

OVH [ns3024499.ip-149-202-72.eu]

- tailscale
 - webmin
 - docker
 - custom scripts
-
- [1. Vendor Setup](#)
 - [2. Initial Login](#)
 - [3. System Defaults](#)
 - [4. Core System Packages Config](#)
 - [Custom Scripts /usr/local/bin](#)
-
- [5. Core Application Services](#)
 - [Tailscale VPN](#)
 - [Webmin](#)
 - [Docker](#)
-
- [APT Sources](#)
 - [Tailscale APT Sources](#)
 - [Docker APT Sources](#)

1. Vendor Setup

2. Initial Login

[root user/secure/user defaults]

Once the server is deployed, a number of configuration steps are followed to ensure

- base install pre-installed packages are appropriate
- base install is configured for secure remote access for root user
- templating of the /etc/skel user for
 - pre-configure ssl access and keys
 - pre-configure login script
 - pre-configure sudo access and groups

3. System Defaults

- creation of basic folder structure
- installation of components required for folder merging via FUSE
- post-reboot tasks and disk mount automation `/etc/fstab`

4. Core System Packages Config

- SSH Config
- Tailscale
- fail2ban
- UFW

+ custom scripts for automation

4. Core System Packages Config

Custom Scripts /usr/local/bin

[illegible]


```
| Script Name | Description | Code |
|-----|-----|-----|
| | | |
./backup-compose-services.sh
./chmod-safe-downloads.sh
./ntfy.sh
./chmod-safe-media-downloads.sh
./create-docker-dev
./create-container-user.sh
./create-container-user.sh
./validate-pem.sh
./clear-logs.sh
./wipe
./docker-latest-images.sh
./earliest-file-tampstamp-to-parent-folder
./extract-ips.sh
./backup-compose-configs.sh
./uplog
./count-subdir-files.sh
./conf-zip
./commit-configs
./whois-ip.sh
./gh_repo_init
./renew-pkwnl-ssl.sh
./ffmpeg-wbem.sh
```



```
./bashbar

./slickslice

./plex-db-repair-tool.sh

./auto_shutdown_containers

./renew-int-ssl.sh

./youtube_links

./setReleaseDate

./pk-check-squid.sh

./make-release-year-playlists.sh

./renew-internal-ssl.sh

./mkv2mp4.sh

./iptables-allowip.sh

./lazy

./folder-backup

./create-dev-repo

./movies-compile-latest-movies.sh

./archive

./link-portainer

./auto_shutdown_webservice

./pipe-exec

./launch-venv.sh

./ytdl

./list_portainer_templates

./init_repo

./mkv2mp4

./renew-tailscale-ssl.sh

./docker-image-prune-3months.sh

./get_feature.sh

./renew-plex-ssl.sh

./launch_portainer_stack

./merge-iptv-xml.sh

./update-epg-xml.sh
```

```
./ddocker  
  
./webhook-listen.sh  
  
./ban-ip-port.sh  
  
./squid  
  
./wait-for-ssh.sh  
  
./chmod-safe.sh  
  
./webhook-processor.sh
```

5. Core Application Services

- docker
- XRDP

5. Core Application Services

Tailscale VPN

Tailscale VPN from <https://tailscale.com/>

- ☐ configures a host interface `tailscale0`
- ☐ installs auto-start for tailscale daemon `/etc/systemd/system/multi-user.target.wants/tailscaled.service`
- ☐ starts service at boot allocating IP address 100.100.69.2 to the tailscale0 nic
- ☐ attaches tailscale0 nic to the shared VPN
 - ☐ makes accessible 100.100.69.X addresses
 - ☐ makes the HOST available as an exit node

configured to use account pkswansea@outlook.com via the admin console via <https://login.tailscale.com/admin>

pngtree-banner-with-important-icon-vector-pictures-7826342-244107159.png	<p>The server SSH service running on port 69 is only exposed on the tailscale0 interface via the IP 100.100.69.2 once the daemon has started via <code>/etc/systemd/system/ssh-after-tailscale.service</code> and can only be accessed when connected to a valid VPN client</p>	<pre>#!/bin/bash while ! ip addr show tailscale0 grep -q "inet "; do sleep 10 done systemctl start ssh</pre>
--	---	--

5. Core Application Services

Webmin

Webmin from <https://webmin.com/>

- ☐ installed onto host system via 3rd party apt repository
- ☐ installs and auto-configured for start at boot-time
- ☐ OOB installation listens on all interfaces `https://<ip>:10000`
 - ☐ post install modify the `/etc/webmin/miniserv.conf`
 - ☐ we will only listen on internally accessible networks
 - ☐ we will disable SSL
 - ☐ we will reverse proxy via nginx proxy manager https to http:10000

```
Notable changes for /etc/webmin/miniserv.conf
```

```
port=10000
```

```
sockets=172.22.20.1:*
```

```
ssl=0
```

```
no_ssl2=1
```

```
bind=172.22.22.1
```

```
ipv6=0
```

```
no_tls1_1=1
```

```
webprefixnoredir=1
```

```
no_tls1=1
```

```
no_ssl3=1
```

