Verktyg

- NTP
- Routing och IP-adressering (netplan)
- Certifikat via Lets Encrypt och Certbot
- SNMP
- rsyslog
- nmap
- Secure Copy Protocol (SCP)
- FRRouting

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
```

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
```

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

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

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

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.

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

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

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
```