Ubuntu

- Bra att ha paket
- Paketshantering
- Öka disk
- Ethtool
- Applikationer
    - Netbox
    - Technitium
    - Akovrado
    - Bookstack
    - Wireguard
- Verktyg
    - NTP
    - Routing och IP-adressering (netplan)
    - Certifikat via Lets Encrypt och Certbot
    - SNMP
    - rsyslog
    - nmap
    - Secure Copy Protocol (SCP)
    - FRRouting
- Rensning av filer
- NFS
- IPTables
- Advanced Packet Tool (APT)
- Användarhantering

Bra att ha paket

`net-tools` = Ex. "route", "mtr", "netstat" och andra nätverkskommandon
`inetutils-ping` = Installera ping

Paketshantering

Kräver sudo behörighet.

`sudo apt update` = Leta efter uppdateringar till installerade paket
`sudo apt upgrade` = Uppgradera installerade paket
`sudo apt autoremove` = Avinstallera onödiga paketsberoenden

Öka disk

Efter resize i ex. Proxmox, slå nedan:

```
sudo pvs
sudo pvresize /dev/sda3
df -h

# expand logical volume
sudo lvextend -r -l +100%FREE /dev/mapper/ubuntu--vg-ubuntu--lv

# resize partition
sudo growpart /dev/sda 3
```

Ethtool

Går att felsöka fysiska nic med ethtool. Ex. ethtool -t eth0 offline och ethtool --cable-test eth0

Applikationer

Applikationer

Netbox

Snabbstart med Docker, glöm inte att välja vilken port den ska lyssna på (default 8000):

```
git clone -b release https://github.com/netbox-community/netbox-docker.git
cd netbox-docker
tee docker-compose.override.yml <<EOF
version: '3.4'
services:
  netbox:
    ports:
      - 8000:8080
EOF
docker compose pull
docker compose up
```

Skapa adminanvändare: `docker compose exec netbox /opt/netbox/netbox/manage.py createsuperuser`

Applikationer

Technitium

DNS-server med möjlighet till att prenumerera på blocklistor. Sätts upp enkelt med Docker, följ Github-sida.

Sätt upp två stycken för redundans på olika Docker-hostar. På primära, skapa två stycken NS-records, en mot sig själv och en mot sin sekundär. Specifisera glue IP-adresser. Skapa en sekundär zon på sekundären och peka på primären, nu ska sekundären synkronisera med en zone transfer (AXFR) från primären.

Akovrado

Akovrado är en open source netflow-lösning. Sätts upp med docker.

Systembeskrivning

Akovrado består av tre komponenter:

- inlet service, denna tar emot flows från netflow exportrar och kontaktar exportrar via SNMP.
- orchestrator, konfigurerar interna och externa komponenter.
- console, hanterar access från användare, web-gui.

Konfiguration

Systeminstallation

Följ instruktioner här, modifiera docker-compose.yml fil som man vill. Ta bort demo-data.

Systemkonfiguration

Systemet konfigureras genom att redigera yaml-filer i `akvorado/config`.

Länkar

Github

demo.akovrado.net

Applikationer

Bookstack

Går att installera direkt på en Linux-server eller med docker.

Installationsplats

Vid vanlig installation på Ubuntu-server så installeras allt i `/var/www/bookstack`.

Används MySQL databas så finns den informationen i `/var/lib/mysql/bookstack`.

Backup

Backup av databasen gör enklast med mysqldump:

```
# Syntax
## Only specify the `-p` option if the user provided has a password
mysqldump -u {mysql_user} -p {database_name} > {output_file_name}



# Example
mysqldump -u benny bookstack > bookstack.backup.sql

# Om root mysql-användaren används
sudo mysqldump -u root bookstack > bookstack.backup.sql
```

Applikationer

Wireguard

Server

Jag har följt den här guiden för uppsättning av Wireguard.

Installera först med `sudo apt install wireguard`.

För att inte NAT:a bakom serverns egna IP kan följande tas bort från WG-konfen: `iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE`

Konfigurationsfil på Wireguard-serven är `/etc/wireguard/wg0.conf`. Observera att syntax-fel i filen gör att tjänsten inte startar.

Verifiera att tjänsten är igång med `sudo systemctl status wg-quick@wg0`.

För att se anslutna klienter, slå `sudo wg`.

Klient

På Linux-klienten finns även konfen i /etc/wireguard/wg0.conf.

