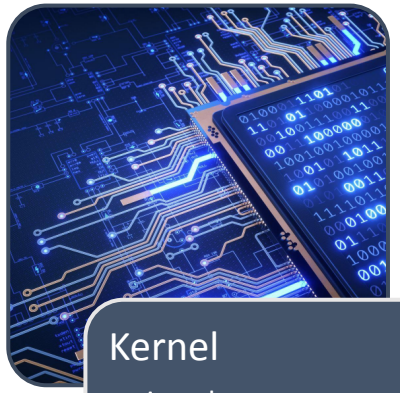
Systems Advanced II
Linux

NFS & Clustering

DE HOGESCHOOL MET HET NETWERK

Elfde-Liniestraat 24, 3500 Hasselt, www.pxl.be

Systems Advanced II - Linux



Kernel

- virtual memory
- user space and kernel space
- processes and threads
- sockets & pipes
- systemd
- Docker

Networking

- routing
- firewall
- eBPF
- certificates
- ssh
- latency
- bandwidth

Services

- nfs
- load balancers and reverse proxies
- databases
- logging

Clusters

- Storage
- Kubernetes

Licensing & open-source software

Systems Advanced II - Linux

kernel

- virtual memory
- user space and kernel space
- processes and threads
- sockets & pipes
- systemd
- Docker

Networking

- routing
- firewall
- eBPF
- certificates
- ssh
- latency
- bandwidth

Services

- nfs
- load balancers and reverse proxies
- databases
- logging

Clusters

- Storage
- Kubernetes

Systems Advanced II - Linux

kernel

- virtual memory
- user space and kernel space
- processes and threads
- sockets & pipes
- systemd
- Docker

