



**HBO-ICT:
TCSB-V2AUT-16
TCSB-DU2AUT-19**

Automation

Werkboek





Inhoudsopgave

1	Inleiding:.....	4
1.1	Theorie en tentamen	4
1.2	Vorbereiding	5
1.2.1	Introductie	5
2	Opdrachten	6
2.1	Week 1, College 1: Python en Sockets.....	6
2.1.1	Lesopdracht	6
2.1.2	Weekopdracht	7
2.1.3	Lesstof tentamen	9
2.1.4	Informatiebronnen	9
2.2	Week 1, College 2: 'Bare Metal' → OS Deployment.....	10
2.2.1	Lesopdracht	10
2.2.2	Weekopdracht	10
2.2.3	Lesstof tentamen	17
2.2.4	Informatiebronnen	18
2.3	Week 2, College 1: Modules Security.....	19
2.3.1	Om de les mee te beginnen.....	19
2.3.2	Lesopdracht	19
2.3.3	Weekopdracht	20
2.3.4	Lesstof tentamen	22
2.3.5	Informatiebronnen	22
2.4	Week 2, College 2: Configuration Management Using SaltStack.....	23
2.4.1	Lesopdracht	23
2.4.2	Weekopdracht	23
2.4.3	Lesstof tentamen	28
2.4.4	Informatiebronnen	28
2.5	Week 3, College 1: Database CGI	29
2.5.1	Lesopdracht	29
2.5.2	Weekopdracht	29
2.5.3	Lesstof tentamen	31
2.5.4	Informatiebronnen	31
2.6	Week 3, College 2: Configuration Management (NextCloud).....	32
2.6.1	Lesopdracht	32
2.6.2	Weekopdracht	32



2.6.3	Lesstof tentamen	37
2.6.4	Informatiebronnen	37
2.7	Week 4, College 1: XML Encoding	38
2.7.1	Lesopdracht	38
2.7.2	Weekopdracht	39
2.7.3	Lesstof tentamen	39
2.7.4	Informatiebronnen	40
2.8	Week 4, College 2: Desired State Configuration (DSC)	41
2.8.1	Lesopdracht	41
2.8.2	Weekopdracht	41
2.8.3	Lesstof tentamen	46
2.8.4	Informatiebronnen	46
2.9	Week 5, College 1: Object Oriented Matplotlib	47
2.9.1	Lesopdracht	47
2.9.2	Weekopdracht	47
2.9.3	Lesstof tentamen	50
2.9.4	Informatiebronnen	50
2.10	Week 5, College 2: Test Automation.....	51
2.10.1	Lesopdracht	51
2.10.2	Weekopdracht	52
2.10.3	Lesstof tentamen.....	56
2.10.4	Informatiebronnen	56
2.11	Week 6, College 1: PowerShell.....	57
2.11.1	Lesopdracht	57
2.11.2	Weekopdracht	57
2.11.3	Lesstof tentamen.....	58
2.11.4	Informatiebronnen	59
2.12	Week 6, College 2: OS Virtualization using Containers.....	60
2.12.1	Lesopdracht	60
2.12.2	Weekopdracht	60
2.12.3	Lesstof tentamen.....	64
2.12.4	Informatiebronnen	64
2.13	Bijlage1: Antwoorden 2.1.1, opdracht 1	65
2.14	Bijlage2: Antwoorden 2.3.1, functies.....	67



1 Inleiding:

“Automation” leer je pas echt goed kennen door ermee te werken op basis van “echte” praktijksituaties. Dat is ook wat we gaan doen deze cursus en dit werkboek helpt je erbij. Je vindt in dit document alle opdrachten die horen bij deze cursus. Elke les heeft ongeveer dezelfde opbouw:

- **Lesopdracht:** Dit is een opdracht die je uitvoert in de les, daardoor breng je de theorie direct in praktijk.
- **Weekopdracht:** Deze opdrachten mag je alvast thuis maken, maar dat hoeft niet: Je maakt deze opdrachten tijdens het college om de lesstof die de docent dan uitlegt gelijk te oefenen tijdens de les. Het is te veel werk om af te krijgen tijdens de les, dus thuis afmaken.
- **Lesstof tentamen:** Bestudeer deze informatie heel goed, want bij dit vak hoort een schriftelijk tentamen. De vragen kunnen uit dit werkboek, de presentaties en de documenten op Canvas komen. Kortom: Alles op Canvas is belangrijk hiervoor.
- **Informatiebronnen:** Dit bevat extra informatie over het onderwerp. Hierdoor begrijp je de stof beter.

We gaan ervan uit dat je gebruik kunt maken van een eigen laptop.

1.1 Theorie en tentamen

In de studentenhandleiding staat hoe deze cursus is opgebouwd.

Bij het praktische gedeelte, hoort ook een theorietoets. Hiervoor moet je alle documenten op Canvas bestuderen inclusief dit werkboek met het kopje “lesstof tentamen”).

Wk	Onderwerp	Bestuderen
1.	Inleiding automation	1a-automation_isnt_just_for_scale.pdf (5 blz.) 1b- Building Your DevOps Engine – Rackspace.pdf (9 blz.)
2.	Configuration Management	2a- introduction-to-configuration-management.pdf (8 blz.) 2b- Declarative vs Imperative.pdf (4 blz.)
3.	Configuration Management	3a- gettingstartedwithsaltstack.pdf (39 blz.)
4.	Desired State Configuration	4a- configuration-management-windows-powershell-desired-state-configuration.pdf (26 blz.)
5.	IT Process Automation	5a- A-Guide-to-IT-Process-Automation-Tools (18 blz.) 5b- byte_of_python hst12 only.pdf (alleen hst. 12: OO, 11 blz.)
6.	Managing Containers	6a- Infrastructure as Code.pdf (3 blz.) 6b- Docker.pdf (12 blz.) 6c- Containers_ Docker, Windows and Trends _ Blog _ Microsoft Azure (5 blz.) 6d- Infrastructure as Code OReilly.pdf (19 blz.)



1.2 Voorbereiding

1.2.1 Introductie

Bij de labs maken we gebruik van PyCharm of Visual Studio Code (VSC), Python versie 3 (tenzij anders aangegeven), RHEL 8 (Linux), Windows Server 2019, Recente Windows 10 Client en recente VMWare Workstation op je eigen laptop.

Hoewel er verschillende mogelijkheden zijn geven wij de voorkeur aan de onderstaande configuratie:

Je eigen laptop met 8GB of meer geheugen, een interne SSD, meest recente “Vmware Workstation 16) voor Windows”, Python 3.x (meest recente) en PyCharm of VSC.

Wat betreft PyCharm: je kan PyCharm professional gebruiken. Als je dat wilt, lees dan hier hoe dat moet:

<https://blog.jetbrains.com/blog/2018/09/18/free-jetbrains-licenses-as-part-of-github-student-developer-pack/>

Heb je nog vragen? Stel ze gerust aan je docent...

Je bent zelf verantwoordelijk voor het creëren van de juiste omgeving, maar we helpen je wel een beetje op weg. Als je kiest voor een andere configuratie zoals Linux waar het kan, of het gebruik van Apple besturingssystemen, dan is dat niet verboden, maar je krijgt daarbij geen ondersteuning en je moet wel zorgen dat je de geboden opdrachten goed kan uitvoeren. Dat geldt ook voor het gebruik van een andere IDE zoals Visual Studio Code (VSC), misschien moet je iets meer zelf uitzoeken.

Nog meer informatie over de cursus is te vinden in de studentenhandleiding.



12) Maak onderstaande code werkend.

```
def doeIets(x):
    # als x kleiner dan 10, dan moet y zijn: "x is kleiner dan 10"
    # anders, als x groter of gelijk aan 10, maar kleiner dan 20: y="x tussen 10 en 20"
    # anders: y="x is groter dan 19"
    # geef de variabele y terug
# vraag de gebruiker om een getal
# maak er een integer van
# roep de functie aan. Wat de functie teruggeeft moet naar de variabele z
# druk z af op het scherm
```

13) Maak een lijst met daarin twee lijstjes met ieder 3 getallen. print uit de tweede lijst het tweede getal.

14) Print de huidige werkdirectory (linux: pwd). Tip gebruik de module os en functie getcwd...

Opdracht 2

De leraar heeft je zojuist uitgelegd via de presentatie hoe je met sockets kan werken in Python. Probeer nu in tweetallen via Python een connectie te maken (TCP) en data te sturen vanaf de ene computer naar de andere.

Werk als volgt:

Lees eerste deze pagina http://www.bogotobogo.com/python/python_network_programming_server_client.php

Gebruik de informatie uit de voorbeelden om het nu zelf te programmeren:

- Student 1 maakt de server (een while loop hoeft nog niet. Als je 1 zin kan ontvangen en printen op het scherm is het al goed).
- Student 2 maakt de client (maak een connectie en stuur een zin).

Opmerkingen hierbij:

- Gebruik je windows? Denk aan de firewall.
- Is je Server af? Test 'm eerst met telnet vanaf linux, netcat, op Putty (raw mode).
- Is het gelukt? Kijk dan of je als server ook nog informatie terug kan sturen naar de client of maak je eigen "whatsapp" – 1:1 chat client!

2.1.2 Weekopdracht

Functionele eis:

Je gaat een service bouwen in Python die luistert op een poort (TCP). Vanaf je management systeem kan je een verbinding maken met deze server en via de service op afstand software starten op de externe host. Op die manier kan je het systeem met deze Python-server op afstand "beheren". Natuurlijk moet je er voor zorgen dat deze service geen veiligheidsrisico's oplevert voor de host en daarom moet je de service beveiligen.

Je hebt nodig:

De code hieronder heb je nodig voordat je kan beginnen met deze opdracht. Het bevat een hele eenvoudige tcp-service waarmee de basis is gelegd voor deze opdracht. Je maakt dus alleen de server-code. De client code heb je niet nodig, want client ben je door bijvoorbeeld putty in raw mode te gebruiken.

Hieronder de code die je kan gebruiken om te starten met de opdracht:

```
import socket
import sys
import subprocess
```



```
HOST = '' # Alle beschikbare interfaces
PORT = 8888 # Willekeurige poort (denk aan firewall bij Windows Systemen)

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((HOST, PORT))
s.listen(10)
print('> Socket luistert op poort:', PORT)

# Wacht op connecties (blocking)
conn, addr = s.accept()

# Er is een client verbonden met de server
print('> Verbonden met ' + addr[0] + ':' + str(addr[1]))

# De server meldt zich aan de client
conn.sendall(b'WelkomOpMijnServer, vertel me iets, dan zeg ik hetzelfde terug:\n')

# Wacht op input van de client en geef deze ook weer terug (echo service)
data = conn.recv(1024)
data=str(data.decode('ascii')).rstrip() # # Remove \r | \n | \r\n

print('> Client data ontvangen:'+data+'<eindeData>')
conn.sendall(b"jeStuurdeMijDezeData:"+data.encode())
conn.sendall(b'\n')

# Verbreek de verbinding en sluit de socket
conn.close()
s.close()
```

Stappenplan:

Bouw het script als volgt op en test steeds ieder onderdeel voordat je verder gaat!

1. Zorg allereerst dat je vanaf een andere client kan communiceren met de bovenstaande Python-service (mag in tweetallen). Let hierbij op de firewall en gebruik als client netcat, telnet (niet de windows telnet client, linux versie kan meestal wel) of putty (in raw mode). **Kortom, zorg dat dit script “werkt”.**
2. Als de service werkt: Er is geen foutafhandeling opgenomen in het script bij het maken, binden en luisteren op een poort. **Voeg goede foutafhandeling toe** aan de server voor deze stappen in het proces en test ook dat dit werkt. (Je kan dit zelf testen door als PORT een getal op te geven dat al wordt gebruikt door windows, je krijgt een overzicht met “netstat -an”.)
3. In het script verbreekt de server de verbinding nadat de client iets heeft verzonden. Zorg ervoor dat de server blijft luisteren en echo-en tenzij de client het woord "stop" zendt, in dat geval mag het script stoppen. **Kortom: er is een loop nodig.**
4. Maak het nu mogelijk dat een client **een extern commando** verstuurt dat op de server wordt uitgevoerd. Dit is natuurlijk een veiligheidsrisico: als je niet oppast kan een ander je harde schijf formatteren! Implementeer dit daarom als volgt: Als de client het woord “CALC” verstuurt dan start je de calculatoren als de client het woord “NOTEPAD” verstuurt start je notepad. Start alle executables asynchroon.



5. Je hebt het aantal commando's al beperkt, maar bedenk nog **een extra beveiligingslaag**, zodat niet iedereen deze commando's kan uitvoeren, waarmee de oplossing nog veiliger wordt. Bepaal zelf hoe je dit wilt doen.
6. Iedere keer als een programma wordt gestart moet dit worden gelogd in **een logfile op de server**. Vermeld telkens een regel met de datum/tijd en het commando dat werd gestart.
7. Optioneel: **Volg de communicatie met een packet analyzer** zoals whireshark (hoef je niet te laten zien aan je docent).

2.1.3 Lesstof tentamen

Wat moet je weten voor het tentamen over dit onderwerp, dus hoe kan je je het beste voorbereiden?

- **Leer** alle informatie die je vindt in de powerpoint presentatie over dit onderwerp.
- **Lees** alle informatie die wordt aangeboden onder het kopje "Informatiebronnen".
- **Zorg dat je een fout kan halen** uit de Python code waarin een eenvoudige TCP-server of TCP-client wordt getoond.
- Zorg dat je zelf clients en servers hebt gemaakt zodat je **getoonde code van anderen goed begrijpt** en kan **aanpassen/verbeteren**.

2.1.4 Informatiebronnen

- Youtube: <https://www.youtube.com/watch?v=XiVVYfgDoIU>
- Perfecte website voor allerlei modules, dus dit ook <https://pymotw.com/2/socket/tcp.html>
- Alle details, simpel http://www.tutorialspoint.com/python/python_networking.htm
- simpele voorbeelden, wat uitgebreidere uitleg <https://pythontips.com/2013/08/06/python-socket-network-programming/>



2.2 Week 1, College 2: 'Bare Metal' → OS Deployment

Er bestaan verschillende methoden om hardware te voorzien van een besturingssysteem: Met “bootable media”, zoals een installatie CD, in een Cloud omgeving gebruik maken van een template, via multicast een hele groep computers deployen met software zoals “Symantec Ghost” etc. De uitdaging bij deployment is altijd om flexibel te blijven zodat je niet direct een nieuwe image hoeft te maken als je een extra pakket wilt installeren bijvoorbeeld. In deze week ga je ervaring opdoen met een “Zero Touch” Installatiemethode met behulp van de Preboot Execution Environment (PXE) in een Linux omgeving. Hierbij wordt de installatie over het netwerk gestart. Deze methode is ook van toepassing op het Windows platform. De gebruikte software heet dan “WDS”, “WAIK”, “WDT”, maar het principe is min of meer gelijk: Boot over het netwerk, laad vervolgens een klein besturingssysteem waarmee je de installatie kan uitvoeren van het eigenlijke besturingssysteem aan de hand van een bestand waarin alle antwoorden staan op de vragen die je anders tijdens een handmatige installatie zou hebben beantwoord.

In een volgende weekopdracht gaan we ervoor zorgen dat we het systeem na de eenmalige installatie kunnen configureren en up-to-date kunnen houden via “configuratie management software”.

2.2.1 Lesopdracht

Welke methoden ken je om een systeem (fysiek, dus “bare metal”, of virtueel in een private of public cloud) te voorzien van een besturingssysteem? Maak in tweetallen een lijstje van de methode en bijbehorende voor- en nadelen.

2.2.2 Weekopdracht

We gaan een “zero Touch” installatie uitvoeren van het RHEL8 besturingssysteem, dat wil zeggen: “Het systeem wordt geboot waarna het volledig automatisch wordt voorzien van een besturingssysteem, zonder handmatige stappen”.

We maken hierbij gebruik van VMware workstation en een tweetal systemen: Het eerste systeem bevat de software om het tweede systeem te kunnen uitrollen.

RedHat Enterprise Linux versie 8 (RHEL8)

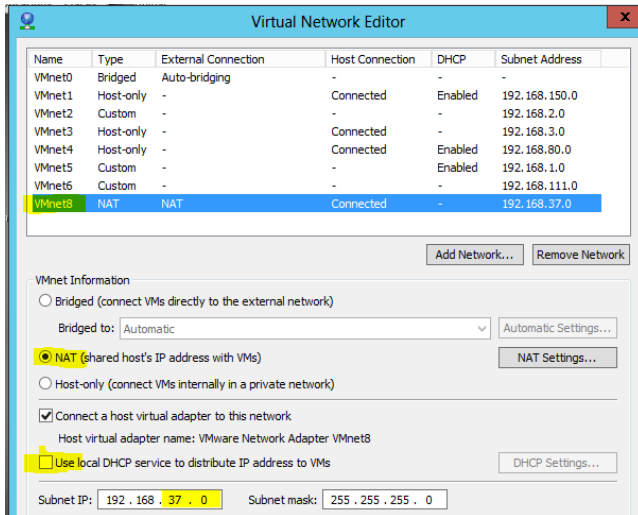
Maak eerst een account aan op de Redhat website om de iso te kunnen downloaden: <https://access.redhat.com/> Kies voor een “Personal” account type.

Download de laatste Red Hat Enterprise Linux versie op <https://developers.redhat.com/products/rhel/download>

Bij het installeren: uncheck “Connect to Red Hat Insights”. Meer details over “subscription management” vind je hier: <https://www.linuxbabe.com/redhat/download-install-rhel8>

VMware Workstation

Het is belangrijk dat jezelf IP-adressen uitdeelt (DHCP) en dat je toegang hebt tot het Internet tijdens de installatie. Middels VMware Workstation kan je dat realiseren door gebruik te maken van een VMnet waarbij NAT wordt gebruikt, maar waarbij je de DHCP server van VMware uitschakelt.



