

Linux Server

D. Leeuw

16 februari 2023

v.0.1.0



© 2021-2023 Dennis Leeuw

Dit werk is uitgegeven onder de Creative Commons BY-NC-SA Licentie en laat anderen toe het werk te kopiëren, distribueren, vertonen, op te voeren, en om afgeleid materiaal te maken, zolang de auteurs en uitgever worden vermeld als maker van het werk, het werk niet commercieel gebruikt wordt en afgeleide werken onder identieke voorwaarden worden verspreid.

Over dit Document

Dit document behandelt Linux voor het middelbaar beroepsonderwijs in Nederland, maar kan breder ingezet worden, daar het gericht is op het behalen van het LPI Linux Essentials examen. De doelgroep is niveau 4 van het MBO, met enige kennis van computers.

Versienummering

Het versienummer van elk document bestaat uit drie nummers gescheiden door een punt. Het eerste nummer is het major-versie nummer, het tweede nummer het minor-versienummer en de laatste is de nummering voor bug-fixes.

Om met de laatste te beginnen als er in het document slechts verbeteringen zijn aangebracht die te maken hebben met type-fouten, websites die niet meer beschikbaar zijn, of kleine foutjes in de opdrachten dan zal dit nummer opgehoogd worden. Als docent of student hoeft je je boek niet te vervangen. Het is wel handig om de wijzigingen bij te houden.

Als er flink is geschreven aan het document dan zal het minor-nummer opgehoogd worden, dit betekent dat er bijvoorbeeld plaatjes zijn vervangen of geplaatst/weggehaald, maar ook dat paragrafen zijn herschreven, verwijderd of toegevoegd, zonder dat de daadwerkelijk context is veranderd. Een nieuw cohort wordt aangeraden om met deze nieuwe versie te beginnen, bestaande cohorten kunnen doorwerken met het boek dat ze al hebben.

Als het major-nummer wijzigt dan betekent dat dat de inhoud van het boek substantieel is gewijzigd om bijvoorbeeld te voldoen aan een nieuw kwalificatiedossier voor het onderwijs of een nieuwe versie van Linux Essentials van de LPI. Een nieuw major-nummer betekent bijna altijd voor het onderwijs dat in het nieuwe schooljaar men met deze nieuwe versie aan de slag zou moeten gaan. Voorgaande versies van het document zullen nog tot het einde een schooljaar onderhouden worden, maar daarna niet meer.

Document ontwikkeling

Het doel is door middel van open documentatie een document aan te bieden aan zowel studenten als docenten, zonder dat hier hoge kosten aan verbonden zijn en met de gedachte dat we samen meer weten dan alleen. Door samen te werken kunnen we meer bereiken.

Bijdragen aan dit document worden dan ook met alle liefde ontvangen. Let u er wel op dat materiaal dat u bijdraagt onder de CC BY-NC-SA licentie vrijgegeven mag worden, dus alleen origineel materiaal of materiaal dat al vrijgegeven is onder deze licentie.

De eerste versie is geschreven voor het ROC Horizon College.

Versienummer	Auteurs	Verspreiding	Wijzigingen
0.1.0	Dennis Leeuw	Wim Onrust	Initieel document
0.2.0	Dennis Leeuw	HEITO18IB-A	Toegevoegd: versie-nummering, de shell, begin van werken met bestanden

Tabel 1: Document wijzigingen

Inhoudsopgave

Over dit Document	i
1 Inleiding	1
2 Debian installatie	3
3 Werken met daemons	9
3.1 init en systemd	9
3.2 Je eigen startup script	10
4 Logging	13
4.1 Log rotation	13
4.2 syslog	14
4.3 journalctl	15
5 Monitoring services	17
5.1 Proces monitoring	17
5.2 Daemons en netwerken	18
6 Printing	21
6.1 CUPS	21
7 Wireless	23
8 SMARTd	25
9 Remote management	27
9.1 SSH	27
9.2 Ansible	27
9.3 Puppet	27
9.4 SaltStack	27

10 Linux in een Windows Domain	29
10.1 SaMBa	29
11 Mailserver	31
11.1 postfix	31
12 LAMP	33
12.1 Apache	33
12.2 MariaDB	33
12.3 PHP	34
13 NextCloud	35
13.1 NextCloud installatie	36
Index	37

Hoofdstuk 1

Inleiding

Deze Linux cursus beoogt aan te sluiten bij het Linux Essentials examen van de LPI (Linux Professional Institute) en dient als voorbereiding op het MBO ICT Systems and Devices Expert examen. Voor het leren gebruiken van de grafische interface en de command line maken we gebruik van CentOS en om kennis te maken met het gebruik van Linux als server installeren we Debian. De keuze om CentOS als werkstation te installeren en Debian als server is volledig willekeurig. Het doel is dat de studenten kennis maken met de rpm en dpkg package managers en leren dat het ene Linux systeem het andere niet is.