Networking

- routing
- firewall
- eBPF
- certificates
- ssh
- latency
- bandwidth

Services

- nfs
- load balancers and reverse proxies
- databases
- logging

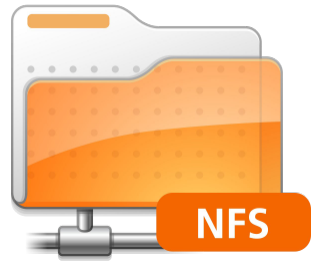
Clusters

- Storage
- Kubernetes

Overzicht

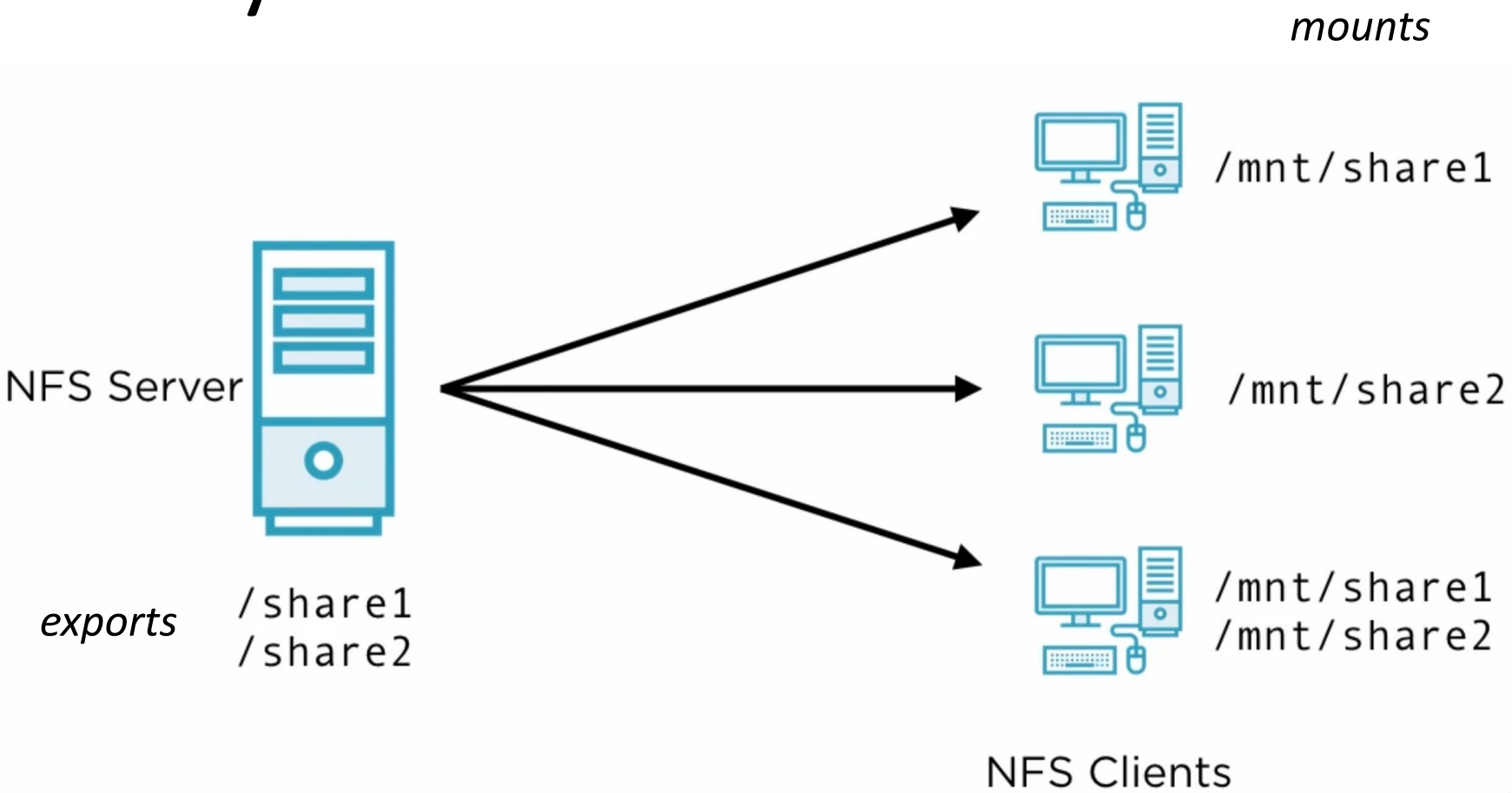
- Introductie
- Architectuur
- Client en Server configuration
- LAB nfs
- File Security
- LAB nfs
- LAB High Availability NFS (gluster)
- LAB Kerberos

Network File System (NFS)



- Network File System (NFS)
- Protocol voor een gedistribueerd file systeem
- 1984 door Sun Microsystems (Sun) ontwikkeld
- Client computer kan via een netwerk toegang krijgen tot remote file systemen (exports) via een local mount point.
- Internet standaard
  RFC 7530 - Network File System (NFS) Version 4 Protocol (ietf.org)
- ook ingebouwd in Windows Server

