

 JeroenJean / Labo_bestandssystemen Public

☆ 0 stars 🍴 4 forks

 Star Watch ▾[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#) main ▾

...



Tiebevn ...

25 days ago

[View code](#) README.md

Labo_bestandssystemen

Download volgende ovf's:

- Voor Intel/AMD:
 - [Debian zonder partities](#)
 - [Debian met partities](#)
- Voor MAC M1/M2 (dit downloadt ZIP-files die je moet unpacken met WinZip):
 - [Debian zonder partities](#)
 - [Debian met partities](#)

Importeer deze ova's in VmWare. Indien dit gelukt is, zou je nu twee nieuwe virtuele machines in je lijst moeten hebben.

Harde schijf zonder partities

Start de VM met naam "Debian_zp". Je zal merken dat dit een linux server is zonder GUI. Om in te loggen mag je volgende gegevens gebruiken:

- Username: ucll
- Password: ucllrocks

Het eerste wat we gaan doen is nagaan hoeveel harde schijf ruimte er beschikbaar is op deze servers. Dit doe je door het commando `df` te gebruiken. Normaal krijg je volgende overzicht:

```
uc11@debian-zp:~$ df
Filesystem      1K-blocks      Used Available Use% Mounted on
udev            981152         0     981152   0% /dev
tmpfs           199604         712     198892   1% /run
/dev/sda2       18964304 1501568  16474064   9% /
tmpfs           998004         0     998004   0% /dev/shm
tmpfs           5120          0        5120   0% /run/lock
/dev/sda1       523244        3496     519748   1% /boot/efi
tmpfs           199600         0     199600   0% /run/user/1000
```

Hoewel dit een duidelijk overzicht geeft van alle partities die aangemaakt zijn op de harde schijf, moet je zelf nog aan de slag om uit te rekenen wat de exacte schijfruimte is. Het commando `df` heeft echter een optie om dit zelf te berekenen alvorens het overzicht te tonen. Hiervoor heb je in vorige labo's geleerd om gebruik te maken van de `--help` optie. Deze geeft het gebruik van het commando weer. Er bestaat ook een andere manier om de details van een commando te achterhalen: `man`. Dit gebruik je steeds als volgt: `man <commando>`. Probeer nu zelf te achterhalen welke optie je moet meegeven aan het commando `df` om het overzicht in een leesbaar formaat te hebben.

Je kan nu zien dat er nog voldoende ruimte vrij is op de harde schijf. Hier gaan we verandering in brengen. Voer volgende commando's uit:

```
$ su -
```

Dit commando zorgt dat we root worden. `su` wil namelijk zeggen 'switch user', '-' wil zeggen root (en nog wat extra zaken, die nu nog niet belangrijk zijn). Root is de user dat in linux alle rechten heeft en dus ook alles mag doen op het systeem. Na het uitvoeren van het commando zal er gevraagd worden achter het wachtwoord. Dit is net hetzelfde als het wachtwoord van de 'uc11' user, behalve op de ARM (MacOS M1/M2) images, waar het wachtwoord 'root' is. *(dit is inderdaad niet veilig en het wachtwoord zelf ook niet, maar we zitten in een testomgeving waar we geen belangrijke documenten hebben staan of toegang hebben tot kritische onderdelen. Toch fijn dat je al zelf die gedachten hebt gemaakt. Dat wil zeggen dat je security hoog in het vaandel draagt)*

Vervolgens voeren we volgende commando uit:

```
# for foo in {1..100}; do dd of=/home/uc11/$foo.full if=/dev/zero bs=1M count=400; do
```



Met wat je ondertussen hebt geleerd over Python en 'Linux Unhatched' zou je dit normaal volledig moeten begrijpen. Zoniet, zal `man` en 'Google' je al een heel eind verder brengen om het volledig te begrijpen.

Achterhaal hoeveel ruimte er nog vrij is op de harde schijf na het uitvoeren van dit commando.