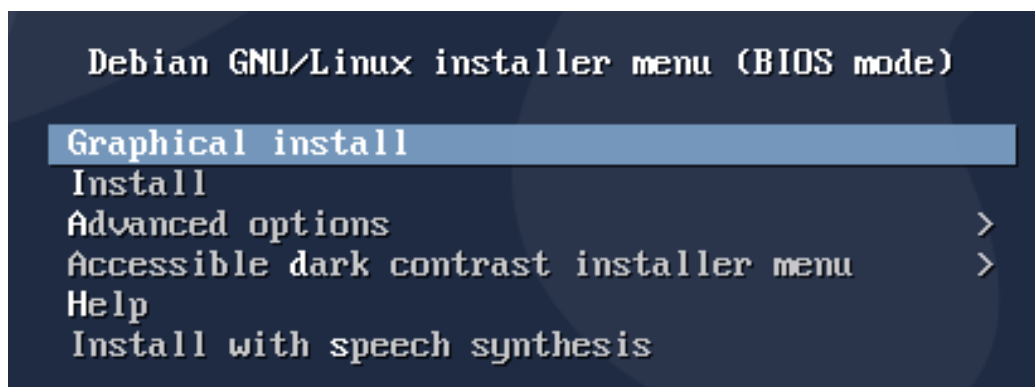
Alle Linux systemen zullen geïnstalleerd worden als virtuele machines op Virtual Box (<https://www.virtualbox.org/>). Door gebruik te maken van virtuele machines zijn we niet afhankelijk van de onderliggende hardware. De keuze voor VirtualBox heeft te maken met het feit dat dit product gratis te gebruiken is en beschikbaar is voor zowel Windows, Mac OS X als Linux.

Voor de CentOS machine is 15G vrije schijfruimte nodig en voor het Debian systeem 8G, wat een totaal aan 23G vrije schijfruimte vereist. Voor elke machine hebben we 2G RAM nodig, dus een totaal van 4G RAM moet vrij beschikbaar zijn.

Hoofdstuk 2

Debian installatie

Selecteer de Grafische installatie interface in de Debian Installer Menu (2.1).



Figuur 2.1: Debian Installer Menu

Selecteer daarna de Engelse taal. Dit is de taal die gebruikt wordt voor de installatie. De taal van het OS kiezen we later.

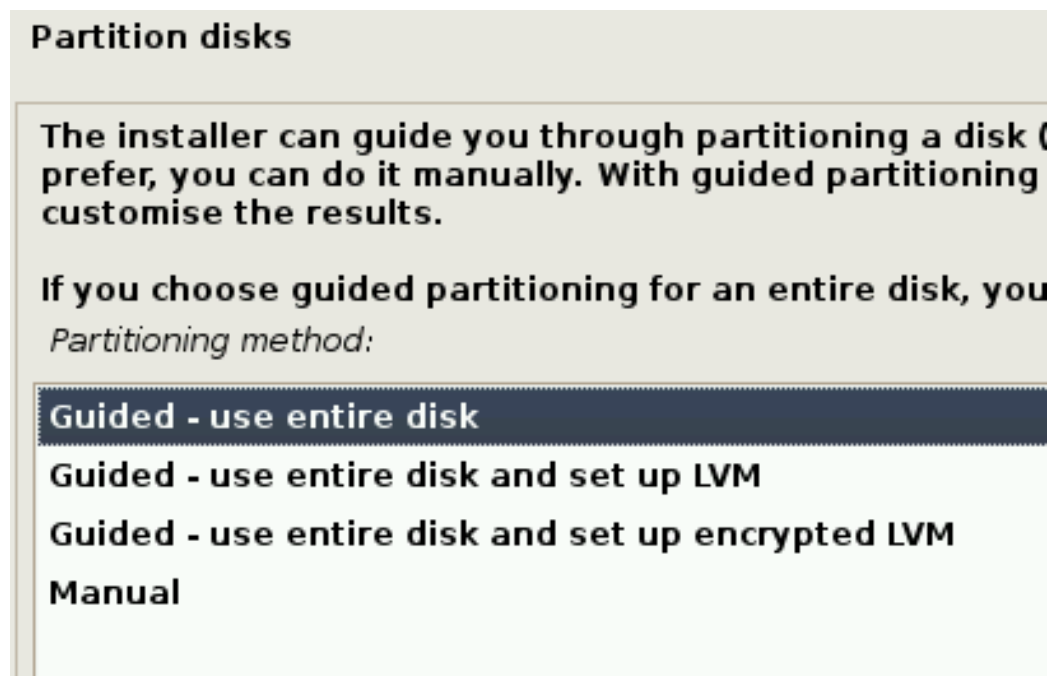
Bij Location kiezen we eerst other en daarna Europe en dan Netherlands. Voor de Locales laten we staan en_US.UTF-8. Voor de keyboard houden we American English aan, tenzij je een specifiek toetsenbord hebt.