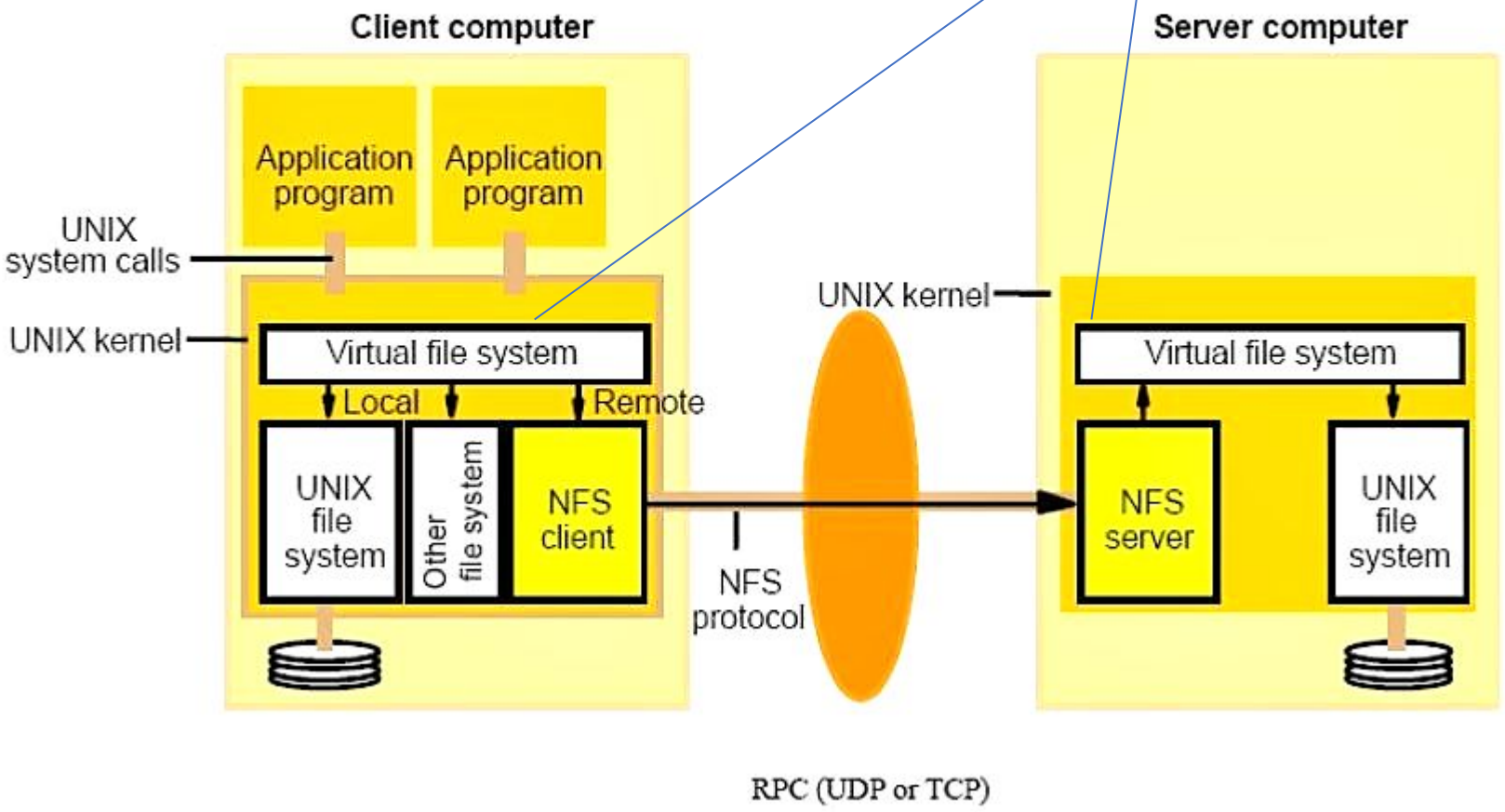
NFS system


mounts
/mnt/share1
/mnt/share2
/mnt/share1
/mnt/share2
NFS Server
exports
/share1
/share2
NFS Clients

NFS architecture

linux virtual file system (vfs)

- software layer in the kernel that provides the **filesystem interface** to user space programs.
- also provides an **abstraction layer** within the kernel which allows different filesystem implementations to coexist.



NFS Networking

- NFSv3
  - UDP / dynamic ports
  - sneller - interessant in low-latency, super-reliable netwerken
  - makkelijker om op te zetten
  - security issues
- NFSv4
  - TCP/2049
  - more reliable
  - meer security features zoals Kerberos support
  - file locking support
  - complex

NFS Server Components

- **NFS Version 3:**
  - **`nfs-server` (`nfsd`)**: Provides basic file storage services
  - **`rpcbind`**: Manages RPC port reservations and connections.
  - Service discovery: Utilizes `rpcbind` for discovering NFS services.
- **NFS Version 4:**
  - **`nfs-server` (`nfsd`)**: Provides file storage services, with enhancements like statefulness.
  - TCP based service: Exclusively uses TCP/IP for communication.
  - Service discovery: Incorporates more advanced mechanisms integrated into the protocol.
  - **`rpcbind`** can run for compatibility purposes, but is not required

NFS Server configuration

```
/nfs-share      192.168.20.0/24(rw,no_root_squash)
```

- **exports**
  - shared directory resources
  - beschreven in **`/etc/exports`**:
  - `/export  <host>(options)`
    - bv. `/share1 server1.pxl.local`
    - **`/export`** - directory die geshared wordt
    - **`<host>`** - host of network die access hebben tot deze export
    - **`(options)`** - opties voor die host/network
- **`exportfs`** - commando om runtime config tabel van exported file systems te maintainen
  - `/var/lib/nfs/etab` - runtime configuratie van de exports