Je zal merken dat alle beschikbare ruimte op de harde schijf is ingenomen. Op het eerste zicht lijkt het alsof het systeem nog perfect werkt. Probeer nu om de applicatie 'screen' te installeren. Met wat je hebt geleerd in de vorige labo's zou dit geen probleem mogen zijn.

? Kan het systeem de applicatie installeren?

? Waarom wel/niet?

Je mag de vm afsluiten door gebruik te maken volgend commando:

```
# shutdown -h now
```

Harde schijf met partities

Start nu de vm 'Debian_mp' op. Voer volgende stappen uit in deze vm:

- Log in met de ucll gebruiker
- Word 'root'
- Kijk na hoeveel vrije ruimte er op de harde schijf/partities is.
- Voer het volgende commando uit: `for foo in {1..100}; do dd of=/home/ucll/$foo.full if=/dev/zero bs=1M count=400; done`
- Kijk na hoeveel ruimte er nog beschikbaar is op de harde schijf/partities
- Installeer de applicatie 'screen'

? Lukt het nu wel om de applicatie te installeren?

? Waarom wel/niet?

Na dit deel van het labo zou het duidelijk moeten zijn wat één van de voordelen is van het gebruik van partities.

Partities

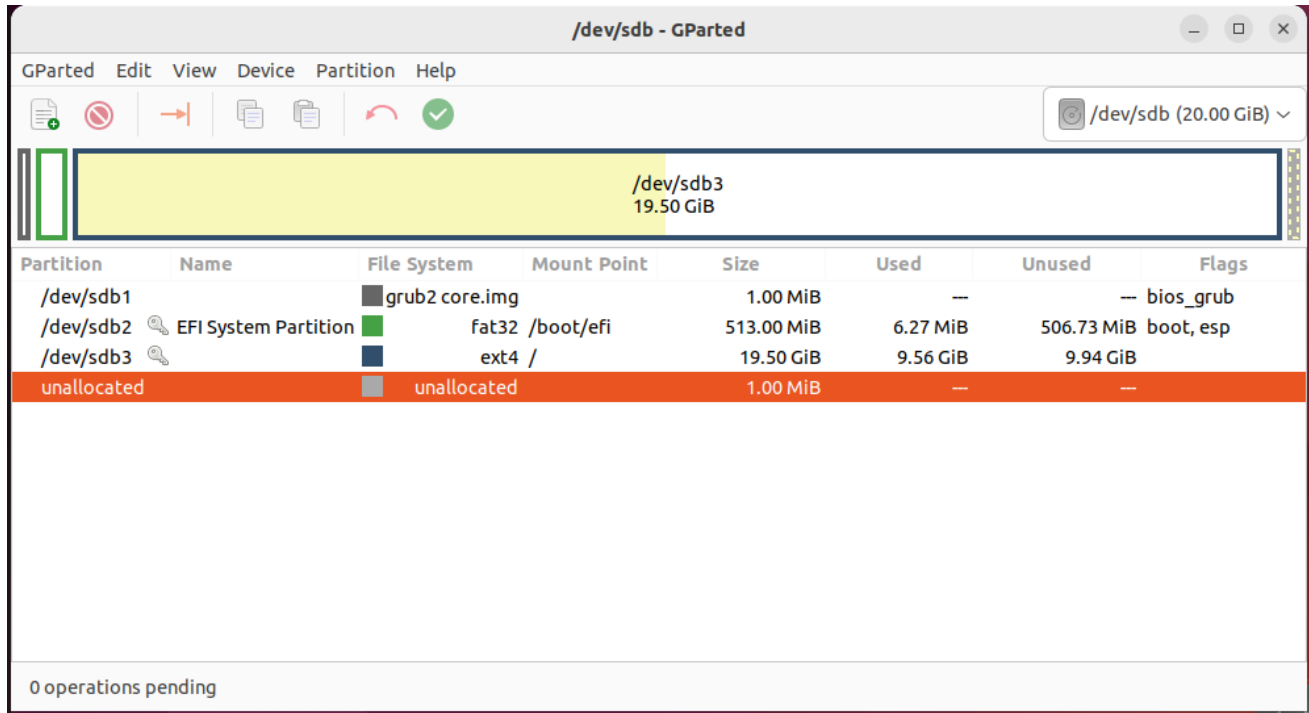
Nu we het nut van partities kennen, wordt het tijd om zelf te leren hoe je partities moeten aanmaken, hetgeen niet zo moeilijk is. Hiervoor bestaan er heel wat tools die zowel in Linux als Windows beschikbaar zijn. Voor dit labo gaan wij gebruik maken van `parted`. Het voordeel van `parted` is, dat hiervoor ook een GUI applicatie is geschreven, dewelke je in Ubuntu kan gebruiken: `GParted`.