Om man använder network manager för att hantera nätverk (ex. på Fedora) och man vill stänga av, eller sätta upp, automatisk uppkoppling av WG gör man det med:

```
nmcli connection show
nmcli connection modify CON_NAME autoconnect no
```

Erfarenheter

Vid uppgradering till Ubuntu 26 så slutade Wireguard att fungera. Det visade sig vara IP-forwarding som hade stängts av:

```
! För att temporärt aktivera
jehrlander@wireguard:~$ sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 0
jehrlander@wireguard:~$ sudo sysctl -w net.ipv4.ip_forward=1
[sudo: authenticate] Password:
net.ipv4.ip_forward = 1

! Permanent
echo 'net.ipv4.ip_forward=1' | \
  sudo tee /etc/sysctl.d/99-wireguard-forward.conf

sudo sysctl --system
```

Verktyg

Verktyg

NTP

Installera paket:

```
sudo apt-get update
sudo apt-get install ntp
```

Ändra i fil och lägg till önskade servrar, iburst gör att synkronisering blir snabbare vid uppstart:

```
sudo vim /etc/ntp.conf
pool 0.server1.ntp.org iburst
pool 1.server2.ntp.org iburst
```

Starta om tjänst och lägg till uppstart av tjänst vid boot:

```
sudo systemctl restart ntp
sudo systemctl enable ntp
```

Verifiera ntp:

```
ntpq -p
```

Verktyg

Routing och IP-adressering (netplan)

Ubuntu använder sig av netplan.

Konfiguration sker i YAML-fil, `/etc/netplan/00-installer-config.yaml`. DHCP eller fast adress pekas ut på samma nivå. DNS-servrar pekas ut här.
Exempelkonfiguration med fast IPv4-adress och DHCPv6:

```
network:
  ethernets:
    ens18:
      dhcp6: true
      addresses:
      - 10.1.1.12/30
      nameservers:
        addresses:
        - 10.0.1.11
        search:
        - example.net
      routes:
      - to: 0.0.0.0/0
        via: 10.0.1.11
  version: 2
```

Efter konfigurering av fil slå kommandot `sudo netplan apply`. Verifiering av YAML-filen sker innan den installeras och installeras ej vid misslyckad validering.

Ex. konf med fast IPv4 och fast IPv6 adress:

```
network:
  ethernets:
    ens18:
      addresses:
      - 10.10.12.14/30
```

```
      - 2001:db8::1/128
      nameservers:
        addresses:
        - 10.10.22.13
        search:
        - example.net
      routes:
      - to: default
        via: 10.10.12.13/30
      - to: "::/0"
        via: fe80::d676:a0ff:fe6b:e587
        on-link: true
  version: 2
```

Det går även att byta namn på nätverksinterface i netplan. Se konf nedan, mac-adress har tagits fram med `lshw -C network` där serial är mac-adress. Här blir alltså det nya namnet eth0.

```
network:
  ethernets:
    ens18:
      addresses:
      match:
        macaddress: aa:bb:cc:dd:ee:ff
      set-name: eth0
```

Verktyg

Certifikat via Lets Encrypt och Certbot

1. Först behövs snapd. Kommer per default med Ubuntu. Kontrollera installerade snap paket med `snap list`
2. Ta bort eventuella installation av certbot-auto och Certbot OS med `sudo apt-get remove certbot`
3. Installera certbot med `sudo snap install --classic certbot`
4. Förbered maskinen så att certbot-kommandot kan köras: `sudo ln -s /snap/bin/certbot /usr/bin/certbot`
5. Få ett certifikat och uppdatera apache-konf med `sudo certbot --apache`, alternativt få enbart ett certifikat och uppdatera apache manuellt med `sudo certbot certonly --apache`
6. När du har fått ett certifikat, gör en backup av apache-konfen och skapa en ny apache-konffil: `sudo mv /etc/apache2/sites-available/bookstack.conf /etc/apache2/sites-available/bookstack.conf.old`, `sudo vim /etc/apache2/sites-available/bookstack.conf`
7. Ändra relevanta värden och skapa en ny apache-konfiguration (vhost):

```
<VirtualHost *:80>
    ServerName YOUR-DOMAIN-HERE
    RewriteEngine On
    RewriteRule ^(.*)$ https://%{HTTP_HOST}$1 [R=301,L]
</VirtualHost>

<VirtualHost *:443>
ServerName YOUR-DOMAIN-HERE
ServerAdmin webmaster@localhost
DocumentRoot /var/www/bookstack/public/

    SSLEngine on
    SSLCertificateFile      /etc/letsencrypt/live/YOUR-DOMAIN-HERE/fullchain.pem
    SSLCertificateKeyFile   /etc/letsencrypt/live/YOUR-DOMAIN-HERE/privkey.pem

    <Directory /var/www/bookstack/public/>
        Options Indexes FollowSymLinks
        AllowOverride None
```

```
        Require all granted
        <IfModule mod_rewrite.c>
            <IfModule mod_negotiation.c>
                Options -MultiViews -Indexes
            </IfModule>
            RewriteEngine On
            # Handle Authorization Header
            RewriteCond %{HTTP:Authorization} .
            RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]
            # Redirect Trailing Slashes If Not A Folder...
            RewriteCond %{REQUEST_FILENAME} !-d
            RewriteCond %{REQUEST_URI} (.+)/$
            RewriteRule ^ %1 [L,R=301]
            # Handle Front Controller...
            RewriteCond %{REQUEST_FILENAME} !-d
            RewriteCond %{REQUEST_FILENAME} !-f
            RewriteRule ^ index.php [L]
        </IfModule>
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

8. Aktivera apaches SSL modul: `sudo a2enmod ssl`
9. Verifiera om det finns några fel med `sudo apachectl configtest` och åtgärda vid behov
10. Redigera bookstacks .env-fil: `sudo vim /var/www/bookstack/.env`, ändra `APP_URL` till HTTPS istället för HTTP
11. Slutligen, starta om apache2! `sudo systemctl restart apache2`

Skapa certifikat manuellt

Först behövs ett värde för en dns-01 challenge:

```
sudo certbot certonly --manual --manual-auth-hook /etc/letsencrypt/acme-dns-auth.py --preferred-challenges dns --debug-challenges -d \*.your-domain -d your-domain
```

Byt ut värden ovanför till det som önskas. Exemplet ber om ett wildcard. Man kommer behöva lägga till ett värde i sin externa DNS-zon för validering. Certifikaten hamnar sedan i en undermapp

till `/etc/letsencrypt/live/`.

Certifikat som har hämtats med certbot kan kontrolleras med `sudo certbot certificates`. Certifikat kan tas bort med `sudo certbot delete`.

Vid förnyelse hade jag problem med att få ut en ny TXT-sträng för DNS validering. Borttagande av certifikat och skapande med nedan syntax för att få TXT värdet fungerade:

```
sudo certbot certonly --manual --preferred-challenges=dns --email admin@jehrlander.net --server https://acme-v02.api.letsencrypt.org/directory --agree-tos -d jehrlander.net -d *.jehrlander.net

Saving debug log to /var/log/letsencrypt/letsencrypt.log
Requesting a certificate for jehrlander.net and *.jehrlander.net

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Please deploy a DNS TXT record under the name:

_acme-challenge.jehrlander.net.

with the following value:

xxxxxx

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Verktyg

SNMP

SNMPv2

1. Installera SNMP `sudo apt install snmpd snmp libsnmp-dev`
2. Gör en kopia av originalkonf-filen `sudo cp /etc/snmp/snmpd.conf /etc/snmp/snmpd.conf.bak`
3. Redigera konf-filen, `sudo vi /etc/snmp/snmpd.conf`, ändra `agentAddres` till enbart tillåtna IP-adresser, ex. `agentAddress udp:192.168.1.5:161`
4. Ändra SNMP-communities `public` från `rocommunity public` och `rocommunity6 public` till önskat värde.
5. Starta om tjänsten med `sudo service snmpd restart`

Ändra även följande värden i snmpd.conf till något passande:

```
sysLocation    Sitting on the Dock of the Bay
sysContact     Me <me@example.org>
```

SNMPv3

Verktyg

rsyslog

rsyslog är installerat per default. rsyslog är en Linux-tjänst som hanterar skickande och/eller mottagande av syslog. Saknas den så installera med `apt install rsyslog`.

Redigera filen `/etc/rsyslog.conf`. Lägg här till en rad i stil med *.* @syslog.domain.net:514. rsyslog använder sig av TCP 514.

Vid förändringar i filen måste tjänsten startas om med `sudo service rsyslog restart`. Kontrollera tjänstens status med `sudo service rsyslog status`.