FIGUUR 1: WEL NAT, GEEN DHCP.

Tweede Linux systeem (client.school.test)

Dit wordt het “kale” systeem met alleen een harddisk en ten minste (!) 2 GB memory (liever te veel dan te weinig). Maak hiervoor een nieuwe VM. Paar aanwijzingen hierbij:

- “I will install the operating system later”.
- Linux, RHEL 8 64-bit
- Create a new virtual disk.
- Customize hardware: “Remove Printer” (niet heel belangrijk).
- Netwerk: VMnet3 (of ander vmnet)

Maak hiervan een snapshot met de naam “bare metal”. Dat test gemakkelijk. De verdere installatie gaan we straks geautomatiseerd uitvoeren. Voor nu even klaar met dit systeem. Eigenlijk heb je nu een lege, “kale” server.

Eerste Linux systeem (salt.school.test)

Hieronder wordt de configuratie toegelicht (“echte” installatie volgt later op deze pagina).

Het eerste RHEL-8 systeem bevat de software waarmee deployments kunnen worden uitgevoerd:

DNS: Zodat de nieuwe systemen de server kunnen vinden.

DHCP: Om een ip-adres te krijgen en het adres van de TFTP server door te geven.

TFTP: Om de Linux kernel te downloaden waarmee de installatie van het echte besturingssysteem kan starten.

FTP: Om het besturingssysteem te downloaden.

Kickstart: Om het besturingssysteem te configureren.

Installeer nu eerst RHEL8 met de DVD is, bijvoorbeeld: rhel-8.4-x86_64-dvd.iso ¹

- Software: “Minimal install”
- Klik niet op: “Connect to Red Hat”
- Netwerkaart in “NAT”.
- “local media” als installatie source.

We gaan achtereenvolgens het systeem voorbereiden en vervolgens de verschillende softwarecomponenten installeren en configureren. De kennis die je al hebt opgedaan tijdens eerdere cursussen (Basis ICT-Platformen

¹ Dit is in juni 2021 de meest recente RHEL versie.



TCSB-V1BIPL-18) zoals starten en installeren van services en kennis van nano, dns en dhcp wordt als bekend verondersteld.

Vorbereiding:

Om de installatie wat te vergemakkelijken zetten we de firewall en SeLinux ('disabled' of 'permissive') uit. Dat doe je natuurlijk niet in een productieomgeving! Vergeet niet je server bij RedHat te registreren op je eigen subscription.

```
>>> dnf update
>>> setenforce 0
>>> nano /etc/sysconfig/selinux
        #SELINUX=enforcing
        SELINUX=permissive
>>> hostnamectl set-hostname salt.school.test
```

Geef dan je server een tweede netwerkkaart in een vmnet. Zorg voor deze instellingen:

Instelling	Ens160 (eerste nic)	Ens224 (tweede nic)
Boot protocol	DHCP	Static ip adres: 192.168.3.100
vmnet	NAT	Vmnet3 (mag ook een ander zijn, maar wel gelijk aan tweede server)

Via VMnet3 gaan we straks je client connecten, die kan het Internet bereiken via deze server, dus moeten we nog "NAT" en routing regelen, dat kan onder linux heel makkelijk met:

- `firewall-cmd --change-interface=ens160 --zone=external`
- `firewall-cmd --permanent --change-interface=ens160 --zone=external`
- `firewall-cmd --change-interface=ens224 --zone=internal`
- `firewall-cmd --permanent --change-interface=ens224 --zone=internal`
- `firewall-cmd --set-default-zone=internal`

Bekijk de firewall instellingen nu maar eens via onderstaande commando's en let op "masquerade":

- `firewall-cmd --list-all --zone=internal --zone=external`
- `firewall-cmd --list-all --zone=internal --zone=internal`

DNS:

Je nieuwe client moet straks je server kunnen vinden. Nu minder belangrijk, maar wel noodzakelijk voor de volgende lessen. Kortom, we hebben DNS nodig.

```
>>> dnf makecache
>>> dnf install bind bind-utils
Maak een nieuwe zone aan: "school.test". (bestandsnaam: /var/named/school.test.db)
```

Zorg ervoor dat er tenminste een entry wordt aangemaakt voor:

- Salt.school.test → 192.168.3.100
- Client.school.test → 192.168.37.4

Pas het bestand /etc/named.conf aan.

Test de configuratie van de DNS voordat je verder gaat:

```
>>> named-checkconf /etc/named.conf
>>> named-checkzone school.test school.test.db
```

Start en enable nu je name daemon (named).

Zorg ervoor dat je server nu je eigen DNS server gaat gebruiken ipv de queries forward naar VMWare.



Werkt dit nu goed? (staat het IP-adres in de 'answer section' bij het onderstaande command?)

```
>>> dig client.school.test
```

Werkt het niet? Wat staat er in /etc/resolv.conf ?

Staat de firewall open voor toegang op poort 53 (udp/tcp) vanuit vmnet3?

DHCP

Met Dynamic Host *Configuration* Protocol (DHCP) kan je je nieuwe host configureren met allerlei opties. Uit OSS1 weet je dat dit een router kan zijn, een time server of een DNS server. We maken nu gebruik van een nieuwe optie, namelijk de optie "next server" en "filename" waarmee je het IP adres en bestandsnaam doorgeeft van de TFTP server waar de host de PXE-linux omgeving kan downloaden en starten: Een mini-os dus. Met dit mini-os kan via FTP de echte installatie worden gestart.

```
>>> dnf install dhcp-server
```

```
>>> nano /etc/dhcp/dhcpd.conf
```

```
ddns-update-style interim;  
ignore client-updates;  
authoritative;  
allow booting;  
allow bootp;  
allow unknown-clients;
```

```
# A slightly different configuration for an internal subnet.
```

```
subnet 192.168.3.0 netmask 255.255.255.0 {  
    range 192.168.3.200 192.168.3.254;  
    option domain-name-servers 192.168.3.100;  
    option routers 192.168.3.100;  
    default-lease-time 7200;  
    max-lease-time 14400;  
    next-server 192.168.3.100; # PXE/TFTP server IP  
    filename "pxelinux.0";  
}
```

TFTP

De PXE ROM maakt gebruik van UDP/IP (voor transport), DHCP (voor configuratie) en TFTP (voor bestandsoverdracht). Trivial File Transfer Protocol (TFTP) maakt gebruik van UDP en is een protocol dat veel gebruikt wordt om computers vanaf een netwerk te starten. TFTP ken je misschien doordat je het ook kan gebruiken om andere apparatuur zoals routers, switches, ADSL- en kabelmodems van firmware en configuraties te voorzien. TFTP werd voor het eerst gedefinieerd in 1980. Door de eenvoud van dit protocol zijn clients en servers relatief eenvoudig te implementeren in kleine hoeveelheden geheugen. De TFTP server wordt gestart vanuit xinetd. Dit is een soort "super-server daemon" waarin meerdere services kunnen worden ondergebracht. Wij gebruiken alleen TFTP.

We gaan nu een TFTP daemon installeren en de juiste bestanden kopiëren. Die bestanden komen "gewoon" vanaf de RHEL-8 iso.

```
>>> dnf install tftp tftp-server syslinux wget
```



Dit is dit nog niet genoeg. Je moet ook het bestand `tftp.service` aanpassen:

```
>>> nano /usr/lib/systemd/system/tftp.service
[Unit]
Description=Tftp Server
Requires=tftp.socket
Documentation=man:in.tftpd

[Service]
ExecStart=/usr/sbin/in.tftpd -s /tftpboot
StandardInput=socket

[Install]
Also=tftp.socket
WantedBy=multi-user.target
```

Zorg nu voor de juiste data in je `/tftpboot` directory. Deze data heb je verkregen via het pakket “`syslinux`”:

```
>>> mkdir -p /tftpboot/netboot/
>>> mkdir /tftpboot/pxelinux.cfg
>>> cp -v /usr/share/syslinux/pxelinux.0 /tftpboot
>>> cp -v /usr/share/syslinux/menu.c32 /tftpboot
>>> cp -v /usr/share/syslinux/memdisk /tftpboot
>>> cp -v /usr/share/syslinux/mboot.c32 /tftpboot
>>> cp -v /usr/share/syslinux/chain.c32 /tftpboot
>>> cp -v /usr/share/syslinux/ldlinux.c32 /tftpboot
>>> cp -v /usr/share/syslinux/libutil.c32 /tftpboot
```

Starten maar!

```
>>> systemctl status tftp
>>> systemctl enable tftp
>>> firewall-cmd --zone=internal --add-service=tftp --permanent
>>> firewall-cmd --reload
```

Enable en start de service “`tftp`” (indien nodig). Test je configuratie met:

```
>>> cd /tmp/; tftp 127.0.0.1 -c get memdisk
```

Zie je de file ‘`memdisk`’ nu op disk staan? Zo niet, herstart je server en probeer het opnieuw.

PXE Menu

Als het nieuwe systeem (client.school.test) straks boot ziet de gebruiker een menu. We doen zero touch installation, dus de gebruiker hoeft niet te kiezen, maar de gebruiker wil wel zien wat hij/zij heeft gekozen. Kijk goed naar de laatste regels in het bestand. Snap je wat ze betekenen? Let op, **de rode regel is één enkele regel!**

```
>>> nano /tftpboot/pxelinux.cfg/default
default menu.c32
prompt 0
timeout 30
MENU TITLE HU.School.Test PXE Menu

LABEL RHEL8_x64
```



```
MENU LABEL RHEL 8 X64
KERNEL /netboot/vmlinuz
APPEND initrd=/netboot/initrd.img inst.repo=ftp://192.168.3.100/pub/RHEL8
inst.ks=ftp://192.168.3.100/pub/ks.cfg
```

VSFTP

Nu nog de ftp-website voor het transport van de RHEL-8-iso naar de nieuwe client maken.

```
>>> dnf install vsftpd
>>> mkdir /var/ftp/pub/RHEL8
```

Mount de DVD met de RedHat Source op de plaats /var/ftp/pub/RHEL8 en wel zodanig dat je het volgende ziet:

```
>>> ls /var/ftp/pub/RHEL8/
```

```
[root@salt tmp]# ls /var/ftp/pub/RHEL8/
AppStream  EULA          RPM-GPG-KEY-redhat-release  images
BaseOS     GPL           TRANS.TBL                   isolinux
EFI        RPM-GPG-KEY-redhat-beta  extra_files.json            media.repo
```

Zorg dat de dvd na reboot ook weer gemount is! Met onderstaande tips kom je er wel uit:

- Met blikd vind je het UUID van de dvd-speler
- Voeg een regel toe aan /etc/fstab met daarin de kolommen: UUID, pad (/var/ftp/pub/RHEL8), filesystem (iso9660), en verder nog een paar attributen (ro,user,auto 0 0)

Je hebt nog een paar bestanden uit de ftp-directory nodig voor het mini-os dat wordt gestart via tftp:

```
>>> cp /var/ftp/pub/RHEL8/images/pxeboot/vmlinuz /tftpboot/netboot/
>>> cp /var/ftp/pub/RHEL8/images/pxeboot/initrd.img /tftpboot/netboot/
```

Ten slotte wil je waarschijnlijk nog anonymous toegang toestaan. Dat kan bij RHEL door een aanpassing te maken in de file: /etc/vsftpd/vsftpd.conf Daar kom je vast wel uit.

Alles gedaan? Enable en start ten slotte de vsftpd daemon.

Denk ook aan de firewall! Tip: --add-service=ftp

Wil je weten of de ftp server goed werkt? Install dan een ftp client (dnf install ftp) en log aan met user "anonymous", cd dan naar de directory pub/RHEL8/. Zie je daar je bestanden?

Kickstart

Bijna klaar! Met deze iso kan je veel verschillende systemen uitrollen: desktop, server, grafisch of tekst. Door middel van een kickstart file (zie TCSB-V1BIPL-18) kan je daar zelf invulling aangeven.

Hieronder de kickstartfile die we gebruiken tijdens dit lab voor de uitrol van een niet-grafisch systeem (minder installatietijd). Maak het onderstaande kickstart bestand aan:

```
>>> nano /var/ftp/pub/ks.cfg
# Network information
network --bootproto=dhcp --device=ens160 --activate --hostname=client.school.test
# Firewall configuration
firewall --enabled
# Reboot after installation
```



```
reboot
# Use FTP installation media
url --url="ftp://192.168.3.100/pub/RHEL8"
# Root password [i used here "Pa$$w0rd"]
rootpw --plaintext Pa$$w0rd
# Use graphical install
graphical
firstboot --disable
# System keyboard
# of deze u.s. international? keyboard us-aRHEL
keyboard us
# System language
lang en_US
# SELinux configuration
selinux --permissive
# Installation logging level
logging --level=info
# System timezone
timezone Europe/Amsterdam
# System bootloader configuration
bootloader --location=mbr
clearpart --all --initlabel
part swap --asprimary --fstype="swap" --size=1024
part /boot --fstype xfs --size=200
part pv.01 --size=1 --grow
volgroup rootvg01 pv.01
logvol / --fstype xfs --name=lv01 --vgname=rootvg01 --size=1 --grow

%packages
@core
net-tools
%end
%post
%end
```

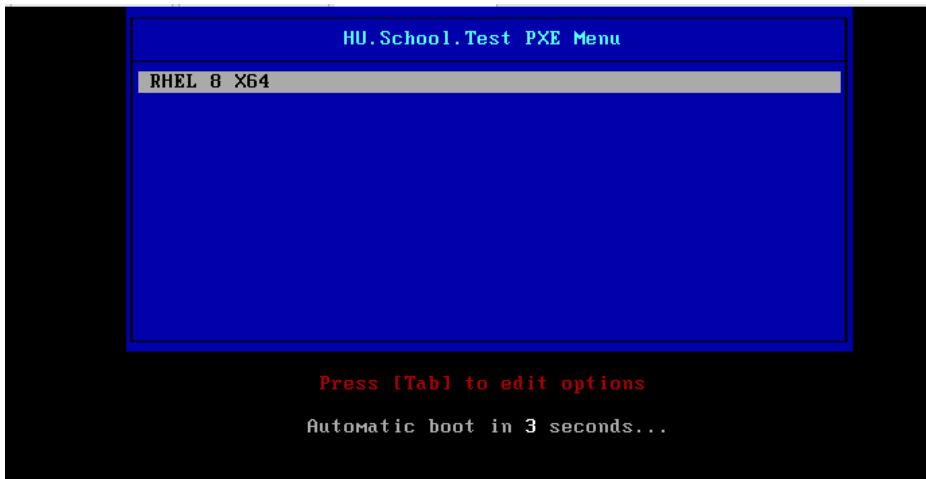
Testen

Draaien dhcpd, tftp, vsftpd en is de iso gemount in /var/ftp/pub? Dan is het tijd om de installatie van de client te testen. Start de client-VM (druk indien nodig op F12)



FIGUUR 2: PXE BOOT MIDDELS DHCP.

Als alles goed is gegaan zie je het onderstaande PXE menu en wordt een linux kernel geladen waarmee de installatie zal beginnen:



FIGUUR 3: BOOTMENU.

Nazorg

Neem na afloop van de installatie de host client.school.test op in de DNS. Hmm, is dat wel heel slim met een dynamisch ip adres? Weet je hoe het beter zou kunnen? Iets met dhcp en reservering?!
Maak een snapshot van beide systemen. Je hebt ze nodig in een volgende opdracht.

Optioneel (als je het leuk vindt)²

Je kan hetzelfde nu bouwen onder Windows: Gebruik hiervoor naast de ISO van een Windows client ook: Windows AD, DNS, DHCP, WDS en eventueel WSIM voor het maken van de answer file. Op het Internet bestaan hiervoor veel tutorials...

2.2.3 Lesstof tentamen

Wat moet je weten voor het tentamen over dit onderwerp, dus hoe kan je je het beste voorbereiden?

- **Leer** alle informatie die je vindt in de powerpoint presentatie en dit werkboek over dit onderwerp. De precieze inhoud van de configuratiefiles hoeft je niet te kennen. Wel welke services nodig zijn, waar ze voor dienen en hoe je deze globaal configureert.
- **Lees** alle informatie die wordt aangeboden onder het kopje "Informatiebronnen".

² Mocht iemand dit willen doen met Windows 7 moet er rekening gehouden worden met het toevoegen van imagex aan windowspe om de install.wim te kunnen applyen. Tussen Windows 7 en 8 wordt er overgegaan naar een DISM commando die dit doet.

Daarnaast nog moet een Windows image altijd met dezelfde versie development kit gemaakt worden (op dezelfde versie Windows). Je kunt niet een Windows 8 image met de Windows 7 WAIK kit maken.

Bij het maken van een image gebruik Windows DISM. Als hierbij een andere versie DISM wordt gebruikt dan bij de Windows versie van de image dat gaat dit problemen geven tijdens de installatie. Dit geldt ook voor bijvoorbeeld Windows 7 en 7.1.

Als dit met een development kit gedaan wordt kun je ook Windows uitrollen met de RHEL server. Dit vereist wel syslinux 5.01 of hoger.



2.2.4 Informatiebronnen

- Vergelijkbaar met de configuratie hierboven, maar wel minder volledig!
<https://www.unixmen.com/install-pxe-server-RHEL-7/>
- Meer informatie over tftp: <http://www.bo-yang.net/2015/08/31/RHEL7-install-tftp-server>



2.3 Week 2, College 1: Modules | Security

2.3.1 Om de les mee te beginnen

Eerst nog even dit: Functies zijn belangrijk in ieder stuk Python code. Kortom, basiskennis ook voor SNE-ers. Hieronder daarom wat oefeningen van (relatief makkelijk) tot moeilijk!

