Installation

Alle install scripte
wget http://PXE-FILE-SERVER/config/fullinstaller.sh
curl -fsSL http://PXE-FILE-SERVER/config/fullinstaller.sh | bash
curl -fsSL http://PXE-FILE-SERVER/config/ncaio.sh | bash

- Install Base Software
- Install Webmin
- Install PHP
- Install Docker
- Install NodeJS
- Install DotNet 6
- Install WireGuard
- Install Redis
- Prometheus Exporter

Install Base Software

Base Software

```
#!/bin/bash
# Update and Upgrade
apt update && apt upgrade -y && apt full-upgrade -y
#All the base Stuff
apt-get install -y apt-show-versions apt-transport-https bc binutils bsdmainutils bzip2 ca-certificates curl file git
gpg gzip htop jq lib32gcc-s1 lib32stdc++6 libauthen-pam-perl libio-pty-perl libnet-ssleay-perl libpam-runtime
lsb-release openssl perl rsync screen software-properties-common sudo tar tmux unzip util-linux wget
```

Webserver + Datenbankserver

```
#!/bin/bash
# Update and Upgrade
apt update && apt upgrade -y && apt full-upgrade -y
#Apache + MariaDB
apt-get install -y apache2 libmariadb3 mariadb-client mariadb-server
```

Apache Module aktivieren:

```
a2dismod mpm_event
a2enmod core so watchdog http log_config logio version unixd access_compat alias auth_basic authn_core
authn_file authz_core authz_host authz_user autoindex cgi cgid deflate dir env filter headers mime mpm_prefork
negotiation php proxy proxy_connect proxy_fcgi proxy_http proxy_wstunnel reqtimeout rewrite setenvif
socache_shmcb ssl status xml2enc cgi
```

Qemu Agent (Nur wenn VM)

```
#!/bin/bash
#If VM guest additions
apt-get install -y qemu-guest-agent
```

ResolvConf (Mainly for WireGuard)

> **Nach der Installation von ResolvConf muss die Maschine neu gestartet werden, da sonst keine Domains auflösbar sind!**

```
#!/bin/bash
#If VM guest additions
apt-get install -y resolvconf
```

Install Webmin

Webmin

```
#!/bin/bash
apt-get update && apt-get upgrade -y
curl -o setup-repos.sh https://raw.githubusercontent.com/webmin/webmin/master/setup-repos.sh
sh setup-repos.sh -f
# Update apt list
apt-get update -y && apt-get upgrade -y
#install webmin
apt-get install webmin --install-recommends -y
```

Apache V-Host Config:

```
<VirtualHost **HOSTIP**:443>
    ServerName srv-v-**HOST**.jwebbi.de
    SSLEngine on
    SSLCertificateFile /etc/cloudflare/jwebbi.de.pem
    SSLCertificateKeyFile /etc/cloudflare/jwebbi.de.key
    ProxyPreserveHost Off
    ProxyPass / http://127.0.0.1:10000/
ProxyPassReverse / http://127.0.0.1:10000/
</VirtualHost>
```

Webmin:

In /etc/webmin/config

```
referers=srv-v-*changeme*.jwebbi.de
referer=1
webprefixnoredir=1
```

In /etc/webmin/miniserv.conf

```
bind=127.0.0.1
ipv6=0
host=srv-v-*changeme*.jwebbi.de
ssl=0
ssl_redirect=0
```

Install PHP

PHP Packete

```
#!/bin/bash
# add key and update apt list
echo "deb https://packages.sury.org/php/ $(lsb_release -sc) main" | tee /etc/apt/sources.list.d/sury-php.list
wget -qO - https://packages.sury.org/php/apt.gpg | apt-key add -
apt-get update && apt-get upgrade -y
PHPVERSION=8.1
apt-get install -y "php${PHPVERSION}" "libapache2-mod-php${PHPVERSION}" "php${PHPVERSION}-bcmath" "php${PHPVERSION}-bz2" "php${PHPVERSION}-cgi" "php${PHPVERSION}-cli" "php${PHPVERSION}-common" "php${PHPVERSION}-curl" "php${PHPVERSION}-fpm" "php${PHPVERSION}-gd" "php${PHPVERSION}-gmp" "php${PHPVERSION}-imap" "php${PHPVERSION}-intl" "php${PHPVERSION}-mbstring" "php${PHPVERSION}-mysql" "php${PHPVERSION}-opcache" "php${PHPVERSION}-readline" "php${PHPVERSION}-soap" "php${PHPVERSION}-xml" "php${PHPVERSION}-xmlrpc" "php${PHPVERSION}-zip"
```

Install Docker

Docker Install Commands

```
#!/bin/bash
# add key and update apt list
mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/debian $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list
> /dev/null
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

Installing ctop

```
sudo apt-get install ca-certificates curl gnupg lsb-release
curl -fsSL https://azlux.fr/repo.gpg.key | sudo gpg --dearmor -o /usr/share/keyrings/azlux-archive-keyring.gpg
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/azlux-archive-keyring.gpg]
http://packages.azlux.fr/debian \
  $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/azlux.list >/dev/null
sudo apt-get update
sudo apt-get install docker-ctop
```

Install NodeJS

NodeJS Install Script

```
#!/bin/bash
# add key and update apt list
curl -fsSL https://deb.nodesource.com/setup_19.x | bash - &&apt-get install -y nodejs
```

Install DotNet 6