De volgende keuze die je moet maken is die voor een hostname, kortom hoe gaat je computer heten. Ik heb gekozen voor DNC01, wat staat voor Debian NextCloud eerste machine. De keuze voor het domein waarin de machine zit is lastig omdat we geen domein hebben. Het makkelijkst is om hier voorlopig localdomain in te vullen. We kunnen het later altijd nog wijzigen.

Nu moeten we een root wachtwoord kiezen, omdat we diensten gaan draaien op deze machine is het verstandig om een sterk wachtwoord te kiezen.

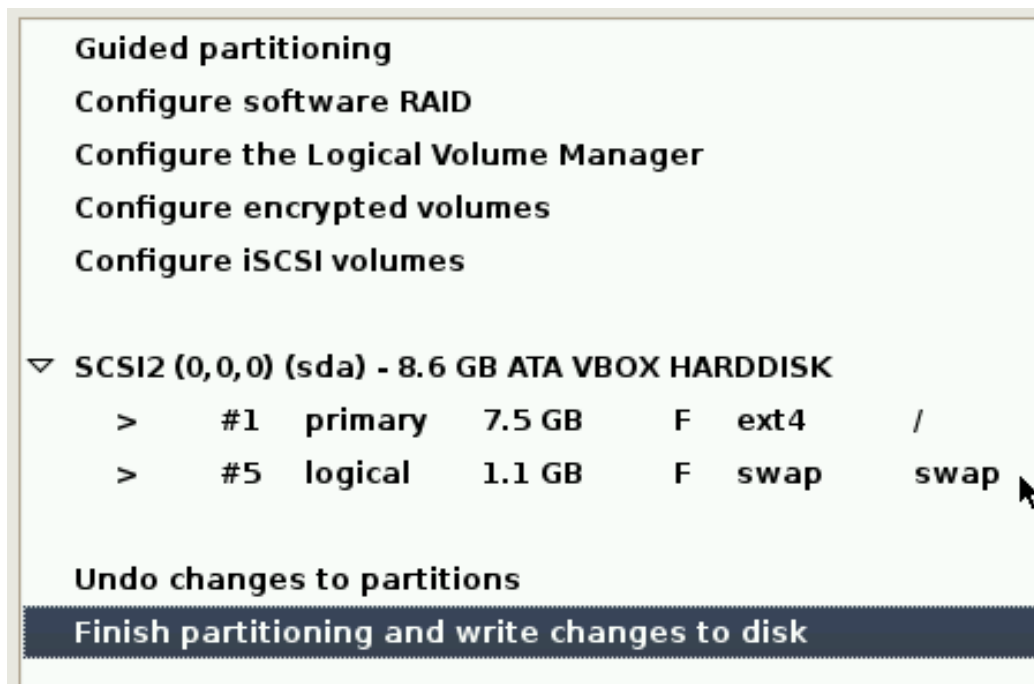
Ook voor de gebruiker is dat een goed plan. Gebruik een password manager om een wachtwoord te generen of om je gekozen wachtwoorden erin op te slaan.

We houden de harddisk indeling simpel, dus kiezen we Guided als onze optie voor de harddisk indeling zoals aangegeven in [2.2](#).



Figuur 2.2: Debian Disk Partitioning

We kiezen de enige harddisk die er is en kiezen voor All files in one partition. In het overzicht dat we te zien krijgen zien we dat er naast een bestandssysteem ook een swap-partitie aangemaakt is zoals weergegeven in [figuur 2.3](#).



Figuur 2.3: Debian Disk Overview

Nadat we hebben gekozen voor Yes bij de vraag of we het zeker weten zal Debian beginnen met de installatie van het systeem. Na de installatie van het base system (een minimaal systeem), krijgen we de vraag of we nog meer software van andere CD's willen installeren. Dat willen we niet. Debian gaat nu opzoek naar updates en additionele software die geïnstalleerd kan worden. De repo's van Debian staan over de wereld verspreid, om geen data op te halen van bijvoorbeeld servers in de Verenigde Staten, maar alleen van servers in Nederland, zodat de belasting van het Internet alleen lokaal is, moeten we een keuze maken vanaf welke servers wij onze software willen ophalen. Kies Netherlands en deb.debian.org. We gebruiken geen proxy dus bij die vraag mag je continue geven.

Debian gaat nu de databases ophalen van alle software die er voor Debian beschikbaar is en gaat daarna meteen je systeem updaten met de laatst beschikbare versies. Afhankelijk van hoe nieuw de ISO was waarvan de hebt geboot kan dit even duren. Je kan af en toe een vraag krijgen van de installer, lees de tekst dan goed en bepaal of je het wel of niet wil wat er gevraagd wordt. Als de update klaar is krijg je de keuze om nog extra software te installeren (zie [2.4](#)).