Voor we zelf partities gaan aanmaken, gaan we eerst even een partitietabel van dichterbij bekijken.

GParted

Zorg dat alle voorgaande VM's correct afgesloten worden en start daarna je Ubuntu VM van het eerste labo. 'GParted' is normaal niet standaard geïnstalleerd op je vm. Het eerste wat we dus moeten doen is 'GParted' installeren.

Als je dat gedaan hebt en je start 'GParted' op, krijg je normaal volgend scherm te zien.



Dit geeft je een overzicht van alle partities die op de harde schijf ter beschikking zijn. In dit overzicht krijgen we al heel wat informatie:

- **Partition:** hier krijgen we de naam zoals dit gekend is op hardware niveau in het systeem.
- **Name:** Bij het aanmaken van een partitie kunnen we een meer gebruiksvriendelijkere naam configureren
- **File system:** Zoals we gezien hebben in de theorie moet elke partitie geformateerd worden in een bepaald filesysteem voor we gebruik kunnen maken van de partitie om files en directories aan te maken.
- **Mount Point:** Naast een hardware verwijzing moet er ook een 'mount point' geconfigureerd worden. Dit is een directory waarin het filesysteem van die partitie beschikbaar wordt gemaakt voor het OS.
- **Flags:** Voor elke partitie kunnen we bepaalde vlaggen instellen die een extra functionaliteit toevoegen aan een partitie. Zo moet er minstens 1 partitie met de 'boot flag' geconfigureerd worden om ervoor te zorgen dat het OS kan opstarten. De BIOS zal na zijn post immers deze partitie gebruiken om het OS te kunnen laten opstarten.

Indien je nog meer details wilt weten over een bepaalde partitie volstaat om deze partitie te selecteren, rechter muisklik, 'Information'. Dit zal ook het GUID laten zien, alsook de eerste en laatste sector die gebruikt is voor de partitie.



Doe dit ook werkelijk voor alle partities!

Parted

Ook in cli kan je de partities bekijken door gebruik te maken van `parted`. Zorg dat je de vm 'Debian_mp' opstart. Eenmaal opgestart voer volgende commando uit als root:

```
root@debian-mp:~# parted /dev/sda
GNU Parted 3.4
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

Typ nu `print`. Dit zou je normaal een overzicht moeten geven van alle partities die zich op je vm bevinden:

```
Model: VMware, VMware Virtual S (scsi)
Disk /dev/sda: 21.5GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:
```

Number	Start	End	Size	File system	Name	Flags
1	1049kB	538MB	537MB	fat32		boot, esp
2	538MB	4862MB	4324MB	ext4		
3	4862MB	6636MB	1774MB	ext4		
4	6636MB	7661MB	1024MB	linux-swap(v1)		swap
5	7661MB	8046MB	385MB	ext4		
6	8046MB	21.5GB	13.4GB	ext4		

Analyse

Hoewel Parted en GParted ons een mooi overzicht geven, is het ook mogelijk dit nog een stapje verder te zetten: puur naar de bytes gaan kijken. Alle applicaties voor het aanmaken of aanpassen van partities doen dit immers ook. Dit gaan we echter in verschillende stappen moeten doen:

GPT tabel extracten:

We hebben in de theorie gezien dat de GPT partitietabel zich in de eerste LBA's van een harde schijf bevindt. Door gebruik te maken van `dd` kunnen we de nodige bytes kopiëren naar een bestand. De meeste belangrijkste bytes bevinden zich in de GPT Header. Standaard zal deze zich in LBA 1 bevinden. Het commando om deze te kopiëren is:

```
# dd if=/dev/sda of=/root/header bs=512 skip=1 count=1
```

Na het uitvoeren van dit bestand bevindt de volledige inhoud van LBA 1 zich in de file 'header'.

GPT header bekijken

Probeer deze file nu uit te lezen. Het commando hiervoor hebben we al geleerd.

Echter dit geeft enkel volgende output (? kan je ook verklaren waarom?):

```
EFI PART\?????"??P?d5?;D??K?????}a
```

Om de header file correct te kunnen uitlezen, kan je gebruik maken van het commando `hexyl`. Indien je dit correct gedaan hebt, zou je volgende output moeten hebben:

00000000	45 46 49 20 50 41 52 54	00 00 01 00 5c 00 00 00	EFI PART	00•0\000
00000010	a8 b4 ef be 00 00 00 00	01 00 00 00 00 00 00 00	xxxx0000	•0000000
00000020	ff ff 7f 02 00 00 00 00	22 00 00 00 00 00 00 00	x••0000	"0000000
00000030	de ff 7f 02 00 00 00 00	50 9e 64 1f 35 db 3b 44	x••0000	Pxd•5x;D
00000040	b1 b0 f6 4b e5 fa 8e b5	02 00 00 00 00 00 00 00	xxxKxxxx	•0000000
00000050	80 00 00 00 80 00 00 00	d2 14 7d 61 00 00 00 00	x000x000	x•}a0000
00000060	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	00000000	00000000
*				
00000200				

Dit is letterlijk de hexadecimale output van de headerfile (en dus eigenlijk van LBA 1 waar de header van de GPT tabel zich bevindt). Om alle verschillende onderdelen uit de tabel te halen, zou je kunnen tellen om zo de juiste bytes eruit te kunnen halen. Het commando `hexyl` heeft echter een aantal opties waardoor het niet nodig is om handmatig te tellen.

Offset	Length	Contents
0	8 bytes	Signature ("EFI PART", 45 46 49 20 50 41 52 54)
8	4 bytes	Revision (For GPT version 1.0 (through at least UEFI version 2.3.1), the value is 00 00 01 00)
12	4 bytes	Header size in little endian (in bytes, usually 5C 00 00 00 meaning 92 bytes)
16	4 bytes	CRC32 of header (0 to header size), with this field zeroes during calculation
20	4 bytes	Reserved; must be zero
24	8 bytes	Current LBA (location of this header copy)
32	8 bytes	Backup LBA (location of the other header copy)
40	8 bytes	First usable LBA for partitions (primary partition table last LBA + 1)
48	8 bytes	Last usable LBA (secondary partition table first LBA – 1)
56	16 bytes	Disk GUID (also referred to as UUID on <u>UNIXes</u>)
72	8 bytes	Partition entries starting LBA (always 2 in primary copy)
80	4 bytes	Number of partition entries
84	4 bytes	Size of partition entry (usually 128)
88	4 bytes	CRC32 of partition array
92	*	Reserved; must be zeroes for the rest of the block (420 bytes for a 512-byte LBA)
LBA size	Total	

Probeer aan de hand van bovenstaande tabel en het `hexy1` command volgend gegevens uit de header file te krijgen:

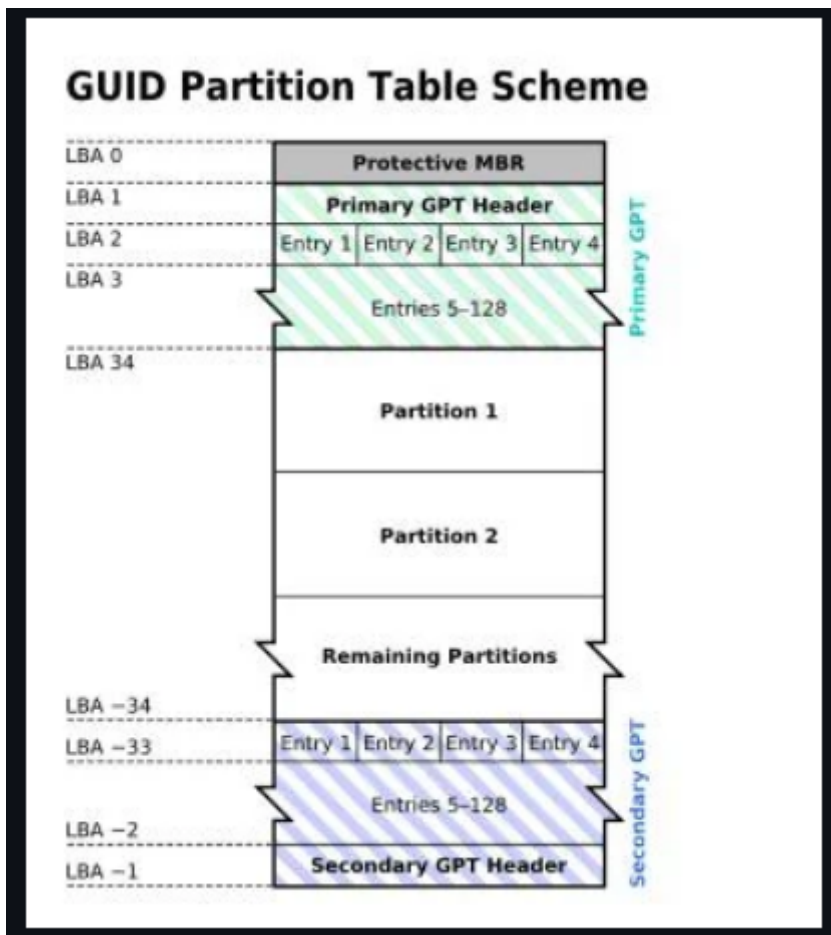
- Signature
- Current LBA
- Backup LBA
- Aantal partities in de tabel
- Grootte van item in partietitabel

? Hoeveel partitie items bevinden zich in de tabel?

? Wat is de grootte van een partitie item in de tabel?

Voor deze vragen zal je je kennis moeten gebruiken van de talstelsels dat we in de eerste week hebben gezien.


GPT entry bekijken



Pas volgende commando correct aan zodat je de inhoud van LBA2 in een nieuwe file zet

