TECHNISCHE DOCUMENTIE

TEAM 3 - YELLOWHOST

IT Factory
CCS

Luka, Jorn, Dries, Kateryna, Jonathan, Lennerd, Henk

Academiejaar 2017-2018

Campus Geel, Kleinhoefstraat 4, BE-2440 Geel

Inhoud

1. TECHNISCHE DOCUMENTATIE

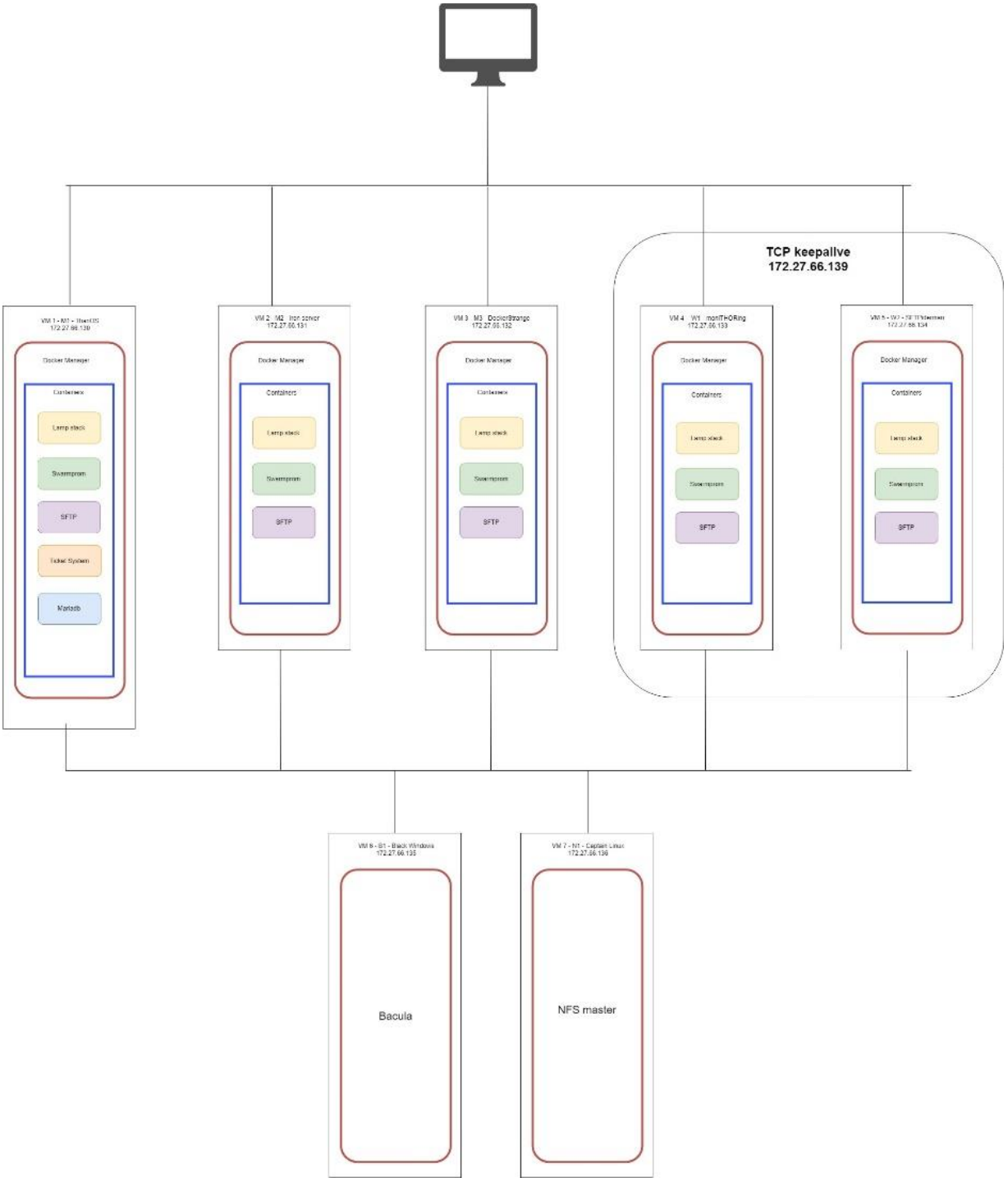
1.1. Inleiding

In dit deel vind je een overzicht van onze hosting opstelling. Hier kan je onder meer zien hoeveel VM's we gebruiken, het bijhorend IP, de services die op elke VM draaien en tot slot de naam die we aan elke VM hebben gekoppeld. Hieronder volgt ook nog een schema dat onze opstelling visualiseert.

1.2. Overzicht virtuele machines

VM nr.	SERVICES	IP-address	VM-naam
VM 1	Docker manager (SFTP, LAMP, OTRS, Monitoring, MariaDB)	172.27.66.130	M1-ThanOS
VM 2	Docker manager (SFTP, LAMP, Monitoring)	172.27.66.131	M2-Iron server
3nVM 3	Docker manager (SFTP, LAMP, Monitoring)	172.27.66.132	M3-DockerStrange
VM 4	Docker Worker (SFTP, LAMP, Monitoring)	172.27.66.133 172.27.66.139	W1-moniTHORing
VM 5	Docker Worker (SFTP, LAMP, Monitoring)	172.27.66.134 172.27.66.139	W2-SFTPiderman
VM 6	Bacula	172.27.66.135	B1-Black Windows
VM 7	NFS	172.27.66.136	N1-Captain Linux
VM 8	NFS (master)	172.27.66.137	N2-Hulk Storage
VM 9	QA & development Docker manager2	172.27.66.138	L1-Stan Lee
VM10	QA & development Docker worker2		L2-Ant Swarm

1.3. Infrastructure-diagram



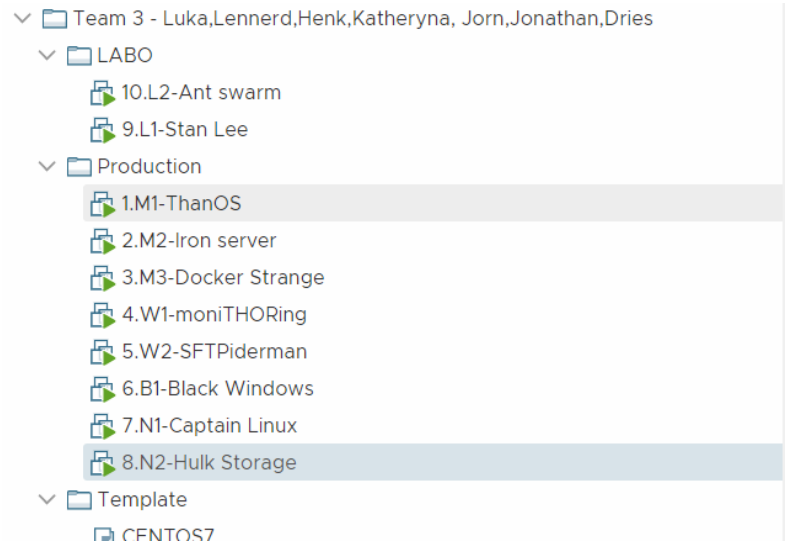
2. CENTOS INSTALLATIE

2.1. Inleiding

We hebben CentOS gekozen als Operating System. Hiervoor is de installatie van CentOS op elke VM vereist. In dit deel kan je het installatieproces dat we hebben toegepast stapsgewijs volgen.

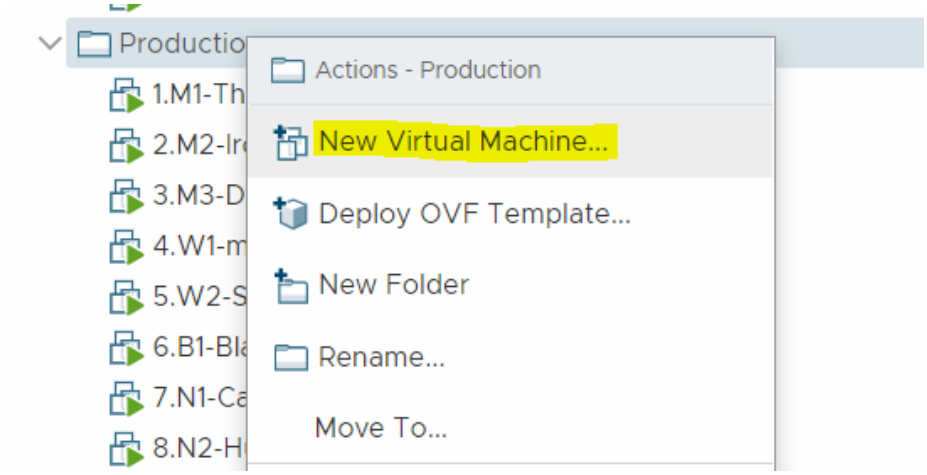
2.2. Installatie in vSphere

In onderstaande foto zie je een overzicht van onze virtuele omgeving die we gedurende deze week hebben opgesteld.

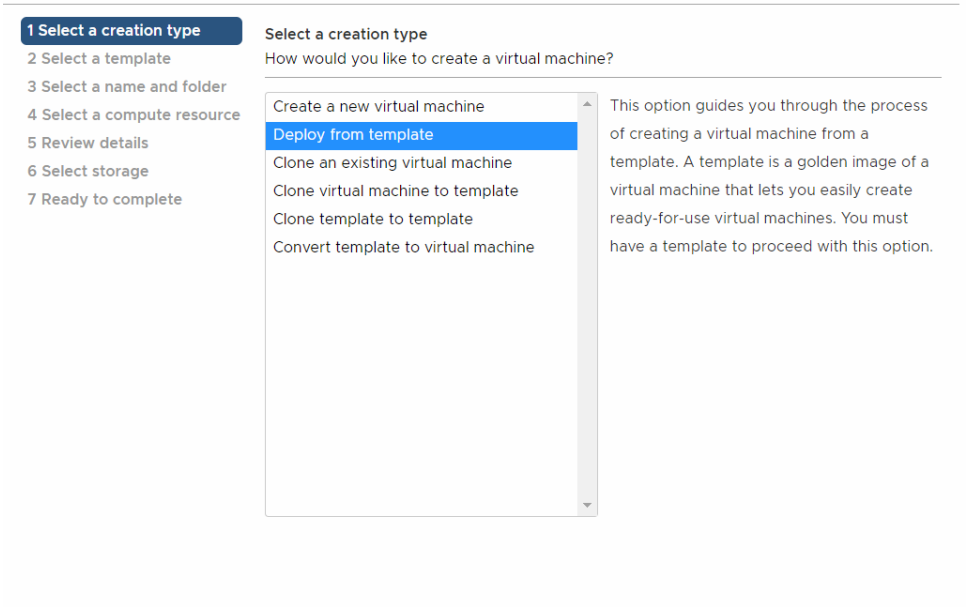


2.2.1. Aanmaken van een virtuele machine

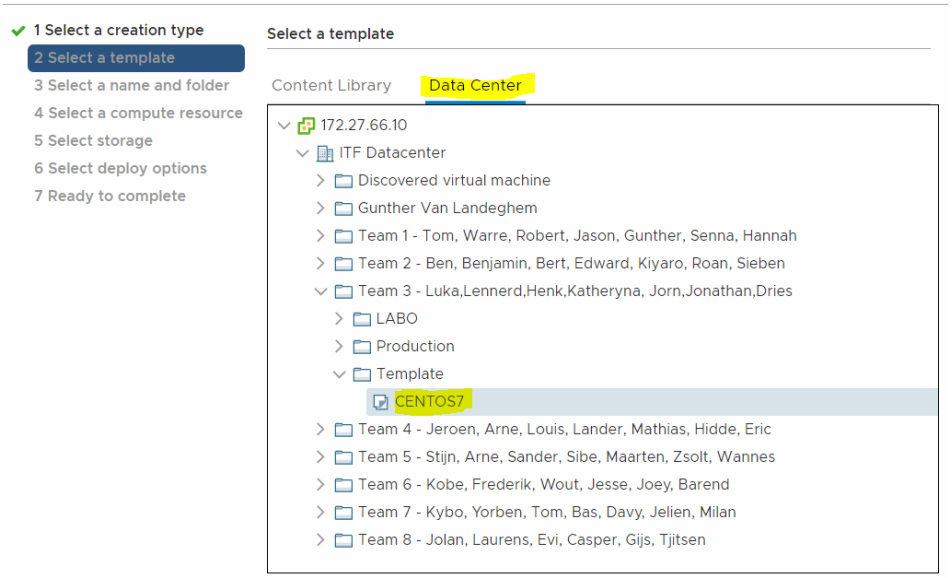
Indien we een map selecteren en vervolgens rechtermuis klikken, kunnen we kiezen voor "New Virtual Machine" vervolgens opent er een 7 stappen proces.