Figuur 2.4: Debian Additional Software Choice

We willen geen desktop en ook geen printserver, dus deze twee opties moeten we uit zetten. De standard system utilities willen we wel, dus die laten we aan staan.

Nadat Debian klaar is met de installatie kiezen we ervoor om GRUB te installeren in de master boot record, we selecteren `/dev/sda` en als alles klaar is rebooten we het systeem.

Login als root en installeer het sudo pakket. Voeg daarna jezelf als gebruiker toe aan de sudo groep:

```
# usermod -a -G sudo dennis
```

Vervang **dennis** door je eigen gebruikersnaam die je aangemaakt het bij de installatie.

We loggen uit als root en loggen vanaf nu alleen nog in als gewone gebruiker.

Hoofdstuk 3

Werken met daemons

In de Unix wereld zeggen we altijd 'Everything is a file and if it is not a file it is a process'.

Processen die op de achtergrond draaien heten daemons in de Unix-wereld. Ze zijn ook bekend als services, zeker in de Windows wereld. Daemons zijn dus processen waar je weinig van merkt en die rustig hun werk staan te doen. Deze processen draaien zonder dat ze input van de gebruiker nodig hebben.

Bekende daemons zijn bijvoorbeeld een web- of mailservers, maar het kan ook een printserver zijn.

3.1 init en systemd

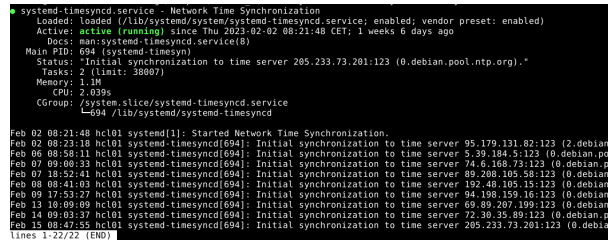
Nadat de kernel in het geheugen geladen is zal deze gestart worden. De Linux kernel zorgt ervoor dat de beschikbare hardware klaar is voor gebruik en dat er processen gestart kunnen worden. Het eerste proces dat de kernel start is **systemd**. Vroeger was dit **init** en dat kan je op vele ander Unix-achtige besturingssystemen nog tegen komen, maar de meeste Linux distributies zijn over naar **systemd**.

systemd is het proces dat ervoor zorgt dat alle ander processen gestart worden. De **systemd** daemon heeft proces nummer 1. Het is de eerste daemon die start bij het opstarten van het systeem en de laatste die afgesloten wordt bij het afsluiten van het systeem.

Naast de **systemd** daemon die de moeder is van alle processen, zijn er ook commando's waarmee je (als root) daemons kunt starten, stoppen en herstarten. Het commando dat doorvoor beschikbaar is heet **systemctl**. Een standaard met de installatie meegekomen daemon in de **systemd-timesyncd** daemon. We gaan deze daemon gebruiken om een beetje vertrouwd te raken

met `systemctl`.

```
sudo systemctl status systemd-timesyncd
```



```
systemd-timesyncd.service - Network Time Synchronization
Loaded: loaded (/lib/systemd/systemd-timesyncd.service; enabled; vendor preset: enabled)
Active: active (running) since Thu 2023-02-02 08:21:48 CET; 1 weeks 6 days ago
Docs: man:systemd-timesyncd.service(8)
Main PID: 694 (systemd-timesyn)
Status: "Initial synchronization to time server 205.233.73.201:123 (0.debian.pool.ntp.org)."
```

Figuur 3.1: Status output van `systemctl`

Als je net als in het voorbeeld (Figuur 3.1) een lijn hebt met END dan kan je gebruik maken van 'q' om weer op de command-prompt terecht te komen.

Door aan `systemctl` de commando's stop of start mee te geven kunnen we daemons op ons systeem stoppen en starten. De tijd synchronisatie daemon is op dit moment gestart, dus het eerste wat we kunnen doen is hem stoppen:

```
sudo systemctl stop systemd-timesyncd
```

Met `systemctl status` kunnen nu de status van de daemon zien en dan zien we dat deze gestopt is. Het opnieuw opstarten doen we met start:

```
sudo systemctl start systemd-timesyncd
```

3.2 Je eigen startup script

Om een service te starten of te stoppen gebruikt `systemd` bestanden waarin beschreven wordt hoe een service heet en welke commando's er nodig zijn om te starten en stoppen. Deze beschrijving van een service wordt binnen `systemd` een **unit** genoemd. Units kan je onder andere terug vinden in de `/etc/systemd/system` directory. Een voorbeeld van een unit-file in Debian 11 is de `syslog.service`. Deze ziet er zo uit:

```
[Unit]
Description=System Logging Service
Requires=syslog.socket
Documentation=man:rsyslogd(8)
Documentation=man:rsyslog.conf(5)
Documentation=https://www.rsyslog.com/doc/
```



```
[Service]
Type=notify
ExecStart=/usr/sbin/rsyslogd -n --iNONE
StandardOutput=null
Restart=on-failure

# Increase the default a bit in order to allow many simultaneous
# files to be monitored, we might need a lot of fds.
LimitNOFILE=16384

[Install]
WantedBy=multi-user.target
Alias=syslog.service
```