```
# dd if=/dev/sda of=/root/header bs=512 skip=1 count=1
```

GUID partition entry format

Offset	Length	Contents
0 (0x00)	16 bytes	Partition type GUID (mixed endian ^[7])
16 (0x10)	16 bytes	Unique partition GUID (mixed endian) 
32 (0x20)	8 bytes	First LBA (little endian)
40 (0x28)	8 bytes	Last LBA (inclusive, usually odd)
48 (0x30)	8 bytes	Attribute flags (e.g. bit 60 denotes read-only)
56 (0x38)	72 bytes	Partition name (36 UTF-16LE code units)

Zorg dat je volgende gegevens uit de juist aangemaakte file kan halen:

- Eerste LBA
- Laatste LBA

Partities aanmaken

Fysieke Harde schijf toevoegen

- Voeg een nieuwe harde schijf toe aan de 'Debian_zp' van 30GB toe.
- Start de vm op
- Eenmaal opgestart, voer dan volgende command uit: `lsblk`. Dit toont je onder andere een overzicht van alle harde schijven die aangesloten zijn. Je nieuwe harde schijf moet hier normaal ook tussen staan. Zorg dat je duidelijk weet wat je nieuwe toegevoegde harde schijf is. Dit zal normaal `/dev/sda` zijn.

