

2.1 CLI and text file handling

Command Line Interface

In deze les gaan we voor het eerst kennis maken met de command line. We bespreken wat commando's en gaan daarmee oefenen. Het doel van deze les is een kennismaking met de command line.

CLI is de command line interface van Linux.

Linux bestaat eigenlijk uit de CLI en optioneel zit daarbij een grafische gebruikers interface (GUI). Linux Servers die in datacenters staan zijn vaak **headless**. Dat betekent dat er geen monitor en toetsenbord is aangesloten op de server. Vaak bestaat één fysieke server ook meerdere virtuele servers. Als je zelf zo'n server wilt dan moet je zoeken naar VPS, dat staat voor **Virtual Personal Server**. Een VPS met Linux is erg goedkoop omdat het OS gratis is. Ook is de performance vaak zeer redelijk tot goed. Deze website draait ook op een VPS van een paar euro per maand en samen met deze website draaien er nog een paar websites op dezelfde Virtual Server. Er draaien meerdere virtuele (web) hosts op één virtuele server. Wat een virtual web host is leren we over een paar lessen.

Headless

In data centers staan dus veel headless Linux servers.



Op zich kan er wel een GUI worden toegevoegd, maar dit wordt vaak als omslachtig en traag gezien. Bovendien hebben echte 'hackers' een voorkeur voor een CLI. Je zult leren dat de CLI van Linux erg rijk is en dat je hele gave dingen met een paar commando's kan uitvoeren. Binnen Linux kun je ook vrij eenvoudig zaken automatiseren. Een goede Linux systeembeheerder automatiseert dus als zijn taken en hoeft daarna alleen nog maar updates uit te voeren en te monitoren dat alles goed gaat.

Linux en tekst bestanden (text files)

Linux heeft veel configuratie in plain text files staan. Daarom zijn er veel commando's die iets met files doen.

In de Linux lessen gaan we af en toe een paar handige Linux CLI commando's behandelen.

Om te beginnen gaan we samen in de les naar de volgende commando's kijken. In deze les een paar handige commando's die te maken hebben met files.

Belangrijke commando's voor file handling

commando	wat doet het?	voorbeeld
man	laat een help pagina van een commando zien	man cat
cat	laat de inhoud van één of meerdere files zien	cat /var/log/syslog
grep	zoekt in een file naar een woord	grep error /var/log/syslog
head	laat de eerste regels van een file zien	head /var/log/syslog
tail	laat de laatste regels van een file zien	tail /var/log/syslog
touch	maak een lege file	touch test
echo	print iets	echo "Hallo"
cp	CoPy - kopieer een file	cp file01 file02
mv	MoVe - verplaats/hernoem een file	mv file01 file02
rm	ReMove - verwijder een file	rm file01

Belangrijk: Linux is case sensitive!

Redirect output

Linux heeft een paar handige concepten. Zo kan je de output die naar het scherm gaat omleiden naar een file. Dat heet *redirect output*.

Neem bijvoorbeeld het commando `head /var/log/syslog`. Dat laat de eerste paar regels van de Linux log zien. Je kunt deze output naar een file wegschrijven. Bijvoorbeeld `head /var/log/syslog > mijnTestFile`

Het `>` teken schrijft dus de output van een commando naar een file. Het `>>` teken voegt de output van een commando toe aan de file.

Voer de volgende commando's uit en probeer te begrijpen wat er gebeurt en waarom.

```
echo 0123456789 > file01
cat file01 file01 > file02
cat file02
cp file02 file01
cat file01 >> file02
ll
cat file02
```

Haal alle files weg: `rm file01 file02`

Commando's aan elkaar rijgen

In Unix/Linux kun je commando's aan elkaar rijgen, dat kan super handig zijn en we komen daar nog in detail op terug, maar kijk eens naar het volgende voorbeeld:

```
cat /etc/passwd | cut -d ":" -f1 | sort
```

Wat hier staat is het volgende. Laat de inhoud van de file `/etc/passwd` zien en gebruik de output als input voor het commando `cut`. Het commando `cut` haalt de eerste kolom uit de input en deze output wordt weer gebruikt voor de input van het `sort` commando. Deze sorteert de output en stuurt de output naar het scherm

Je kunt natuurlijk de output ook naar een file sturen zoals we in het vorige hoofdstuk hebben geleerd.

```
cat /etc/passwd | cut -d ":" -f1 | sort > user_names.txt
```

Het commando is nu uitgebreid met `> user_names.txt` en dat zorgt ervoor dat de output niet naar het scherm gaat, maar naar een file.

Opdrachten

Zorg ervoor dat je VM met Ubuntu Server draait en log aan onder jouw eigen gemaakte user met MobaXterm.