Als je een service hebt gewijzigd of toegevoegd moet je systemctl vertellen dat er iets nieuws is. Dat doe je met het volgende commando:

```
$ sudo systemctl daemon-reload
```


Hoofdstuk 4

Logging

Omdat daemons op de achtergrond draaien en geen direct contact met een gebruiker of met root hebben zullen ze op een andere manier moeten laten weten of het goed met ze gaat, wat ze aan het doen zijn en of de beheerder misschien moet ingrijpen omdat er iets fout gaat. Daemons laten ons weten wat er aan de hand is door hun berichten te loggen. Loggen is het wegschrijven van de berichten naar een logbestand.

Linux systemen verzamelen alle log bestanden in een directory `/var/log`. Een daemon kan zelfstandig zijn eigen log wegschrijven naar een bestand, of de daemon kan gebruik maken van een log-server. Het voordeel van een log-server is dat de daemon alleen de API van de log-server hoeft te kennen. Alle andere logica rond het loggen wordt afgehandeld door de log-server. Voor meer informatie over de log-server zie de sectie over syslog (4.2).

Om direct te kunnen volgen wat er op een systeem gebeurt wordt veel het `tail` commando gebruikt. `tail` heeft een optie `-f` die follow betekent. Het blijft volgen welke nieuwe regels er aan een bestand worden toegevoegd. Een manier om bijvoorbeeld de `messages` log te blijven volgen is:

```
$ sudo tail -f /var/log/messages
```

Omdat alle log-bestanden tekst bestanden zijn kan je ook bijvoorbeeld `grep` erop los laten en zo zoeken naar bepaalde patronen in de log-bestanden.

4.1 Log rotation

Als applicaties berichten loggen naar bestanden dan worden de log bestanden steeds langer en daarmee het systeem steeds trager. Dat is niet handig. Eens in de zoveel tijd moet een log-bestand dan ook geleegd worden, maar we willen de oude logs niet kwijt raken. Het proces dat ervoor zorgt dat logs op

de juiste manier behandeld worden zodat ze niet te groot worden heet log rotate.

Er zijn verschillende manieren om het roteren van logs voor elkaar te krijgen. Natuurlijk kunnen we het handmatig doen. We moeten dan:

1. de daemon stoppen,
2. het log-bestand verplaatsen naar een andere naam (bijvoorbeeld, log-bestand.backup),
3. een nieuw leeg log-bestand aanmaken met de juiste rechten,
4. de daemon opstarten.

Dit is een hoop werk en tijdens het werk staat de daemon uit. Tevens is het elke keer dezelfde handeling, dus een prima klus om te automatiseren.

We kunnen een script schrijven dat dit doet en dat via cron laten draaien. Een andere optie is dat we een tool gebruiken die het voor ons regelt zoals bijvoorbeeld `logrotate`.

4.2 syslog

Het proces dat op de achtergrond draait en dat verantwoordelijk is voor het schrijven van de verschillende logbestanden heet syslog, een afkorting van system logging. Het gebruik van syslog heeft als groot voordeel dat niet elke software ontwikkelaar alle code hoeft te schrijven om logbestanden te schrijven, hij hoeft alleen maar te weten hoe hij data moet aanlever bij de syslog server. Voor de systeembeheerder is het makkelijk omdat hij niet elk programma hoeft te vertellen waar deze zijn meldingen weg moet schrijven, die hoeft alleen maar de syslog-server te beheren en het laatste voordeel is dat de output van syslog altijd dezelfde is, dus de logbestanden worden nog makkelijker leesbaar ook.

Vele Linux distributies gebruiken rsyslog als hun syslog server. De belangrijkste logbestanden in de `/var/log` die beheerd worden door de syslog server zijn:

`messages` Alle logs

`mail (SuSE) maillog (RedHat) of mail.log (Debian)` Mail logging

`mail.info` Mail info messages

`mail.warn` Mail warning messages

`mail.err` Mail error messages

`daemon.log` (**Debian**) Berichten van Daemons

`auth.log` (**Debian**) of `secure` (**RedHat**) Authenticatie gerelateerde berichten

4.3 journalctl

`systemd` is het eerste proces (daemon) dat opgestart wordt op een Linux systeem. Het is daarmee de moeder van alle processen. Als er een daemon opgestart moet worden dan wordt dat gedaan via `systemd`. `systemd` is dus de plek waar ook in de gaten gehouden kan worden wat er opgestart is en wat niet en daarmee kan `systemd` een bijzondere inkijk geven in de status van een systeem. Om meer inzicht te krijgen in wat `systemd` allemaal voorbij heeft zien komen is er een tool met de naam `journalctl`.