NFS /etc/exports - <host>

- /<export> <host>(options)

```
/nfs-share      192.168.20.0/24(rw,no_root_squash)
```

  - enkele machine
    - fully qualified domain name (FQDN)
    - hostname
    - ip address
  - IP networks
    - CIDR notation (Classless Inter-Domain Routing met prefix voor netwerk bits)
      - bv. 192.168.0.0/28 (==192.168.0.0 tot en met 192.168.0.15)
  - wildcards
    - * of ?
      - *.example.com, *.*.example.com
      - server?.pxl.local
    - [a-z]

NFS /etc/exports - (options)

```
/nfs-share      192.168.20.0/24(rw,no_root_squash)
```

- /<export> <host>(options)
  - Default options
    - **ro** - read only
    - **sync** - reply pas wanneer writes naar disk commited zijn
    - **wdelay** - houdt writes tegen wanneer er wschlk. nog writes gaan volgen
    - **rootsquash** - uid 0 wordt gemapt naar anonymous user om root access to voorkomen
    - **sec=sys** - security model gebruikt mapping van user ids tussen hosts (gevaarlijk!)
  - Andere options
    - **rw/ro** - read-write vs read-only
    - **sync/async** - safety vs performance
    - **all_squash** - alle user ids worden gemapt naar anonymous users
    - **no_root_squash** - geen enkele user wordt gemapt op anonymous user of group id
    - **sec=krb5** (kerberos 5), **sec=krb5i** (integrity checking) of **sec=krb5p** (integrity, encryption)

NFS Client: mount methods

- runtime mounting (weg na reboot)
  - **`mount -t nfs -o rw server0.pxl.local:/share1 /mnt/share1`**
- persistent mounting
  - **`/etc/fstab`**
    - `server0.pxl.local:/share1 /mnt/share1 nfs rw 0 0`
- dynamic on-demand mounting
  - autofs
- `/etc/nfsmount.conf`
  - globale opties voor NFS mounts

Default NFS client options

- **`rw`** - read write
- **`sync`** - wachten op write confirmations
- **`suid`** - gebruik setuid programma's
  - allow users to run an executable file with the file system permissions of the executable's owner or group respectively
  - causes new files and subdirectories created within a directory to inherit its group ID, rather than the primary group ID of the user who created the file (the owner ID is never affected, only the group ID)
- **`dev`** - mount op filesysteem als block of char device
- **`exec`** - uitvoeren van binaries toegestaan
- **`auto`** - mountable met mount -a
- **`hard`** - blocking waits wanneer server offline is
- **`sec=sys`** - gebruik UID/GID security model
- **`realtime`** - update inode access times

zie `man nfs`

veel gebruikte NFS client options

- **`ro`** - read only
- **`async`** - niet wachten op write confirmations
- **`nosuid`** - gebruik setuid applicaties niet toegestaan
- **`noexec`** - uitvoeren van binaries niet toegestaan
- **`soft`** - error wanneer server offline is
- **`sec=krb5`**, **`krb5i`**, **`krb5p`** - gebruik Kerberos authentication
- **`port`** - gebruik specifieke poort

zie `man nfs`

NFS LAB

LAB contents

server0 (File Server)

- Set up a service to share a folder over the network.
- Allow network access through the firewall.
- Share the `/share1` folder with other machines.

server1 (Client Machine)

- Connect to the shared folder on the server.
- Access it temporarily or set it to connect automatically at startup.

Autofs (Automatic Access)

- Set up so the shared folder only connects when needed.
- Saves system resources by disconnecting when idle.

nfs Lab

Hostname	IP Address	Fully Qualified Domain Name (FQDN)
server0	192.168.99.100	server0.psdemo.local
server1	192.168.99.101	server1.psdemo.local
server2	192.168.99.102	server1.psdemo.local
client0	192.168.99.103	server2.psdemo.local

192.168.1.0/24

- Update existing environment
  - `cd ~/sysadv2-labs`
  - `git pull origin main`
- or clone new environment
  - `cd ~`
  - `git clone https://github.com/PXLSystemsAdvancedII/sysadv2-labs`
- `cd ~/sysadv2-labs/nfs_lab`
- `./deploy.sh`