Exempelkonfiguration, skicka loggfiler

Här är en exempelkonfiguration på hur rsyslog kan skicka loggfiler över UDP 514.

```
! Lägg först till nedan modul i /etc/rsyslog.conf
module(load="imfile" PollingInterval="10")

! Skapa ny konffil /etc/rsyslog.d/ för applikationen
# Definiera en mall för syslog
template(name="TraditionalFormatWithHost" type="string" string="%timegenerated% %HOSTNAME%
%syslogtag%%msg%\n")

# Lägg till inputs för dina loggfiler från olika mappar
input(type="imfile"
      File="/data/docker/dns1/dnsserver_config/_data/logs/*.log"
      Tag="app-dns1"
      Severity="info"
      Facility="local0")

input(type="imfile"
      File="/data/docker/ntp1/_data/logs/*.log"
      Tag="app-ntp1"
```

```
      Severity="info"
      Facility="local0")

# Dynamisk vidarebefordran till Graylog för alla taggar som börjar med "app-"
if $programname startswith 'app-' then {
    action(
        type="omfwd"
        Target="graylog-docker.jehrlander.net"
        Port="514"
        Protocol="udp"
        template="TraditionalFormatWithHost"
    )
    stop
}

! Apparmor blockerar per default tillgång till mapparna för rsyslog. Den behöver ha åtkomst till överliggande mappar. Lägg till mapparna i  /etc/apparmor.d/usr.sbin.rsyslogd, ex:
/data/docker**
/data/docker/dns1/dnsserver_config/_data/logs/* r,

! Ladda om apparmor därefter
sudo apparmor_parser -r /etc/apparmor.d/usr.sbin.rsyslogd
```

Verifiera med `sudo tail -f /var/log/syslog | grep rsyslog`. Starta om rsyslog med `sudo systemctl restart rsyslog` vid behov och förändringar.

Verktyg

nmap

Ett verktyg för att scanna nät, använd med ex. `nmap -v 192.168.171.0/24` för att skanna det nätet.

Flaggor:

-v	verbose
-r	utan DNS uppslag
-R	alltid med DNS uppslag

Verktyg

Secure Copy Protocol (SCP)

Filöverföringar över SSH (tcp 22).

Användning

Enkelt exempel: scp LOKALT-FILNAMN.txt destination-fqdn.jehrlander.net:mapp/remote-filnamn.txt

Finns många bra flaggor:
-r = Recursive, skicka allt i nuvarande katalog
-l = Speca användarnamn, men går också att göra med användarnamn@destination-fqdn.jehrlander.net
-p = Ändra destinationsport från default 22 till något annat
-O = O för Old. Använd för klassisk SCP, mot ex. allt

Verktyg

FRRouting

FRRouting (FRR) är en open source mjukvara för att utföra routing på Linux-plattformar. Se deras hemsida här.

När FRR är installerat är CLI:t ganska likt Ciscos. Slå `sudo vtysh` för att komma till CLI:t.

Exempelkonfiguration:

```
jehrlander@ubuntu:~$ sudo vtysh

Hello, this is FRRouting (version 8.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

reverse-proxy# show run
Building configuration...

Current configuration:
!
frr version 8.1
frr defaults traditional
hostname ubuntu
log syslog informational
service integrated-vtysh-config
!
ip route 10.10.2.0/24 Null0
!
router ospf
 redistribute kernel
 redistribute static
 redistribute table
 redistribute vnc
 network 10.0.0.0/8 area 0
exit
!
router ospf6
```

```
exit
!
end
```

Rensning av filer

Rensning av stora filer

Om man som jag inte har jättemycket utrymme att leka med kan det vara bra att rensa filer ibland. Vissa loggfiler kan bli otroligt stora.

Hitta stora filer och sortera dem med: `sudo find / -size +50M -type f -exec du -h {} \; | sort -n`

Önskar man exkludera sökvägar från kommandot ovan kan man använda någon av varianterna nedanför, den översta exkluderar en sökväg och den andra flera:

```
sudo find / \
  -path /proc -prune -o \
  -size +50M -type f -exec du -h {} \; | sort -n


sudo find / \
  \( -path /proc -o -path /sys -o -path /dev \) -prune -o \
  -size +50M -type f -exec du -h {} \; | sort -n
```

Behöver man städa journalctl-loggar går det med följande kommandon:

```
journalctl --vacuum-size=100M


journalctl --vacuum-time=10d
```

Man bör även köra fstrim ibland. fstrim tar bort oanvända block på ett filsystem. Det gör att överliggande virtualisering (ex. Proxmox) får tillbaka diskutrymme. `sudo systemctl start fstrim.service`. discard ska även vara aktiverat på diskarna i Proxmox för att kunna få tilllbaka data.

Rensning av viss filtyp

Hitta först filerna, i det här exemplet .mru filer:

`find . -name "*.m3u" -type f`

Verifiera att det är rätt filer. Ta sedan bort dem genom att slänga på `-delete:`

`find . -name "*.m3u" -type f -delete`

NFS

Klient

Installera paket för att vara NFS-klient, `sudo apt install nfs-common`.

För att testa kan man använda kommandot mount.

Mounte:a sen mappen: `sudo mount host_ip:/var/nfs/general /nfs/general`. Observera att mappen som man mountar i måste skapas lokalt först med mkdir.