1. Write a Python function to return the highest of a list of numbers provided.
Sample List : (8, 2, 3, -1, 7)
Expected Output : 8
2. Schrijf een functie die je een leeftijd als input heeft en die je geboortjaar teruggeeft (je mag een variabele `year_now=2017` gebruiken), waarna je het print.
Sample input for function in 2017: 50
Expected Output: 1967
3. Zelfde als de vorige vraag, maar nu het huidige jaar berekenen (iets met `import datetime`, google maar even?).
4. Write a Python function to check whether a number is in a given range.
5. Write a Python program to print the even numbers from a given list.
Sample List : [1, 2, 3, 4, 5, 6, 7, 8, 9]
Expected Result : [2, 4, 6, 8]
6. Write a Python function to multiply all the numbers in a list.
Sample List : (8, 2, 3, -1, 7)
Expected Output : -336
7. Write a Python program to reverse a string.
Sample String : "1234abcd"
Expected Output : "dcba4321"
8. Write a Python function that takes a number as a parameter and check whether the number is prime (priemgetal) or not.

Antwoorden achterin dit werkboek.

2.3.2 Lesopdracht

Je hebt in de les gezien hoe je een module kan maken met daarin je eigen functies.

1. Maak nu je eigen module "mymodule" en zorg ervoor dat je de functie "dobbelsteen" vanuit een python script kunt aanroepen.
2. Als dat werkt maak je nog een nieuwe functie in deze module met de naam "yesno". Als je deze functie aanroept dan moet de functie altijd "yes" of "no" teruggeven. Welke van de twee je terugkrijgt vanuit de functie is willekeurig. Tip: gebruik uit de module `random` de functie "choice" in je eigen code van de module (zie ook http://www.tutorialspoint.com/python/number_choice.htm).



3. Ten slotte: kies logische variabele namen en documenteer iedere functie zodat de gebruiker ook later nog snapt welke argumenten je kan meegeven en wat de functie teruggeeft.

2.3.3 Weekopdracht

1. Beheerders hebben vaak te maken met scripts waarin wachtwoorden staan. Niet zo veilig:
 - Maak een scriptje met de naam 'write_password.py'. Dit script bevat de volgende functionaliteit: Vraag de gebruiker om zijn/haar wachtwoord. Schrijf dit wachtwoord in het bestand 'password.txt'.
 - Maak nu een tweede scriptje met de naam 'read_password.py'. Dit script bevat de volgende functionaliteit: Vraag de gebruiker om zijn/haar wachtwoord en print "Wachtwoord OK" als het wachtwoord juist is en "Wachtwoord FOUT" als het wachtwoord fout is. Je moet daarbij het bestand password.txt gebruiken.
 - Oei, het wachtwoord is leesbaar in het bestand, dat is natuurlijk niet zo veilig. Daarom moet je de beide scripts zodanig aanpassen dat niet het wachtwoord, maar een (veilige) hash van het wachtwoord wordt geschreven in het bestand 'password.txt'.
2. Een hacker heeft een bestand 'password.txt' gevonden. Dit bestand bevat de volgende inhoud:
f00bdeb2cd9da240a57c951fdflbcba509fd0cd83c5e5ad9a669de12
 - De hacker weet dat de inhoud tot stand is gekomen met behulp van de onderstaande code waarbij op de plaats van de vraagteken een hoofdletter heeft gestaan. Weet jij welke hoofdletter dit zou is geweest?

```
print(hashlib.sha224(b"?").hexdigest())
```
 - De vorige opdracht kon je nog wel oplossen door alle letters te proberen. Dat wordt een stuk lastiger bij de volgende opdracht. Het wachtwoord heeft nu namelijk drie letters. De onderstaande code vraagt je om 100x een oplossing te proberen.

```
import hashlib
p="326f13e25448c99ccae78b0afalc84e6fae2d8a587976df3a86628ae"

for i in range(100):
    ww=input("Welk wachtwoord van 3 letters hoort hierbij?")
    if hashlib.sha224(ww.encode()).hexdigest() == p:
        print("Je hebt het wachtwoord gevonden! Het was:", ww)
        break
    else:
        print("Helaas, nog een poging (poging", i, ")")
```

Pas nu de loop aan zodat je alle opties automatisch probeert dus van AAA tot ZZZ.

Tip: Gebruik de onderstaande tabel. Daarin zie je dat A hetzelfde is als het getal 65. Je kan de A ook printen door chr(65) te printen. Zo kan je een loop maken van A-Z.



Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	\$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(72	48	H	104	68	h
9	09	Horizontal tab	41	29)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	61	3D	=	93	5D]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

TABEL 1: BRON:

[HTTP://WWW.CDRUMMOND.QC.CA/CEGEP/INFORMAT/PROFESSEURS/ALAIN/IMAGES/ASCII1.GIF](http://www.cdrummond.qc.ca/cegep/informat/professeurs/alain/images/ascii1.gif)

3. Je wilt Alice een bestand sturen met de onderstaande inhoud:

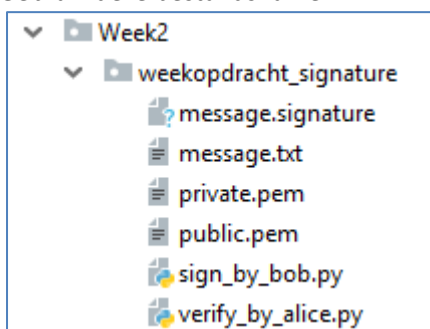
Hi Alice, mijn bankrekeningnummer is NL43RABO032233445, maak maar 870 euro over voor onze vakantie naar deze IBAN. Hartelijke groet, Bob

Je wilt natuurlijk niet dat tijdens het transport van het bestand het bankrekeningnummer kan worden aangepast. Dit kan je voorkomen door het bestand te ondertekenen. Maak daarom 2 python scripts: Een script dat een keypair maakt en vervolgens het bestand leest en er een digitale ondertekening bij maakt.

Een script dat de digitale ondertekening leest en met behulp van de publieke sleutel de ondertekening controleert. Als de ondertekening in orde is, print je "Signature OK". Als het bestand is aangepast print je "Signature NOK!!!".

Een paar tips om je te helpen:

Gebruik deze bestandsnamen:





Hoe een key pair (public + private) moet maken vind je hier:

<https://www.pycryptodome.org/en/latest/src/examples.html#generate-public-key-and-private-key>

Je hebt de module “pycryptodome” nodig.

Hoe je met een key pair een bestand kan ondertekenen vind je hier:

https://pycryptodome.readthedocs.io/en/latest/src/signature/pkcs1_v1_5.html

Let op, je moet de tekst eerst in een bestand zetten bv. met notepad in het bestand “message.txt”, daarna lezen als binary file:

```
f = open(message_file, "rb") # 'rb' !
```

en daarna ondertekenen.

Het resultaat”, de variabele “signature” moet je naar een bestand schrijven. Doe dit weer binary, dus bijvoorbeeld: f=open(signature_file, 'wb')

Hoe je een signature kan controleren lees je ook hier:

https://pycryptodome.readthedocs.io/en/latest/src/signature/pkcs1_v1_5.html

Let op, lees nu ook weer eerst je message.txt bestand (binary mode weer), maak dan de hash en kijk of de hash gelijk is zoals beschreven op de webpagina.

Er staat wel een foutje op de webpagina bij de regel waar de hash wordt gemaakt. Dit moet zijn:

```
h = SHA256.new(message) # not SHA as written in the documentation
```

2.3.4 Lesstof tentamen

Wat moet je weten voor het tentamen over dit onderwerp, dus hoe kan je je het beste voorbereiden?

- **Leer** alle informatie die je vindt in de powerpoint presentatie over dit onderwerp.
- **Leer** de algemene theorie zoals te lezen op: <http://www.laurentluce.com/posts/python-and-cryptography-with-pycrypto/> (de python code hoeft je niet te kennen, maar je moet de code wel kunnen begrijpen).

2.3.5 Informatiebronnen

Modules: <https://docs.python.org/3/tutorial/modules.html>

Security: <http://www.blog.pythonlibrary.org/2016/05/18/python-3-an-intro-to-encryption/>



2.4 Week 2, College 2: Configuration Management Using SaltStack

De servers zijn geïnstalleerd, maar wat nu?! Natuurlijk wil je extra software installeren, users aanmaken, services starten etc. Hiervoor maken we gebruik van “configuration management” (CM) software. Tijdens deze cursus maken we gebruik van SaltStack. In het lab gaan we deze week software installeren en daarna gebruiken om op afstand commando’s uit te voeren op de clients. Het gaat allemaal stap-voor-stap. In de komende week moet je meer zelf gaan uitzoeken. De kennis van deze week is daarbij onmisbaar.

2.4.1 Lesopdracht

Welke Configuration Management software ken jij en wat zijn de verschillen tussen deze tools?

2.4.2 Weekopdracht

Linux1: Master installatie.

De master kan je vergelijken met de “server”, de minions zijn de te beheren systemen, oftewel de “clients”. We installeren eerst de master. Hierbij gaan we uit van het gebruik van een shell met root rechten.

```
# salt zit niet in de base repo
>>> dnf install https://repo.saltproject.io/py3/redhat/salt-py3-repo-
latest.el8.noarch.rpm

# install salt-master
>>> dnf install salt-master

>>> systemctl enable salt-master
>>> systemctl start salt-master

# zet de firewall open op de nic in de default (=internal) zone
>>> firewall-cmd --permanent --add-service=salt-master
>>> firewall-cmd --reload
```

We maken van deze server ook nog een “repository” waar onze client straks de software vandaan kan halen. Daarvoor zet je in de ftp-directory de namen en locaties van de twee repositories die je nodig hebt. Dit gaan we straks vanaf de client laden, dus:

```
>>> cat > /var/ftp/pub/salt_school_test.repo
[salt_school_test-BaseOS]
name=Local RHEL 8 BaseOS repository on salt.school.test
baseurl=ftp://salt.school.test/pub/RHEL8/BaseOS/
enable=1
gpgcheck=1
gpgkey=ftp://salt.school.test/pub/RHEL8/RPM-GPG-KEY-redhat-release

[salt_school_test-AppStream]
name=Local RHEL 8 AppStream repository on salt.school.test
baseurl=ftp://salt.school.test/pub/RHEL8/AppStream/
enable=1
gpgcheck=1
gpgkey=ftp://salt.school.test/pub/RHEL8/RPM-GPG-KEY-redhat-release
```



Linux client: Minion installatie RHEL8

Nu de client, ook wel minion genoemd. De installatie is vergelijkbaar met die van de master. Echter, maak je geen zorgen om de firewall. De client heeft alleen uitgaande connecties naar de master en niet andersom.

Voordat je gaat installeren: Geef je client nu een vast ip adres, bv 192.168.37.4. Je kan de client nu ook toevoegen aan je dns door een entry toe te voegen aan `/var/named/school.test.db`

Test de configuratie voordat je verder gaat: is het internet bereikbaar op je client en kan je de master pingen?

```
>>> ping salt.school.test
```

Zo ja, ga dan verder:

```
# Voeg onze server toe aan de lijst met repositories:
>>> curl -o /etc/yum.repos.d/salt_school_test.repo
ftp://192.168.3.100/pub/salt_school_test.repo
>>> dnf install https://repo.saltproject.io/py3/redhat/salt-py3-repo-
latest.el8.noarch.rpm

# installeer nu python
>>> dnf install -y https://repo.saltstack.com/py3/redhat/salt-py3-repo-
latest.el8.noarch.rpm

# nu kan je de salt-software installeren
>>> dnf -y install salt-minion

# pas de '#master: salt'-regel aan in /etc/salt/minion indien nodig.
>>> vi /etc/salt/minion
>>> systemctl enable salt-minion
>>> systemctl start salt-minion
```

Je client heeft via DHCP een IP-adres gekregen en een key opgestuurd naar de master. Deze moet je nog accepteren vanaf de master. Bekijk eerst alle keys met:

```
>>> salt-key -L
```

Accepteer nu een specifieke key³ met `salt-key -a <hostname>`, of accepteer alle keys met:

```
>>> salt-key -A
```

Als je vanaf de master je minion niet kan bereiken moet je deze opnemen in de DNS, waarschijnlijk in `/var/named/school.test.db` of anders toevoegen via je `/etc/hosts` file. Natuurlijk allemaal niet zo'n fraaie oplossing, immers dit is extra handmatig werk en als je client een ander IP-adres krijgt klopt het niet meer. Weet jij hoe dit beter zou kunnen?

Test nu je installatie vanaf de master met:

```
>>> salt '*' test.ping
```

³ Je kunt ook een autosign file maken waarbij je bijvoorbeeld wildcards gebruikt `*.school.test` Als minions die met deze domeinnaam connectie maken met de master, dan worden deze automatisch geaccepteerd.



Krijg je een response van de client? Bij een 'True' response is de installatie gelukt.

Derde Linux VM: Minion installatie RHEL8 met behulp van kickstart

Wel een beetje jammer natuurlijk dat je na een automatische installatie met PXE en kickstart nu handmatig de saltstack-client moet installeren! Dat moet beter kunnen en dat is ook zo. Doe dit door de onderstaande kickstart configuratie toe te voegen aan je **kickstart file**. Denk ook aan de naam van de host, kies voor client.school.test. Verwijder evt. eerst de huidige client.school.test van de master met behulp van: `salt-key -d client.school.test` Zorg je er wel voor dat de minion straks het Internet kan bereiken? De commando's worden automatisch uitgevoerd nadat je het OS hebt geïnstalleerd en vervolgens kan je je clients op afstand en automatisch beheren!

Test dan je OS-installatie van een nieuwe RHEL client. Hier de info voor je kickstart file:

```
%post
curl -o /etc/yum.repos.d/salt_school_test.repo ftp://192.168.3.100/pub/salt_school_test.repo
dnf install -y https://repo.saltstack.com/py3/redhat/salt-py3-repo-latest.el8.noarch.rpm
dnf install -y salt-minion
sed -i "/#master:/c\master: salt.school.test" /etc/salt/minion
systemctl enable salt-minion
%end
```

Denk ook weer aan het accepteren van de key, de namenresolutie en het testen via salt '*' test.ping

Windows1: Minion installatie voor Windows

Je kan van veel verschillende besturingssystemen een Saltstack client maken. Dat geldt ook voor MS Windows Server 2019. Hieronder de stappen die nodig zijn voor de -handmatige- installatie:

- Installeer een Windows 2019 Server of maak gebruik van een VM die je nog hebt van een eerdere cursus.
- VMnet 3 gebruiken.
- Zorg ervoor dat je server een vast ip adres heeft in het 192.168.3.0/24 netwerk, bv 192.168.3.205. Denk ook aan je default route naar de master en laat ook de DNS via de master verlopen.
- Als het goed is kan je naar www.google.nl met IE en krijg je response als je pingt naar de master:

```
PS C:\Users\Administrator> ping salt.school.test

Pinging salt.school.test [192.168.37.3] with 32 bytes of data:
Reply from 192.168.37.3: bytes=32 time<1ms TTL=64
Reply from 192.168.37.3: bytes=32 time<1ms TTL=64
```

- Download nu de SaltStack minion software:
<https://docs.saltstack.com/en/latest/topics/installation/windows.html> (Python3 AMD64:)
- Installeer de software (minion name: "windows1.school.test", start de service..).
- Pas de naam van de master aan met notepad in de file C:\salt\conf\minion en herstart de service.
- Accepteer de key op de master.
- Test je clients met: salt '*' test.ping



Remote Execution

Je kan nu op afstand commando's uitvoeren. Voer de onderstaande commando's uit en schrijf op wat deze eigenlijk doen. Niet alles werkt goed, snap je waarom sommige commando's niet zo goed werken op alle minions?

```
# ping
salt windows1.school.test test.ping
salt '*' test.ping

# remote command
salt client.school.test cmd.run pwd
salt '*' cmd.run pwd # niet alles wordt ondersteund onder windows

# Disk Usage
salt '*' disk.usage

# Permissions
salt client.school.test acl.getfacl /root/

# Status of all minions
salt-run manage.status

# grains, list all items
salt '*' grains.ls

# Grain(s)
salt '*' grains.get ip4_interfaces
salt 'windows1.school.test' grains.get osfullname
salt 'windows1.school.test' grains.items
salt '*' grains.item saltversion os_family
```

Configuration Management

Mooi, we kunnen onze minions op afstand “besturen”. Door gebruik te maken van state-files kan je ervoor zorgen dat je in een bestand kan opgeven hoe je minion geconfigureerd is. Als voorbeeld gaan we ervoor zorgen dat je een python script plaatst op je minion en deze vervolgens uitvoert op de minion.

Default verwacht de master je State-files en andere bestanden onder `/srv/salt/`, maar dat kan je veranderen in `/etc/salt/master`.

Maak nu eerst een bestand `/srv/salt/scriptje.py` aan op de master met de onderstaande inhoud:

```
#!/bin/python3
f=open('output.log', 'a')
f.write('Hello world\n')
f.close()
```

Snap je wat hierboven gebeurt? Waar wordt `output.log` geplaatst denk je als je het start?

We gaan de acties “bestand kopiëren” en “bestand runnen” beschrijven via YAML in het bestand `/srv/salt/example.sls`. Dat is een “state-file”:

```
mijnpython:
```



```
file.managed:
  - name: /tmp/scriptje.py
  - source: salt://scriptje.py
  - mode: 777
cmd.run:
  - name: /tmp/scriptje.py
  - cwd: /tmp/
```

Snap je wat hierboven gebeurt? (salt:// verwijst trouwens naar de root van je bestanden, default /srv/salt/)

Je kan de goede werking nu testen vanaf de master met het commando (let op, niet de extensie “.sls”) gebruiken bij de uitvoer:

```
>>> salt client.school.test state.apply example
```

Wat gebeurt er als je het bestand /tmp/scriptje.py verwijdert op de minion en je doet dit commando nog een keer?