```
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda         8:0    0   30G  0 disk
sdb         8:16    0   20G  0 disk
|-sdb1      8:17    0  512M  0 part /boot/efi
|-sdb2      8:18    0 18.5G  0 part /
`-sdb3      8:19    0  976M  0 part [SWAP]
sr0        11:0    1 1024M  0 rom
```

Mount point aanmaken

- Maak een nieuwe directory aan genaamd: `/Party`

Nieuwe partitietabel aanmaken

Het eerste wat we moeten doen is een partitietabel aanmaken in parted:

```
# parted /dev/sda
(parted) mklabel gpt
(parted) print
Model: VMware, VMware Virtual S (scsi)
Disk /dev/sda: 32.2GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start  End  Size  File system  Name  Flags

(parted)
```

Nieuwe partitie aanmaken met correct filesystem

Nu de partitietabel is aangemaakt op de hdd, gaan we een partitie aanmaken:

```
(parted) mkpart
Partition name? []? Party
File system type? [ext2]? ext4
Start? 1
End? 10000
(parted) print
Model: VMware, VMware Virtual S (scsi)
```

```
Disk /dev/sda: 32.2GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:
```

Number	Start	End	Size	File system	Name	Flags
1	1049kB	10.0GB	9999MB	ext4	Party	

```
(parted) quit
```

De partitie is nu aangemaakt, maar om deze te kunnen gebruiken moeten we de partities formateren in het correcte filesystem. Met het `lsblk -f` commando is het meteen duidelijk dat de nieuwe aangemaakte partitie nog geen filesystem heeft. Om dit in orde te maken, maken we gebruik van `mkfs`:

```
root@debian-zp:~# lsblk -f
NAME FSTYPE FSVER LABEL UUID                                 FSAVAIL FSUSE% MOUNTPOINT
sda
└─sda1

sdb
├─sdb1
│   vfat    FAT32      8E96-C30A                507.6M    1% /boot/efi
├─sdb2
│   ext4    1.0        d07c0dac-6789-46f0-ae72-7cf548295042  15.7G    8% /
└─sdb3
    swap    1          357d776f-78af-4e0f-9ee7-88fad06138bb      [SWAP]

sr0
```

```
root@debian-zp:~# mkfs -t ext4 /dev/sda1
mke2fs 1.46.2 (28-Feb-2021)
Creating filesystem with 2441216 4k blocks and 610800 inodes
Filesystem UUID: fc2c44f2-382b-488b-b82a-eb4b36b641a9
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632
```

```
Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```



Met `lsblk -f` kunnen we controleren of het aanmaken van het filesystem op de partitie is gelukt.

```
root@debian-zp:~# lsblk -f
NAME FSTYPE FSVER LABEL UUID                                 FSAVAIL FSUSE% MOUNTPOINT
sda
└─sda1 ext4    1.0        fc2c44f2-382b-488b-b82a-eb4b36b641a9
```

```
sdb
├─sdb1 vfat    FAT32      8E96-C30A                507.6M    1% /boot/efi
├─sdb2 ext4    1.0        d07c0dac-6789-46f0-ae72-7cf548295042  15.7G    8% /
└─sdb3 swap    1          357d776f-78af-4e0f-9ee7-88fad06138bb          [SWAP]
sr0
```

Partitie mounten

Het enige wat ons nu nog rest is de geconfigureerde partitie mounten zodat we ze werkelijk kunnen gebruiken in het OS:

```
root@debian-zp:/# mount -t auto /dev/sda1 /Party
```