Verifiera att mappen är mountad med `df -h`.

Önskar man ha NFS-montering permanent behöver man uppdatera `/etc/fstab`.

Ex:

```
asustor.jehrlander.net:/volume1/testmapp   /mnt/testmapp   nfs   defaults,_netdev   0   0
```

Efter tillägg i fstab behöver mount ske med `sudo mount -a`.

Skulle NFS-sessionen blvit stale av någon anledning får man avmounta med `umount -f /sök/väg` och sedan mounta på nytt.

Om en applikation, ex. Docker, är beroende av NFS för att fungera korrekt så är det en bra idé att servicen för docker väntar på att NFS finns innan den drar igång.

Först behöver man lägga till flaggan bg i /etc/fstab. Ex:

```
asustor.jehrlander.net:/volume1/testmapp   /mnt/testmapp   nfs   defaults,bg,_netdev   0   0
```

Starta sen om tjänster:

`systemctl daemon-reload` followed by:
`systemctl restart remote-fs.target` and
`systemctl restart local-fs.target`

Hitta sedan tjänsten som monterad NFS-katalogen med `systemctl list-units -t mount`. Den heter oftast `mapp-mapp.mount` baserad på var den är monterad, med streck istället för snedsträck.

Redigera sedan tjänsten med `systemctl edit docker`.

Lägg till:

```
[Unit]
Requires = data-mapp.mount
After = data-mapp.mount
```

IPTables

Brandväggshantering för Linux.

Verifiera version:

```
jehrlander@docker-mgmt:~$ sudo iptables --version
iptables v1.8.10 (nf_tables)
```

Kontrollera regler:

```
sudo iptables -L -v -n
```

Advanced Packet Tool (APT)

APT är Debians och Debian-baserade Linux-OSs pakethanterare.

Avinstallera paket

Avinstallera paket med `sudo apt purge PAKETNAMN`. Kör sedan `sudo apt autoremove` för att ta bort oanvända beroenden.

APT tar inte bort konfigurationsfiler på servern, de får man ta bort manuellt.

APT tar bort paket som används

I mitt fall försökte `apt autoremove` ta bort Docker efter att jag hade ominstallerat docker.

`apt autoremove` tar bort paket som:

- installerades **automatiskt** som beroenden
- **inte längre krävs** av något annat installerat paket

Med kommandot `apt-mark showauto` kan man se dessa. För att dessa ej ska tas bort av APT kan man markera dessa som manuellt installerade med `sudo apt-mark manual PAKETNAMN`. Verifiera sen igen med `apt-mark showauto` (gärna med en grep på slutet).

Användarhantering

Skapa användare

Användare skapas med kommandot `useradd`. Ex:

`sudo useradd -m -s /bin/bash TESTUSER`

- `-m`: Skapar en hemkatalog för användaren.
- `-s /bin/bash`: Sätter standardskalet till Bash.

Lösenord sätts med kommandot `sudo passwd TESTUSER`. I en interaktiv ruta får man fylla i lösenordet.

Ska användaren har sudo-möjlighet ska den läggas till i sudo-gruppen. Det görs med `sudo usermod -aG sudo TESTUSER`.'

En SSH-mapp ska skapas och en SSH-nyckel kan läggas till om man så önskar:

```
sudo mkdir -p /home/TESTUSER/.ssh
sudo chmod 700 /home/TESTUSER/.ssh
sudo touch /home/TESTUSER/.ssh/authorized_keys
sudo chmod 600 /home/TESTUSER/.ssh/authorized_keys
! Lägg till SSH-nyckel
sudo chown -R TESTUSER:TESTUSER /home/TESTUSER/.ssh
```

Om man vill ha sudo utan lösenord: `echo "TESTUSER ALL=(ALL) NOPASSWD:ALL" | sudo tee -a /etc/sudoers`

Det går även att göra med `sudo visudo`, vilket är ett säkrare sätt att redigera sudo-filen. Lägg till samma rad:

`TESTUSER ALL=(ALL) NOPASSWD:ALL`

Verifiera

Verifiera användaren med `id`-kommandot.

```
jehrlander@docker-mgmt-2:~$ id TESTUSER
uid=997(TESTUSER) gid=997(TESTUSER) groups=997(TESTUSER),27(sudo)
```

Det går även att verifiera genom att innehållet i filen `/etc/passwd`.

```
jehrlander@docker-mgmt-2:~$ cat /etc/passwd | grep TESTUSER
TESTUSER:x:997:997::/home/TESTUSER:/bin/bash
```