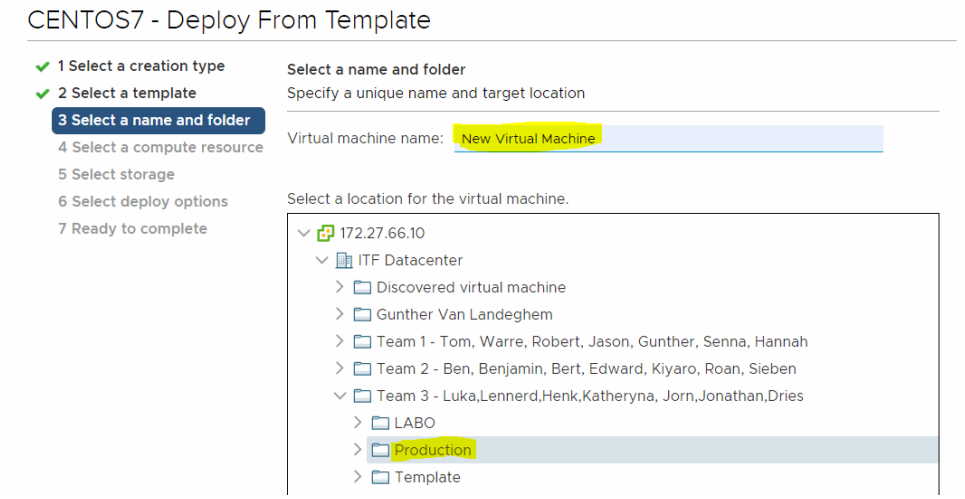


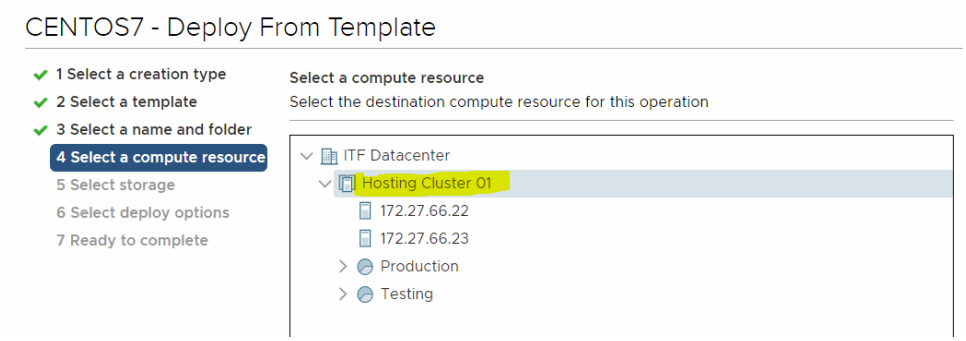
We kiezen hier deploy from template aangezien we reeds een template met CentOS ter beschikking hebben.

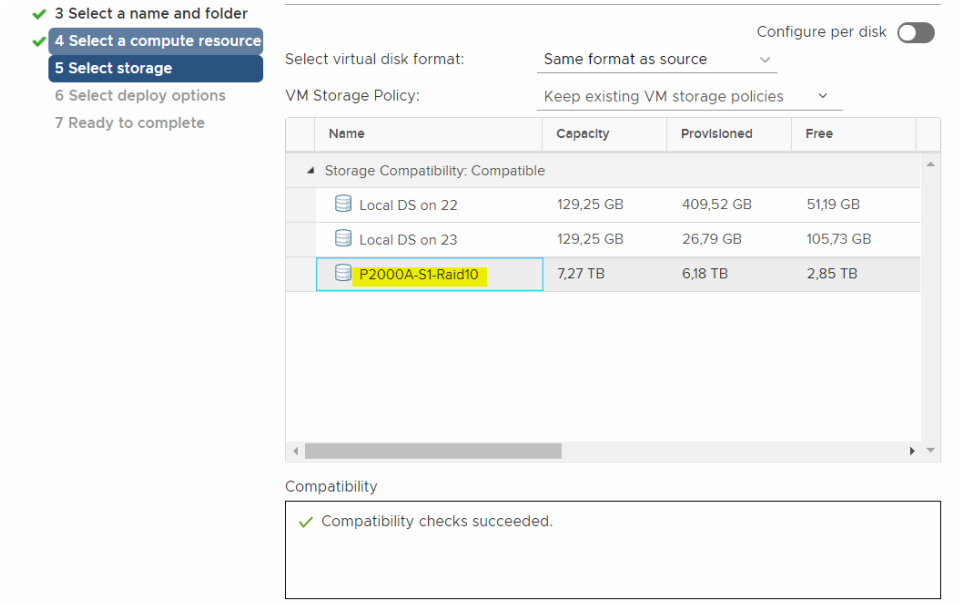


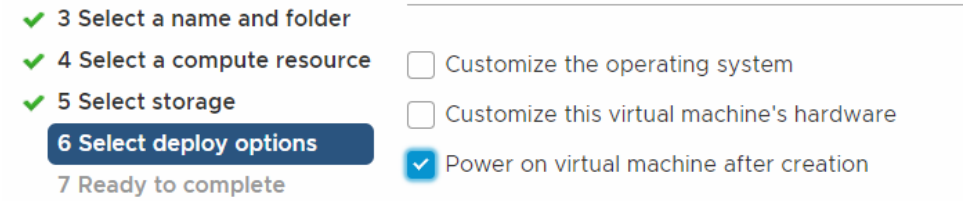
Onderstaande foto's tonen het overige proces nog.

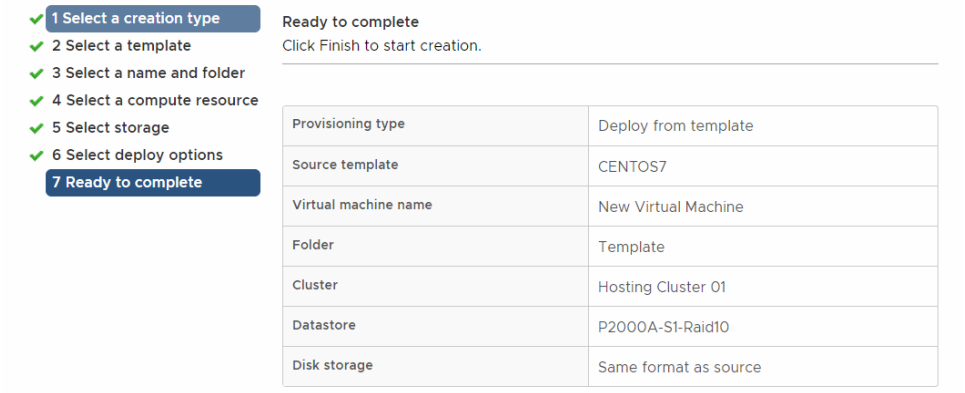





CENTOS7 - Deploy From Template
1 Select a creation type
2 Select a template
3 Select a name and folder
4 Select a compute resource
5 Select storage
6 Select deploy options
7 Ready to complete
Select a compute resource
Select the destination compute resource for this operation
ITF Datacenter
Hosting Cluster 01
172.27.66.22
172.27.66.23
Production
Testing

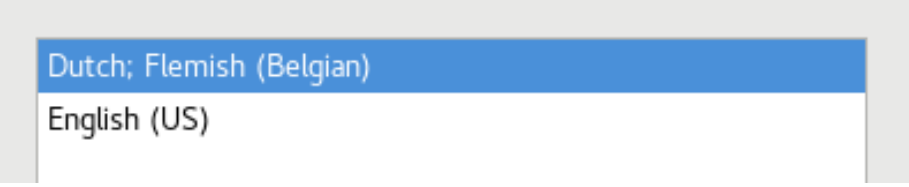

3 Select a name and folder
4 Select a compute resource
5 Select storage
6 Select deploy options
7 Ready to complete
Configure per disk
Select virtual disk format:
Same format as source
VM Storage Policy:
Keep existing VM storage policies
Name
Capacity
Provisioned
Free
Storage Compatibility: Compatible
Local DS on 22
129,25 GB
409,52 GB
51,19 GB
Local DS on 23
129,25 GB
26,79 GB
105,73 GB
P2000A-S1-Raid10
7,27 TB
6,18 TB
2,85 TB
Compatibility
Compatibility checks succeeded.


3 Select a name and folder
4 Select a compute resource
5 Select storage
6 Select deploy options
7 Ready to complete
Customize the operating system
Customize this virtual machine's hardware
Power on virtual machine after creation

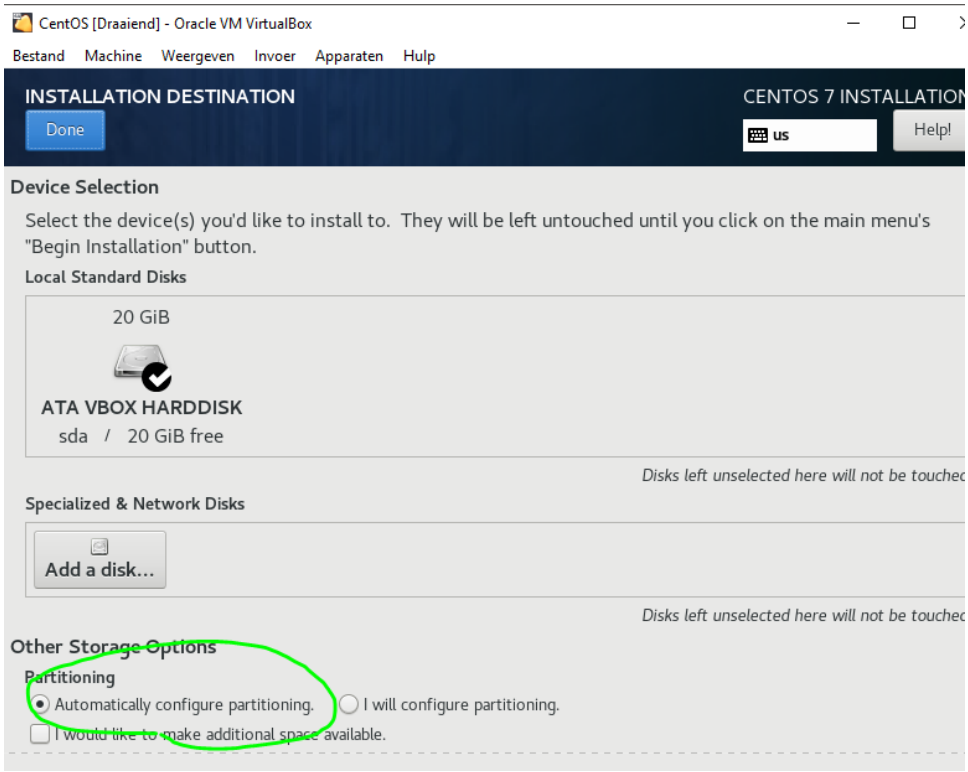

1 Select a creation type
2 Select a template
3 Select a name and folder
4 Select a compute resource
5 Select storage
6 Select deploy options
7 Ready to complete
Ready to complete
Click Finish to start creation.
Provisioning type
Deploy from template
Source template
CENTOS7
Virtual machine name
New Virtual Machine
Folder
Template
Cluster
Hosting Cluster 01
Datastore
P2000A-S1-Raid10
Disk storage
Same format as source

2.3. Installeren van CentOS

1. Installatie taal kiezen (English)
2. DATA & TIME -> Tijdzone juist instellen (Brussels)
3. Keyboard -> op + duwen en Dutch; Flemish (Belgian) toevoegen (zorg ervoor dat dutch bovenaan staat met pijlen naast + knoop)



4. INSTALLATION DESTINATION -> Zorg dat "Automatically configure partitioning" vanonder aangeduid is:



5. Klik op "Begin Installation" en wacht totdat het geïnstalleerd is.
6. Kies wachtwoord voor de root gebruiker: (vb root)
7. Geen user aanmaken
8. Klik op "Finish configuration" en vervolgens op "Reboot".

