/assets.conf  
- /etc/nginx/conf.d/include/ip_ranges.conf  
- /etc/nginx/conf.d/include/proxy.conf  
- /etc/nginx/conf.d/include/log.conf  
- /etc/nginx/conf.d/include/force-ssl.conf
```


- /etc/nginx/conf.d/include/letsencrypt-acme-challenge.conf
- /etc/nginx/conf.d/include/block-exploits.conf
- /etc/nginx/conf.d/production.conf
- /etc/nginx/conf.d/default.conf

Enabling IPV6 in hosts in: /data/nginx

- /data/nginx/redirection_host/10.conf
- /data/nginx/redirection_host/11.conf
- /data/nginx/redirection_host/7.conf
- /data/nginx/redirection_host/1.conf
- /data/nginx/redirection_host/2.conf
- /data/nginx/redirection_host/8.conf
- /data/nginx/redirection_host/9.conf
- /data/nginx/redirection_host/4.conf
- /data/nginx/custom/root-robots.conf
- /data/nginx/proxy_host/32.conf
- /data/nginx/proxy_host/26.conf
- /data/nginx/proxy_host/10.conf
- /data/nginx/proxy_host/31.conf
- /data/nginx/proxy_host/41.conf
- /data/nginx/proxy_host/38.conf
- /data/nginx/proxy_host/36.conf
- /data/nginx/proxy_host/33.conf
- /data/nginx/proxy_host/15.conf
- /data/nginx/proxy_host/37.conf
- /data/nginx/proxy_host/43.conf
- /data/nginx/proxy_host/3.conf
- /data/nginx/proxy_host/23.conf
- /data/nginx/proxy_host/1.conf
- /data/nginx/proxy_host/12.conf
- /data/nginx/proxy_host/2.conf
- /data/nginx/proxy_host/6.conf
- /data/nginx/proxy_host/16.conf
- /data/nginx/proxy_host/35.conf
- /data/nginx/proxy_host/28.conf
- /data/nginx/proxy_host/42.conf
- /data/nginx/proxy_host/18.conf
- /data/nginx/proxy_host/25.conf
- /data/nginx/default_host/site.conf
- /data/nginx/stream/3.conf

```
- /data/nginx/stream/1.conf
- /data/nginx/stream/6.conf
- /data/nginx/stream/5.conf
- /data/nginx/stream/4.conf
```

```
> Docker secrets ...
```

```
-----
- - - - -
| \ | | _ \ | \ | | | |
| \ | | |_) | | \ |
| | \ | _/ | | |
|_| \_|_| |_| |_|
-----
```

```
User: npm PUID:0 ID:0 GROUP:0
```

```
Group: npm PGID:0 ID:0
-----
```

```
> Starting nginx ...
```

```
> Starting backend ...
```

```
[6/21/2025] [7:24:00 PM] [Global ] > [ ] info Using Sqlite: /data/database.sqlite
```

```
[6/21/2025] [7:24:04 PM] [Migrate ] > [ ] info Current database version: none
```

```
[6/21/2025] [7:24:04 PM] [Global ] > [ ] debug CMD: [ -f
```

```
'/etc/letsencrypt/credentials/credentials-3' ] || { mkdir -p /etc/letsencrypt/credentials
```

```
2> /dev/null; echo 'dns_ovh_endpo
```

```
int = ovh-eu
```

```
dns_ovh_application_key = cb81d5c8327179df
```

```
dns_ovh_application_secret = f0893715412c7a54752c89441c9c5cf4
```

```
dns_ovh_consumer_key = c14b3c1e4723d77e341c5d7499b2a76c' >
```

```
'/etc/letsencrypt/credentials/credentials-3' && chmod 600
```

```
'/etc/letsencrypt/credentials/credentials-3'; }
```

```
[6/21/2025] [7:24:04 PM] [Global ] > [ ] debug CMD: [ -f
```

```
'/etc/letsencrypt/credentials/credentials-4' ] || { mkdir -p /etc/letsencrypt/credentials
```