Het opstarten van `journalctl` zonder enige opties of argumenten laat zien wat `systemd` heeft verzameld sinds het opstarten van het systeem, of sinds de laatste logrotate.

Om alleen de berichten te zien van een specifiek proces kunnen we het volgende commando gebruiken:

```
$ sudo journalctl -f -u systemd-journald
```

De `-f` optie doet hetzelfde als bij `tail`, het zegt dat de log gevolgd (follow) moet worden. Met `-u` geef je op welke 'unit' er gemonitord moet worden.

Hoofdstuk 5

Monitoring services

In de sectie over `journalctl` hebben we gezien hoe we informatie over een proces kunnen opvragen. We kunnen zien of een daemon gestart of gestopt is. Daemons zijn processen die op de achtergrond draaien en die allerlei taken kunnen uitvoeren. Zo is `cron` een daemon, maar een daemon kan ook een mail-server of web-server zijn. Deze laatste twee luisteren op het netwerk naar inkomende verbindingen. De mail-server doet dat voor inkomende mail op poort 25 en de web-server op poort 80. Het is handig als we kunnen zien welke processen er draaien en wat ze aan het doen zijn. Dit hoofdstuk beschrijft enkele commando's die we daarvoor kunnen gebruiken.

5.1 Proces monitoring

In het document 'Linux CLI' hebben we `ps` behandeld. Met `ps` kan je zien welke processen er draaien op een systeem. In deze sectie gaan we `ps` met andere opties gebruiken dan we eerder hebben gedaan. De Linux variant van `ps` kent de mogelijkheid om ook opties te gebruiken zonder gebruik van het min teken voor de optie, dit is de zogenaamde BSD-style:

```
$ ps aux
```

doet bijna hetzelfde als `ps -ef`. De output ziet er ook bijna hetzelfde uit:

```
$ ps aux
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START
TIME COMMAND
root           1   0.0   0.0 167316 11924 ?        Ss   Feb02
0:48 /sbin/init
root           2   0.0   0.0      0      0 ?        S    Feb02
0:00 [kthreadd]
root           3   0.0   0.0      0      0 ?        I<   Feb02
0:00 [rcu_gp]
```

root	4	0.0	0.0	0	0 ?	I<	Feb02
0:00	[rcu_par_gp]						
root	9	0.0	0.0	0	0 ?	I<	Feb02
0:00	[mm_percpu_wq]						
root	10	0.0	0.0	0	0 ?	S	Feb02
0:00	[rcu_tasks_rude_]						
root	11	0.0	0.0	0	0 ?	S	Feb02
0:00	[rcu_tasks_trace]						
root	12	0.0	0.0	0	0 ?	S	Feb02
1:25	[ksoftirqd/0]						
root	13	0.0	0.0	0	0 ?	I	Feb02
6:07	[rcu_sched]						
root	14	0.0	0.0	0	0 ?	S	Feb02
0:03	[migration/0]						
root	16	0.0	0.0	0	0 ?	S	Feb02
0:00	[cpuhp/0]						
root	17	0.0	0.0	0	0 ?	S	Feb02
0:00	[cpuhp/1]						
root	18	0.0	0.0	0	0 ?	S	Feb02
0:03	[migration/1]						
root	19	0.0	0.0	0	0 ?	S	Feb02
0:10	[ksoftirqd/1]						

De extra informatie is voornamelijk de extra data over het gebruik van de CPU en van het geheugen (MEM). Deze informatie kennen we ook uit de `top` applicatie. Ook dit commando is behandeld in de 'Linux CLI' handleiding.

5.2 Daemons en netwerken

Veel daemons worden gebruikt om diensten aan het netwerk aan te bieden. Om een dienst aan te kunnen bieden moet een daemon luisteren op een port en zodra er een verbinding wordt gemaakt op de port moet de daemon een nieuw proces laten luisteren. Veel daemons kan je vertellen op welke interface of welk IP-adres ze moeten luisteren, zo kan je er bijvoorbeeld voor zorgen dan een MySQL server alleen luistert op de interface van localhost zodat hij niet van buitenaf beschikbaar is.

Als we een daemon opstarten willen we kunnen controleren of de daemon daadwerkelijk luistert op een interface. Van oudsher werd hiervoor het `netstat` commando gebruikt, nieuwere systemen hebben meestal het `ss` commando. Beide commando's hebben de dezelfde functionaliteit. Je kan ze gebruiken om te zien welk proces er luistert op welke interface en je kan zien welke verbindingen er op dit moment gaande zijn. We zullen voor de compleetheid beide commando's laten zien.