Bijna klaar! In het bovenstaande commando geef je heel expliciet aan welke state-file je wilt toepassen op welke minion. Als je veel state files hebt is dat niet handig. Je kan een bestand maken, top.sls waarin je verwijst naar alle onderliggende state-bestanden. Door vervolgens top.sls uit te voeren worden alle configuraties gechecked en uitgevoerd:

Maak het bestand /srv/salt/top.sls aan:

```
base:
  '*':
    - example
```

Hier staat eigenlijk: *voer op alle minions in de base-environment de “state” door zoals omschreven in het bestand example.sls*

Je kan dit doorvoeren op client.school.test met het onderstaandecommando waarin je de state-file niet noemt:

```
>>> salt client.school.test state.apply
```

Ten slotte willen we er nog voor zorgen dat deze configuratie alleen wordt uitgevoerd op de RHEL systemen.

Voer het onderstaande commando uit:

```
>>> salt '*' grains.get os
```

Je leert van de uitvoer van dit commando dat de linux systemen van het os “RHEL” zijn. Verander nu de top.sls naar:

```
base:
  'G@os:RedHat':
    - example
```

En run salt '*' state.apply. De foutmelding kan je negeren: er wordt alleen aangegeven dat er voor het windows1 systeem geen state-files zijn gevonden en dat klopt.

Kan je er nu voor zorgen dat het script ook onder windows wordt uitgevoerd?

Je mag gebruik maken van de onderstaande informatie:



```
rienk@salt:/srv/salt
File Edit View Search Terminal Help
GNU nano 2.3.1 File: win.sls
mijnpython:
  file.managed:
    - name: c:\\temp\\scriptje.py
    - source: salt://scriptje.py
    - makedirs: True

  cmd.run:
    - name: c:\\salt\\bin\\python.exe c:\\temp\\scriptje.py
    - cwd: c:\\temp
```

We zijn nu klaar: je kan met 1 commando (salt '*' state.apply) alle state files activeren op je minions.

2.4.3 Lesstof tentamen

Wat moet je weten voor het tentamen over dit onderwerp, dus hoe kan je je het beste voorbereiden?

- **Leer** alle informatie die je vindt in de powerpoint presentatie en dit werkboek over dit onderwerp. De precieze inhoud van de configuratiefiles hoeft je niet te kennen. Wel welke services nodig zijn, waar ze voor dienen en hoe je deze globaal configureert.
- **Leer** de theorie over SaltStack zoals te vinden op Canvas over dit onderwerp.

2.4.4 Informatiebronnen

- Tutorial SaltStack: <https://blog.woohoovcs.com/2019/11/introduction-to-saltstack/>



2.5 Week 3, College 1: Database | CGI

Tot nu toe heb je gegevens opgeslagen in een bestand, in memory of in een CSV-bestand. Voor bepaalde toepassingen, zoals het schrijven van logging is dat voldoende, maar als je applicatie wat groter wordt of je wilt bepaalde gegevens aanpassen, dan moet je gaan werken met een database.

Tijdens deze week gaan we dit oefenen. Dit gaat het meest eenvoudig met SQLite, maar de gebruikte technologie is ook goed toepasbaar voor grotere databases zoals MySQL.

Bovendien gaan we aan de slag met de Common Gateway Interface (CGI): Een manier om webapplicaties te bouwen met behulp van Python. Voor beheerders komt dit goed van pas, want niet iedereen heeft Python geïnstalleerd op zijn/haar computer, maar een webbrowser heeft iedereen op elk platform.

Alles bij elkaar kan je al een prima applicatie bouwen: website als front-end, python code voor de functionaliteit en de data wordt opgeslagen in een database!

Vorbereiding

- Installeer SQLite op je computer: <https://www.sqlite.org/download.html> . Kies in het geval van Windows niet voor de DLL, maar voor de zip-file waar sqlite3.exe in is te vinden.
- Daarnaast kan je nog een grafische manager voor de SQLite database installeren, zoals bijvoorbeeld de Firefox add-on <https://addons.mozilla.org/addon/sqlite-manager>
- Voor de CGI-opdracht is het nodig om te beschikken over een webserver waarmee je Python-scripts kunt uitvoeren. Voor details, zie de presentatie.

2.5.1 Lesopdracht

Misschien ook al genoemd in de presentatie:

- Installeer SQLite op je computer. Gebruik daarvoor de package manager “dnf”.
- Maak een SQLite database met daarin een enkele tabel voor het opslaan van Naam/Adres/Woonplaats data. Gebruik hiervoor de SQLite3.exe console applicatie.
- Vul de tabel met drie regels (dus ook drie personen).
- Klaar? Maak nu een Python programma waarin je iemand vraag om zijn/haar voornaam. Daarmee zoek je de bijbehorende gegevens zoals adres en woonplaats op en toon deze informatie vervolgens op het scherm. Gebruik eventueel de informatie van: <https://docs.python.org/2/library/sqlite3.html>

2.5.2 Weekopdracht

SQLite

1. Maak een script dat random 1000 getallen genereert. De getallen moeten >0 en <500 zijn. Zorg ervoor dat deze getallen in een SQLite database worden geplaatst. Tabel heeft als definitie: CREATE TABLE nummers (getal INTEGER)
2. Zorg ervoor dat deze getallen *middels een database query* worden **opgeteld** en print het resultaat op de console.
3. Print daaronder het **gemiddelde** van deze getallen daarnaast.
4. Print daaronder het laagste en hoogste getal (ook dat kan met een database query).

Common Gateway Interface (CGI)

Met de module Tkinter kan je een mooie GUI maken die platformonafhankelijk is. Helaas is dat niet zo heel eenvoudig en bovendien draait je applicatie alleen maar als Python is geïnstalleerd. Daarom is het soms



verstandiger om je Python applicatie als webapplicatie aan te bieden, dan is het direct toegankelijk vanaf iedere client. Dit is precies wat we gaan doen tijdens deze opdracht.

1. Zorg ervoor dat het onderstaande script goed functioneert in combinatie met jouw webserver en web client en zorg dat je de output bekijken met een webbrowser. Informatie over hoe je dit kan doen kan je onder andere hier vinden: <http://support.microsoft.com/kb/276494> of <http://www.w3resource.com/python/cgi-programming.php> of <https://techexpert.tips/iis/enable-python-iis/>

```
print ('Content-type: text/html\n\n')
print ('<HTML><HEAD><TITLE>Python Sample CGI</TITLE></HEAD>')
print ('<BODY>')
print ('<H1>This is a header</H1>')
print ('<p>') #this is a comment
print ('See this is just like most other HTML')
print ('<br>')
print ('</BODY>')
```

2. Zorg ervoor de onderstaande webpagina's goed functioneren in jouw omgeving:

- a. Gebruiker.html

```
<html>
  <head>Hello Wie?</head>
  <body>
    <form action="gebruiker.py" method="post">
      First Name: <input type="text" name="first_name"><br />
      Last Name: <input type="text" name="last_name" />
      <input type="submit" value="Submit" />
    </form>
  </body>
</html>
```

- b. Gebruiker.py

```
#!/usr/bin/python
import cgi

# Create instance of FieldStorage
form = cgi.FieldStorage()

# Get data from fields
firstname = form.getvalue('first_name')
lastname = form.getvalue('last_name')

print ("Content-type:text/html\n\n")
print ("<html><head><title>Hello - Second CGI Program</title></head><body>")
print ("<h2>Hello %s %s</h2>" % (firstname, lastname))
print ("</body></html>")
```

3. Maak nu een fout in de webpagina door bijvoorbeeld geen "print", maar "pint" te gebruiken op een bepaalde regel. Wat valt je op aan de foutmelding in de webbrowser?
4. Wat valt je op aan de foutmelding in de webbrowser als je je webpagina verandert van `import cgi` naar:
`import cgi, cgitb`
`cgitb.enable()`



5. Maak je eigen -eenvoudige- webmail-applicatie. Dat wil zeggen: Maak een HTML-webformulier waarin je een ontvanger, subject en tekstveld voor de body kunt invoeren en zorg ervoor dat het bericht per mail kan worden afgeleverd.

Wat betreft het verzenden van e-mail. Hieronder wat adviezen:

- Stuur alleen mail naar jezelf.
- Binnen of buiten de HU: Het gedrag van de HU-SMTP-mailserver is verschillend. Het kan dus zijn dat je script thuis goed werkt, maar op school niet. Gmail gebruiken werkt altijd hetzelfde, maar is iets minder gemakkelijk.
- Voor tips: Lees deze pagina en gebruik de voorbeelden: <https://julien.danjou.info/sending-emails-in-python-tutorial-code-examples/>

2.5.3 Lesstof tentamen

Wat moet je weten voor het tentamen over dit onderwerp, dus hoe kan je je het beste voorbereiden?

- **Leer** alle informatie die je vindt in de powerpoint presentatie over dit onderwerp.
- **Lees** de informatie die wordt aangeboden onder het kopje "Informatiebronnen".
- **Zorg dat je een fout kan halen** uit de Python code waarin een CGI-script / HTML-pagina wordt getoond.
- Zorg dat je zelf CGI-scripts hebt gemaakt zodat je **getoonde code van anderen goed begrijpt** en kan **aanpassen/verbeteren**.
- Begrijp hoe je veilig moet programmeren met databases om SQL-injectie te voorkomen.
- Zorg dat je **alle stappen kent** van het maken van een SQLite-database en -tabel tot de INSERT en SELECT van data.

2.5.4 Informatiebronnen

SQLite: <https://docs.python.org/2/library/sqlite3.html>

SQLite: <https://www.youtube.com/watch?v=o-vsdfCBpsU> en <https://www.youtube.com/watch?v=qfGu0fBfNBs>

CGI: <http://www.tutorialspoint.com/python/python CGI programming.htm> &

CGI: <http://docs.python.org/3/library/cgi.html?highlight=cgi#cgi>

CGI op IIS: <https://techexpert.tips/iis/enable-python-iis/>

Mail sturen vanuit Python: <https://julien.danjou.info/sending-emails-in-python-tutorial-code-examples/>



2.6 Week 3, College 2: Configuration Management (NextCloud)

Vorige week heb je ervaring opgedaan met de basisprincipes van SaltStack. Deze week gaan we een stap verder. Op het Internet staan veel “tutorials”, “how to’s” en “filmpjes” waarin de configuratie van software wordt uitgelegd. Zou het niet veel handiger zijn als je een paar SaltStack files hebt voor jouw platform waarna de hele installatie en configuratie wordt uitgevoerd? Aan de hand van de YAML-files kan je ook precies zien wat er gebeurt, kortom een werkende omgeving en ook nog gedocumenteerd! Door middel van deze weekopdracht doe je de noodzakelijke kennis op die je nodig hebt voor de eindopdracht.

2.6.1 Lesopdracht

--

2.6.2 Weekopdracht

Tijdens deze weekopdracht ga je de software “**NextCloud**” installeren op het RHEL8 platform. Dit is een vervanging voor GoogleDrive, Dropbox en Onedrive zonder mogelijke “pottenkijkers” zou je kunnen zeggen.

Je installeert aan de hand van een tutorial die je “omzet” naar Saltstack configuratie. De NextCloud server gebruiken we nog in een volgende les.

Het uiteindelijke doel is dat je een ingerichte NextCloud server kan installeren op het RHEL8 platform zonder dat je daarvoor handmatige acties moet uitvoeren. Zorg dat je in je configuratie gebruik maakt van: YAML, Grains, State files en Pillars.

Zet de onderstaande tutorial om naar een automatische Saltstack configuratie als vervolgstap op de installatie uit de vorige les...

Bron: https://docs.nextcloud.com/server/latest/admin_manual/installation/example_centos.html

Paar opmerkingen hierbij:

- Een eventuele presentatie in de les is bedoeld om je een beetje op weg te helpen. Daarmee wordt lang niet alles duidelijk, het is de bedoeling dat je zelf gaat puzzelen.
- **Tip:** Maak een snapshot van je client, installeer het eerst handmatig. Werkt ‘t? Ga dan alle stappen automatiseren met saltstack, steeds 1 stap erbij.
- **Tip2:** Maak verschillende saltstack files en test ze individueel, bijvoorbeeld: state-file voor apache met php, state-file voor mariadb en een state-file voor NextCloud.
- **Tip3:** Werkt ‘t? Maak het dan “mooi” door pillars toe te voegen, bijvoorbeeld voor database naam, user, password.
- **Tip4:** Niet verplicht, maar voor degenen die echt alles willen automatiseren: kijk eens in de directory `/var/www/html/nextcloud` daar zit de file `occ` en als je deze aanroept met de optie `maintenance:install` kan je ook de configuratie van de database automatiseren, dat is een stap van de handmatige setup, zie hieronder.
- **Tip5:** Denk ook aan “include” om ervoor te zorgen dat de verschillende statefiles worden verwerkt, bv: nextcloud state file kan de statefiles van de database en apache “includen”.
- **Tip6:** Denk ook aan “require” om ervoor te zorgen dat een module eerder wordt uitgevoerd dan een andere module.

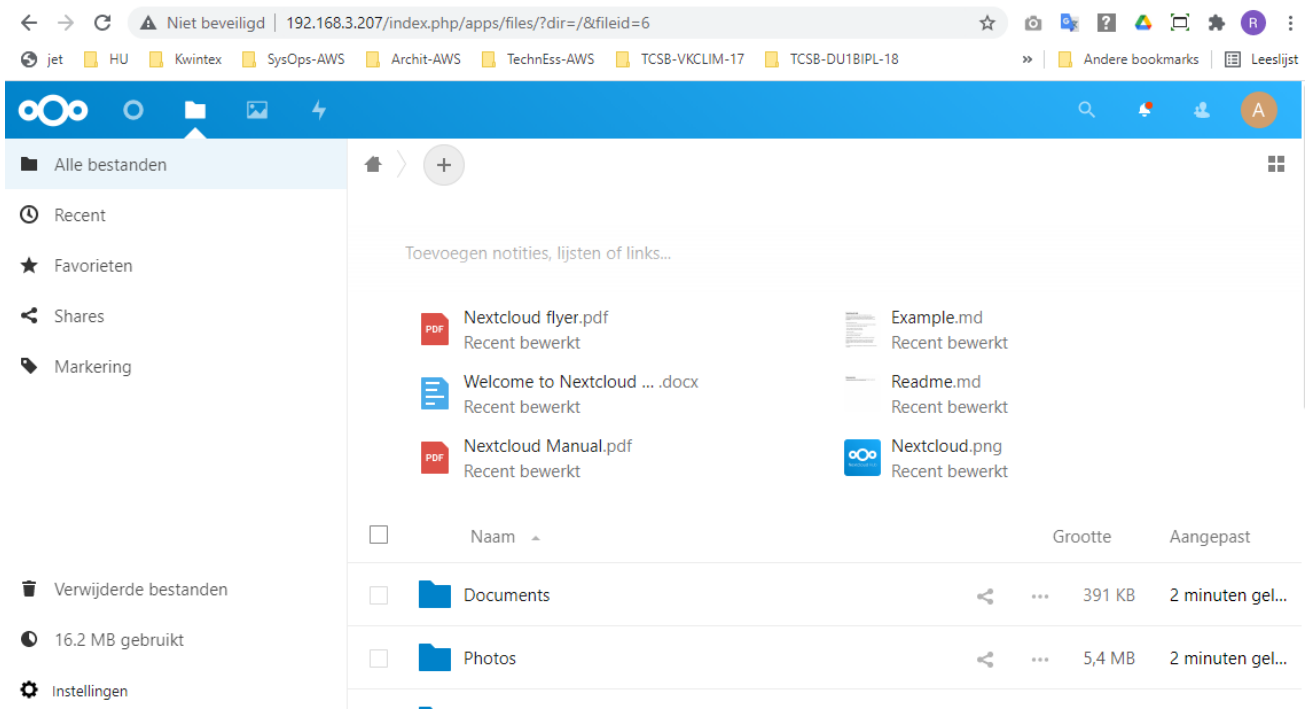


Benodigde packages:	<pre>dnf install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm</pre>
	<pre>dnf install -y epel-release yum-utils unzip curl wget \ bash-completion policycoreutils-python-utils mlocate bzip2</pre>
Apache	<pre>dnf install -y httpd</pre>
	Maak een virtual host in /etc/httpd/conf.d/nextcloud.conf met daarin als informatie:
	<pre><VirtualHost *:80> DocumentRoot /var/www/html/nextcloud/ ServerName client.school.test <Directory /var/www/html/nextcloud/> Require all granted AllowOverride All Options FollowSymLinks MultiViews <IfModule mod_dav.c> Dav off </IfModule> </Directory> </VirtualHost></pre>
	<pre>systemctl enable httpd.service systemctl start httpd.service</pre>
	<pre>firewall-cmd --add-service=http --permanent firewall-cmd --reload</pre>
PHP	<pre>dnf -y module switch-to php:7.4</pre>
	<pre>dnf install -y php-7.4.* php-gd-7.4.* php-mbstring-7.4.* php-intl-7.4.* php-pecl-apcu php-mysqlnd-7.4.* php-opcache-7.4.* php-json-7.4.* php-pecl-zip</pre>
Mariadb/Mysql	<pre>dnf install -y mariadb mariadb-server python3-PyMySQL</pre>
	<pre>systemctl enable mariadb.service systemctl start mariadb.service</pre>
Maak je database	<pre>mysql -u root -p create database nextcloud; grant all on nextcloud.* to 'dbuser'@'localhost' identified by '<<hierjewachtwoord>>'; FLUSH PRIVILEGES; exit</pre>
Redis (voor caching)	<pre>dnf install -y redis systemctl enable redis.service systemctl start redis.service</pre>



NextCloud	Download de Server software (Download for server) van https://nextcloud.com/install Kies voor het '.tar.bz2'-archief, bijvoorbeeld: https://download.nextcloud.com/server/releases/nextcloud-21.0.2.tar.bz2
Uitpakken in de juiste dir	<pre>tar -xjvf nextcloud-21.0.2.tar.bz2 -C /var/www/html/ mkdir /var/www/html/nextcloud/data chown -R apache:apache /var/www/html/nextcloud systemctl restart httpd.service</pre>
we'll set SELinux. Many tutorials will tell you to disable SELinux (because it is a difficult component to manage). Instead, I suggest you to keep it on and add the rules for NextCloud:	<pre>semanage fcontext -a -t httpd_sys_rw_content_t '/var/www/html/nextcloud/data(/.*)?' semanage fcontext -a -t httpd_sys_rw_content_t '/var/www/html/nextcloud/config(/.*)?' semanage fcontext -a -t httpd_sys_rw_content_t '/var/www/html/nextcloud/apps(/.*)?' semanage fcontext -a -t httpd_sys_rw_content_t '/var/www/html/nextcloud/.htaccess' semanage fcontext -a -t httpd_sys_rw_content_t '/var/www/html/nextcloud/.user.ini' semanage fcontext -a -t httpd_sys_rw_content_t '/var/www/html/nextcloud/3rdparty/aws/aws-sdk- php/src/data/logs(/.*)?' restorecon -R '/var/www/html/nextcloud/' setsebool -P httpd_can_network_connect on</pre>
Open firefox	http://192.168.3.207/
Choose Mariadb/MySQL	Maak een beheerders account Stel de database in, niet SQLite! Krijg je een foutmelding: let op de url, ga naar 192.168.3.207 Helpt dat?