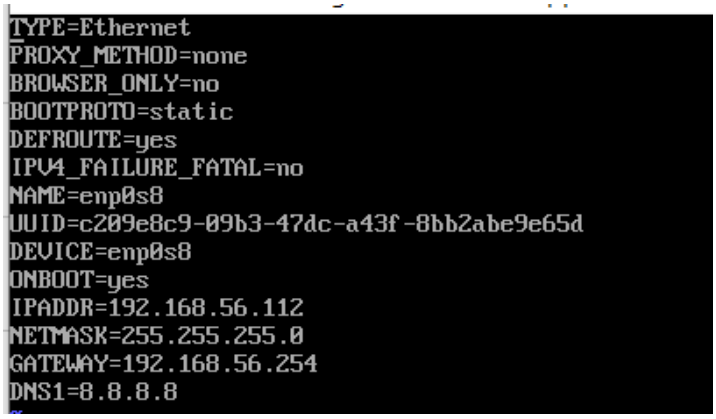
2.4. Netwerkinstellingen aanpassen

Nu de installatie rond is, gaan wij de netwerkinstellingen aanpassen.

```
vi /etc/sysconfig/network-scripts/ifcfg-enpXsX
```

Druk op i (insert mode) om het bestand te kunnen aanpassen. Verander laatste lijn:
Onboot=no naar yes

Verander het IPADD naar één van de IP's uit de gekregen range. (bv 172.27.66.133),voeg gateway, dns en netmask toe + :wq om op te slaan:



Voorbeeld in virtual box

```
reboot now
```

Ping 8.8.8.8 zou nu moeten werken (ctrl + c om te stoppen)

Installeer nu ssh of download een programma zoals "mRemoteNG" zodat copy paste werkt

```
yum install openssh openssh-server openssh-clients openssl-libs
```

3. TIJDZONE INSTELLEN

3.1. Inleiding

De tijdzone op alle servers juist in te stellen is belangrijk voor het monitoren van systemen en het instrumenteel voor het oplossen van problemen wanneer ze ontstaan.

3.2. NTP-server instellen

Eerst zullen wij de NTP-service installeren op de server met dit commando.

```
yum install ntp
```

Nu de NTP-service geïnstalleerd is op de server gaan we de service starten en ook deze enablen zodat de service bij het opstarten vanzelf start.

```
sudo systemctl start ntpd
```

```
sudo systemctl enable ntpd
```

Als tijdserver gebruiken wij de volgende servers.

```
server 0.be.pool.ntp.org
server 1.be.pool.ntp.org
server 2.be.pool.ntp.org
server 3.be.pool.ntp.org
```

Om deze in te stellen, moeten wij een bestand aanpassen met volgend commando.

```
sudo vi /etc/ntp.conf
```

Hierin staan al enkele ntp servers in. Deze gaan wij aanpassen naar de servers die wij gekozen hebben.

Nu gaan we nog twee opties bijvoegen namelijk

```
kod limited
```

Deze zet je in hetzelfde bestand achter noquery vanboven. Eens dit gedaan is gaan wij uit het bestand door ctrl+c in te drukken en het volgende te typen.

```
!wq
```

Om de wijzigingen door te voeren, voeren wij dit uit.

```
sudo systemctl restart ntpd
```

Na enkele minuten kunnen wij nakijken of het gewerkt heeft door het volgende te doen.

```
ntpq –p
```

```
date –R
```

4. NFS (NETWORK FILE SYSTEM)

4.1. Inleiding

NFS maakt het mogelijk om bestanden op te vragen over het netwerk op dezelfde manier als men dit zou doen voor bestanden op de lokale schijf.

Wij gebruiken nfs om ervoor te zorgen dat de bestanden vanuit elke client-server naar de master nfs-server worden gestuurd. Elke client is gemount met de master. De bestanden zullen via een share automatisch worden opgeslagen op de nfs master.

In onze situatie starten we een nieuwe CentOS 7 machine waar we een NFS-server op installeren. Het ip adres van master is 192.168.56.8. Daarnaast wordt ook nog een client-server geïnstalleerd met het volgende ip adres 192.168.56.15.

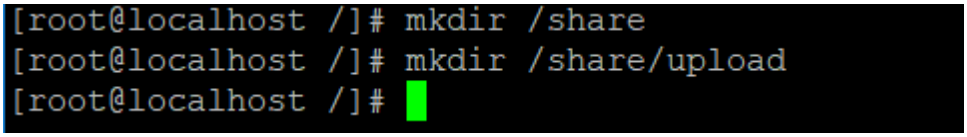
4.2. Installeren NFS op master CentOS 7

Als eerste stap installeren we NFS op de CentOS server met yum:

```
yum install nfs-utils
```

Nu creëren we het map die geshared zal worden door NFS:

```
mkdir /share/upload
```



De rechten van de map veranderen:

```
chmod -R 777 /share/upload
chown nfsnobody:nfsnobody /share/upload
```

We zullen de map /var/nfsshare gebruiken als shared folder. Het is belangrijk dat we bijvoorbeeld geen /home map gebruiken, want dan zullen de permissies een groot probleem kunnen worden in de hiërarchie van het systeem. Mochten we wel de map /home willen sharen dan moeten we de rechten aanpassen.

De volgende stappen zullen services starten en activeren, zodat ze opstarten bij het booten:

```
systemctl enable rpcbind
systemctl enable nfs-server
systemctl enable nfs-lock
systemctl enable nfs-idmap
systemctl start rpcbind
systemctl start nfs-server
systemctl start nfs-lock
systemctl start nfs-idmap
```

Nu gaan we de NFS-map sharen over het netwerk:

```
nano /etc/exports
```

We maken één punt waar we mee kunnen sharen. De bovenstaande file /etc/exports gaan we moeten aanpassen. Zie hieronder de inhoud van de file.

```
/share/upload
172.27.66.130(rw,sync,no_root_squash,no_all_squash)

/share/upload
172.27.66.131(rw,sync,no_root_squash,no_all_squash)

/share/upload
172.27.66.132(rw,sync,no_root_squash,no_all_squash)
/share/upload
172.27.66.133(rw,sync,no_root_squash,no_all_squash)

/share/upload
172.27.66.134(rw,sync,no_root_squash,no_all_squash)

/share/upload
172.27.66.135(rw,sync,no_root_squash,no_all_squash)

/share/upload
172.27.66.136(rw,sync,no_root_squash,no_all_squash)

share/upload
172.27.66.138(rw,sync,no_root_squash,no_all_squash)
```

Als laatste gaan we NFS-service herstarten:

```
systemctl restart nfs-server
```

& we moeten de NFS-service firewall-cmd public zone overschrijven in CentOS7:

```
firewall-cmd --permanent --zone=public --add-service=nfs
firewall-cmd --permanent --zone=public --add-service=mountd
firewall-cmd --permanent --zone=public --add-service=rpc-bind
firewall-cmd –reload
```

NFS CLIENT INSTALLATIE

In dit geval gebruiken we een CentOS7 als client. Nogmaals installeren we de nfs-utild package:

```
yum install nfs-utils
```

Nu maken we de NFS-mappen aan die worden gebruikt als mount point:

```
mkdir -p /mnt/nfs/share/upload
```

Hierna zullen we de NFS gedeelde /share/upload/ map mounten in de client machine als volgt:

```
mount -t nfs 172.27.66.137:/home /mnt/nfs/share/upload/
```

Nu zijn we geconnecteerd met de NFS-share, we zullen een dubbele check doen om te kijken of het is gelukt:

```
df –kh
```

We zijn geconnecteerd met de NFS-share.

We zullen een Read/Write check doen met het shared pad. Probeer het volgende commando:

```
touch /mnt/nfs/var/nfsshare/test_nfs
```

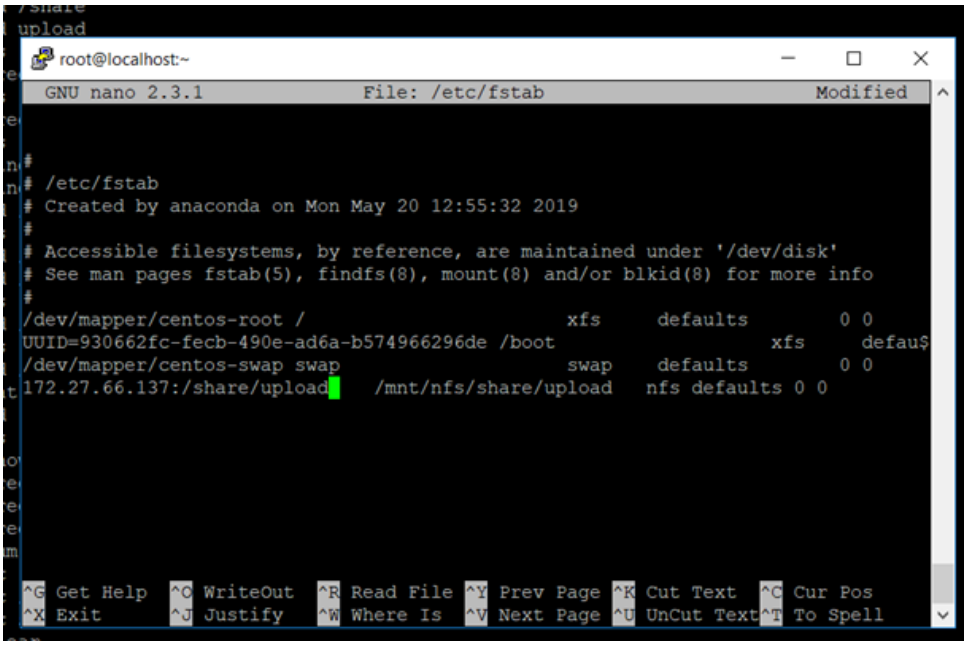
De file is terug te vinden op de master:

```
test_nfs
```

De NFS-share is succesvol geconfigureerd.

Nu gaan we de share definitief maken door in /etc/fstab een regel toe te voegen:

```
172.27.66.137:/share/upload    /mnt/nfs/share/upload    nfs defaults 0 0
```



5. DOCKER INSTALLATIE

5.1. Inleiding

We hebben ervoor gekozen om onze opstelling via Docker Swarm te implementeren. Hiervoor is een installatie van Docker vereist op de bepaalde VM's. In dit deel geven we het installatieproces voor Docker en vervolgens hoe je Docker Swarm kan implementeren.

5.2. Docker installatie

Eerst installeren wij de componenten die docker nodig heeft om te werken:

```
sudo yum install -y yum-utils \ device-mapper-persistent-data \ lvm2
```

Eens we de componenten hebben, kunnen wij de repository instellen vanwaar wij docker zullen installeren:

```
sudo yum-config-manager \ --add-repo \
https://download.docker.com/linux/centos/docker-ce.repo
```

Installeer nu docker van deze repository:

```
sudo yum install docker-ce docker-ce-cli containerd.io
```

Docker start niet automatisch op boot dus voer deze commando's uit na de installatie:

```
sudo systemctl start docker
```

```
sudo systemctl enable docker
```

6. DOCKER SWARM AANMAKEN

6.1. Inleiding

Om de creatie en vernietiging van onze containers te automatiseren gebruiken we docker swarm. Docker swarm is inbegrepen bij docker en is ideaal voor kleine tot middel grote omgevingen door gemakkelijkheid van gebruik. Intern doet docker swarm ook aan load balancing waardoor wij dit niet meer moeten doen.

6.2. Configuratie

Voer het onderstaande commando uit op de VM waarvan je wilt dat deze swarm manager wordt:

```
docker swarm init --advertise-addr <ip-addr>
```

<ip-addr> is het 192.168 address

Na dit commando krijg je een docker join commando om uit te voeren op vm's die je wilt toevoegen aan de swarm. Je mag deze toevoegen als workers of als managers.

Nu gaan we poorten moeten openzetten.

Voor je start, controleer status van de firewall:

```
systemctl status firewalld
```

Het zou nog niet moeten runnen, dus start het:

```
systemctl start firewalld
```

Enable het nu zodat het start bij de boot:

```
systemctl enable firewalld
```

Op de node dat een Swarm manager gaat zijn, gebruik de volgende commando's om de nodige poorten te openen:

```
firewall-cmd --add-port=2376/tcp --permanent
firewall-cmd --add-port=2377/tcp --permanent
firewall-cmd --add-port=7946/tcp --permanent
firewall-cmd --add-port=7946/udp --permanent
firewall-cmd --add-port=4789/udp --permanent
```

Nota: Als je een fout maakt en een entry moet verwijderen, type:

```
firewall-cmd --remove-port=port-number/tcp —permanent
```

Daarna, herlaad de firewall:

```
firewall-cmd –reload
```

Herstart Docker opnieuw:

```
systemctl restart docker
```

Voer dan op elke node dat gaat functioneren als een Swarm worker de volgende commando's uit:

```
firewall-cmd --add-port=2376/tcp --permanent
firewall-cmd --add-port=2377/tcp --permanent
firewall-cmd --add-port=7946/tcp --permanent
firewall-cmd --add-port=7946/udp --permanent
firewall-cmd --add-port=4789/udp --permanent
```

Daarna, herlaad de firewall:

```
firewall-cmd --reload
```

Herstart Docker:

```
systemctl restart docker
```

Je hebt succesvol de used FirewalID gebruikt om de nodige poorten te openen voor Docker Swarm.