Installation Steps

1. install webmin repo

```
wget -O - https://raw.githubusercontent.com/webmin/webmin/master/webmin-setup-repo.sh  
| sudo bash
```

2. update repo

```
sudo apt update
```

3. install webmin package

```
sudo apt install -y webmin
```

4. start and verify service

```
sudo systemctl status webmin  
sudo netstat -anp|grep 10000 | grep LISTEN | awk '{print $4}' | awk '{print  
"https://"$1}'
```

5. access initial webmin UI and login as root

[Screenshot 2025-06-23 at 21.07.46.png](#)

Installation CLI commands copy & Paste

```
wget -O - https://raw.githubusercontent.com/webmin/webmin/master/webmin-setup-repo.sh |  
sudo bash  
sudo apt update  
sudo apt install -yq webmin  
sudo systemctl status webmin  
sudo netstat -anp|grep 10000 | grep LISTEN | awk '{print $4}' | awk '{print "http://"$1}'  
| xargs -n1 open
```

Update OOB installation

We can update via the webmin UI to change a minimal set of options to use Webmin behind a local IP which we access via https proxy through the nginx proxy

For the following configuration to be enabled, follow the setup steps below

[image.png](#)

Pre-Setup Requirements

- ☐ Docker installed on HOST
- ☐ Docker networks configured on HOST
- ☐ Nginx Proxy Manager container setup as per [The NGINX Proxy Manager Install Guide](#)
- ☐ Access from your local machine to the server Tailscale IP address via Tailscale VPN

NGINX Proxy Host Configuration

Your NGINX Docker compose file should be setup to listen on your VPN (Tailscale) Server IP Address

- 100.100.69.2:80:80
- 100.100.69.2:443:443
- 100.100.69.2:81:81

We now setup an inbound host to listen on HTTP and HTTPS, setting the Domain Name and then routing traffic to one of the internal IP addresses that Webmin is listening on

Next you want to request an SSL certificate or use the wildcard cert that should be available; ensure to **enable Force SSL** so all connections are secure; as a final check, we setup the advanced nginx config to check source IP ranges - if the address is not local or VPN, it is denied

Screenshot 2025-06-23 at 21.57.23.png	Screenshot 2025-06-23 at 21.57.44.png	Screenshot 2025-06-23 at 21.57.51.png
---	---	---

Post Install Configuration (WebUI)

<div><input type="checkbox"/> Open Console</div> <div><input type="checkbox"/> Login as root</div>	Screenshot 2025-06-23 at 21.07.46.png
Post Login Error	<div>When loading using only the proxied address (https://webmin.admin.tld.com) it may redirect to https://webmin.admin.tld.co.uk:10000 - which will cause an error (as we should have blocked access externally to 10000) - simply remove the port from the URL and hit enter to load the page</div> <div>image.png</div>

<input type="checkbox"/> Open the Webmin Config Page	Screenshot 2025-06-23 at 21.17.38.png
<input type="checkbox"/> update IPs <input type="checkbox"/> leave internal IP <input type="checkbox"/> remove external <input type="checkbox"/> modify listen ports as required (only change if there are conflicts)	Screenshot 2025-06-23 at 21.10.48.png
<input type="checkbox"/> disable SSL as the NGINX proxy will receive the SSL connection and terminate it using HTTP internally (optional but easier) <input type="checkbox"/> Setup SSL Certs if you use SSL - use your *.admin wildcard SSL cert	Screenshot 2025-06-23 at 21.10.16.png
<input type="checkbox"/> Update the approved referer DNS names	Screenshot 2025-06-23 at 21.18.04.png

Docker

APT Sources

Tailscale APT Sources

```
#/bin/bash -e

GPG_URL=https://pkgs.tailscale.com/stable/ubuntu/$(lsb_release -cs).noarmor.gpg
GPG_KEYFILE=/usr/share/keyrings/tailscale-archive-keyring.gpg
APT_URL=https://pkgs.tailscale.com/stable/ubuntu/$(lsb_release -cs).tailscale-keyring.list
APT_LIST=/etc/apt/sources.list.d/tailscale.list


which lsb_release || apt install -yq lsb-release
echo [ -f ${GPG_KEYFILE} ] || curl -fsSL ${GPG_URL} | sudo tee ${GPG_KEYFILE} >/dev/null
echo curl -fsSL ${APT_URL} | sudo tee ${APT_LIST}
```

Docker APT Sources

```
#!/bin/bash -e

for pkg in docker.io docker-doc docker-compose docker-compose-v2 podman-docker containerd
runc; do sudo apt-get remove -y $pkg; done

sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /usr/share/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /usr/share/keyrings/docker.asc
sudo chmod a+r /usr/share/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-
compose-plugin
```