```
#!/bin/bash
# Get key/dpkg
wget https://packages.microsoft.com/config/debian/10/packages-microsoft-prod.deb -O packages-microsoft-prod.deb
# Add key/dpkg
dpkg -i packages-microsoft-prod.deb
# Update and Upgrade
apt-get update && apt-get upgrade -y
# Install runtime
apt-get install -y dotnet-runtime-6.0
```

Install WireGuard

WireGuard

```
#!/bin/bash
# Update and Upgrade
apt update && apt upgrade -y && apt full-upgrade -y
#WireGuard Installation
apt-get install -y resolvconf sudo wireguard
```

> Zum autostarten mit wg-quick: "*systemctl enable wg-quick@CONFIG*"

> Umbedingt im Service File eintragen: " ***Before=apache2.service mariadb.service redis-server.service ssh.service*** "

Eigentliche Datei liegt unter /var/run/resolvconf/resolv.conf

Inhalt:

```
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
# 127.0.0.53 is the systemd-resolved stub resolver.
# run "resolvectl status" to see details about the actual nameservers.

nameserver 1.1.1.1
nameserver 1.0.0.1
options timeout:1 attempts:1 rotate
```

Wenn als intern.conf gespeichert:

```
nano /etc/wireguard/intern.conf && wg-quick up intern && systemctl enable wg-quick@intern
```

Haupt VPN fürs Management:

```
nano /etc/wireguard/main.conf && wg-quick up main && systemctl enable wg-quick@main
```

```
wg-quick@main.service
```

Monitoring VPN:

```
nano /etc/wireguard/monitor.conf && wg-quick up monitor && systemctl enable wg-quick@monitor && nano /lib/systemd/system/prometheus-node-exporter.service && systemctl daemon-reload
```

```
After=wg-quick@monitor.service
```

```
echo "ARGS=\"--collector.tcpstat --collector.network_route --collector.logind --collector.mountstats --collector.meminfo_numa --collector.processes --web.listen-address=\"172.26.77.:9100\"\"" > /etc/default/prometheus-node-exporter && systemctl restart prometheus-node-exporter
```

Install Redis

```
#!/bin/bash
# Update and Upgrade
apt update && apt upgrade -y && apt full-upgrade -y
#Redis Installation
curl -fsSL https://packages.redis.io/gpg | sudo gpg --dearmor -o /usr/share/keyrings/redis-archive-keyring.gpg
echo "deb [signed-by=/usr/share/keyrings/redis-archive-keyring.gpg] https://packages.redis.io/deb $(lsb_release
-cs) main" | sudo tee /etc/apt/sources.list.d/redis.list
apt-get update
apt-get install -y redis
```

Redis Config -> /etc/redis/redis.conf

Prometheus Exporter

Grundinstallation für Monitoring mit Prometheus und Grafana.

Base Exporter:

```
#!/bin/bash
apt-get install -y prometheus-node-exporter prometheus-process-exporter
```

Exporter for Apache Webserver:

```
#!/bin/bash
apt-get install -y prometheus-node-exporter prometheus-process-exporter prometheus-apache-exporter
```

Exporter for Mysql Datenbankserver:

```
#!/bin/bash
apt-get install -y prometheus-node-exporter prometheus-process-exporter prometheus-mysqld-exporter
```

Exporter for Redis Datenbankserver:

```
#!/bin/bash
apt-get install -y prometheus-node-exporter prometheus-process-exporter prometheus-redis-exporter
```

Exporter for IPMI:

```
#!/bin/bash
apt-get install -y prometheus-node-exporter prometheus-process-exporter prometheus-ipmi-exporter
```

Exporter for NGINX Webserver:

```
#!/bin/bash
apt-get install -y prometheus-node-exporter prometheus-process-exporter prometheus-nginx-exporter
```

Exporter for PostGRES Datenbankserver:

```
#!/bin/bash
apt-get install -y prometheus-node-exporter prometheus-process-exporter prometheus-postgres-exporter
```

Exporter for SNMP:

```
#!/bin/bash
apt-get install -y prometheus-node-exporter prometheus-process-exporter prometheus-snmp-exporter
```

Default Ports

Node Exporter	9100
MariaDB	9101
Redis	9102
Apache	9103

Firewall Regeln

Einzelne Exporter

Node Exporter

```
iptables -A INPUT -p tcp -m tcp -s 209.16.144.27/32 --dport 9100 -j ACCEPT && iptables -A INPUT -p tcp -m tcp --dport 9100 -j DROP
```

MariaDB Exporter

```
iptables -A INPUT -p tcp -m tcp -s 209.16.144.27/32 --dport 9101 -j ACCEPT && iptables -A INPUT -p tcp -m tcp --dport 9101 -j DROP
```

Redis Exporter

```
iptables -A INPUT -p tcp -m tcp -s 209.16.144.27/32 --dport 9102 -j ACCEPT && iptables -A INPUT -p tcp -m tcp --dport 9102 -j DROP
```

Node+Redis+MariaDB Zusammen

```
iptables -A INPUT -p tcp -m tcp -m multiport -s 209.16.144.27/32 -j ACCEPT --dports 9100,9101,9102 && iptables -A INPUT -p tcp -m tcp -m multiport -j DROP --dports 9100,9101,9102
```