We zullen nu een worker aanmaken. Om een worker of manager token op te vragen, voer je het volgende commando uit:

```
docker swarm join-token <worker of manager>
```

(worker of manager afhankelijk van welke token je wilt krijgen)

De token die je krijgt paste je vervolgens in de VM die je wilt toevoegen (CTRL+c)

7. LAMP STACK SERVICE IN DOCKER SWARM

7.1. Inleiding

Wij hebben er voor gekozen om een LAMP stack te deployen in een container die via onze docker swarm wordt verspreid over het netwerk. Onze website/web applicatie runnen bovenop deze stack. (Linux, Apache, MariaDB, PHP)

7.2. Configuratie

Maak een map om php projecten in te steken:

```
mkdir /www
```

Maak een php bestand aan:

```
vi /www/index.php
```

Kopieer deze inhoud in het php bestand.

```
<html>
 <head>
  <title>PHP Test</title>
 </head>
 <body>
 <?php echo '<p>Hello World</p>'; ?>
 </body>
</html>
```

Kijk na of de swarm werkt:

```
docker node ls
```

Er zouden twee machines moeten zijn.

Maak een service aan voor de docker manager. Deze verspreid dan de container op de verschillende machines. Het commando zal naast het aanmaken van de service ook de map binden met php bestanden.

```
docker service create  --name lamp  --replicas 1 --publish 443:443  --mount type=bind,src=/mnt/nfs/share/upload/php,destination=/srv/http  lostmylife/lamp_test:firsttry
```

Surf naar https://<ip-addr>.

8. SFTP SERVICE IN DOCKER SWARM

8.1. Inleiding

Secure file transfer protocol service (SFTP) zal ervoor zorgen dat de gebruikers via een veilige manier kunnen connecteren (via encryptie). Deze service hebben we gecontaineriseerd via Docker.

8.2. Configuratie

Maak een service aan voor de docker manager deze verspreid de container op de verschillende machines.

```
docker service create --name sftp --replicas 4 --publish 2222:22 --mount type=bind,src=/mnt/nfs/share/upload/php,destination=/home --mount type=bind,src=/mnt/nfs/share/upload/users/users.conf,destination=/etc/sftp/users.conf,ro atmoz/sftp
```

Onder de gemounte upload folder moet door root nog een folder worden aangemaakt

Maak op elke docker swarm server de volgende folders aan:

```
mkdir -p /host/upload
```

```
mkdir -p /users
```

Maak op elke server in de folder /users een file met de volgende inhoud:

```
Cd /users
```

```
Touch users.txt
```

```
Vi users.txt
```

foo:123:1001:100:/home/foo/upload

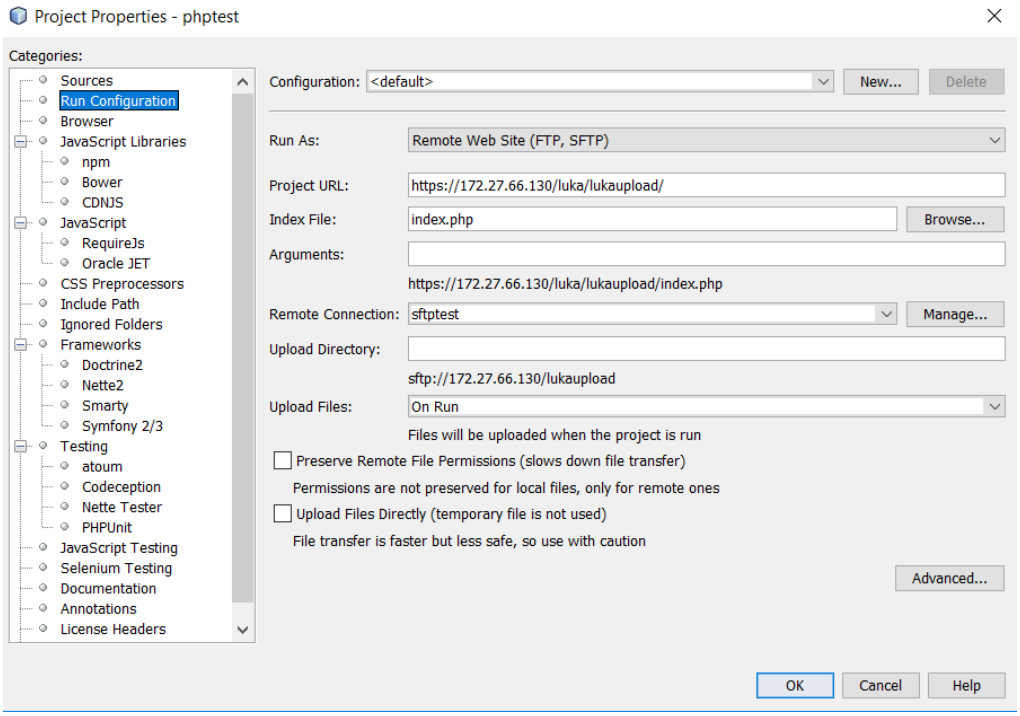
bar:abc:1002:100:/home/bar/upload

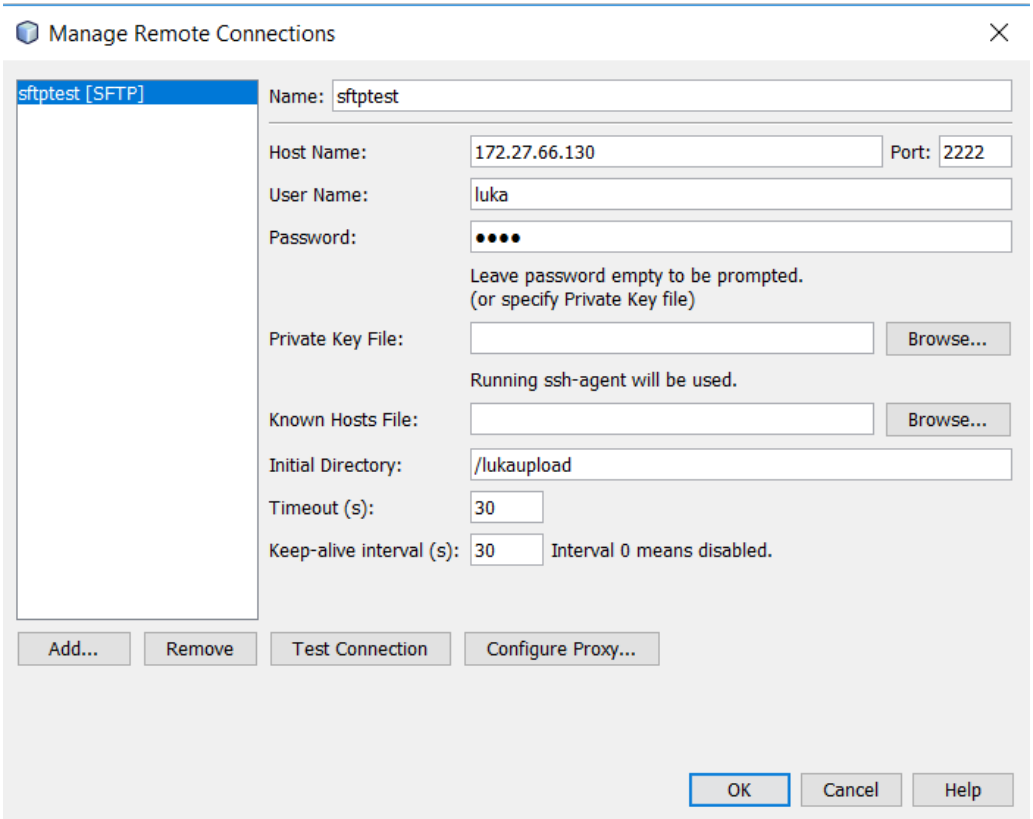
baz:xyz:1003:100:/home/baz/upload

Syntax:

username:wachtwoord:userid:groupid:folderwaarinzekunnenuploaden

Netbeans connectie instellingen





9. PHPMYADMIN EN SQL DATABASE

9.1. Inleiding

phpMyAdmin is een webapplicatie waarmee MySQL-databases via een browser beheerd en geraadpleegd kunnen worden. In onze MariaDB worden onder meer backups en user credentials geplaatst.

9.2. Configuratie

Aanmaken van de Mariadb service. Uitvoeren op een manager:

```
docker service create --name mariadb --mount
type=bind,src=/mnt/nfs/share/upload/mariadb,destination=/var/lib/mys
ql -e MYSQL_ROOT_PASSWORD=root123 –mode global --publish
3306:3306 mariadb:latest
```

Aanmaken van de phpmyadmin service. Uitvoeren op een manager.

```
docker service create --name myadmin -d -e PMA_HOST=172.27.66.139
-p 8888:80 phpmyadmin/phpmyadmin
```

10. OSTICKET

10.1. Inleiding

OS ticket is een ticketingsysteem waarmee gebruikers van ons systeem problemen kunnen melden aan administrators. Dit doen ze door een virtueel ticket aan te maken waarop de admins via een specifieke admin panel een overzicht krijgen van deze tickets en vervolgens kunnen oplossen.

10.2. Configuratie

Met dit commando maken we een container aan met de OSticket service in:

```
docker service create --name Osticket --replicas 1 --publish 8080:80 -e MYSQL_HOST=172.27.66.130 -e MYSQL_USER=root -e MYSQL_PASSWORD=root123 campbellsoftwaresolutions/osticket
```

Na deze installatie kan je nakijken of OSticket bereikbaar is. Ga naar volgend adres in je browser. De standaard logingegevens zijn:

http://localhost:8080/scp/.

- **username:** ostadmin
- **password:** Admin1

11. USER ROLE AANMAKEN

11.1. Inleiding

Omdat we niet op elke server apart een login wouden maken met een nieuw wachtwoord en onszelf root rechten geven, hebben we dit vergemakkelijkt door een scriptje te schrijven.

11.2. Configuratie en naming-convention

De naming-convention die wij gekozen hebben kan je in deze tabel bekijken.

Naming-Convention	Afkorting	User name
System	Sys-	Service naam
Admin	ADM-	Voornaam Achternaam
User	U-	Voornaam Achternaam

11.3. Script

```
#!/bin/sh
declare -a namen=(ADM-LGeentjens ADM-LKolb ADM-JPeeters ADM-DJanssen ADM-HHinrichs ADM-JKitooto ADM-KMekhova)
for i in "${namen[@]}";
do
    sudo useradd -p $(openssl passwd -1 test) $i
    echo "$i ALL=(ALL) ALL" >> /etc/sudoers
    passwd --expire $i
    chage -l $i
done
```

Dit is het script. We hebben een array gemaakt met al onze namen in, en daarna een for-loop om alle namen hetzelfde te geven.

Het script maakt users aan met een wachtwoord 'test' om het te beveiligen voor de eerste login. Daarna zet hij de users in de groep 'etc/sudoers' om deze sudo-rechten te geven.

Daarna zorgde het scriptje dat de wachtwoorden gingen expire en dat hij een nieuw wachtwoord moet ingeven na de eerste login.

We hebben deze shellscript op onze servers gezet. Op de servers doen we dan eerst: chmod +x namen.sh, waar 'namen.sh' ons naam is van onze shellscript.

Hierna voeren we de shellscript uit door './namen.sh' te doen.
Als dit errors geeft kunnen we 2 dingen doen: het bestand op unix-formaat zetten en terug op onze server zetten, of in onze server het

commando geven: ‘sed -i -e 's/\r$//' namen.sh’. Dit zet het script ook in unix formaat. Hierna kunnen we het bestand wel uitvoeren door ‘./namen.sh’ uit te voeren.

Eerste login:

Username	**Eerste login wachtwoord**
ADM-LGeentjens	test
ADM-LKolb	test
ADM-JPeeters	test
ADM-DJanssen	test
ADM-HHinrichs	test
ADM-JKitooto	test
ADM-KMekhova	test

De eerste keer inloggen met dit account doe je door 2x als wachtwoord ‘test’ te ingeven. Hierna moet je zelf een wachtwoord kiezen. Je wordt hierna uitgelogd van de server. Als je je inlogt dan moet dit vanaf nu met het nieuwe wachtwoord.