Als alles goed is gegaan, zou je normaal een nieuwe partitie moeten hebben die je in je systeem kan gebruiken:

```
root@debian-zp:~# lsblk -f
NAME      FSTYPE FSVER LABEL UUID                                 FSAVAIL FSUSE% MOUNTPOINT
sda
├─sda1 ext4    1.0          fc2c44f2-382b-488b-b82a-eb4b36b641a9    8.6G      0% /Party
sdb
├─sdb1 vfat    FAT32      8E96-C30A                507.6M    1% /boot/efi
├─sdb2 ext4    1.0        d07c0dac-6789-46f0-ae72-7cf548295042  15.7G    8% /
└─sdb3 swap    1          357d776f-78af-4e0f-9ee7-88fad06138bb          [SWAP]
sr0
```

```
root@debian-zp:/# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            959M     0  959M   0% /dev
tmpfs           195M  712K  195M   1% /run
/dev/sdb2        19G   1.5G   16G   9% /
tmpfs           975M     0  975M   0% /dev/shm
tmpfs           5.0M     0   5.0M   0% /run/lock
/dev/sdb1        511M   3.5M  508M   1% /boot/efi
tmpfs           195M     0  195M   0% /run/user/1000
/dev/sda1        9.1G   24K   8.6G   1% /Party
```



Ga nu zelf aan de slag om nog 2 extra partities aan te maken op HDD waar we hierboven een partitie hebben aangemaakt (De stap van partietabel aanmaken, moet je nu niet meer doen!):

- Partitie 2:
 - 10GB
 - Mount point: /UCLL
 - Filesystem: Fat32

- Partitie 3:
 - 10GB
 - Mount point: /Rocks
 - Filesystem: NTFS

💡 Je zal merken dat je een aantal packages moet installeren om de partities te kunnen formateren in NTFS of FAT32. Om te achterhalen welke packages je moet installeren, kan je 2 methodes gebruiken:

- Ofwel ga je zelf op het internet opzoek naar de juiste pakketten die je hiervoor moet installeren.
- Ofwel log je via SSH in op leia.ucillabs.be op poort 22345 met je rnummer. De lijst met packages kan je vinden in /tmp/LabPartities

Na het toevoegen van de partitie zou je ongeveer deze output moeten hebben:

```
root@debian-zp:~# lsblk -f
NAME        FSTYPE FSVER LABEL UUID                                 FSAVAIL FSUSE% MOUNTPO
sda
├─sda1 ext4    1.0          fc2c44f2-382b-488b-b82a-eb4b36b641a9    8.6G      0% /Party
├─sda2 vfat     FAT32        4333-EE1E                               9.3G      0% /UCLL
└─sda3 ntfs                    596DB3E17D1F7047                       9.3G      1% /Rocks
sdb
├─sdb1 vfat     FAT32        8E96-C30A                               507.6M     1% /boot/e
├─sdb2 ext4    1.0          d07c0dac-6789-46f0-ae72-7cf548295042    15.7G     8% /
└─sdb3 swap      1          357d776f-78af-4e0f-9ee7-88fad06138bb
sr0
```

Raid

Raid 1

Een raid cluster aanmaken is niet zoveel verschillend als een partitie aanmaken.

- Maak 2 nieuwe HDD's aan voor je 'Debian_zp' met dezelfde capaciteit.
- Start je VM op en kijk na of je systeem de twee toegevoegde HDD's kan herkennen. Noteer de correcte naam van de twee harde schijven.
- Om je RAID 1 cluster aan te maken gebruik je volgend commando (zorg dat je 'X' vervangt door de correcte waarden):

```
# mdadm --create /dev/md0 /dev/sdX /dev/sdX --level=1 --raid-devices=2
```