Lab: NFS export - configure nfs server

- **server0** (`vagrant ssh server0`)
  - `sudo -i`
  - installeer nfs service
    - `dnf -y install nfs-utils`
    - `systemctl status nfs-server`
    - `systemctl enable --now nfs-server`
  - firewall config
    - `systemctl enable --now firewalld`
    - `firewall-cmd --permanent --zone public --add-service nfs`
    - `firewall-cmd --reload`

Lab: NFS export - maak een export

```
[root@server0 ~]# cat /var/lib/nfs/etab
/share1 server1(rw,sync,wdelay,hide,nocrossmnt,secure,root_squash,no_all_squash,no_subtree_check,secure_
locks,acl,no_pnfs,anonuid=65534,anongid=65534,sec=sys,rw,secure,root_squash,no_all_squash)
[root@server0 ~]#
```

- **server0**
  - maak een export
    - `mkdir /share1`
    - `vi /etc/exports`

      `/share1 192.168.56.0/24(rw,sync)`
  - maak de export actief
    - `exportfs -arv`
  - check runtime configuration met default options
    - `cat /var/lib/nfs/etab`

sync: writes are committed to disk before the server replies (safer, default)

Lab: NFS export - mount export op client

- vagrant ssh server**1**
  - `sudo dnf -y install nfs-utils`
  - runtime mount
    - `sudo mount -t nfs server0.pxldemo.local:/share1 /mnt/`
    - `mount | grep server0`
    - `ls /mnt/`

```
[vagrant@server1 ~]$ mount | grep server0
server0:/share1 on /mnt type nfs4 (rw,relatime,vers=4.2,rsize=131072,wsize=131072,namlen=255,hard,proto=
tcp,timeo=600,retrans=2,sec=sys,clientaddr=192.168.56.101,local_lock=none,addr=192.168.56.100)
[vagrant@server1 ~]$ ls /mnt/
[vagrant@server1 ~]$
```

Lab: NFS export - welke exports zijn er available op de server?

- server**0**
  - welke exports zijn er available op de server? rpcbind kan service discovery aanbieden.
    - `sudo systemctl enable --now rpcbind`
    - `sudo systemctl status rpcbind`
    - firewall openzetten voor rpcbind
      - `sudo firewall-cmd --permanent --zone public --add-service=rpc-bind`
      - `sudo firewall-cmd --permanent --zone public --add-port=20048/udp`
      - `sudo firewall-cmd --reload`
- server**1**
  - vraag export list van `server0` op
    - `showmount -e server0`

```
Last login: Sun May  4 11:46:58 2025 from 10.0.2.2
[vagrant@server1 ~]$ showmount -e server0
Export list for server0:
/share1 192.168.56.0/24
[vagrant@server1 ~]$
```

Lab: NFS export - persistent NFS mounts

- server**1**
  - `sudo vi /etc/fstab`
    - `192.168.56.0/24:/share1  /mnt  nfs  defaults,rw,_netdev  0  0`
      - `server:/path/to/export  /local_mountpoint  nfs  <options>  0  0`
      - **`_netdev`** optie - wacht met mounten totdat het netwerk online is
  - unmount vorige runtime mount
    - `sudo umount /mnt/`
  - herlees /etc/fstab config file
    - `sudo mount -a`
    - `mount | grep server0`

Lab: NFS export - autofs

- autofs
  - automount daemon
  - mount share enkel bij access
  - unmount na ingestelde tijd
  - minder bandbreedte nodig dan statische mounts
- server**1**
  - enable service
    - `sudo yum -y install autofs`
    - `sudo systemctl enable --now autofs`
  - configure: voeg export toe
    - `sudo vi /etc/auto.misc`

      ```
      share1   -fstype=nfs,rw   server0:/share1
      ```
    - `sudo systemctl restart autofs`
  - test
    - `ls /misc/`
    - `ls /misc/share1`
    - `ls /misc/`