Gebruikte Documentatie:

https://www.digitalocean.com/community/tutorials/how-to-create-a-sudo-user-on-centos-quickstart

12. SWARMPROM (MONITORING)

1.1. Inleiding

Eén van de opsreportcards is het bijhouden van maandelijkse waardes/gegevens van onze servers om deze later te kunnen analyseren en om makkelijker fouten/bugs in het systeem te kunnen vinden. Hiervoor hebben wij Swarmprom gekozen aangezien dit een bekende monitoring tool is voor Docker implementaties en bovendien een mooie GUI heeft.

Voorwaarde:

- Docker geïnstalleerd
- Swarm met minstens 1 manager en 1 worker.

Installeren van swarmprom

Eerst installeren we git.

```
sudo yum install git
```

Vervolgens git repository ophalen.

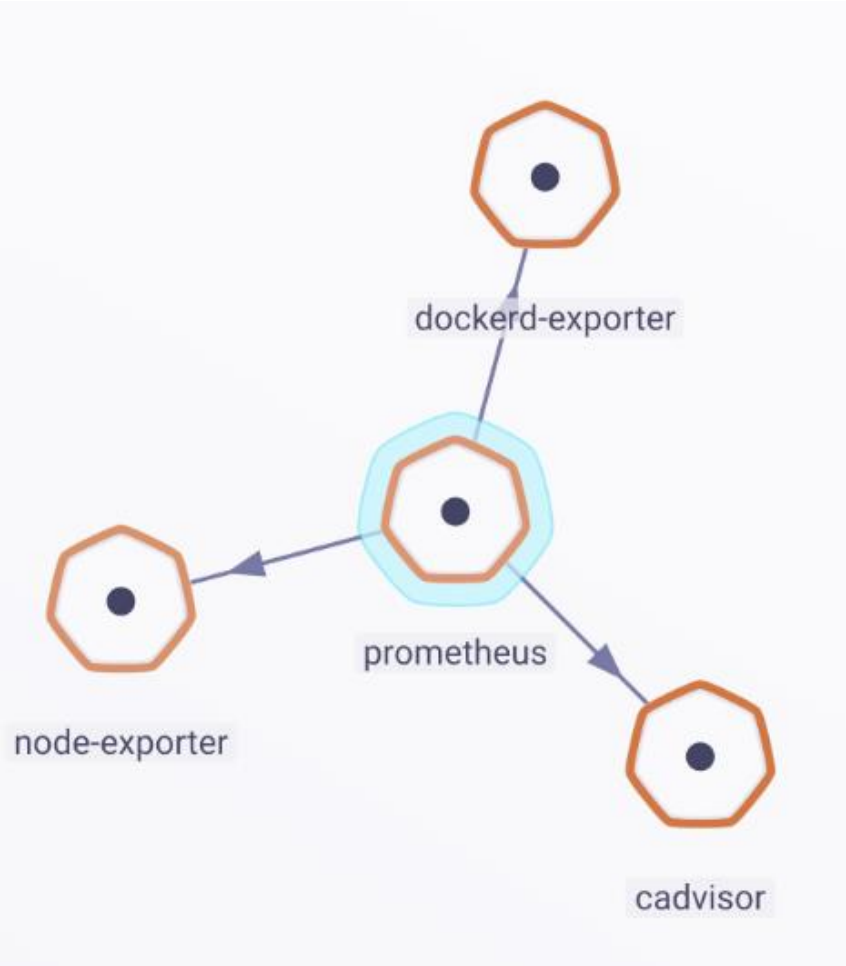
```
git clone https://github.com/stefanprodan/swarmprom.git
```

```
cd swarmprom
```

Tot slot:

```
ADMIN_USER=admin \ADMIN_PASSWORD=admin
\SLACK_URL=https://hooks.slack.com/services/TOKEN
\SLACK_CHANNEL=devops-alerts \SLACK_USER=alertmanager \docker
stack deploy -c docker-compose.yml mon
```

Swarmprom is een starterskit voor Docker Swarm monitoring met Prometheus, Grafana, cAdvisor, Node Exporter, Alert- Manager en Unsee.



13. CLAMAV

13.1. Inleiding

Clam AntiVirus (ClamAV) is een opensource-antivirusprogramma die met name trojans, virussen en malware detecteert. In deze tutorial zullen we illustreren hoe wij ClamAV installeren op onze CentOS 7. Daarnaast installeren wij ook nog Rkhunter (rootkithunter)/ . Dit is een tool die de servers scant op rootkits en algemene beveiligingslekken.

13.2. Installatie

Eerst gaan we het systeem updaten.

```
yum -y update
```

Vervolgens gaan we de EPEL repository toevoegen in ons systeem aangezien ClamAV niet beschikbaar is in de standaard repository van CentOS 7. Dit doe je met volgende commando's:

```
yum -y install epel-release
```

```
yum -y update
```

```
yum clean all
```

```
sudo yum install clamav clamav-update clamav-scanner-systemd clamav-server-systemd
```

Daarna passen we de configuratie aan door de Example-tekst van 2 files uit commentaar te plaatsen.

```
sudo sed -i -e "s/^Example/#Example/" /etc/freshclam.conf
```

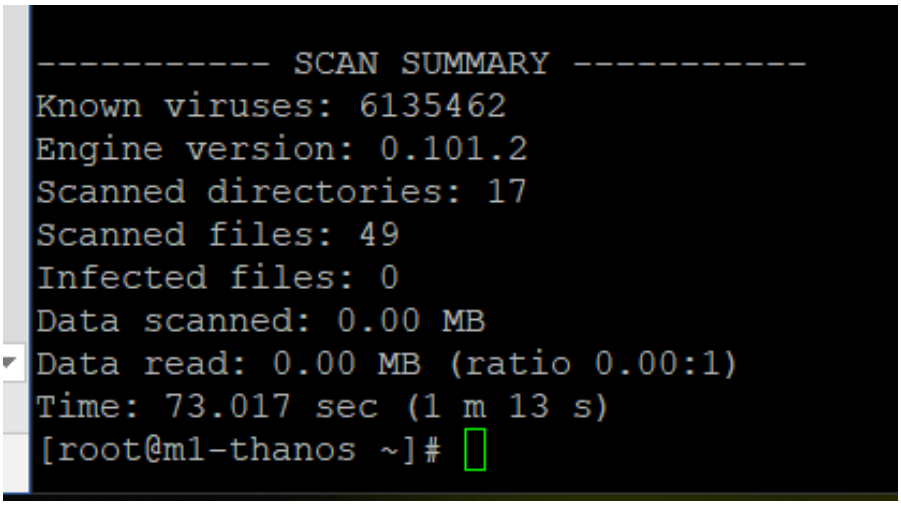
```
sudo sed -i -e "s/^Example/#Example/" /etc/clamd.d/scan.conf
```

Virus database voor ClamAV updaten doen we door volgend commando in te geven.

```
sudo freshclam
```

Wanneer je klaar bent met het updaten van de virusdatabase kan je een testscan uitvoeren in je home directory om te testen of het scannen werkt.

```
sudo clamscan -r /home
```



RKhunter is een gebruikelijke optie om je systeem te scannen op rootkits en algemene kwetsbaarheden. Het kan gemakkelijk geïnstalleerd worden van het package manager op CentOS door gebruik te maken van volgend commando:

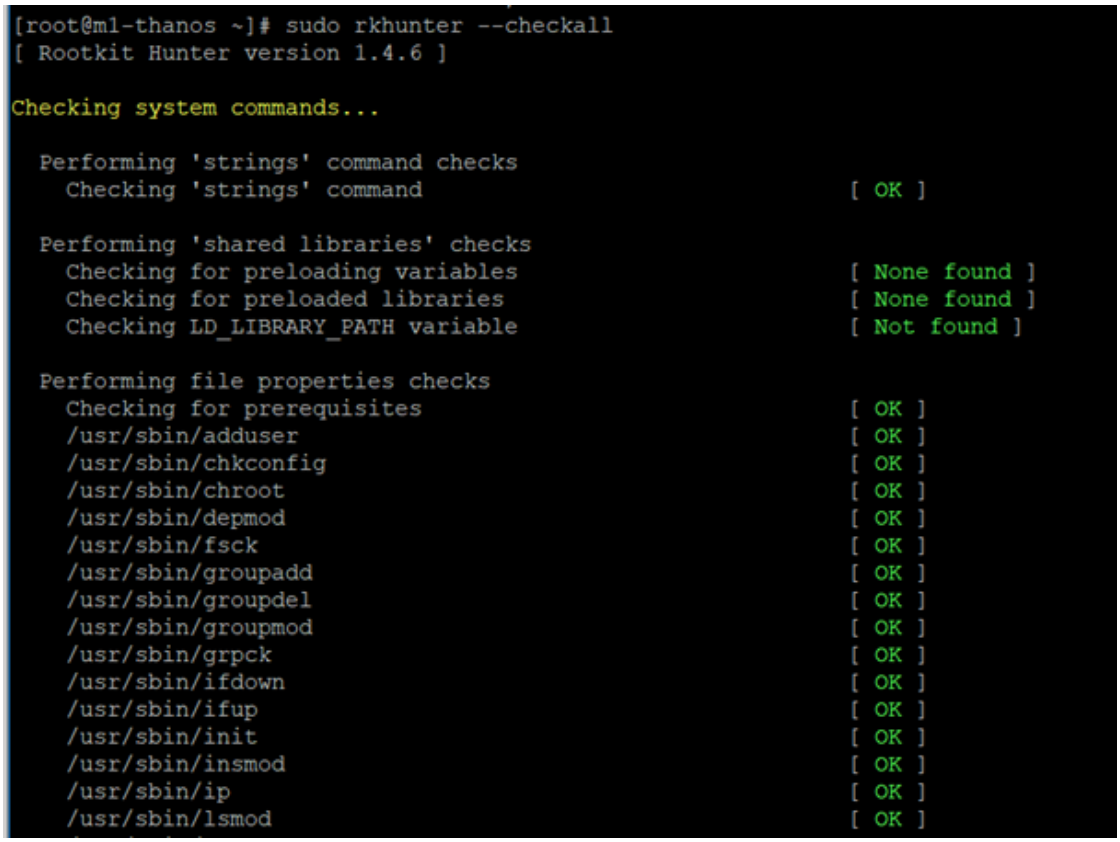
```
sudo yum install rkhunter
```

Update database:

```
sudo rkhunter --propupd
```

Deze commando gaat door system commando’s; zoekt actief voor rootkits en malware, netwerk en local host settings en geeft tot slot ook een summary alsook een recording in de logfiles.

```
sudo rkhunter –checkall
```



Logfiles:

`sudo cat /var/log/rkhunter/rkhunter.log | grep -i warning`

`sudo cat /var/log/rkhunter/rkhunter.log | grep -i warning`

14. BACULA

14.1. Inleiding

Voor de backup gaan we Bacula gebruiken. Bacula is een open-source netwerk die de system administrator de mogelijkheid geeft om backups te maken, data recovery te doen en data over een netwerk te verifiëren. Deze is ook gebaseerd op de Client/Server-model. Aangezien Bacula flexibel is, past het in verschillende netwerken waar een backup-oplossing moet komen. Het is uit te breiden van één enkele computer tot een system bestaande uit honderden computers. Backup is belangrijk om het verlies van data te voorkomen.

14.2. Installatie & configuratie

Installatie en configuratie gebeurt aan de hand van volgende stappen met telkens uitleg.

14.2.1. Configuratie van de bacula server (deel 1)

Stap 1:

Instaleer en start (indien het nog niet gebeurd is), de MySQL service op:

```
sudo yum install mariadb-server
```

```
sudo systemctl start mariadb
```

Om niet telkens de MySQL service zelf op te starten bij boot kunnen we deze automatisch laten starten bij boot, dit gebeurt aan de hand van volgende commando.

```
sudo systemctl enable mariadb
```

Stap 2:

Installeer de Bacula server packages. Dit gebeurt a.d.h.v. het volgende commando:

```
sudo yum install -y bacula-director bacula-storage bacula-
console bacula-client
```

We moeten ook nog 3 poorten openzetten die bacula gaat gebruiken. Deze stap moet op de server en op de client gebeuren. Het gebeurt a.d.h.v. volgende commando's:

```
 sudo firewall-cmd --zone=public --add-port=9101/tcp --permanent
 sudo firewall-cmd --zone=public --add-port=9102/tcp --permanent
sudo firewall-cmd --zone=public --add-port=9103/tcp --permanent
```

```
 sudo firewall-cmd --reload
```

Stap 3:

Creëer de 'Bacula databases and user' tables met behulp van volgende scripts:

```
/usr/libexec/bacula/grant_mysql_privileges
```

```
/usr/libexec/bacula/create_mysql_database -u root
```

```
/usr/libexec/bacula/make_mysql_tables -u root
```

Stap 4:

Log in op de MySQL console als root:

```
mysql -u root
```

Stap 5:

In de MySQL console stel je het wachtwoord voor de Bacula database user in. Alles tot en met ';' moet op 1 lijn ingevoerd worden:

```
UPDATE mysql.user SET
Password=PASSWORD('geef_hier_een_zelfgekozen_ww_in') WHERE
User='bacula';
```

```
FLUSH PRIVILEGES;
```

Verlaat hierna het MySQL prompt met het "exit" commando.