1. Ubuntu WS heeft een soort notepad/kladblok.
Met de editor `nano` kan je iets in de file zetten of veranderen. Open een nieuwe file met `nano file01.txt`
Zet jouw naam in de file en bewaar (save) de file.

Zoek uit de lijst hierboven met welk commando je de inhoud van de file kunt afdrucken, probeer of het werkt en zet het commando op je antwoordenblad.

2. Voer de onderstaande stappen uit en schrijf op wat de uitkomst van stap 4 is. Je ziet drie getallen, beschrijf wat deze getallen betekenen.
 1. Maak met het touch commando een file aan en noem die *file.txt*
 2. Gebruik het echo commando om *Hello world this is nice!* in de file.txt te plaatsen
 3. Gebruik de geleerde kennis om de tekst *Hello world this is nice!* vier keer onder elkaar in de file.txt te krijgen.
 4. Voer het commando `wc file.txt` uit. Wat zie je en wat betekent dat? Als je niet zeker bent, controleer dan met `man wc` hoe wc precies werkt.
3. Zoek op hoe je met het commando tail de laatst 20 regels moet afdrukken. Druk de laatste 20 regels van `/var/log/syslog` af en plaatst die in een file `laatst20.txt`. Voer het commando `cat laatst20.txt laatst20.txt laatst20.txt > laatste.txt` uit. Hoeveel regels staat er in laatste.txt? Schrijf op hoe je dit hebt gedaan en schrijf je antwoord op.
4. Als ubuntu is opgestart dan wordt er een regel in de `/var/log/syslog` geplaatst, bijvoorbeeld;
`Jan 12 21:27:40 ubuntu_vm systemd[1121]: Startup finished in 102ms.`
Gebruik grep en zoek uit hoe lang jouw Ubuntu er de laatste keer over deed om te starten. Schrijf het commando dat je hebt gebruikt op.
5. Laat alleen alle regels uit de syslog file zien van vandaag. Schrijf het commando dat je hebt gebruikt op.
6. Begin met een file01 en zet daar text in. Het commando `cat file01 > file01` doet iets vreemds; de file01 is leeg, leg uit waarom.
7. Met het commando `sudo apt-get install sl` installeren we een programmaatje. Omdat we iets selecteren moeten we 'elevated rights' krijgen. Dat is een mooie manier om te zeggen dat we root toegang moeten hebben. Sudo staat voor "Super User DO". In een vervolg les gaan we hier nog verder op in. Installeer het programmaatje en start het op.

Vraag: *Kan je de stoomlocomotief ook laten vliegen? Ja, hoe dan - en hoe ben je hier achter gekomen?*

Met `sudo apt-get remove sl` kunnen we het programmaatje weer verwijderen.

8. Kijk naar het volgende commando en test het uit.

```
cat /var/log/syslog | wc | awk '{print $1}'
```

Het commando is aan elkaar geregen, je kunt er stapjes afhaken om te kijken wat de vorige stapjes deden.

Omschrijf kort op je antwoordenblad wat commando laat zien.

9. Met `ifconfig` konden we ons ipnummer vinden weet je nog? Probeer maar.

Alle ipnummers in onze VM's beginnen met 192. Met `grep 192` kun je dus de juiste regel vinden.

Rijg de twee commando's aan elkaar, zodat je in één keer de regel met het juiste ipnummer vind.

Schrijf het commando op dat je hebt gebruikt.

10. Het volgende commando kan ook korter.

```
cat /var/log/syslog | tail
```

Hoe doe je hetzelfde, maar dan met minder commando's? Schrijf het verkorte antwoord op.

11. Maak een file met het commando `echo test > test.txt`

Je wilt de file kopiëren en je typt in:

```
mv test.txt test-bak.txt
```

Helaas is dat niet goed gegaan want je wilde de file kopiëren en je hebt een verkeerd commando gebruikt. Kijk met `ls -la` welk(e) files je nu hebt.

Met welk commando corrigeer je dit en zorg je ervoor dat je alsnog een kopie hebt?

12. Maak drie files, `file01.txt`, `file02.txt` en `file03.txt` en zet in alle drie de files de volgende regel tekst: "Dit is een test bestand!". Voer het commando `cat file01.txt file02.txt >> file03.txt` uit.

(1) Hoeveel keer staat de test-regel nu in het bestand `file03.txt` ?

Voer nu het commando `cat file01.txt file02.txt > file03.txt` uit.

(2) Hoeveel keer staat nu de test-regel in het bestand `file03.txt` ?

Leg uit waarom het tweede geval minder test-regels laat zien dan het eerste geval.

Revision #25

Created 2020-01-14 13:10:04 UTC by Admin

Updated 2025-11-12 19:36:29 UTC by Max