De onderstaande stappen **hoef je niet te automatiseren**, maar voer je uit vanaf de webbrowser:
Test nu of de omgeving “werkt” door te browsen naar: <http://192.168.3.207/> en kijk of je bij je bestanden kan komen, bv:



Zo, dat was handmatig, maar nu met saltstack, volledig geautomatiseerd. Hoe dat moet moeten jullie zelf uitzoeken. In ieder geval met state-files en pillars. Hieronder wat tips:

COMMAND	TIP
dnf install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm	pkg.installed , gebruik daarbij de parameter sources
dnf install -y epel-release yum-utils unzip curl wget bash-completion policycoreutils-python-utils mlocate bzip2	
dnf install -y httpd	pkg.installed
Maak een virtual host in /etc/httpd/conf.d/nextcloud.conf met daarin als informatie:	
<pre><VirtualHost *:80> DocumentRoot /var/www/html/nextcloud/ ServerName client.school.test <Directory /var/www/html/nextcloud/> Require all granted AllowOverride All Options FollowSymLinks MultiViews <IfModule mod_dav.c> Dav off </IfModule> </Directory> </VirtualHost></pre>	<p>file.managed</p> <p>zet informatie hiernaast in een file bij je state files</p>



<pre>systemctl enable httpd.service systemctl start httpd.service</pre>	service.running
<pre>firewall-cmd --add-service=http --permanent firewall-cmd --reload</pre>	firewalld.present
<pre>dnf module switch-to php:7.4</pre>	Lukt niet echt anders dan met cmd.run
<pre>dnf install -y php-7.4.* php-gd-7.4.* php-mbstring-7.4.* php-intl-7.4.* php-pecl-apcu php-mysqlnd-7.4.* php-opcache-7.4.* php-json-7.4.* php-pecl-zip</pre>	
<pre>dnf install -y mariadb mariadb-server python3-PyMySQL systemctl enable mariadb.service systemctl start mariadb.service</pre>	
<pre>mysql -u root -p create database NextCloud_db; grant all on NextCloud_db.* to 'dbuser'@'localhost' identified by '<<hierjewachtwoord>>'; FLUSH PRIVILEGES; exit</pre>	mysql_database.present mysql_user.present mysql_grants.present als het werkt: zet password/user/database in een pillar file
<pre>dnf install -y redis systemctl enable redis.service systemctl start redis.service</pre>	
<p>Download de Server software (Download for server) van https://nextcloud.com/install Kies voor het '.tar.bz2'-archief, bijvoorbeeld: https://download.nextcloud.com/server/releases/nextcloud-21.0.2.tar.bz2</p>	archive.extracted met parameter "skip_verify: True"
<pre>tar -xjvf nextcloud-21.0.2.tar.bz2 -C /var/www/html/</pre>	archive.extracted, zie hierboven
<pre>mkdir /var/www/html/nextcloud/data chown -R apache:apache /var/www/html/nextcloud</pre>	archive.extracted, zie hierboven kan allemaal in dezelfde functie
<pre>systemctl restart httpd.service</pre>	
<pre>semanage fcontext -a -t httpd_sys_rw_content_t '/var/www/html/nextcloud/data(/.*)?' semanage fcontext -a -t httpd_sys_rw_content_t '/var/www/html/nextcloud/config(/.*)?' semanage fcontext -a -t httpd_sys_rw_content_t '/var/www/html/nextcloud/apps(/.*)?' semanage fcontext -a -t httpd_sys_rw_content_t '/var/www/html/nextcloud/.htaccess' semanage fcontext -a -t httpd_sys_rw_content_t '/var/www/html/nextcloud/.user.ini'</pre>	selinux.fcontext_policy_present met parameter "sel_type: httpd_sys_rw_content_t"



<pre>semanage fcontext -a -t httpd_sys_rw_content_t '/var/www/html/nextcloud/3rdparty/aws/aws-sdk- php/src/data/logs(/.*)?'</pre>	
<pre>restorecon -R '/var/www/html/nextcloud/' setsebool -P httpd_can_network_connect on</pre>	selinux.fcontext_policy_applied selinux.boolean
http://192.168.3.207/	
Maak een beheerders account Stel de database in, niet SQLite!	

2.6.3 Lesstof tentamen

- **Leren:** Zie 2.4
- **En bovendien:** Powerpoint van deze les.

2.6.4 Informatiebronnen

- Saltstack modules <https://docs.saltproject.io/en/latest/ref/states/all/index.html>



2.7 Week 4, College 1: XML | Encoding

XML is overall: Dit document, opgeslagen als office-docx document, is een XML-bestand bijvoorbeeld.

Tijdens het eerste college gaan we aan de slag met XML. We leren dat het bestaat uit “gewoon” tekst. Toch is tekst helemaal niet zo “gewoon”. Als encoding wordt besproken snap je wat daar ingewikkeld aan is...

2.7.1 Lesopdracht

- 1) Maak met een teksteditor als notepad/textpad/notepad++ een XML file. Doe dit door de onderstaande Cisco configuratiefile om te schrijven naar XML. Bekijk daarna het resultaat met je browser (liefst Chrome). Is je XML Well Formed? Maak logisch gebruik van elementen, nesting en attributen. Je mag de naamgeving van elementen/attributen aanpassen als je daar een goede reden voor hebt.

```
interface Ethernet0
  nameif test
  ip address 10.10.88.50 255.255.255.254
  !
interface Ethernet1
  nameif inside
  ip address 10.86.194.176 255.255.254.0
  !
enable password 8Ry2YjIyt7RRXU24 encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
hostname XXX
!
policy-map xxx_global_fw_policy
  class inspection_default
    inspect dns
    inspect ftp
    inspect http
```

- 2) Voeg bij Interface Ethernet 0 een description element of attribuut toe met de onderstaande tekst en bekijk met de browser of dit is gelukt: Herman's network: "<Games & Other stuff>"

- 3) Voeg op een logische plaats commentaar toe aan de XML-file.

- 4) Voeg het onderstaande tekstblok ongewijzigd toe aan de XML-file door gebruik te maken van een CDATA element:

```
for (c in colors) {
  var item = $('<div/>').css("background-color", '#' + colors[c]).addClass("colorbox");
  item.click((function () {
    var col = colors[c];
    return function () {
      currentColor = col;
    };
  })());
```

- 5) Verander de description van interface Ethernet 1 in: 请收藏 Bekijk in Chrome of de tekst leesbaar is.



2.7.2 Weekopdracht

1. Maak met behulp van notepad of een andere eenvoudige text editor zoals Textpad of Notepad++ een XML-bestand waarin je een eenvoudig netwerk beschrijft. De onderstaande informatie moet ten minste voorkomen in het document:
In het netwerk komen 5 clients voor en 2 servers. Van alle systemen is het ip-adres opgenomen in het document. Het client OS is Windows 10, behalve bij een client die nog Windows 7 als besturingssysteem heeft. De servers zijn Windows 2016 Standard. Het XML bestand beschrijft van de servers ook nog het aantal harddisks (3 per server) en de driveletters die op elke schijf zijn gemaakt (geen RAID), bv "C" of "D" etc.
2. Maak vervolgens een Python script waarin je de XML leest en waarvan je van ieder systeem zowel het OS als ook het IP-adres print op de console.
3. Nu een nuttiger toepassing voor een SNE-er: Maak een **XML-configuratie-bestand** aan voor FTP-client. In het XML-bestand moeten tenminste de ftp-server, username, wachtwoord, doeldirectory, bronbestand zijn opgenomen op een logische manier.
4. Zorg voor een FTP server (gebruik bijvoorbeeld Filezilla), maak gebruik van Python-FTP-client code die je vindt via Google. Zorg ervoor dat je FTP-script gebruik maakt van je client-configuratiefile. De configuratiefile zorgt ervoor dat je je bestand op ieder systeem kan gebruiken zonder wijzigingen. De systeemspecifieke informatie staat alleen in je configuratiebestand.
5. Voer het PowerShell "Get-EventLog -LogName System -Newest 5 -EntryType Error | Export-Clixml test.xml" -commando uit. Wat doet dit commando eigenlijk?
6. Verwijder nu de namespace uit het XML-document, dat is het attribuut "xmlns". (dit onderdeel gaan we namelijk niet behandelen tijdens deze cursus):

```
<Objx Version="1.1.0.1" xmlns="http://schemas.microsoft.com/powershell/2004/04">
```


Wordt daarmee:

```
<Objx Version="1.1.0.1">
```
7. Zorg dat je de XML kan lezen en wel zodanig dat je met behulp van Python-code dat je de omschrijving van de events presenteert op het scherm. Gebruik hierbij de volgende informatie als je dat handig vindt:

De onderstaande XPath-query gaat op zoek in het XML-bestand naar alle elementen met de naam "ip_address", die een attribuut "mask" hebben dat de waarde "255.255.254.0" bezit.

```
'//ip_address[@mask="255.255.254.0"]'
```

Meer over XPath? <http://effbot.org/zone/element-xpath.htm>

2.7.3 Lesstof tentamen

Wat moet je weten voor het tentamen over dit onderwerp, dus hoe kan je je het beste voorbereiden?

- **Leer** alle informatie die je vindt in de powerpoint presentatie over zowel XML als over Encoding.
- **Lees** de informatie die wordt aangeboden onder het kopje "Informatiebronnen".
- **Zorg dat je een fout kan halen** uit de Python code waarin een XML wordt gelezen.
- **Begrijp XPath-expressies.**
- Zorg dat je zelf XML-parsing-scripts hebt gemaakt zodat je **getoonde code van anderen goed begrijpt** en kan **aanpassen/verbeteren**.



2.7.4 Informatiebronnen

- XML v1.0.ppt (W3Schools)
- Test je kennis op: http://www.w3schools.com/xml/xml_quiz.asp
- <http://www.diveintopython3.net/xml.html> , maar niet: “namespaces”, ook niet lxml
- Meer info: <https://docs.python.org/3/library/xml.etree.elementtree.html>



2.8 Week 4, College 2: Desired State Configuration (DSC)

Desired State Configuration (DSC) is een Microsoft methode om declaratief configuratie management uit te voeren. In dit lab behandelen we de concepten waaronder de Push en Pull modellen. Ten slotte moet je zelf proberen om een DNS Server in te richten met DSC.

2.8.1 Lesopdracht

Geen.

2.8.2 Weekopdracht

We gaan een DSC-omgeving nu echt zelf bouwen. Dat geeft ons bovendien extra mogelijkheden zoals het gebruik van een Pull Server.

Nodig

- 2x Windows 2019 systemen: dc1 en server1. Zorg dat beide systemen onderdeel zijn van het domain hu.test waarbij de dc1 de domain controller is. Dc1 wordt straks ook een Pull Server en Server1 is steeds de "target", kortom je client.
- Je mag ook gerust gebruik maken van een eerder gebruikte omgeving.
- Download de 64-bit msi-versie van 7-zip op <https://www.7-zip.org/download.html>

Push Configuration

- Kijk eerst eens naar de resources die je kan gebruiken en je lokale configuratie van de LSM:
PS > Get-DscResource
Welke resources kan je gebruiken?

PS > Get-DscLocalConfigurationManager
Push of Pull?
- Nu een nuttige toepassing. We gaan 7-zip installeren door de file eerst te kopiëren naar het doelsysteem en daarna te installeren. De kopie actie is niet echt nodig, maar wel nuttig om meer van DSC te leren. Bij het installeren van een MSI-bestand moet je de naam en het productId kennen. Krijg je foutmeldingen? Installeer 7-zip dan eerst zelf en noteer het ProductId en de Naam voor gebruik in je configuratie.

```
PS C:\Users\Administrator> Get-WmiObject Win32_Product | Format-Table IdentifyingNumber, Name
IdentifyingNumber      Name
-----
{5FCE6D76-F5DC-37AB-B2B8-22AB8CEDB1D4} Microsoft Visual C++ 2008 Redistributable - x64 9.0.30729.6161
{23170F69-40C1-2702-1602-000001000000} 7-Zip 16.02 (x64 edition)
{02A869FC-38AC-4835-BB1C-9BB4758A8C92} VMware Tools
{1F1C2DFC-2D24-3E06-BCB8-725134ADF989} Microsoft Visual C++ 2008 Redistributable - x86 9.0.30729.4148
```

- Start PowerShell ISE en type het onderstaande. Zorg dat het werkt zodra je het script start. Bij de eerste stap wordt de mof-file gemaakt, zie hieronder.



```
Administrator: Windows PowerShell
File Edit View Tools Debug Add-ons Help
test1.ps1 X
1 Configuration Install7Zip
2 {
3   param(
4     [Parameter(Mandatory=$true)]
5     [String]$Servers
6   )
7
8   Node $Servers
9   {
10    File ZipToTarget
11    {
12      Ensure = "Present"
13      Type = "File"
14      SourcePath = "\\dc1\SHARE\7z1602-x64.msi"
15      DestinationPath = "C:\DSC\Packages\7z1602-x64.msi"
16    }
17
18    Package ZipInstall
19    {
20      Ensure = "Present"
21      DependsOn = "[File]ZipToTarget"
22      Name = "7-Zip 16.02 (x64 edition)"
23      ProductId = "23170F69-40C1-2702-1602-000001000000"
24      Path = "C:\DSC\Packages\7z1602-x64.msi"
25    }
26  }
27 }
28
29 Install7Zip -Output c:\DSC\zip -Servers "server1"
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

Directory: C:\DSC\zip

Mode                LastWriteTime         Length Name
----                -
-a-----          9/24/2016   8:29 PM           3064 server1.mof

PS C:\Users\Administrator>
```

- Maak nu de benodigde share aan en plaats daar de msi.
- Bij de volgende stap deploy je de configuratie op je target: server1.
`Start-DscConfiguration -Path C:\DSC\zip -wait -verbose -Force`

Pull Configuration

- We maken nu een Pull Configuratie Server. Je kan deze maken met DSC, wel zo handig! Het is bovendien een goede manier om weer wat meer van DSC te leren.
- 1 Tijdens de eerste stap installeer je een module hiervoor uit de PowerShell Gallery op DC1: <http://www.powershellgallery.com/> Misschien wordt je gevraagd om NuGet te installeren voor het eerste gebruik en dat is prima. Het geef ook niet dat de repository untrusted is. Je hebt hiervoor wel een Internetverbinding nodig.

`Install-Module -Name xPSDesiredStateConfiguration`



- 2 Start vervolgens de ISE en kopieer de onderstaande configuratie en voer het script uit. Kijk hierna of de website van de Pull Server bereikbaar is.

```
configuration CreatePullServer {
    param (
        [string[]]$ComputerName = 'localhost' # Een andere computernaam in het domein kan ook.
    )

    # Importeer de resource uit de powershellgallery die je net hebt geïnstalleerd
    Import-DSCResource -ModuleName xPSDesiredStateConfiguration

    # Configureer nu de node "$computername", in dit geval de "localhost"
    Node $ComputerName {
        # Installeer eerst de windowsFeature die hiervoor bestaat
        WindowsFeature DSCServiceFeature {
            Ensure = "Present"
            Name = "DSC-Service"
        }

        # Installeer vervolgens de website en configureer deze
        xDscWebService PSDSCPullServer {
            Ensure = "Present"
            EndpointName = "PSDSCPullServer"
            Port = 8080
            PhysicalPath = "$env:SystemDrive\inetpub\wwwroot\PSDSCPullServer"
            CertificateThumbPrint = "AllowUnencryptedTraffic"
            ModulePath = "$env:PROGRAMFILES\WindowsPowerShell\DscService\Modules"
            ConfigurationPath = "$env:PROGRAMFILES\WindowsPowerShell\DscService\Configuration"
            State = "Started"
            DependsOn = "[WindowsFeature]DSCServiceFeature"
            UseSecurityBestPractices = $false
        }
    }
}

# Roep nu de configuratie zoals hierboven gemaakt aan en zet de output mof-files in de output
directory
CreatePullServer -OutputPath c:\DSC\CreatePullServer\

# Apply de configuratie nu op de Server (hier localhost)
Start-DscConfiguration -Path "c:\DSC\CreatePullServer" -wait -verbose

# Alles gelukt?
# Heb je de windows feature geïnstalleerd?
# Get-WindowsFeature -Name DSC-Service
#
# En de website?
# Get-Website
#
# Hoe ziet je configuratie er nu uit?
# Get-DscConfiguration
#
# Bekijk nu of de website beschikbaar is:
# http://localhost:8080/PSDSCPullServer.svc/
```

- 3 De PullServer is nu klaar. Nu moeten we de LCM op de target configureren zodat deze omschakelt van Push mode naar Pull node. In Pull node gebruiken we niet meer de naam van de MOF-file als aanduiding van de server, zoals "server1.mof", maar gebruiken we een GUID die we zelf moeten maken:

```
PS > $guidServer1 = [GUID]::NewGuid()
PS > $guidServer1
```

- 4 Vervolgens gaan we de configuratie van de LCM op Server1 aanpassen zodat deze gaat "pullen". Ook dat doen we met DSC.

```
configuration LCMPullMode {
```



```

param (
    [string]$ComputerName, [string]$GUID
)
node $ComputerName {
    LocalConfigurationManager {
        ConfigurationID = $GUID
        ConfigurationMode = 'ApplyAndAutocorrect'
        RefreshMode = 'Pull'
        DownloadManagerName = 'WebDownloadManager'
        DownloadManagerCustomData = @{
            ServerUrl = 'http://DC1:8080/PSDSCPullServer.svc'
            AllowUnsecureConnection = 'true'
        }
        RebootNodeIfNeeded = $true
    }
}
}
}

LCMPullMode -ComputerName Server1 -GUID <<TYPEHIERJOUWGUID,dus: $guidServer1>>

```

Je hebt nu een mof gemaakt.

```

Directory: C:\Users\Administrator\LCMPullMode

Mode                LastWriteTime         Length Name
----                -
-a-----          9/25/2016  11:18 AM         2044 Server1.meta.mof

```

- De mof gaan we nog eenmaal “pushen” naar de client. Niet met “start-dsconfiguration”, maar met “set-dsclocalconfigurationmanager”:

```
PS> Set-DscLocalConfigurationManager -Path .\LCMPullMode
```

Bekijk op de server1 of dat is gelukt:

```
Get-DscLocalConfigurationManager
```

Staat er: RefreshMode: Pull ?