```
2> /dev/null; echo 'dns_ovh_endpo
```

```
int = ovh-eu
```

```
dns_ovh_application_key = cb81d5c8327179df
```

```
dns_ovh_application_secret = f0893715412c7a54752c89441c9c5cf4
```

```
dns_ovh_consumer_key = c14b3c1e4723d77e341c5d7499b2a76c' >
```

```
'/etc/letsencrypt/credentials/credentials-4' && chmod 600
```

```
'/etc/letsencrypt/credentials/credentials-4'; }
```

```
[6/21/2025] [7:24:04 PM] [Global ] > □ debug      CMD: [ -f
'/etc/letsencrypt/credentials/credentials-5' ] || { mkdir -p /etc/letsencrypt/credentials
2> /dev/null; echo 'dns_ovh_endpo
int = ovh-eu
dns_ovh_application_key  = cb81d5c8327179df
dns_ovh_application_secret = f0893715412c7a54752c89441c9c5cf4
dns_ovh_consumer_key  = c14b3c1e4723d77e341c5d7499b2a76c' >
'/etc/letsencrypt/credentials/credentials-5' && chmod 600
'/etc/letsencrypt/credentials/credentials-5'; }
[6/21/2025] [7:24:04 PM] [Global ] > □ debug      CMD: [ -f
'/etc/letsencrypt/credentials/credentials-8' ] || { mkdir -p /etc/letsencrypt/credentials
2> /dev/null; echo 'dns_ovh_endpo
int = ovh-eu
dns_ovh_application_key  = cb81d5c8327179df
dns_ovh_application_secret = f0893715412c7a54752c89441c9c5cf4
dns_ovh_consumer_key  = c14b3c1e4723d77e341c5d7499b2a76c' >
'/etc/letsencrypt/credentials/credentials-8' && chmod 600
'/etc/letsencrypt/credentials/credentials-8'; }
[6/21/2025] [7:24:04 PM] [Global ] > □ debug      CMD: [ -f
'/etc/letsencrypt/credentials/credentials-9' ] || { mkdir -p /etc/letsencrypt/credentials
2> /dev/null; echo 'dns_ovh_endpo
int = ovh-eu
dns_ovh_application_key  = cb81d5c8327179df
dns_ovh_application_secret = f0893715412c7a54752c89441c9c5cf4
dns_ovh_consumer_key  = c14b3c1e4723d77e341c5d7499b2a76c' >
'/etc/letsencrypt/credentials/credentials-9' && chmod 600
'/etc/letsencrypt/credentials/credentials-9'; }
[6/21/2025] [7:24:04 PM] [Global ] > □ debug      CMD: [ -f
'/etc/letsencrypt/credentials/credentials-11' ] || { mkdir -p /etc/letsencrypt/credentials
2> /dev/null; echo 'dns_ovh_endp
oint = ovh-eu
dns_ovh_application_key  = cb81d5c8327179df
dns_ovh_application_secret = f0893715412c7a54752c89441c9c5cf4
dns_ovh_consumer_key  = c14b3c1e4723d77e341c5d7499b2a76c' >
'/etc/letsencrypt/credentials/credentials-11' && chmod 600
'/etc/letsencrypt/credentials/credentials-11'; }
[6/21/2025] [7:24:04 PM] [Global ] > □ debug      CMD: [ -f
'/etc/letsencrypt/credentials/credentials-12' ] || { mkdir -p /etc/letsencrypt/credentials
2> /dev/null; echo 'dns_ovh_endp
```