Als een daemon luistert op een poort en we willen kunnen zien of dat

daadwerkelijk gebeurt dan moeten we de commando's vragen om ons de LISTENING status te laten zien:

```
$ netstat -l
```

```
$ ss -l
```

Als je beide commando's geïnstalleerd hebt staan dan zal je zien de de output iets verschillend is.

Er luisteren over het algemeen veel processen naar allerlei verschillende zaken. Vaak willen we alleen weten of ze op TCP of UDP luisteren. Met de `-u` optie kunnen we kiezen voor UDP en met `-t` voor TCP. We kunnen ze ook tesamen gebruiken om zowel de TCP als de UDP porten te laten zien:

```
$ netstat -lut
```

```
$ ss -lut
```

Je ziet dat voor beide commando's de opties identiek zijn.

Tot slot van het bekijken van de luisterende poorten willen we ook kunnen zien welk proces (welke daemon) de poort gebruikt. Dit kan alleen root zien, dus moeten we `sudo` gebruiken:

```
$ netstat -tulp
```

```
$ ss -tulp
```

Beide commando's kunnen ook gebruikt worden om te zien welke verbindingen er nu open staan met je computer:

```
$ netstat -tuanp | grep -vE 'LISTEN|:\*' 
```

```
$ ss -tu -o state connected
```

Gebruik de man-page van de verschillende commando's om te zien wat de opties betekenen.

Hoofdstuk 6

Printing

Een van de complexere taken is printen. Als het om één printer gaat en één werkstation dan valt het wel mee, maar als je in je netwerk meerdere verschillende printers hebt van verschillende merken en je hebt verschillende besturingssystemen die moeten kunnen printen dan wordt het al snel complex. Om deze complexiteit op te lossen kan je gebruik maken van een printserver.

De printserver is een stuk software dat een print opdracht van een client aanneemt, deze in de rij plaatst en als de printer beschikbaar is de opdracht naar de printer stuurt. Dit lijkt een simpel proces, maar er zijn verschillende printertalen en verschillende document standaarden. Op de één of andere manier moeten die verschillende documenten om gezet worden naar de taal die de betreffende printer snapt.

De verschillende document formaten die een printer snapt zijn:

- PostScript (EPS)
- PDF
- PCL

PostScript™ is ooit bedacht door Adobe Systems tussen 1982 en 1984. Het is jarenlang de standaard geweest als beschrijving van een document zodat een printer het kon afdrukken. PDF heeft het sinds 1992 langzaam aan verdrongen. Ook PDF is bedacht door Adobe Systems.

PCL is een taal bedacht door de mensen bij Hewlett-Packard, maar wordt allang niet meer alleen op HP printers gebruikt.

6.1 CUPS

CUPS, of wel de Common Unix Printing System is een open source print systeem waarvan de code primair ontwikkeld wordt door Apple.

CUPS draait standaard op port 631 en kan geconfigureerd worden via configuratie bestanden of via een webserver die draait op <http://127.0.0.1:631/>.

Hoofdstuk 7

Wireless

Hoofdstuk 8

SMARTd

Hoofdstuk 9

Remote management

9.1 SSH

9.2 Ansible

9.3 Puppet

9.4 SaltStack

Hoofdstuk 10

Linux in een Windows Domain

10.1 SaMBa

Hoofdstuk 11

Mailserver

11.1 postfix

Hoofdstuk 12

LAMP

LAMP is een afkorting voor Linux, Apache, MySQL, PHP. Het is een veel gebruikte 'stack' voor het opzetten van webservern op het Internet. In de loop van de tijd zijn er veel variaties op deze afkorting gekomen door vervanging van bijvoorbeeld Linux door Windows (WAMP), of door MacOS X (MAMP), maar er kan ook een andere database gebruikt worden in plaats van MariaDB (was MySQL). We zouden bijvoorbeeld ook PostgreSQL kunnen gebruiken (LAPP) of andere webservern bijvoorbeeld NginX (waarbij we de E van engine gebruiken) LEMP, en de P kan ook Perl of Python zijn. Wij houden het bij de traditionele stack van LAMP, namelijk Linux, Apache, MariaDB en PHP.

Apache, MariaDB en PHP vormen samen de Middleware in onze PAAS omgeving.

12.1 Apache

De meest bekende webserver in gebruik op het Internet is de Apache webserver.