Stap 6:

Alvorens verder te gaan moeten we eerst instellen dat Bacula de MySQL libraries gebruikt. Standaard gebruikt Bacula de PostOgre libraries.

Voer het volgende commando uit:

```
sudo alternatives --config libbaccats.so
```

U zult volgend scherm te zien krijgen:

```
Output
There are 3 programs which provide 'libbaccats.so'.

  Selection    Command
-----------------------------------------------
   1           /usr/lib64/libbaccats-mysql.so
   2           /usr/lib64/libbaccats-sqlite3.so
*+ 3           /usr/lib64/libbaccats-postgresql.so

Enter to keep the current selection[+], or type selection number: 1
```

Geef als invoer "1" in. Druk vervolgens op enter.

De Bacula server (en client) onderdelen zijn nu geïnstalleerd.

Stap 7:

Bacula heeft een Backup directory nodig.

Met behulp van volgend commando maken we en directory aan. Op de server komt map /bacula/backup/ te staan:

```
sudo mkdir -p /bacula/backup/
```

Stap 8:

We veranderen de bestandpermissies zodat enkel Bacula aan deze directories kan.

```
sudo chown -R bacula:bacula /bacula
```

```
sudo chmod -R 700 /bacula
```

Als je een error "Permission denied" krijgt, kan je SELinux op disabled zetten en kijken of het werkt. Als het dan nog niet werkt, verander permissies op de map /bacula naar 777.

Open **/etc/selinux/config** file en verander de SELINUX mod naar disabled:

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
#       targeted - Targeted processes are protected,
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

En reboot de machine!

Stap 9:

Om correct te werken moeten er enkele onderdelen afzonderlijk geconfigureerd worden. **We beginnen met de Bacula Director.**

Open de config file van de Bacula Director met volgend commando

```
sudo vi /etc/bacula/bacula-dir.conf
```

**Om met vi te werken, heb je volgende commando's nodig:*

i = insert (om te beginnen typen)

esc = om uit insert mode weg te gaan

:wq = om veranderingen te bewaren en uit de file weg te gaan

:q = om vi te verlaten als je geen veranderingen ingevoegd hebt

Stap 10:

Zoek naar de Director Resource en voeg onderstaande lijn eraan toe

```
DirAddress=127.0.0.1
```

U zou volgende resultaat moeten bekomen:

```
Director {                            # define myself
 Name = bacula-dir
 DIRport = 9101         # where we listen for UA connections
 QueryFile = "/etc/bacula/query.sql"
 WorkingDirectory = "/var/spool/bacula"
 PidDirectory = "/var/run"
 Maximum Concurrent Jobs = 1
 Password = "@@DIR_PASSWORD@@"          # Console password
 Messages = Daemon
 DirAddress = 127.0.0.1
}
```

Stap 12:

Maak vervolgens een Local Job aan in de Job Resource. Je kan deze resource snel vinden door te zoeken naar de naam "*BackupClient1"*

Verander hier de “Name” in “BackupLocalFiles”.

```
Job {
 Name = "BackupLocalFiles"
 JobDefs = "DefaultJob"
}
```

Zoek hierna naar de Job met naam “RestoreFiles”. Verander de waarde van “Name” naar “RestoreLocalFiles” en waarde van “Where” naar “/bacula/restore”

Dit was de directory die we hebben aangemaakt voor de restore files.

Na de aanpassingen zou u volgende moeten bekomen:

```
Job {
 Name = "RestoreLocalFiles"
 Type = Restore
 Client=bacula-fd
 FileSet="Full Set"
 Storage = File
 Pool = Default
 Messages = Standard
 Where = /bacula/restore
}
```

Stap 13:

Na het aanmaken van de job gaan we nu de File Set configureren. De File Set geeft aan welke files of directories er mee in de backup moeten worden opgenomen of welke niet moeten worden opgenomen.

Zoek naar de FileSet Resource.

Voeg onder “Options” van de FileSet resource de line ‘*compression = GZIP*’ toe.

Verander de waarde van “File” van ‘`/usr/sbin`’ naar ‘`/`’.

Voeg tenslotte de line ‘`File = /bacula`’ toe aan de File Resource.

Als eindresultaat zou u dit moeten hebben:

```
FileSet {
 Name = "Full Set"
 Include {
   Options {
     signature = MD5
     compression = GZIP
   }
File = /
}
 Exclude {
   File = /var/spool/bacula
   File = /proc
   File = /tmp
   File = /.journal
   File = /.fsck
   File = /bacula
 }
}
```

Stap 14:

Ga opzoek naar de Storage Resource.

Verander hierin de waarde van “Address” naar een private FDQN (of een private IP-adres).

U zou het volgende moeten bekomen:

```
Storage {
 Name = File
# Do not use "localhost" here
 Address = backup_server_ip_address
 SDPort = 9103
```

```
 Password = "@@SD_PASSWORD@@"

 Device = FileStorage

 Media Type = File

}
```

Stap 15

Ga opzoek naar de Catalog Resource.

Verander hier de waarde van "dbpassword" naar dezelfde waarde als het wachtwoord dat u hebt ingesteld (zie stap 5)

U zou volgend resultaat moeten bekomen:

```
# Generic catalog service

Catalog {

 Name = MyCatalog

# Uncomment the following line if you want the dbi driver

# dbdriver = "dbi:postgresql"; dbaddress = 127.0.0.1; dbport
=

 dbname = "bacula"; dbuser = "bacula"; dbpassword =
"geef_hier_een_zelfgekozen_ww_in"

}
```

Stap 16:

Ga opzoek naar de Pool Resource.

Voeg de line "Label Format = Local-"

U zou volgend resultaat moeten bekomen:

```
# File Pool definition

Pool {

 Name = File

 Pool Type = Backup

 Label Format = Local-

 Recycle = yes                        # Bacula can
automatically recycle Volumes
```

```
 AutoPrune = yes                     # Prune expired volumes

 Volume Retention = 365 days         # one year

 Maximum Volume Bytes = 50G          # Limit Volume size to
something reasonable

 Maximum Volumes = 100               # Limit number of
Volumes in Pool

}
```

Sla vervolgens op en exit de text editor.

Stap 17:

Om na te gaan of de Director configuratie geen syntaxfouten bevat, kunt u volgend commando uitvoeren:

sudo bacula-dir -tc /etc/bacula/bacula-dir.conf

Indien u geen 'errors' te zien krijgt, zijn er geen syntaxfouten aanwezig.

Stap 18:

Na de Director configuratie afgerond te hebben, gaan we aan de slag met de Storage Daemon.

Open de SD-configuratie file:

sudo vi /etc/bacula/bacula-sd.conf

Stap 19:

Ga op zoek naar Storage Resource

Voeg de "SDAddress" parameter toe en geef als waarde de private FDQN (of private IP-adres) van de back-up server.

U zou het volgende resultaat moeten bekomen:

```
Storage {                             # definition of myself

 Name = bacula-sd

 SDPort = 9103                  # Director's port

 WorkingDirectory = "/var/spool/bacula"

 Pid Directory = "/var/run"

 Maximum Concurrent Jobs = 20
```

```
 SDAddress = backup_server_private_ip_address

}
```

Stap 20:

Ga op zoek naar de Device Resource met als waarde voor "Name = FileStorage".

Verander de waarde van "Archive Device" naar *"/bacula/backup/".*

U zout het volgende resultaat moeten bekomen:

```
Device {

 Name = FileStorage

 Media Type = File

 Archive Device = /bacula/backup/

 LabelMedia = yes;                   # lets Bacula label
unlabeled media

 Random Access = Yes;

 AutomaticMount = yes;               # when device opened,
read it

 RemovableMedia = no;

 AlwaysOpen = yes;

}
```

Sla vervolgens op en exit de text editor.

Stap 21:

Om na te gaan of de Service Daemon configuratie geen syntaxfouten bevat kunt u volgend commando uitvoeren:

```
sudo bacula-sd -tc /etc/bacula/bacula-sd.conf
```

Indien u geen errors te zien krijgt, zijn er geen syntax fouten aanwezig.

We zijn nu klaar om de Bacula componenten te restarten.

Stap 22:

Elk Bacula component heeft een eigen wachtwoord dat gebruikt wordt voor authenticatie tussen die componenten. We kunnen deze wachtwoorden simpel laten genereren en configureren aan de hand van volgende commando's, of zelf een wachtwoord kiezen, want deze wachtwoord wordt enkel gebruikt voor interne communicatie tussen daemons om backup- en restore-files naar elkaar door te geven.

Het beste is ook dat je overal hetzelfde wachtwoord gebruikt zodat je geen "permission denied"-error krijgt. Je kunt ook hier kijken of alle names en passwords matchen: https://www.bacula.org/7.4.x-manuals/en/problems/Bacula_Frequently_Asked_Que.html#SECTION00260000000000000000

Console Director

```
DIR_PASSWORD=`date +%s | sha256sum | base64 | head -c 33`

sudo sed -i "s/@@DIR_PASSWORD@@/${DIR_PASSWORD}/"
/etc/bacula/bacula-dir.conf

sudo sed -i "s/@@DIR_PASSWORD@@/${DIR_PASSWORD}/"
/etc/bacula/bconsole.conf
```

SD Director

```
SD_PASSWORD=`date +%s | sha256sum | base64 | head -c 33`

sudo sed -i "s/@@SD_PASSWORD@@/${SD_PASSWORD}/" /etc/bacula/bacula-
sd.conf

sudo sed -i "s/@@SD_PASSWORD@@/${SD_PASSWORD}/" /etc/bacula/bacula-
dir.conf
```

FD Director

```
FD_PASSWORD=`date +%s | sha256sum | base64 | head -c 33`

sudo sed -i "s/@@FD_PASSWORD@@/${FD_PASSWORD}/" /etc/bacula/bacula-
dir.conf

sudo sed -i "s/@@FD_PASSWORD@@/${FD_PASSWORD}/" /etc/bacula/bacula-
fd.conf
```

Nu de wachtwoorden zijn ingesteld zijn we klaar om de componenten op te starten.

Stap 23:

Start de Bacula componenten op met volgende commando's

```
sudo systemctl start bacula-dir

sudo systemctl start bacula-sd

sudo systemctl start bacula-fd
```

Na de aanpassingen in de configuratiefiles moet je ook de componenten herstarten. Dat doet je a.d.h.v. de volgende commando's:

```
sudo systemctl restart bacula-dir

sudo systemctl restart bacula-sd

sudo systemctl restart bacula-fd
```

Om niet telkens de componenten op te moeten zetten bij boot kunnen we deze automatisch laten starten bij boot. Dit gebeurt aan de hand van volgende commando's.

```
sudo systemctl enable bacula-dir

sudo systemctl enable bacula-sd

sudo systemctl enable bacula-fd
```

We hebben nu een werken systeem maar nog geen clients dat zal in de volgende stappen aan bod komen.