Hoe vaak wordt de configuratie opgehaald bij de server?

ConfigurationModeFrequencyMins = Zorg ervoor dat de configuratie in de Desired State is.

RefreshFrequencyMins = Haal de configuratie op van de pull server.

- In de volgende stap gaan we een configuratie maken en deze plaatsen op de pull server. We gaan ervoor zorgen dat de “SNMP Trap Service” altijd is gestart. Maak hiervoor een mof-file.

```

configuration SNMPTrapServiceConfig {
    param(
        [string[]] $ComputerName
    )
    node $ComputerName {
        service SNMP {
            Name = 'snmptrap'
            StartupType = 'Automatic'
            State = 'Running'
        }
    }
}

SNMPTrapServiceConfig -ComputerName Server1

```

- Tot zover geen verschil, alleen bij de Pull-mode moeten de mof-files worden hernoemd naar de GUID's van hun server.



```
PS> Rename-Item .\SNMPTrapServiceConfig\Server1.mof <<JOUWGUIDVANSERVER1>>.mof
```

- 8 De volgende stap is het maken van een checksum van alle mof-files, zodat de ontvangende client altijd kan controleren of de configuratie niet is aangepast of half is overgekomen tijdens het transport (integrity). Je gebruikt hiervoor `New-DSCChecksum` in de directory waar de mof-files staan:

```
PS C:\Users\Administrator> Rename-Item .\SNMPTrapServiceConfig\Server1.mof 96866c6f-808e-4518-91c6-9b071a4b93e5.mof
PS C:\Users\Administrator> dir .\SNMPTrapServiceConfig

Directory: C:\Users\Administrator\SNMPTrapServiceConfig

Mode                LastWriteTime         Length Name
----                -
-a----             6/21/2021   9:55 AM             1922 96866c6f-808e-4518-91c6-9b071a4b93e5.mof

PS C:\Users\Administrator> cd .\SNMPTrapServiceConfig
PS C:\Users\Administrator\SNMPTrapServiceConfig> New-DscChecksum *
PS C:\Users\Administrator\SNMPTrapServiceConfig> dir

Directory: C:\Users\Administrator\SNMPTrapServiceConfig

Mode                LastWriteTime         Length Name
----                -
-a----             6/21/2021   9:55 AM             1922 96866c6f-808e-4518-91c6-9b071a4b93e5.mof
-a----             6/21/2021   9:57 AM              64 96866c6f-808e-4518-91c6-9b071a4b93e5.mof.checksum
```

- 9 Bijna klaar. Nu de files verplaatsen naar de PullServer. Kopieer deze naar de directory aangegeven in stap 2 bij "ConfigurationPath" eindigend op "WindowsPowerShell\DscService\Configuration". Bijvoorbeeld: `Copy *.mof* -Destination "C:\Program Files\WindowsPowerShell\DscService\Configuration"`
- 10 Bekijk de configuratie op Server1 met `Get-DscConfigurationStatus .` Doe dit na 15 minuten nog eens. Stop evt. de SNMP Trap service om te zien of DSC dit weer herstelt binnen een half uur. Bekijk de status met: `gsv snmptrap`
Tip: Geen zin om 15 minuten te wachten? Probeer eens: `Update-DscConfiguration`

Nu Zelf!

- Je hebt nu ervaring met configuraties, push en een pull server. Nu wat meer creativiteit. Maak nu een configuratie waarbij je een DNS Server inricht op een server voor de zone "hu.local" met daarin een A-record verwijzend naar www.hu.local met ip-adres 10.0.0.1. Zorg ervoor dat de DNS-server functioneert en test dit met:

```
PS C:\Users\Administrator> resolve-dnsname www.hu.local -server localhost

Name                Type    TTL    Section    IPAddress
----                -
www.hu.local        A       3600   Answer     10.0.0.1
```

- Paar tips:
 - Je kan de pull server en client gebruiken, maar dat test niet lekker. Immers je moet steeds een kwartier wachten bij elke test! Gebruik daarom liever een push configuratie. Dat werkt beter in een testfase. Je kan dit doen door de DNS-server op je dc1 te gebruiken. Kortom, DSC naar localhost, of gebruik een andere Windows Server die nog niet in Pull mode is.
 - <https://www.powershellgallery.com/packages/xDnsServer/2.0.0> → de module die je moet installeren. Dat kan bijvoorbeeld met:



```
Install-Module -Name xDnsServer -Repository PSGallery
```

Lukt het niet? Dit helpt, probeer het daarna opnieuw:

```
Install-Module PowerShellGet -RequiredVersion 2.2.4 -SkipPublisherCheck
```

- <https://github.com/PowerShell/xDnsServer> --> beschrijving van de DNS-server
- Zorg voor de Service "DNS", een PrimaryZone met xDnsServerPrimaryZone en een DNS-A-record via xDnsRecord...

2.8.3 Lesstof tentamen

- **Leer** alle informatie die je vindt in de powerpoint presentatie en dit werkboek over dit onderwerp.
- **Leer** de theorie over DSC zoals te vinden op Canvas over dit onderwerp.

2.8.4 Informatiebronnen

- <https://www.simple-talk.com/sysadmin/powershell/powershell-desired-state-configuration-the-basics/> - Goede beschrijving van onderdelen van deze weekopdracht en de werking van DSC.
- <http://www.powershellgallery.com/> - Extra DSC-resources.



2.9 Week 5, College 1: Object Oriented | Matplotlib

Je hebt gehoord dat Object-georiënteerd programmeren (OOP) veel voordelen biedt en dat OOP door Python ondersteund wordt. We gaan dit nu toepassen bij het gebruik van object georiënteerd programmeren. In het tweede deel gaan we grafieken maken met behulp van matplotlib.

2.9.1 Lesopdracht

Object-Orientatie

Maak een klasse "Train"

1. Bedenk Attributes voor deze klasse:

Fields (=velden)

Methods (=functies)

2. Maak een Constructor die zorgt voor initialisatie van de velden.

3. Maak een functie die de juiste "snelheid" regelt (versnellen/vertragen).

4. Maak een functie die het object kan printen.

Matplotlib

Maak een taartdiagram van het gebruik van de website w3schools.com per type besturingssysteem. Gebruik hiervoor de onderstaande code en data...

```
import matplotlib.pyplot as plt
x= [ 10,20,30]
y=['cats', 'dogs', 'fishes']
z=[.2, .1,0]
plt.pie(x, labels=y, explode=z)
plt.show()
```

2012	Win7	Vista	NT*	WinXP	Linux	Mac	Mobile
October	56.8%	3.0%	1.8%	22.1%	4.8%	9.2%	1.8%

2.9.2 Weekopdracht

OV-chipkaart

Het doel is om de OV-chipkaart als een object op te nemen in je programma. Het eindresultaat moet je met een tweetal Python bestanden laten zien.

Persoonlijke gegevens

Persoonsgegevens worden verwerkt in overeenstemming met het [Privacybeleid Trans Link Systems B.V](#) (pdf).

Geslacht

Man Vrouw

Voorletter(s) * Tussenvoegsel(s) Achternaam * 0 / 26

Geboortedatum *

DD MM JJJJ

De OV-chipkaart die je nodig heb om met de trein te kunnen reizen wordt geleverd door Trans Link Systems. Om een kaart aan te vragen moet je de informatie zoals te zien in de afbeelding invoeren.



1. Maak in Python een minimale klasse "Smartcard" met een docstring waarin je de klasse globaal omschrijft. Sla deze klasse op in een apart bestand met de naam smartcard.py
2. Geef de klasse een constructor (__init__ functie) waarmee de gegevens uit de figuur worden vastgelegd in het object.
3. Maak twee nieuwe persoonlijke smartcards aan. Gebruik hierbij de constructor uit de vorige opdracht.
4. Zorg voor een uitbreiding van de constructor zodat de "creation-date" altijd wordt toegevoegd aan een nieuw object. Gebruik hierbij bijvoorbeeld de module "datetime".
5. Print van iedere smartcard de achternaam en de creation-date.
6. Maak binnen de klasse een methode "check_name" waarmee je controleert of het totale aantal karakters van voorletters+tussenvoegsel(s)+achternaam niet groter is dan 26. Als de naam geldig is geeft de methode "True" terug en anders "False". Roep de functie aan voor iedere user en print het resultaat op het scherm.
7. Maak binnen de klasse een methode "load" waarmee je geld kan toevoegen aan de kaart.
8. Maak ook een methode "withdraw" waarmee je geld afboekt van de kaart. De methode geeft -1 terug als er een negatief saldo zou kunnen ontstaan.
9. Geef elke methode een logische docstring waarin je de functie beschrijft en aangeeft welke parameters als input worden gegeven en wat de functie teruggeeft.
10. Zorg ervoor dat elke smartcard een uniek nummer krijgt door hiervoor een klasse variabele te gebruiken.
11. Zorg ervoor dat het printen van een smartcard object de naam toont van deze klant met spaties op de juiste plaats. Gebruik hiervoor de speciale functie __str__.
12. Maak nu het hoofdsript. Dit maak je in een ander bestand. Het script kent de volgende flow:
 - a. Importeer allereerst je klasse "smardcard".
 - b. Vraag de gebruiker om input. De input gebruik je om een nieuwe smartcard te initialiseren. Het object dat je hiervoor gebruikt is personal_ov_1.
 - c. Vraag de gebruiker om input. De input gebruik je om een tweede smartcard te initialiseren. Het object dat je hiervoor gebruikt is personal_ov_2.
 - d. Zorg ervoor dat op beide kaarten 10 euro wordt gestort.
 - e. Haal van de eerste card nu 5 euro af.
 - f. Haal van de tweede card nu 15 euro af (en toon een logische foutmelding).

Matplotlib | PowerShell

Verzamel data van de Performance Monitor (perfmon). Je mag zelf weten welke data je verzamelt. Hieronder wordt de variabele \$counters gebruikt om de hoeveelheid processortijd te meten gedurende 8 seconden. Werkt het voorbeeld niet bij jou, of wil je liever wat anders meten: Type dan "Get-Counter" op een PowerShell console om een paar voorbeelden te zien op jouw computer.

```
$counters = @("\Processor(_total)\% Processor Time")
```

```
Get-Counter -Counter $counters -SampleInterval 1 -MaxSamples 8 | export-counter proc.csv -FileFormat csv -Force
```




Haal met notepad de eerste twee regels weg:

```
notepad .\proc.csv
```

Maak vervolgens van deze data een grafiek met behulp van de csv-module en matplotlib. Hieronder een aantal voorbeelden die je kan gebruiken in je code. Veel mooiere voorbeelden vind je op:

<http://matplotlib.org/gallery.html>

```
# Import
import matplotlib.pyplot as plt
import numpy as np

# My first Script
plt.plot([1,8,3,4])
plt.show()

# X & Y
plt.plot([1,2,3,4], [1,4,9,16])
plt.show()

# Axis
plt.plot([1,2,3,4], [1,4,9,16])
plt.axis([0, 6, 0, 20]) # [xmin, xmax, ymin, ymax]
# or
plt.axis(xmin=0,xmax=10) # **kwargs
plt.show()

# Axis limits
plt.plot([1,2,3,4], [1,4,9,16])
plt.xlim(-3) # xmin
# or
plt.xlim(xmin=-3) # xmin, xmax
plt.show()

# Color/line-type
plt.plot([1,4,9,16], 'r') # b, g, k, m, r, w, y
plt.plot([1,2,3,4], 'v') # ., o, v, etc.
plt.plot([2,3,4,5], '--') # - - - . :
plt.plot([4,5,6,7], 'k-.v') # all
plt.show()

# Set line properties: keyword args
plt.plot([1,2,3,4], [1,4,9,16], linewidth=4.0)
plt.show()

# Text: title, labels, text, grid, arrow
plt.plot([50,60,70,100,130], [.006,.010,.006,.010,.022])
plt.title('Histogram of IQ')
plt.xlabel('Smarts', fontsize=14, color='red')
plt.ylabel('Probability')
plt.text(60, .025, 'something...')
plt.axis([40, 160, 0, 0.03])
plt.grid(True)
plt.annotate('Interesting', xy=(130, .022),
xytext=(100, .025), arrowprops=dict(facecolor='black'))
plt.show()

# list of lines and set line properties using setp()
x1=(1,2,3,4,5,6,7,8,9)
y1=(2,5,7,4,0,9,7,1,2)
y2=(5,7,4,0,9,7,1,2,1)
lines=plt.plot(x1,y1,x1,y2)
plt.setp(lines, color='r', linewidth=2.0)
plt.show()
```



```
# Bar charts display rectangular bars (either vertical or horizontal)
# with their length proportional to the values they represent.
plt.bar([1,2,3,8],[1,8,3,4], width=.5)
plt.show()

# Histogram charts are a graphical display of frequencies,
# represented as bars
y = np.random.randn(100) # generate random numbers
print(y)
#y = [3,4,5,3,0,6,6]
plt.hist(y)
plt.show()

# Save to a file
x = range(6)
plt.plot(x,x, x,[i**2 for i in x])
plt.savefig('D:\plot124.png', dpi=200)
```

2.9.3 Lesstof tentamen

Wat moet je weten voor het tentamen over deze onderwerpen, dus hoe kan je je het beste voorbereiden?

- **Leer** alle informatie die je vindt in de powerpoint presentatie
- **Leer** hoofdstuk 12 uit het boek “Byte of Python”, over OO. Staat op Canvas: **5b- byte_of_python hst12 only.pdf**
- **Zorg dat je een fout kan halen** uit de Python code.
- **Begrijp Matplotlib scripts.**
- Zorg dat je zelf OO en Matplotlib scripts hebt gemaakt zodat je **getoonde code van anderen goed begrijpt** en kan **aanpassen/verbeteren**.

2.9.4 Informatiebronnen

- Matplotlib → <http://matplotlib.org/> | <http://matplotlib.org/gallery.html>



2.10 Week 5, College 2: Test Automation

Testen is een gedeelde verantwoordelijkheid: Developers maken nieuwe functionaliteit, maar letten soms minder op non-functionele aspecten zoals security of performance. Beheerders zijn gebaat bij een stabiele, betrouwbare omgeving. Kortom: testen doe je samen en tools/scripts helpen daarbij.

Vroeger kwam er misschien ieder jaar een nieuwe release en kon je de software ook handmatig testen.

Tegenwoordig werken we vaak “agile” en dat betekent veel vaker een nieuwe software release. Handmatig testen wordt dan heel vervelend. In deze cursus maak je kennis met “Selenium”, een open source suite aan tools waarmee je websites volledig geautomatiseerd kunt testen.

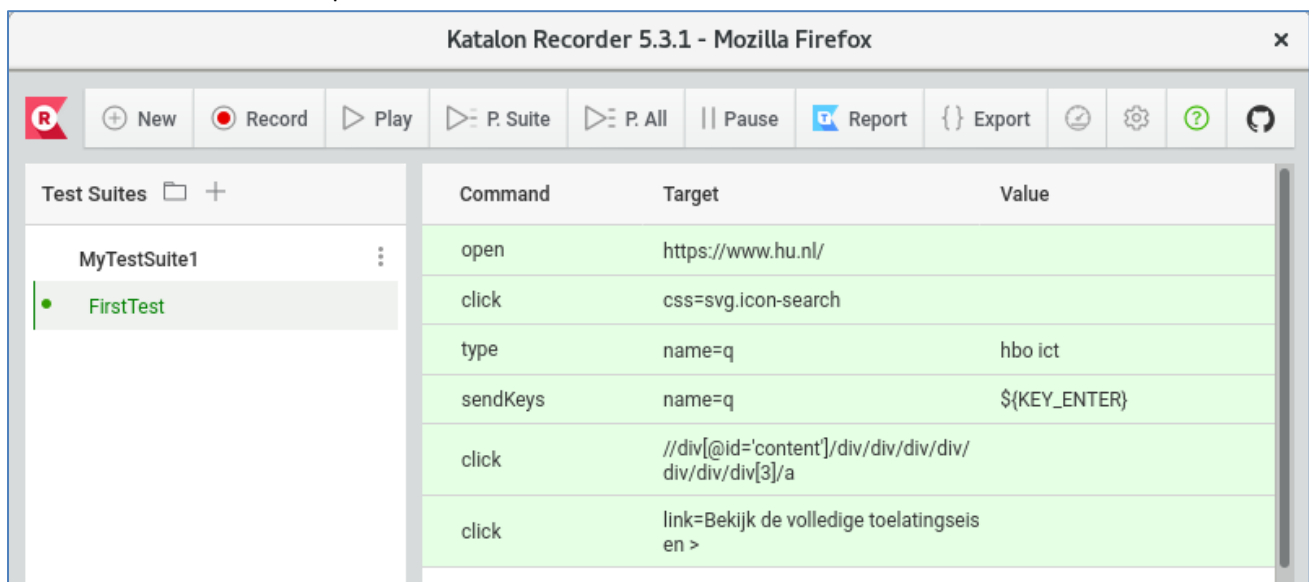
We gaan deze les **in zijn geheel uitvoeren** onder Linux. De les is getest met behulp van: RHEL8.4 (Ootpa) en Firefox (78.11.0esr).