```

oint = ovh-eu
dns_ovh_application_key = cb81d5c8327179df
dns_ovh_application_secret = f0893715412c7a54752c89441c9c5cf4
dns_ovh_consumer_key = c14b3c1e4723d77e341c5d7499b2a76c' >
'/etc/letsencrypt/credentials/credentials-12' && chmod 600
'/etc/letsencrypt/credentials/credentials-12'; }
[6/21/2025] [7:24:04 PM] [Certbot ] > ► start Installing ovh...
[6/21/2025] [7:24:04 PM] [Global ] > □ debug CMD: . /opt/certbot/bin/activate &&
pip install --no-cache-dir acme==$(certbot --version | grep -Eo '[0-9](\.[0-9]+)')
certbot-dns-o
vh==$(certbot --version | grep -Eo '[0-9](\.[0-9]+)') && deactivate
[6/21/2025] [7:24:10 PM] [Certbot ] > ☒ complete Installed ovh
[6/21/2025] [7:24:10 PM] [Setup ] > □ info Added Certbot plugins ovh
[6/21/2025] [7:24:10 PM] [Setup ] > □ info Logrotate Timer initialized
[6/21/2025] [7:24:10 PM] [Global ] > □ debug CMD: logrotate /etc/logrotate.d/nginx-
proxy-manager
[6/21/2025] [7:24:10 PM] [Setup ] > □ info Logrotate completed.
[6/21/2025] [7:24:10 PM] [IP Ranges] > □ info Fetching IP Ranges from online
services...
[6/21/2025] [7:24:10 PM] [IP Ranges] > □ info Fetching https://ip-
ranges.amazonaws.com/ip-ranges.json
[6/21/2025] [7:24:10 PM] [IP Ranges] > □ info Fetching
https://www.cloudflare.com/ips-v4
[6/21/2025] [7:24:10 PM] [IP Ranges] > □ info Fetching
https://www.cloudflare.com/ips-v6
[6/21/2025] [7:24:10 PM] [SSL ] > □ info Let's Encrypt Renewal Timer
initialized
[6/21/2025] [7:24:10 PM] [SSL ] > □ info Renewing SSL certs expiring within 30
days ...
[6/21/2025] [7:24:10 PM] [IP Ranges] > □ info IP Ranges Renewal Timer initialized
[6/21/2025] [7:24:10 PM] [Global ] > □ info Backend PID 214 listening on port 3000
...
[6/21/2025] [7:24:10 PM] [SSL ] > □ info Completed SSL cert renew process
[6/21/2025] [8:24:10 PM] [SSL ] > □ info Renewing SSL certs expiring within 30
days ...
[6/21/2025] [8:24:10 PM] [SSL ] > □ info Completed SSL cert renew process
[6/21/2025] [8:44:10 PM] [Global ] > □ debug CMD: /usr/sbin/nginx -t -g "error_log
off;"
[6/21/2025] [8:44:10 PM] [Nginx ] > □ debug Deleting file:

```

/data/nginx/proxy_host/3.conf

[6/21/2025] [8:44:11 PM] [Global]] > [debug CMD: /usr/sbin/nginx -t -g "error_log off;"

[6/21/2025] [8:44:11 PM] [Global]] > [debug CMD: /usr/sbin/nginx -t -g "error_log off;"

[6/21/2025] [8:44:11 PM] [Nginx]] > [info Reloading Nginx

nginx proxy manager log indicators

| Nginx Proxy Manager theme.park Mod |

useradd warning: npm's uid 0 outside of the UID_MIN 1000 and UID_MAX 60000 range.

Variables set:

'APP_FILEPATH'=/app/frontend/

'TP_DOMAIN'='

'TP_COMMUNITY_THEME'='

'TP_SCHEME'='

'TP_THEME'='

Enabling IPV6 in hosts in: /data/nginx

? Docker secrets ...

| \ | | _ \ | \ |

| \ | | |_) | | \ | |

| | \ | _/ | | |

|_| _|_| |_| |_|

User: npm PUID:0 ID:0 GROUP:0

Group: npm PGID:0 ID:0

? Starting nginx ...

? Starting backend ...

[6/21/2025] [7:24:00 PM] [Global] > ? info Using Sqlite: /data/database.sqlite

[6/21/2025] [7:24:04 PM] [Migrate] > ? info Current database version: none

[6/21/2025] [7:24:04 PM] [Global] > ?

debug CMD: [-f '/etc/letsencrypt/credentials/credentials-3'] || { mkdir -p /etc/letsencrypt/c

int = ovh-eu

dns_ovh_application_key = cb81d5c8327179df

dns_ovh_application_secret = f0893715412c7a54752c89441c9c5cf4

dns_ovh_consumer_key = c14b3c1e4723d77e341c5d7499b2a76c' > '/etc/letsencrypt/credentials/credent

[6/21/2025] [7:24:10 PM] [Certbot] > ? complete Installed ovh

[6/21/2025] [7:24:10 PM] [Setup] > ? info Added Certbot plugins ovh

[6/21/2025] [7:24:10 PM] [Setup] > ? info Logrotate Timer initialized

[6/21/2025] [7:24:10 PM] [Global] > ?

debug CMD: logrotate /etc/logrotate.d/nginx-proxy-manager

[6/21/2025] [7:24:10 PM] [Setup] > ? info Logrotate completed.

[6/21/2025] [7:24:10 PM] [IP Ranges] > ? info Fetching IP Ranges from online services...

[6/21/2025] [7:24:10 PM] [IP Ranges] > ?

info Fetching https://ip-ranges.amazonaws.com/ip-ranges.json

[6/21/2025] [7:24:10 PM] [IP Ranges] > ? info Fetching https://www.cloudflare.com/ips-v4

[6/21/2025] [7:24:10 PM] [IP Ranges] > ? info Fetching https://www.cloudflare.com/ips-v6

[6/21/2025] [7:24:10 PM] [SSL] > ? info Let's Encrypt Renewal Timer initialized

[6/21/2025] [7:24:10 PM] [SSL] > ? info Renewing SSL certs expiring within 30 days ...

[6/21/2025] [7:24:10 PM] [IP Ranges] > ? info IP Ranges Renewal Timer initialized

[6/21/2025] [7:24:10 PM] [Global] > ? info **Backend PID 214 listening on port 3000 ...**

[6/21/2025] [8:44:11 PM] [Nginx] > ? **info Reloading Nginx**

NGINX Proxy Manager should now be accesible via the admin port (81) on the internal tailscale IP address

Admin console

[Screenshot 2025-06-21 at 22.41.58.png](#)

Wildcard Domains Proxy Host Setup

Wildcard Domains SSL Certs via Letsencrypt using DNS Challenge

	Screenshot 2025-06-21 at 22.43.32.png
--	---