- Je RAID 1 cluster is op dit ogenblik aangemaakt. Deze wordt nu helemaal voorbereid om te kunnen gebruiken in je systeem. De status hiervan kan je nagaan door volgende

commando:

```
# cat /proc/mdstat
```

- Je RAID 1 cluster heeft echter nog geen filesystem geïnstalleerd. Hoe je dit moet doen, zou je normaal moeten weten. Kies voor het ext4 filesystem.
- Als je filesystem geïnstalleerd is, moet je dit nog correct mounten in linux. Zorg dat je RAID 1 cluster gemount is in /Cluster1
- Wanneer je nu je VM zou opnieuw opstarten, zou de mount verdwenen zijn. Om dit op te lossen moeten we de mount toevoegen aan de /etc/fstab file. Dit kan je doen door gebruik te maken volgende commando's:

```
# mdadm --detail --scan | tee -a /etc/mdadm/mdadm.conf
```

```
# update-initramfs -u
```

```
# echo '/dev/md0 /Cluster1 ext4 defaults,nofail,discard 0 0' | tee -a /etc/fstab
```



Nu je RAID cluster is aangemaakt gaan we deze ook even uittesten:

- Maak een paar bestanden aan op je cluster
- Sluit je vm correct af.
- Verwijder 1 harde schijf dat je gebruikt hebt in je RAID 1 cluster uit je VM configuratie
- Start je VM terug op
- Kijk je cluster na via `cat /proc/mdstat`

? Kan je nog steeds aan je bestanden aan?

Op dit ogenblik hebben we eigenlijk een defecte harde schijf gesimuleerd. Om optimaal gebruik te maken van je RAID 1 cluster moet je deze terug herstellen in zijn oorspronkelijke staat:

- Maak een nieuwe HDD aan van dezelfde grootte.
- Start je VM terug en kijk of de nieuwe harde schijf herkend wordt.
- Gebruik volgende commando om de nieuwe harde schijf aan de cluster toe te voegen:

```
# mdadm --manage /dev/md0 -a /dev/sdX
```

Via `cat /proc/mdstat` kan je nu duidelijk zien dat de cluster zich aan het herstellen is.

Raid 5

Een RAID 5 cluster aanmaken is praktisch net hetzelfde als een RAID 1. Het enige verschil is dat je gebruik moet maken van minstens 3 harde schijven.

❗ Configureer zelf een RAID 5 cluster dat gemount is op /Cluster2 en EXT4 filesystem heeft. Normaal ga enkel het 'mdadm --create' command correct moeten aanpassen en de fstab file.

💡 Probeer net zoals bij de RAID 1 cluster een aantal files aan te maken, een simulatie van faulty HDD en de RAID 5 cluster te herstellen.

Next Level

Nu je perfect weet hoe partities en RAID werkt en hoe je deze kan aanmaken, gaan we een nieuwe Linux distributie installeren met partities en een RAID 1 cluster:

- Download de Ubuntu server [ISO](#).
- Maak een nieuwe VM aan met 2GB RAM en 20GB HDD
- Installeer Ubuntu server met onderstaande partities.
- Eenmaal correct geïnstalleerd, sluit 2 HDD extra aan van 10GB en mount deze in een RAID1 cluster.
- Zorg dat je RAID cluster automatisch gemount wordt bij het opstarten.

Een goede linux installatie zal minimum steeds bestaan uit volgende partities:

- swap
- /
- /home
- /var
- /tmp
- EFI boot

Deze partities zijn ook geconfigureerd op de 'Debian_mp' vm, je mag gerust de grootte van deze partities overnemen voor je nieuwe VM:

```
root@debian-mp:~# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda           8:0    0   20G  0 disk
├─sda1        8:1    0  512M  0 part /boot/efi
├─sda2        8:2    0    4G  0 part /
├─sda3        8:3    0   1.7G  0 part /var
├─sda4        8:4    0   977M  0 part [SWAP]
├─sda5        8:5    0   367M  0 part /tmp
└─sda6        8:6    0  12.5G  0 part /home
```

❗ Let wel op, sommige partities moeten ook een correcte flag krijgen. Google zal je hierbij graag helpen.

Releases

No releases published

Packages

No packages published

Contributors 2



JeroenJean



Tiebevn Tiebe Van Nieuwenhove