Stap 24:

Om te voorkomen dat we de config files nog ingewikkelder gaan maken, maken we een nieuwe directory aan waarin we config files gaan aanmaken.

```
sudo mkdir /etc/bacula/conf.d
```

Stap 25:

Nu moeten we nog voor zorgen dat deze aparte in de main dir config worden geladen, dit doen we door het volgende stuk code op het einde van de config file te plaatsen.

```
sudo vi /etc/bacula/bacula-dir.conf
```

Voeg dit toe helemaal onderaan de file (let op de spaties):

```
@|"find /etc/bacula/conf.d -name '*.conf' -type f -exec echo @{} \;"
```

Stap 26:

Vervolgens gaat u nog een pool aanmaken om de remote back-up jobs te configureren.

```
sudo vi /etc/bacula/conf.d/pools.conf
```

Hierin plakt u het volgende in en slaag het vervolgens op:

```
Pool {
```

```
 Name = RemoteFile

 Pool Type = Backup

 Label Format = Remote-

 Recycle = yes                        # Bacula can automatically
recycle Volumes

 AutoPrune = yes                      # Prune expired volumes

 Volume Retention = 365 days          # one year

   Maximum Volume Bytes = 50G          # Limit Volume size to
something reasonable

 Maximum Volumes = 100                # Limit number of Volumes
in Pool

}
```

Stap 27:

Best kunt u nu eens controleren of de configuratie file errores heeft:

```
sudo bacula-dir -tc /etc/bacula/bacula-dir.conf
```

14.2.2. Configuratie bacula client

Stap 28:

Nu gaat u over naar het configureren van uw clienthost, we beginnen met Bacula te installeren:

```
sudo yum install bacula-client
```

Voordat we verder gaan met configureren adviseren wij u om volgende gegevens ergens op te schrijven omdat u deze zal nodig hebben in de volgende stappen:

- Client hostname : bv. "ClientHost"
- Client Private FQDN: bv. “likeclienthost.private.example.com” u kan ook gewoon het ip gebruiken (vb 172.27.66.137)
- Bacula Server hostname

Stap 29:

In de volgende stappen gaat u een wachtwoord nodig hebben om de communicatie tussen de director en file daemon tot stand te brengen. Je kan een random wachtwoord genereren met het volgende commando, vergeet het niet op te schrijven:

```
date +%s | sha256sum | base64 | head -c 33 ; echo
```

Stap 30:

Vervolgens gaat u enkele aanpassingen doen in de config files van de file daemon:

```
sudo vi /etc/bacula/bacula-fd.conf
```

Verander de director naam en paswoord:

```
Director {
 Name = bacula-dir
 Password = "Voer hier uw wachtwoord in"
}
```

Stap 31:

Vervolgens moet u enkele parameters aanpassen aan de file daemon resource in dezelfde file:

```
FileDaemon {                          # this is me
 Name = ClientHost-fd
 FDAddress = client_private_ip
 FDport = 9102                  # where we listen for the director
 WorkingDirectory = /var/spool/bacula
 Pid Directory = /var/run
 Maximum Concurrent Jobs = 20
}
```

Stap 32:

Ook moet u in dezelfde file instellen dat de daemon de log messages naar de backup server stuurt:

```
Messages {
 Name = Standard
 director =  bacula-dir = all, !skipped, !restored
}
```

Stap 33:

Sla de file op en controleer vervolgens op errors:

```
sudo bacula-fd -tc /etc/bacula/bacula-fd.conf
```

Stap 34:

Als er geen foutmeldingen zijn kan u de filedaemon herstarten en instellen dat deze opstart bij het opstarten van de host.

```
sudo systemctl restart bacula-fd
```

```
sudo systemctl enable bacula-fd
```

Stap 35:

In deze stap gaat u een directory moeten aanmaken waar de baculaserver de restore files + permissies heen plaatst + (kan zijn dat je 777 permissies moet geven), deze map staat dus op de client.

```
sudo mkdir -p /bacula/restore
```

```
sudo chown -R bacula:bacula /bacula
```

```
sudo chmod -R 700 /bacula
```

14.2.3. Configuratie bacula server (deel 2)

Stap 36:

Nu gaat u FileSets creëren voor welke files u wilt backuppen met de gewenste backup jobs, u doet dit in het volgende bestand:

```
sudo vi /etc/bacula/conf.d/filesets.conf
```

Plak hier volgende tekst is en pas de waardes aan.

```
FileSet {
 Name = "Home and Etc"
 Include {
   Options {
     signature = MD5
     compression = GZIP
```

```
   }
#hier vermeldt je mappen die je op de client wilt backuppen
   File = /home
   File = /share/upload
 }
 Exclude {
   File = /home/bacula/not_important
 }
}
```

Stap 37:

Nu bent u klaar om uw clients aan de bacula server toe te voegen, daarvoor moet je de bacula director configureren met de nieuwe client en job resources in het volgende bestand:

```
sudo vi /etc/bacula/conf.d/clients.conf
```

Hier plakt u het volgende in en verander de waardes:

```
Client {
 Name = ClientHost-fd
 Address = client_private_ip_address
 FDPort = 9102
 Catalog = MyCatalog
 Password = "voer hier uw wachtwoord in"          # password for
Remote FileDaemon
 File Retention = 30 days            # 30 days
 Job Retention = 6 months            # six months
 AutoPrune = yes                     # Prune expired Jobs/Files
}
```

Stap 38:

Nu kunt u back-up jobs gaan aanmaken met een uniek naam en de gewenste details van de client en data dat moet worden geback-upt, dit doet u in hetzelfde bestand.

```
Job {
 Name = "BackupClientHost"
 JobDefs = "DefaultJob"
 Client = ClientHost-fd
 Pool = RemoteFile
 FileSet="Home and Etc"
}
```

Nu u een job hebt aangemaakt, kunt u deze opslaan en controleren op errors.

```
sudo bacula-dir -tc /etc/bacula/bacula-dir.conf
```

Stap 39:

Herstart de director en test de client connectie.

```
sudo systemctl restart bacula-dir
```

```
sudo bconsole (na * gewoon "status client" ingeven of "exit"/"q" om weg te gaan)
```

```
*status client
```

Als alles goed gaat zou u dit terug moeten krijgen, indien niet controleer de error door naar messages te gaan.

```
Select Client resource: ClientHost-fd

The defined Client resources are:

    1: bacula-fd

    2: ClientHost-fd

Select Client (File daemon) resource (1-2): 2
```

Om foutmeldingen te bekijken:

```
*Messages
```

Stap 40:

Nu kunt u de back-up jobs testen door volgende commando's uit te voeren:

```
*Run
```

U zou daarna dit moeten krijgen:

```
Select Job resource: BackupClientHost

The defined Job resources are:

     1: BackupLocalFiles

     2: BackupCatalog

     3: RestoreLocalFiles

     4: BackupClientHost

Select Job resource (1-4): 4
```

Kies voor de pas aangemaakte job

```
Select Job resource (1-4): 4
```

Vervolgens kiest u voor yes

```
Confirmation prompt:

OK to run? (yes/mod/no): yes
```

Hierna kunt u kijken hoe het back-uppen verlopen is bij de messages (syslogs), na * in te geven:

```
Messages
```

Je kunt ook kijken of de job effectief gedaan is door de "list jobs" commando in te geven:

```
*list jobs
```

Volgende output laat je JobStatus zien (T= Terminated, F=Fatal error, R-running)

```
Enter a period to cancel a command.
*list jobs
Automatically selected Catalog: MyCatalog
Using Catalog "MyCatalog"
+-------+------------------+---------------------+------+-------+----------+-------------+-----------+
| JobId | Name             | StartTime           | Type | Level | JobFiles | JobBytes    | JobStatus |
+-------+------------------+---------------------+------+-------+----------+-------------+-----------+
|    46 | BackupClientHost | 2019-05-23 09:41:46 | B    | F     |      209 |   7,760,514 | T         |
```

Stap 41:

Nu kan u ook de restorejobs testen (nog altijd op de server):

```
restore all
```

Kies vervolgens voor 5 "Select the most recent backup"

Kies dan welke client er moet worden gerestored (2 ClientHost-fd), hierna type je "done" in de console en de client zal gerestored worden. U kunt de "messages" of "list jobs" controleren voor de status van de job.

U hebt nu een werkend backup systeem.

15. OPS REPORT CARDS

15.1. Pager Rotation Schedule

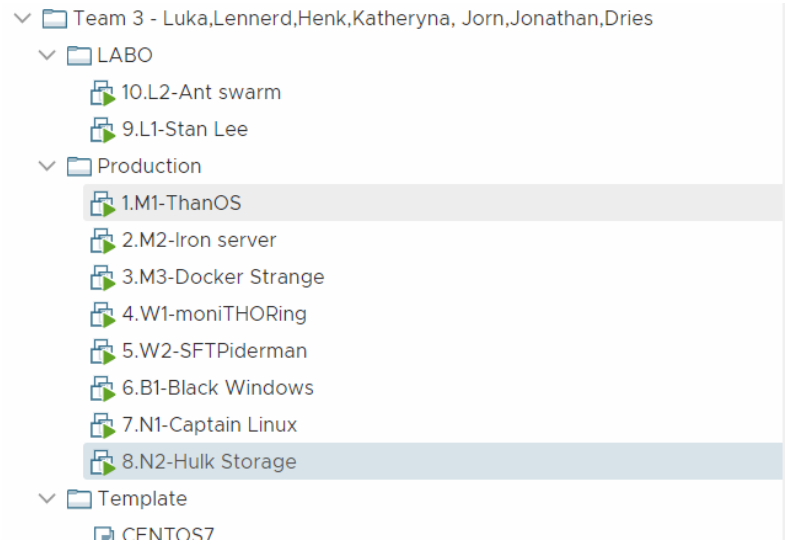
Eén van de report cards bestond uit het opmaken van een rotation schedule. Hier kan je zien wie verantwoordelijk is voor ons systeem doorheen de week, met telkens de nodige contactgegevens.

Pager rotation schedule

Achternaam	Voornaam	Functie	Periode	Nummer privé	Nummer werk	E-mailadres
Janssen	Dries	beheerder	25/05/2019	471589654	014/45/15/89	dries@gmail.com
Mekhova	Kateryna	beheerder	26/05/2019	478512698	014/45/15/90	kateryna@gmail.com
Kolb	Luka	beheerder	27/05/2019	471415498	014/45/15/91	luka@gmail.com
Kitooto	Jonathan	beheerder	28/05/2019	477856698	014/45/15/92	jonathan@gmail.com
Hinrichs	Henk-Sjoerd	beheerder	29/05/2019	497785685	014/45/15/93	sjoerd@gmail.com
Peeters	Jorn	beheerder	30/05/2019	447895651	014/45/15/94	jorn@gmail.com
Geentjens	Lennerd	beheerder	31/05/2019	454698512	014/45/15/95	lennerd@gmail.com

15.2. Canary Roll out

Voor er een rollout gedaan wordt van een nieuwe functionaliteit, zal deze in het labo eerst geconfigureerd en getest worden.



15.3. Do automatic tasks run under a rol account

Automatic tasks runnen onder een sys-* account kort voor system.

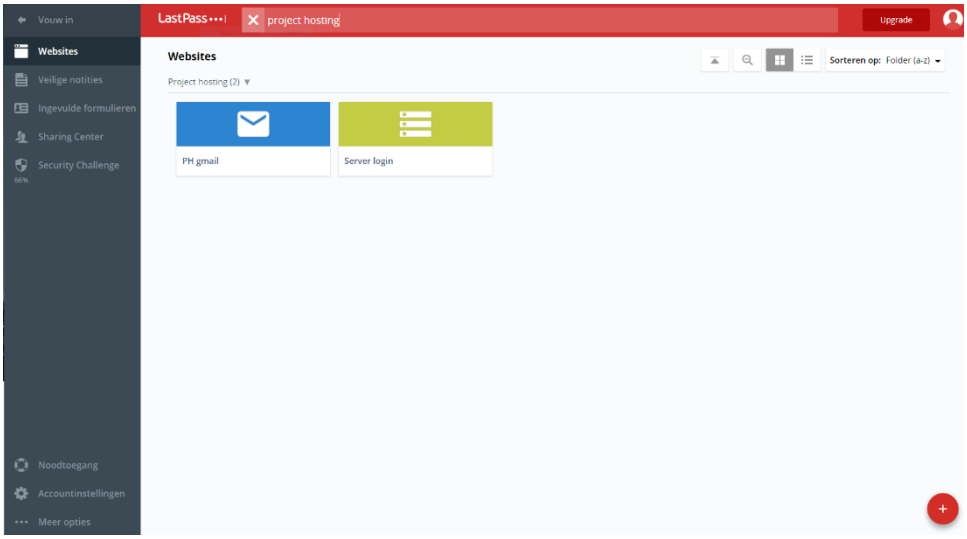
15.4. Seperate QA, production and development servers

We hebben een aparte ontwikkelingsomgeving omgezet om onze producten te testen. Dit zijn twee extra VM's waar CentOS op is geïnstalleerd. Deze twee VM's zijn identiek aan de productie-omgeving. Het voordeel van deze omgeving is het voorkomen van kritieke fouten die zich kunnen voordoen als we aan het testen zijn.

VM 9	QA & development Docker manager2	172.27.66.138	L1-Stan Lee
VM10	QA & development Docker worker2		L2-Ant Swarm

15.5. Password safe

Als password safe maken wij gebruik van lastpass. Elke administrator heeft zijn eigen kluis waarin hij zijn wachtwoorden kan zien en beheren.



15.6. Patch software accross the fleet automatically

Dockerswarm is een clustering en scheduling tool voor docker containers. Je hebt hierin Swarm Mode, een laag tussen de OS en de container images. Clustering zorgt ervoor dat de groepen coöperatief zijn en redundancy voorzien. Als één of meerdere nodes uitvallen, nemen de andere nodes hun taken over. Swarm zorgt zelf voor updates van de laatste versie.