2.10.1 Lesopdracht

Automatiseer de volgende test: *“Browse met Firefox naar www.hu.nl. Zoek daar naar de tekst ‘hbo ict’ klik dan op het plaatje (dus op de opleiding) en navigeer vervolgens naar de toelatingseisen.”* Het kan zijn dat de site van de HU inmiddels is aangepast, misschien moet je deze test zelf een beetje aanpassen.

Installeer om dit mogelijk te maken een versie van Firefox en de juiste plugin: <https://addons.mozilla.org/en-US/firefox/addon/katalon-automation-record/>. Als alternatief kan je de tests trouwens ook runnen vanaf je eigen laptop waar je firefox uitvoert met de plugin en vervolgens connect naar je linux website.

Automatiseer vervolgens de hierboven beschreven test. Als je er tijdens de les zelf niet zo snel uitkomt, gebruik de onderstaande screendump:





2.10.2 Weekopdracht

De weekopdracht bestaat uit drie verschillende stappen: In de eerste stap installeer je een website die je vervolgens gaat testen met behulp van de Katalon Automation Recorder. Als dat is gelukt ga je hiervoor een Python script schrijven zodat je de test automatisch kunt starten, bijvoorbeeld vlak na de deploy op een acceptatieomgeving. We doen alles onder linux.

Je hebt hiervoor de volgende omgeving nodig:

- Een RHEL8-systeem als VM met Internet connectiviteit. Een grafische server is het gemakkelijkst, dan doe je het hele lab op deze server.
- Geef de server 3GB memory.
- Een Firefox browser, je kan deze starten vanuit de grafische VM.

Stap 1: Installeer een website zodat we deze kunnen testen.

We maken gebruik van een eenvoudige en lichtgewicht Content Management Systeem (CMS) op basis van PHP en SQLite. Deze website gaan we in de stappen hierna testen met de Katalon Recorder en Selenium WebDriver.

Installatie:

1. Installeer Apache met PHP:

```
dnf install httpd php
dnf install php-gd php-ldap php-pdo php-xml php-xmlrpc php-mbstring curl zlib
systemctl enable httpd
systemctl start httpd
sudo firewall-cmd --permanent --zone=public --add-service=http
sudo firewall-cmd --reload
```

2. Disable SELinux (dit is namelijk een testsysteem), bijvoorbeeld op deze manier:

```
nano /etc/selinux/config → Pas deze regel aan: SELINUX=disabled
setenforce Permissive
```

3. Installeer het CMS, hieronder de grote lijnen. Misschien is een kleine aanpassing nodig:

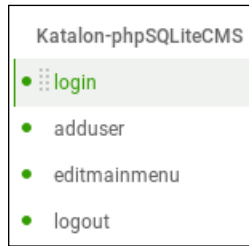
Download de zip van de website: <https://github.com/ilosuna/phpsqlitecms>

```
unzip -d /var/www/html/ ./Downloads/phpsqlitecms-master.zip
mv /var/www/html/phpsqlitecms-master/ /var/www/html/cms-simple/
cd /var/www/html/cms-simple/
chown -R apache:apache cms/
chmod -R 775 cms/
```

Browse naar: <http://localhost/cms-simple> → gelukt?

Stap 2: Test de website met Katalon Recorder (Firefox plugin).

1. Download de Firefox plugin: <https://addons.mozilla.org/en-US/firefox/addon/katalon-automation-record/>
2. Best practices:
 - a. Geen wachtwoorden opslaan in deze browser na het inloggen.
 - b. Dubbelklik op een bepaalde test als je deze wilt bewerken (wordt vetgedrukt).
3. De opdracht: Maak een testsuite aan met daarin de volgende testcases:

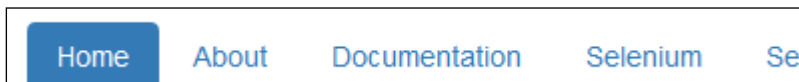


a. **login**



(inloggen kan met admin / admin)

- b. **adduser**: administration → user administration → create user account. Voeg vervolgens een user toe met wachtwoord.
- c. **editmainmenu**: Hierin maak je een aanpassing in het "main-menu". Je voegt een extra knop toe "Selenium" waarin je een link opneemt naar: <http://docs.seleniumhq.org/>. Dit kan je vaker doen overigens, zie de afbeelding hieronder.



d. **logout**: spreekt voor zich.

Stap 3: Test de website nu volledige geautomatiseerd met Selenium WebDriver. Dit doen we helemaal onder Linux omdat dit het meest betrouwbaar en eenvoudig werkt zo lijkt het. Je mag het ook onder Windows proberen.

1. Zorg er eerst voor dat Firefox op het systeem waar je scripts draaien up-to-date is.
2. Installeer selenium op de Linux-VM:

```
# install selenium versie 3.x
sudo pip3 install selenium
# Je kan selenium ook later nog installeren als je pycharm gebruikt.

# install pycharm voor Linux van http://www.jetbrains.com/pycharm/ en installeer dit onder Linux (uitpakken in /opt/). Dit is getest met de "community edition".
Starten via: /opt/pycharm-community-2021.1.2/bin/pycharm.sh

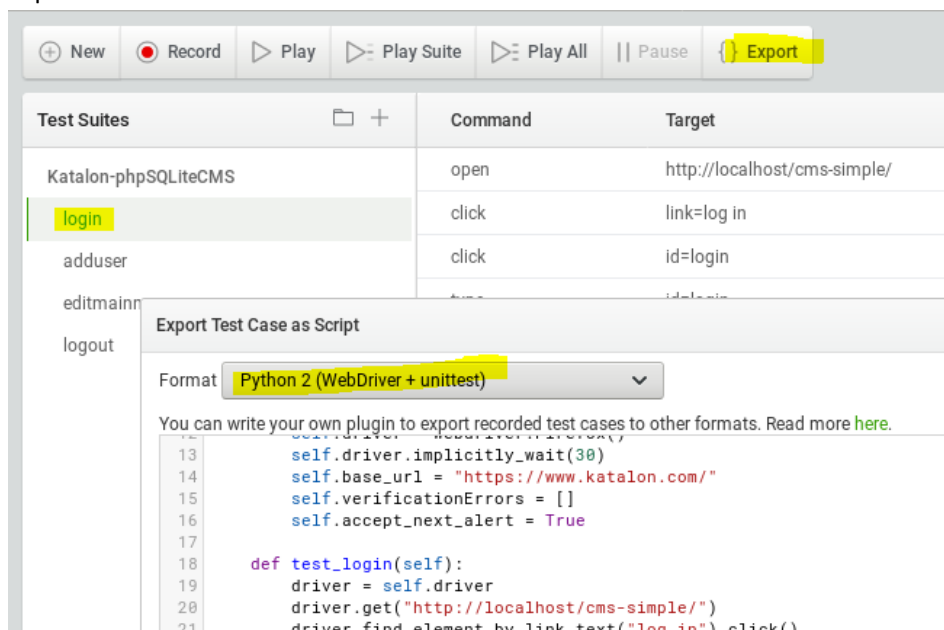
# download de meest recente geckodriver for selenium behorende bij de versie van Firefox die je gebruikt (!) Doe je dat niet dan krijg je rare foutmeldingen:
# Download van: https://github.com/mozilla/geckodriver/releases
# bv. Versie geckodriver-v0.29.1-linux64 past bij Firefox 78.11.0esr (64-bit)
# Uitpakken met tar -xzvf <targetfile> en daarna:
mv geckodriver /usr/local/bin/
# of: export PATH=$PATH:/path/to/geckodriver
# Als je je scripts wilt schrijven onder windows: Plaats de geckodriver.exe in de "current workdir", dus waar je selenium-scripts ook staan.
```



3. Controleer of de installatie is gelukt: Maak het onderstaande script op je Linux-VM en geef het de naam "myscript.py". Als het werkt kan je de rode tekst vervangen door je eigen script in de latere opdrachten. Let ook op het pad naar firefox, als je een eigen firefox hebt gedownload moet je het pad aanpassen. Je start je script als python-versie-3-script: `python3 ./myscript.py`

```
import time
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.firefox.firefox_binary import FirefoxBinary
binary = FirefoxBinary('/usr/bin/firefox')
driver = webdriver.Firefox(firefox_binary=binary, log_path="/tmp/geckodriver.log")
driver.implicitly_wait(30)
driver.maximize_window()
driver.get("https://www.duckduckgo.com")
driver.find_element_by_name("q").clear()
driver.find_element_by_name("q").send_keys("Selenium WebDriver")
driver.find_element_by_name("q").send_keys(Keys.ENTER)
driver.find_element_by_link_text("WebDriver :: Documentation for Selenium").click()
time.sleep(3)
driver.quit()
```

4. Werkt dit? Dan kan je nu de login testcase gaan testen in pyCharm. Selecteer de login test en klik op "{ Export".



Selecteer de python code en **plak delen daaruit in een python file** in pyCharm. Start de test. Werkt het? Zo niet, misschien zie je in de afbeelding hieronder wel een paar goede tips.



```
pyscript.py x cms-lite-test1.py x
1 import time
2 import random
3 from selenium import webdriver
4 from selenium.webdriver.common.keys import Keys
5 from selenium.webdriver.firefox.firefox_binary import FirefoxBinary
6 binary = FirefoxBinary('/usr/bin/firefox')
7 driver = webdriver.Firefox(firefox_binary=binary, log_path="/tmp/geckodriver.log")
8 driver.implicitly_wait(30)
9 driver.maximize_window()
10
11
12 def test_login():
13     driver.get("http://192.168.37.102/cms-simple/")
14     driver.find_element_by_link_text("log in").click()
```

5. Zorg er ten slotte voor dat je alle testcases, gemaakt met Katalon, geautomatiseerd achter elkaar kan uitvoeren. Dat kan je doen door de pythoncode van iedere testcase functie die begint met “test_” te plakken in jouw code. Niet een perfecte oplossing, het kan mooier door unittests te maken, maar het gaat om het idee. Er zijn een paar aanpassingen ten opzichte van de Katalon code gewenst:
- a. **login**: geen aanpassingen
 - b. **adduser**: Let op: de naam van het user account moet nu automatisch worden gegenereerd. Hierdoor worden voorkomen dat je een user probeert aan te maken die al bestaat. Je kan hiervoor iets slims bedenken zoals datum/tijd, iets “randoms” of het mooiste: een teller die steeds met één wordt opgehoogd.
 - c. **adduser**: Nog een aanpassing in adduser, kijk of je het voor elkaar krijgt om te controleren of je gebruiker ook echt is aangemaakt door direct na het aanmaken te controleren of de username voorkomt op de overzichtspagina met gebruikers. Print “gelukt” als dat zo is, en “niet gelukt” als dat niet zo is. Je kan daarvoor de onderstaande code gebruiken (hoeft niet):

```
import time
# soms is het nodig om even te wachten tot de pagina is geladen:
time.sleep(2)

# 'x' is de gevonden waarde, 'username' is de variabele naam voor de user
x = driver.find_element_by_xpath("//td[contains(text(),'+username+')]").text
```
 - d. **editmainmenu**: Deze testcase hoeft NIET te worden geautomatiseerd.
 - e. **<<ietsanders>>**: Automatiseer nog een andere taak met behulp van een script. Je kan ervoor kiezen om eerst de Firefox-plug-in te gebruiken, maar je mag je script ook direct schrijven. Kies uit:
 - i. Nieuwe pagina
 - ii. Een paar Settings aanpassen
 - iii. Een eigen idee.
 - f. **logout**: geen aanpassingen.



2.10.3 Lesstof tentamen

Wat moet je weten voor het tentamen over deze onderwerpen, dus hoe kan je je het beste voorbereiden?

- **Leer** alle informatie die je vindt in de powerpoint presentatie.
- Zorg dat je zelf Katalon Recorder (Firefox Plug-in) en WebDriver-scripts hebt gemaakt zodat je **getoonde code van anderen goed begrijpt** en kan **aanpassen/verbeteren**.

2.10.4 Informatiebronnen

- Firefox Selenium Plugin “download” → <https://addons.mozilla.org/en-US/firefox/addon/katalon-automation-record/>
- User guide: <https://docs.katalon.com/display/KD/KR+User+Guide>
- Selenium documentatie → <http://docs.seleniumhq.org/>



2.11 Week 6, College 1: PowerShell

PowerShell is de manier om op het Windows platform administratieve taken uit te voeren. Meestal gebruik je hiervoor een “oneliner”, maar het is ook mogelijk om grotere scripts uit te voeren.

In de opdrachten hieronder gaan we aan de slag met PowerShell, want daar leer je toch het meeste van.

2.11.1 Lesopdracht

Je hebt tijdens de les een aantal voorbeelden gezien. Probeer deze nu zelf uit, de praktijk is toch anders dan de theorie. Bovendien onthoud je het daardoor veel beter.

Leer jezelf om veel “tab-completion” te gebruiken voor het correct invoeren van cmdlets en parameters, dus zo:

```
get-h<tab> → Get-Help
```

of:

```
Get-Help -ex<tab> → Get-Help -Examples
```

2.11.2 Weekopdracht

PowerShell Oneliners...

- Noem 4 Linux/Unix commando's die ook werken onder Powershell.
- "cd" naar de CA certificaten op de lokale machine. Hoeveel certificaten zie je daar?
- Noem het commando waarmee je de variabelen kan bekijken in de juiste psdrive.
- Welk commando kan je ook gebruiken in plaats van "cd"?
- Welk commando kan je ook gebruiken in plaats van "dir"?
- "cd" naar HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT indien aanwezig. Wat is de ProductName?
- "cls" is een alias van?
- maak een alias aan met de naam "hier" dat verwijst naar "get-location"
- "werkt" de alias nog als je het window met PowerShell afsluit en weer opnieuw opstart?
- Geef het commando waarmee je voorbeelden kan opvragen van het gebruik van "stop-process"
- Met welk commando kan je alle help updaten tot de meest recente versie?
- Met welk commando krijg je online hulp bij het commando "stop-computer"?
- Wat gebeurt er als je een verplichte optie niet opgeeft? Bijvoorbeeld bij "stop-process"?
- Laat alle processen zien met ps en geef deze weer in 4 kolommen.
- Laat alle processen zien met ps en geef deze weer als een lijst (label-waarde).
- Gebruik test-connection om een computer 4x te pingen met steeds 4 seconden tussenpauze.
- Een lastige: maak een lijstje aan met drie computer namen en sla deze op als "hosts.txt". Gebruik test-connection EN get-content om deze hosts achtereenvolgens te "pingen".
- gebruik de pipeline om de nieuwste 10 errors in de eventlog van het systeem te bekijken.
- Gebruik "ps" en converteer de output naar html
- Gebruik "ps" en converteer de output naar csv en sla deze op als file.
- Gebruik "ps" en selecteer verschillende kolommen waaronder TotalProcessorTime
- Gebruik "ps" en sorteer op TotalProcessorTime (aflopend). Welk proces gebruikt de meeste processortijd?
- Idem nu voor het gebruik van virtueel geheugen.
- Vraag een lijst met services op en groepeer op status. Hoeveel services zijn "running" en hoeveel "stopped"?
- Gebruik measure-object om het aantal entries in de c:\windows directory op te vragen.
- Start "notepad" een paar keer. Gebruik nu ps in combinatie met "where" en "like" om alle processen op te vragen die beginnen met een "n".



- Start notepad. Gebruik `get-process` en `stop-process` in een enkele pipeline om notepad te stoppen.

PowerShell Scripts

- Zorg ervoor dat je `executionpolicy` zodanig is aangepast dat je scripts kunt starten.
- Maak dan een script met de naam "np.ps1" dat 3x notepad start, 3 seconden "slaapt", de proces-id's van notepad opvraagt en vervolgens alle notepads stopt.
- Installeer de `telnet-client` met `add-windowsfeature`.
- Zorg dat onderstaande code werkt in PowerShell (kopiëren en uitvoeren).
Port deze nu naar Python en zorg voor dezelfde uitvoer.

```
# Port deze PowerShell code naar Python...
[System.Collections.ArrayList]$leeftijden = 18,22,20,19,21,22,22
$leeftijd = read-host Geef je leeftijd
$leeftijden.add($leeftijd) | out-null
foreach ($i in $leeftijden) {
    write-host Leeftijd: $i
}
$gemiddelde=($leeftijden|measure-object -Average).Average
write-host Het gemiddelde is: $gemiddelde
if ($gemiddelde -lt 20) {
    write-host Allemaal tieners?
} elseif ($gemiddelde -gt 65) {
    write-host Bejaardensoos?
} else {
    write-host Geen jongeren of ouderen...
}
```

Een volledig PowerShell script

- Maak een PowerShell script waarmee je de onderstaande functionaliteit realiseert. Kijk hiervoor naar het bovenstaande script.
 - Vraag de gebruiker om een grootte (in MB) aan te geven.
 - Zoek op de C-drive (of een deel van de drive als het te veel tijd kost) naar bestanden die groter zijn dan de grootte die de gebruiker aangaf.
 - Sorteer deze bestanden op grootte en toon alleen de "top 10".
 - Presenteer deze informatie op een overzichtelijke manier.