Sinds enige tijd is er een webserver die populairder is dan Apache en dat is NGINX die voornamelijk gebruikt wordt in situaties waar performance belangrijk is.

```
$ sudo apt-get install apache2
```

12.2 MariaDB

```
$ sudo apt-get install mariadb-server mariadb-client
```

12.3 PHP

```
$ sudo apt-get install php
```

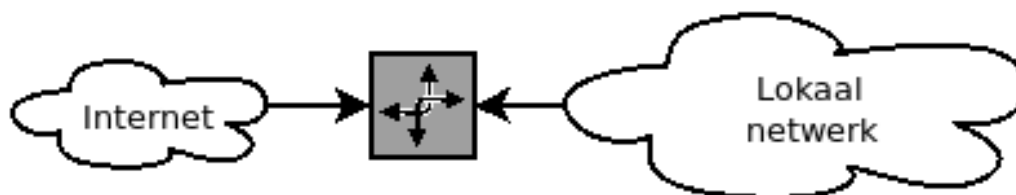

Hoofdstuk 13

NextCloud

Je mobiele telefoon synchroniseert zijn data met de cloud. De meeste cloud oplossingen draaien op Linux, maar wat is dat nu eigenlijk; de cloud? Letterlijk betekent cloud wolk en dat is ook waar de term vandaan komt, in tekeningen van computer netwerken wordt een totaal netwerk vaak vervangen door een wolkje. Zoals bijvoorbeeld het Internet.

De cloud gaat dan ook over diensten die aangeboden worden in een netwerk waarbij het niet meer van belang is op welke machine de data staat, de data bevindt zich ergens in de wolk. Natuurlijk zijn er op de achtergrond fysieke machines waarop de data terecht komt en die beheert moeten worden. In dit hoofdstuk gaan we een machine inrichten met NextCloud. NextCloud is een stukje software geschreven in PHP dat je de mogelijkheid geeft om thuis een cloud systeem te bouwen. Het kan draaien op één server met een paar gebruikers en lijkt dan ook heel erg op een NAS (Network Attached Storage) of je kan er een complete cloud dienst mee ontwikkelen dat 100K+ gebruikers ondersteunt zodat het werkt als DropBox of Google Drive. Wij zullen een simpele installatie doen op een enkele (virtuele) machine.

Een NAS gebruikt meestal FTP of SMB om bestanden te delen, Cloud systemen gebruiken het HTTP(s) protocol om via bijvoorbeeld WebDAV data te delen. Dat hoeven niet alleen bestanden te zijn, dat kan ook kalender informatie (CalDAV) of adresgegevens (CardDAV) zijn. Met NextCloud zou je je telefoon dus volledig kunnen synchroniseren met je eigen server inplaats



van met Google™ of Apple™.

NextCloud is een voortzetting van het ownCloud project. De ontwikkeling van ownCloud begon in 2010 en het ownCloud bedrijf ownCloud Inc. werd in 2011 opgericht door Markus Rex, Holger Dyroff and Frank Karlitschek. Het project was bedoelt als een open source vervanger van DropBox. In 2016 werd de code van ownCloud door Frank Karlitschek geforked en omgedoopt tot NextCloud. Een belangrijk deel van het ownCloud ontwikkelteam ging mee met Frank. Sinds 2017 groeit de aanhang van NextCloud, terwijl die van ownCloud daalt.

13.1 NextCloud installatie

In het vorige hoofdstuk heb je al kennis gemaakt met LAMP (Linux, Apache, MySQL en PHP), daar gaan we nu verder mee werken. We gaan MySQL vullen met de gegevens die nodig zijn voor NextCloud en we installeren de NextCloud software, wat PHP is, binnen de Apache omgeving. Om te kunnen beginnen moeten we eerst NextCloud downloaden of installeren vanuit de package manager. Niet elke distributie levert NextCloud mee en omdat het belangrijk is dat je een NextCloud systeem goed bij houdt met betrekking tot de security-patches is het aan te raden om de software zelf te downloaden vanaf de website van NextCloud.

Ga naar <https://nextcloud.com/> en klik op Get NextCloud en selecteer Download for Server. Op het moment van schrijven is de meest recente versie versie 19.0.1, dus dat versienummer wordt in de rest van dit document gebruikt. Kijk zelf op de pagina van de NextCloud site wat de meest recente versie is en gebruik dat versienummer in de volgende commando's.

```
$ sudo mkdir -p /srv/www/  
$ cd /srv/www/  
$ sudo wget https://download.nextcloud.com/server/releases/  
nextcloud-19.0.1.tar.bz2  
$ sudo tar jxvf nextcloud-19.0.1.tar.bz2
```

Volg verder de aanwijzingen uit het NextCloud installatie document op https://docs.nextcloud.com/server/19/admin_manual/installation/source_installation.html.

Voeg de extra PHP modules toe die nodig zijn voor NextCloud

```
$ sudo apt-get install php-curl php-gd php-mbstring php-intl php7  
.3-mysql php7.3-xml php-zip php-bz2
```

Index

Common Unix Printing System,
 21
CUPS, 21

init, 9

journalctl, 15

log rotate, 14
log-server, 13
loggen, 13
logging, 13
logrotate, 14

netstat, 18

ps, 17

rsyslog, 14

ss, 18
syslog, 14
systemctl, 9
systemd, 9, 15
 timesyncd, 9

timesyncd, 9