15.7. Does each service have appropriate monitoring

Swarmprom is onze monitoring software die de data van verschillende services binnenhaalt. Deze software is te bewerken naar de wensen van de gebruiker. Zo kunnen er nog verschillende plugins toegevoegd worden en kan het monitor-verhaal helemaal custom made worden gemaakt.

15.8. Are your backups automated?

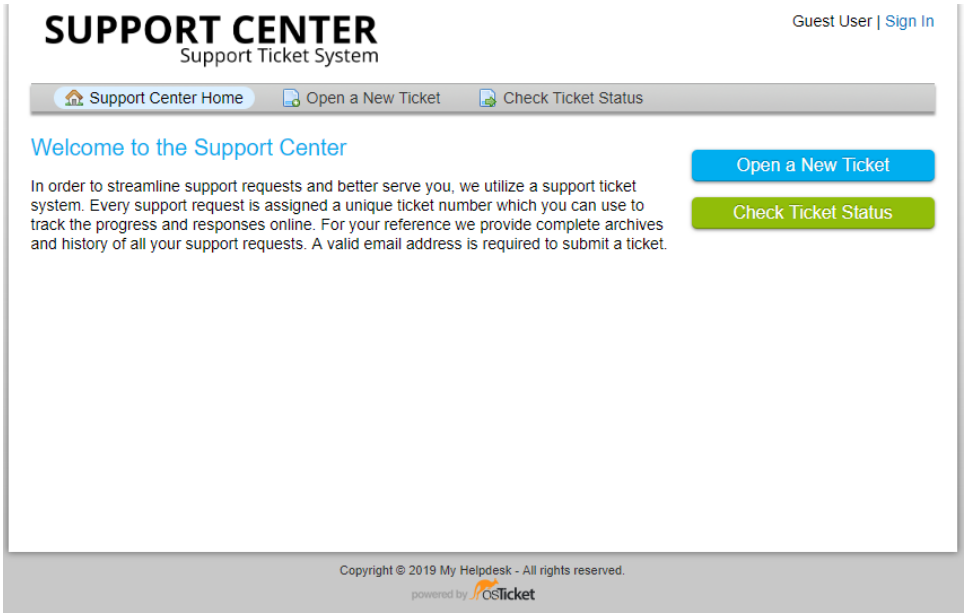
De full backup van de files/mappen gebeurt op de eerste zondag van de maand. De incremental backup van de files/mappen (wijzigingen die gebeurd zijn sinds de full backup), gebeurt elke andere zondag om 23:05. De backup van de catalogus, informatie die intern gebruikt wordt, gebeurt elke zondag en zaterdag om 23:10.

15.9. Do Desktops/laptops/servers run-self-updating, silent, anti-malware software?

Clam AntiVirus (ClamAV) is een opensource-antivirusprogramma die met name trojans, virussen en malware detecteert. Deze hebben wij geïnstalleerd om alle VM's in onze architectuur te beschermen. Daarnaast hebben we een RKhunter geinstalleerd op de servers. Deze scant op rootkits en algemene beveiligingslekken.

15.10. User requests tracking ticket system

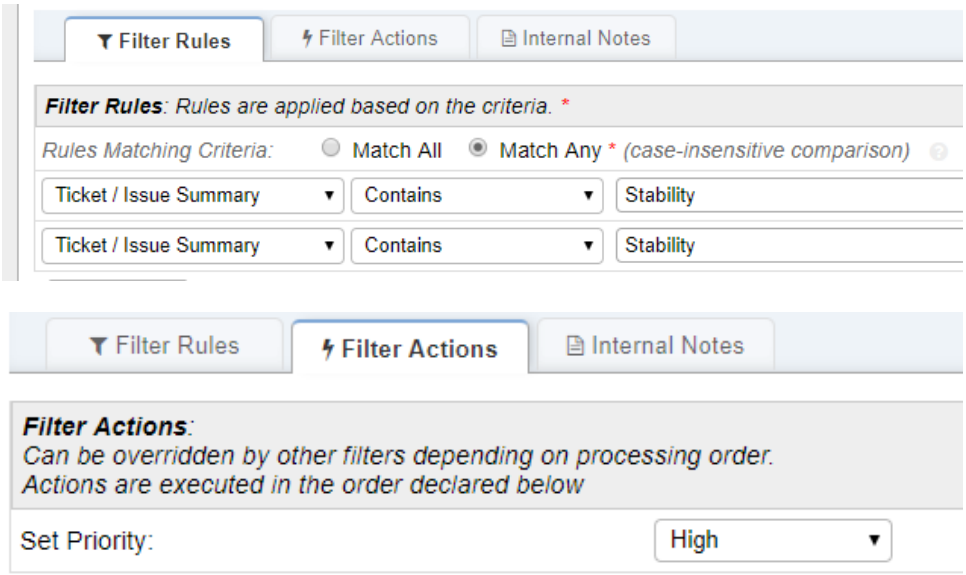
Osticket is een ticketing systeem van Enhancesoft. Met deze applicatie kunnen gebruikers tickets aanmaken voor de support staf.



Osticket heeft een eigen admininstrator panel waar support staf aangemaakte tickets kan bekijken en hierna oplossen.

15.11. In your bugs/tickets, does stability have a higher priority than new features?

In Osticket is het mogelijk om te filteren in de tickets op bijv. ‘Stability’ en de priority hiervan aan te passen naar high.



15.12. Team’s monthly metrics

Wij verzamelen metrics over hoeveel tickets er worden aangemaakt om te meten hoe gemakkelijk gebruikers met het systeem overweg kunnen. Verder meten we de uptime van onze services met Swarmprom.

15.13. Does your team write “design docs"?

Dit document genaamd ‘technische documentatie’ is onze design docs. Hierin staat waarom we bepaalde keuzes hebben gemaakt, wat de keuzes zijn geweest en hoe we deze keuzes hebben geïmplementeerd in de praktijk.

16. TROUBLESHOOTING

16.1. Inleiding

Wanneer dingen verkeerd gaan gedurende het opstellen van het hosting platform, is het altijd handig om een leidraad te hebben. Deze documentatie helpt je vaak voorkomende problemen te troubleshooten.

17. KEEPALIVED

17.1. Inleiding

Keepalived is een routing software met als hoofddoel te loadbalancen en voor high-availibility in ons infrastructuur te zorgen. Voorbeeld: 2 van onze VM's hebben 2 IP's gekregen. Wanneer je met het virtuele .139 IP gaat connecteren verbind deze je door naar 1 van deze 2 VM's afhankelijk van de running status en de priority.

Wegens een bug hebben we deze implementatie niet 100% kunnen laten werken.

17.2. Configuratie manier 1

Eerst doen we de installatie van 'keepalived' met het volgende commando:

```
yum install -y keepalived
```

Daarna gaan we dit configureren, om in de file te gaan doen we:

```
sudo vi /etc/keepalived/keepalived.conf
```

In de file zien we nu een heel scriptje, er zijn 4 dingen waar we op moeten letten. Je gaat dit met 2 servers moeten doen, 1x een master en 1x een backup.

```
vrrp_script chk_httpd {
        script "pidof httpd"
        interval 2
}
vrrp_instance VI_1 {
        # The interface keepalived will manage
        interface eth0 1
        state BACKUP 2
        # How often to send out VRRP advertisements
        advert_int 2
        # The virtual router id number to assign the routers to
        virtual_router_id 51
        # The priority to assign to this device. This controls
        # who will become the MASTER and BACKUP for a given
        # VRRP instance (a lower number get's less priority).
        priority 50 3
        authentication {
                auth_type PASS
                auth_pass SimplePassword
        }
        unicast_src_ip primary_ip_of_web2
        unicast_peer {
                primary_ip_of_web1
        }
        track_script {
                chk_httpd
        }
        # The virtual IP addresses to float between nodes.
        virtual_ipaddress {
                failover_ipv4_address 4
        }
}
```

1: Dit kijkt op welk netwerk er geconfigureerd moet worden. Dit kan je zien door ‘ip a’ te typen en te kijken op welke interface er connectie is met buitenaf, bij ons is dit de ‘ens192’.

2: Je moet weten welke server ja als backup wilt en welke als master. Indien je een master wilt schrijf je ‘state MASTER, als je een backup wilt kan je dit doen aan de hand van het voorbeeld hierboven ‘state BACKUP’.

3: De priority dat het krijgt. Het maakt niet uit welk nummertje het krijgt, je moet enkel letten dat de master een hogere prioriteit heeft dan een backup!

4: Het ip adres dat je wilt gebruiken als virtueel ip.

Je kan nu uit de configuratie file gaan (escape en dan :wq) en ‘keepalived enable’ starten, dit doe je door deze commando’s in te voeren:

```
# systemctl enable keepalived
```

```
# systemctl start keepalived
```

Als je op beide servers wilt bekijken of dit gelukt is, voer het commando ‘ip a’ uit en kijk naar je ip adres. Je ziet nu dat er 2 ip addressen zijn (je gewone en uw virtual ip).

17.3. Configuratie manier 2

```
docker run -d --name keepalived --restart=always \

  --cap-add=NET_ADMIN --net=host \

  -e KEEPALIVED_INTERFACE=NAAM INTERFACE \

  -e KEEPALIVED_UNICAST_PEERS="#PYTHON2BASH:['192.168.56.112',
'192.168.56.111']" \

  -e KEEPALIVED_VIRTUAL_IPS=192.168.56.115 \

  -e KEEPALIVED_PRIORITY=200 \

  osixia/keepalived:1.3.5


 docker run -d --name keepalived --restart=always \

  --cap-add=NET_ADMIN --net=host \

  -e KEEPALIVED_INTERFACE=NAAM INTERFACE \

  -e KEEPALIVED_UNICAST_PEERS="#PYTHON2BASH:['192.168.56.112',
'192.168.56.111']" \

  -e KEEPALIVED_VIRTUAL_IPS=192.168.56.115 \

  -e KEEPALIVED_PRIORITY=100 \

  osixia/keepalived:1.3.5
```

BRONNEN

https://www.serverwatch.com/server-tutorials/slideshows/11-load-balancers-you-need-to-know-in-2016.html

https://www.itzgeek.com/how-tos/linux/centos-how-tos/how-to-install-puppet-4-x-on-centos-7-rhel-7.html

https://www.theforeman.org/manuals/1.20/quickstart_guide.html#QuickstartGuide

https://www.bacula.org/7.4.x-manuals/en/problems/Bacula_Frequently_Asked_Que.html

https://linuxize.com/post/how-to-disable-selinux-on-centos-7/

http://ask.xmodulo.com/open-port-firewall-centos-rhel.html

https://docs.docker.com/config/thirdparty/dsc/

https://www.sumologic.com/blog/using-sumo/kubernetes-vs-docker/

https://docs.microsoft.com/en-us/powershell/dsc/quickstarts/website-quickstart

https://docs.microsoft.com/en-us/powershell/dsc/tutorials/bootstrapdsc

https://docs.microsoft.com/en-us/powershell/dsc/tutorials/dscautomationhostenabled

https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-7/gg318049(v=ws.10)

https://www.linuxjournal.com/content/managing-linux-using-puppet

https://www.digitalocean.com/community/tutorials/how-to-set-up-a-chef-12-configuration-management-system-on-ubuntu-14-04-servers

https://puppet.com/blog/deploying-docker-swarm-puppet

https://forge.puppet.com/puppetlabs/bacula

https://www.softwareadvice.com/crm/it-ticketing-comparison/

https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/

https://www.ibm.com/blogs/bluemix/2018/10/docker-swarm-vs-kubernetes-a-comparison/

https://www.acc-ict.com/kubernetes-vs-docker-swarm-de-voor-en-nadelen-uitgelicht/

https://www.digitalocean.com/community/tutorials/how-to-install-bacula-server-on-centos-7

https://www.youtube.com/watch?v=Y7b8U6_eLDI

https://training.play-with-docker.com/swarm-service-discovery/

https://medium.com/@tiangolo/docker-swarm-with-swarmprom-for-real-time-monitoring-and-alerts-282da7890698

https://phoenixnap.com/kb/how-to-set-or-change-a-hostname-in-centos-7?fbclid=IwAR1THQqXpHUcjwVO-oKBVApQG2qwZdcXRx6NI_PueYfeofM0ISMbAJdjorY

https://www.digitalocean.com/community/tutorials/how-to-configure-ntp-for-use-in-the-ntp-pool-project-on-centos-7

https://www.digitalocean.com/community/tutorials/how-to-configure-the-linux-firewall-for-docker-swarm-on-centos-7#method-1-%E2%80%94-open-docker-swarm-ports-using-firewalld