2.11.3 Lesstof tentamen

Wat moet je weten voor het tentamen over dit onderwerp, dus hoe kan je je het beste voorbereiden?

- **Leer** alle informatie die je vindt in de powerpoint presentatie over dit onderwerp.
- Zorg dat je **de antwoorden weet op de PowerShell vragen** hierboven.
- Zorg dat je van PowerShell **oneliners begrijpt en kan uitleggen** wat er gebeurt.
- **Optioneel:** de informatie die wordt aangeboden onder het kopje "Informatiebronnen". Het gaat daarin niet om nieuwe informatie, maar om verbreding zodat je de stof beter begrijpt.



2.11.4 Informatiebronnen

- Microsoft Virtual Academy: <https://mva.microsoft.com/en-US/training-courses/advanced-tools-scripting-with-powershell-3-0-jump-start-8277>



2.12 Week 6, College 2: OS Virtualization using Containers

Containers zijn de toekomst voor het effectief beheren van systemen. Het is eigenlijk een volgende stap in virtualisatie. Oorspronkelijk alleen mogelijk onder Linux, maar nu ook mogelijk met behulp van Windows Server. Dit laatste besturingssysteem kent twee typen containers: Windows Server en Hyper-V-containers. Je kan ze beheren met PowerShell of met Docker. Oorspronkelijk was het de bedoeling om dit lab uit te voeren onder Windows 2019, maar dat levert nu (nog) te veel problemen op en daarom gebruiken we een RHEL8 distributie en Docker bij de uitvoering van het lab.

2.12.1 Lesopdracht

-

2.12.2 Weekopdracht

Stap 1: Installeer Docker op RHEL

Gebruik je eerste linux server (de saltstack master bijvoorbeeld, mag ook een andere server zijn, gui hoeft niet). We gaan er bij de installatie vanuit dat je dit als “root” uitvoert. Het beheer van docker gaat prima onder je eigen useraccount als je jezelf toevoegt aan de docker group met:

```
usermod -a -G docker <<jouwusername>>
```

Je moet vervolgens de docker repository toevoegen zodat je met de package manager “dnf” de docker software kan installeren:

```
tee /etc/yum.repos.d/docker.repo <<- 'EOF'  
[dockerrepo]  
name=Docker Repository  
baseurl=https://download.docker.com/linux/centos/8/$basearch/stable  
enabled=1  
gpgcheck=1  
gpgkey=https://download.docker.com/linux/centos/gpg  
EOF
```

Installeer de packages en vergeet niet om daarna de docker-service te enablen en te starten:

```
dnf install docker-ce docker-ce-cli
```

Mocht dit niet lukken, gebruik het volgende commando:

```
dnf --allowmissing install docker-ce docker-ce-cli
```

Stap 2: Beheren van Docker

Het beheer van docker gaat met... “docker”. Run het commando

```
docker
```

Uitleg krijg je door een specifiek subcommando, type:

```
docker subcommand --help
```

Stap 3: werken met containers

Om een container te kunnen starten, moet je eerst een image downloaden. Deze images kan je downloaden vanuit de Docker Hub, een registry gemanged door Docker, het bedrijf achter “docker”. Iedereen kan containers bouwen en hosten op de Docker Hub, dus veel applicaties en linux distributies zijn beschikbaar in deze registry.



Controleer of docker goed is geïnstalleerd door een test image te runnen in de container.

```
docker run hello-world
```

Zie je "Hello from Docker!" in de tekst voorkomen? Dan is je installatie geslaagd.

Je kan zoeken naar images op de Docker Hub met het docker commando en search subcommando, bv om centosimages te vinden type je:

```
docker search centos
```

Je ziet allerlei images, maar let op. In de kolom "OFFICIAL" zie je "OK", dat is de image die wordt ondersteund door het bedrijf achter het project. Ook herkenbaar omdat er geen slash in de naam voorkomt: centos

Je kan deze image downloaden naar je computer met het pull subcommando:

```
docker pull centos
```

Omdat je geen versie hebt opgegeven krijg je de nieuwste versie (latest). Let op, dit is dus een Centos container geïnstalleerd op je host RHEL-besturingssysteem!

Bekijk alle images die je nu hebt gedownload met:

```
docker images
```

Tot nu toe heb je nog niets gedaan met de containers. Je kan er niet eens bij! Daarom, start de centos image met de opties -i en -t: je krijgt dan een interactieve shell, bv:

```
docker run -it centos
ps
uname -a
cat /etc/redhat-release
```

Let op de hostname in de vorm van een nummer. Hierdoor zie je dat je werkt binnen de container.

```
[root@client ~]# docker run -it centos
[root@75279c0d8940 /]# ps
  PID TTY          TIME CMD
    1 pts/0    00:00:00 bash
   14 pts/0    00:00:00 ps
[root@75279c0d8940 /]# uname -a
Linux 75279c0d8940 4.18.0-193.el8.x86_64 #1 SMP Fri May 8 10:59:10 UTC 2020 x86_64 x86_64
4 x86_64 GNU/Linux
```

Vergelijk de redhat-release versie met de versie van je host computer. Zie je dat dit een andere versie is! Je werkt met een Linux systeem in een linux systeem! Vergelijk via uname -a op zowel computer als container ook de kernel versie van beiden. Wat valt je op?

Met exit kan je de interactieve shell verlaten en kom je terug op de host.

```
exit
```

Stap 4: MariaDB gebruiken met behulp van Docker.

Zoek met docker naar de naam van mariadb

```
docker search mariadb
```

Download nu de meest recente mariadb image. Het opgeven van "latest" is de default en mag je daarom ook weglaten.

```
docker pull mariadb:latest
```



Bekijk je lijst met images met
docker images

```
[root@client ~]# docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
mariadb             latest         cbbff8572fa8   36 hours ago   406MB
centos              latest         0d120b6ccaa8   5 weeks ago    215MB
hello-world        latest         bf756fb1ae65   8 months ago   13.3kB
```

Een image is geen running proces, het is alleen de software. Om deze te runnen moet je een container maken.

Het juiste commando hiervoor vind je meestal in de documentatie van de image. Bijvoorbeeld om een container te maken met de naam “mariadbtest” type je het volgende commando:

```
docker run --name mariadbtest -e MYSQL_ROOT_PASSWORD=mypass -d mariadb
```

Met de optie “-e” kan je opties doorgeven voor de mysql-daemon.

Met de optie “-d” draait de container in de achtergrond en wordt je container ID geprint. Elke container heeft zijn id...

```
[root@client ~]# docker run --name mariadbtest -e MYSQL_ROOT_PASSWORD=mypass -d mariadb
8e514edee76cd03d2d91a678e2db378a5ae46fcae20f8d0531cf1122478335bc
```

Docker geeft alle containers een id. Om zeker te weten dat de container “draait” kan je een lijst opvragen met running containers:

```
docker ps
```

Je ziet dan alle “draaiende” containers, wil je alle containers zien type dan:

```
docker ps -a
```

Docker containers kan je pauzeren, stoppen, starten en herstarten.

```
docker pause mariadbtest
docker unpause mariadbtest
docker start mariadbtest
docker restart mariadbtest
docker stop mariadbtest
```

Het troubleshooten van een container? Zorg natuurlijk wel dat de container is gestart.

```
docker logs mariadbtest
```

Wil je interactieve toegang tot de container. Voer dan een interactief commando uit op de container. Zorg natuurlijk wel dat de container is gestart.

```
docker exec -it mariadbtest bash
```

Maar nu, wat is een database als je geen toegang krijgt vanuit de buitenwereld?

Installeer eerst de client op de host (!):

```
sudo dnf install mariadb
```

Maak nu een nieuwe container waarbij poort 3306 van de host wordt geforward naar poort 3306 van de container:

```
docker run --name mariadbtest2 -p 3306:3306 -e MYSQL_ROOT_PASSWORD=mypass -d mariadb
```



Kan je de database nu benaderen vanaf de client? (soms moet je even wachten)

```
mysql -u root -pmypass -h 127.0.0.1
```

Meer weten over een specifieke container?

Zoek via het commando hieronder op welk ip-adres je container heeft gekregen (met ifconfig docker0 zie je dat je host ook een ip heeft gekregen):

```
docker inspect mariadbtest2
```

Probeer nu met mysql (andere -h optie) via het ipadres van de container aan te loggen op de database.

Met “docker stop” wordt de container netjes gestopt. Als je de container wilt verwijderen gebruik je “rm”:

```
docker rm mariadbtest
```

Let op, het commando heeft niet de data van de database verwijderd in /var/lib/docker/. Als je deze ook wilt verwijderen gebruik je

```
docker rm -v mariadbtest
```

Stap 5: Zelf je eigen containers maken.

Je kan zelf een nieuwe container maken door een bestaande container te veranderen en de opgeslagen veranderingen te “commit”-ten zodat ze niet verloren gaan.

In het voorbeeld hieronder ga je de teksteditor “nano” installeren op een eigen container en voer je een “commit” uit waarbij je een message (-m) en een naam (auteur) meegeeft.

Maak een copy van de standaard Centos

```
docker create --name mijncentos -t -i centos
```

Start deze kopie, bekijk of hij is gestart en start een interactieve shell.

```
docker start mijncentos
docker ps
docker exec -it mijncentos bash
```

Installeer in de container nano

```
dnf install nano
exit
```

Als je de container stopt ben je alle veranderingen weer kwijt. Maak daarom een image op basis van centos met Nano geïnstalleerd en kijk of deze in de lijst met images staat.

```
docker commit -m "Centos met Nano" -a "test@hu.nl" mijncentos hu/centosnano
docker images
```

```
[root@linux3 rienk]# docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
hu/centosnano       latest         e451128dfc31   5 seconds ago  302.2 MB
mariadb             latest         63d91190a69e   3 days ago     389.8 MB
centos              latest         980e0e4c79ec   5 weeks ago    196.7 MB
hello-world         latest         c54a2cc56cbb   3 months ago   1.848 kB
```

Kan je van deze image nu een container starten?

Kan je een shell starten op de container en is nano daar geïnstalleerd?



Stap 6: Containers maken met een dockerfile.

Dit leek gemakkelijk en heel aantrekkelijk, maar let op, dat kan een flinke klus worden als je jouw eigen container steeds up-to-date wilt houden. Iedere patch moet je weer handmatig doorvoeren en je maakt zomaar een fout. Ook dat zou je natuurlijk moeten automatiseren! Dat kan met behulp van een dockerfile.

We gaan hieronder een apache webserver container maken met een eigen startpagina. Volledig geautomatiseerd. Begin met de standaard en officiële apache image te downloaden

```
docker pull httpd
```

Maak nu een directory op de host, bv: ~/mijndir

Maak nu in die directory een eigen startpagina onder de naam: index.html:

```
<html><body>
<p>Hello World, this is my first Dockerfile created container</p>
</body></html>
```

Maak in dezelfde directory ook een dockerfile aan met de exacte naam 'Dockerfile' met daarin de onderstaande instructies:

```
FROM httpd
MAINTAINER Jan Jansen janjansen@hu.test
ADD index.html /usr/local/apache2/htdocs/
```

Bouw nu je image "mijnhttpd" met:

```
docker build -t mijnhttpd ~/mijndir/
```

Start de image waarbij je de locale poort 8081 gebruikt en test met een browser of jouw pagina bereikbaar is via `http://localhost:8081`

```
docker run -d -p 8081:80 mijnhttpd
```

Stap 7: Nu zelf!

Alles duidelijk tot nu toe? Nu iets creatiever. Probeer nu zelf een image+werkende container te maken van een service naar keuze...

2.12.3 Lesstof tentamen

- **Leer** alle informatie die je vindt in de powerpoint presentatie over dit onderwerp.
- **Leer** de theorie over Docker zoals te vinden op Canvas over dit onderwerp.
- **Optioneel:** de informatie die wordt aangeboden onder het kopje "Informatiebronnen". Het gaat daarin niet om nieuwe informatie, maar om verbreding zodat je de stof beter begrijpt of kan toepassen.

2.12.4 Informatiebronnen

- "Alle" nuttige docker commando's en cheat sheet: <https://github.com/wsargent/docker-cheat-sheet>



2.13 Bijlage1: Antwoorden 2.1.1, opdracht 1

```
"""
Herhalingsopdrachten PYTHON
1) Zet 5 namen in een lijst en ga er met een for-loop doorheen.
   Print alle namen onder elkaar.
"""
l=['jan','piet','ali','amir','truus']
for i in l:
    print(i)

"""
2) Vraag een gebruiker om een getal.
   Als het getal > 100 dan print je "GROOT", anders print je "klein" op het scherm.
"""
# g=int(input("Geef een getal: "))
# if g>100:
#     print("GROOT")
# else:
#     print("klein")

"""
3) Een auto is rood, de trein is geel en de ambulance ook trouwens.
   Zet deze informatie in een dictionary en print deze vervolgens op het scherm.
"""
d={'auto':'rood', 'trein':'geel', 'ambulance':'geel'}
print(d)
for i in d:
    print(i,"-->",d[i])

"""
4) Gegeven de string: a="The client connects to the server by calling socket()".
   Print karakter 5 t/m 10 op het scherm.
"""
a="The client connects to the server by calling socket()"
print(a[4:10])

"""
5) Gegeven de string uit de vorige opdracht:
   print vanaf de laatste 8 karakters tot het einde.
"""
a="The client connects to the server by calling socket()"
print(a[-8:])

"""
6) Print alle letters uit de vorige opdracht met een spatie tussen elke letter
   (tip: gebruik for-loop en bij de printopdracht: end=" ")
"""
a="The client connects to the server by calling socket()"
for i in a:
    print(i, end=" ")

"""
7) Maak een tuple van 10 getallen en print deze gesorteerd van klein naar groot op het scherm.
   Print daaronder het totaal van alle getallen.
"""
t=(7,55,7,0,22,3,3,4456,8,6)
print(sorted(t))
print(sum(t))

"""
8) Maak een eindeloze loop.
   Vraag daarin de gebruiker om input.
   Als de gebruiker "stop" invoert, dan moet je uit de loop springen en is het programma afgelopen.
"""
while(True):
    s=input("type iets...")
    if s == "stop":
        break

"""
9) Maak een functie.
   De functie krijgt een getal mee en
   geeft het dubbele van dit getal terug als het wordt aangeroepen.
   Roep de functie aan voor de getallen 3, 60 en -3 en print het resultaat op het scherm.
"""
```



```
def f(x):
    return x+x

print(f(3))
print(f(60))
print(f(-3))

"""
10) Vraag de gebruiker om een getal.
    Probeer het getal 100 daarna door de opgegeven invoer te delen en print het resultaat.
    Als de invoer "0" was, dan crashed je programma. Zorg dat dit niet gebeurt (gebruik excepties).
"""
g=input("Geef de noemer ")
try:
    print(100/int(g))
except:
    print("Delen door 0 mag niet.")

"""
11) Maak met notepad een tekstbestand van 4 regels. Open dit bestand in Python en print regel voor
regel op het scherm. Zorg ervoor dat er een regelnummer staat voor elke uitvoerregel. Dus bv zo:
1 blabladfalf dsfjsflksjf
2 ddddf dsfjklfjklfsf
3 sfdkfjdkfjdkfjdkfjdkfjdkf
"""
import os
print(os.getcwd())
t=1
f=open('Week1\\README.txt')
b=f.readlines()
t=1
for i in b:
    print(t,i, end='')
    t+=1
f.close()

"""
12) Maak werkende.
"""
def doe_iets(x):
    if x < 10:
        y = "x is kleiner dan 10"
    elif x>=10 and x<20:
        y = "x tussen 10 en 20"
    else:
        y = "x is groter dan 19"
    return y
getal=int(input("Geef een getal "))
z=doe_iets(getal)
print(z)

"""
13) Maak een lijst met daarin twee lijstjes met ieder 3 getallen.
    print uit de tweede lijst het tweede getal.
"""
p=[[1,2,3,4],[11,22,33,44]]
print(p[1][1])

"""
14) Print de huidige werkdirectory (linux: pwd). Tip gebruik de module os en functie getcwd..
"""
import os
print("De huidige directory is:",os.getcwd())
```



2.14 Bijlage2: Antwoorden 2.3.1, functies

<https://www.w3resource.com/python-exercises/>

```
### 1 ### easy
l=(8, 2, 3, -1, 7)
def find_highest(my_list):
    x=max(my_list)
    return x
print(find_highest(l))
```

```
### 2 ### easy
def calculate_birth_year(age):
    year_now=2017
    return year_now-age
print(calculate_birth_year(51))
```

```
### 3 ### easy
import datetime
def calculate_birth_year(age):
    year_now = datetime.datetime.today().year
    return year_now-age
print(calculate_birth_year(51))
```

```
### 4 ### easy
def check_number(number,start,stop):
    if number > start and number < stop:
        return 'yes'
    else:
        return 'no'
r=check_number(23,10,100)
print("Is number in given range?:", r)
```

```
### 5 ### easy
def is_even_num(l):
    even_number = []
    for n in l:
        if n % 2 == 0:
            even_number.append(n)
    return even_number
l=[1, 2, 3, 4, 5, 6, 7, 8, 9]
print(is_even_num(l))
```



```
### 6 ### difficult
l=(8, 2, 3, -1, 7)
def multiply_all(my_list):
    total=1
    for i in my_list:
        total *=i
    return total
print(multiply_all(l))

### 7 ### medium/difficult
s="1234abcd"
l=len(s)
for i in range(l):
    print(s[l-i-1],end='')
print()

### 8 ### difficult
def check_prime(number):
    flag=False
    for i in range(number):
        if i == 1 or i ==0:
            continue
        elif number % i == 0:
            flag=True
            print(number,' is deelbaar door:', i)
    return flag

g=7 # 70=geen priemgetal | 7 is een priemgetal
r=check_prime(g)
if r==False:
    print(g," = een priemgetal")
else:
    print(g," = geen priemgetal")
```