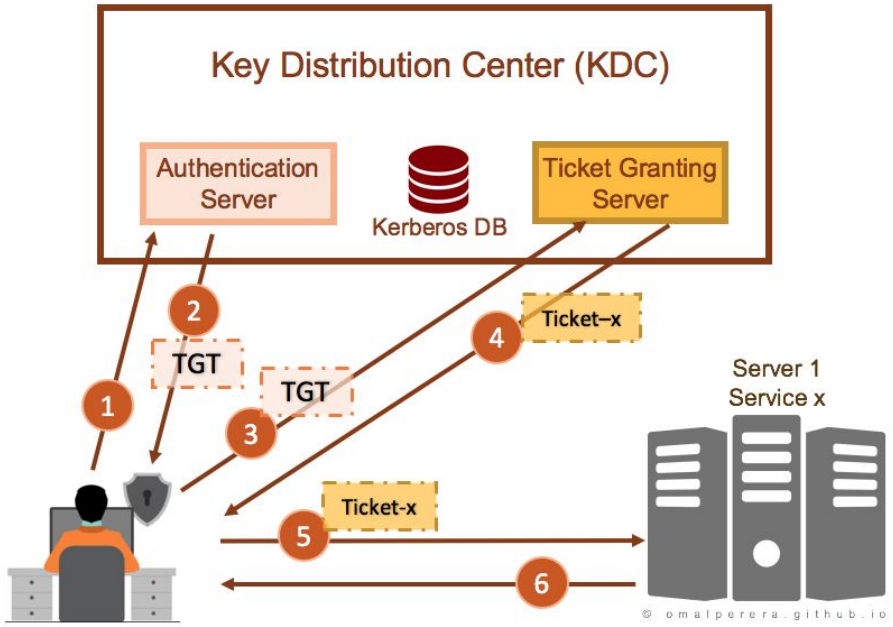
EINDE NFS LAB

NFS File Security

- default: UID/GID security model met AUTH_SYS RPC calls
- gevaarlijk: UIDs en GUID kunnen overlappen!
- Werkt goed met centralized authentication server
- root? (UID 0)
  - **`root_squash`** - enabled by default

Kerberos - korte introductie

- Massachusetts Institute of Technology (MIT) 1980
- Authenticatieprotocol dat werkt op basis van tickets om resources te accessen
- Encryptie beschermt alle access keys and tickets
- Clients kunnen hun identiteit op een veilige manier aantonen en over een onveilig netwerk verbinding maken met services
- Microsoft Active Directory is gebaseerd op de Kerberos Network Authentication Service (V5)
- Key Distribution Center (KDC) *(Active Directory: Domain Controller)*
  - Authentication Service (AS)
    - authenticeert clients en geeft hun Ticket-Granting Tickets (TGT's)
  - Ticket-Granting Service (TGS)
    - aanvaardt authenticated clients en geeft hun tickets om resources zoals files, netwerk, … te accessen
  - Database met gevoelige data
- Principal
  - Unieke identiteit in een Kerberos-systeem waaraan Kerberos tickets kan toewijzen voor access tot Kerberos-aware services
- `keytab`
  - file met pairs van Kerberos principals en encrypted keys



Kerberos Authentication

- AUTH_SYS
  - UID/GID security model
- AUTH_GSS
  - Requirements
    - Kerberos Key Distribution Center (KDC) installed
    - Host en service principals toegevoegd voor client en server
    - Key tabs toegevoegd aan client en server
    - Zowel server exports als client mounts geconfigureerd met sec=krb5:krb5i:krb5p

High Availability?

Systems Advanced II - Linux

kernel

- virtual memory
- user space and kernel space
- processes and threads
- sockets & pipes
- systemd
- Docker

Networking

- routing
- firewall
- eBPF
- certificates
- ssh
- latency
- bandwidth

Services

- nfs
- load balancers and reverse proxies
- databases
- logging

Clusters

- Storage
- Kubernetes

bitrot: gradual corruption of data stored on digital storage devices over time

Case study: Gluster

- What is Gluster ?
  - Gluster is a scalable, distributed file system that aggregates disk storage resources from multiple servers into a **single global namespace**.
- Advantages
  - Scales to several petabytes
  - Handles thousands of clients
  - POSIX compatible
    - POSIX (Portable Operating System Interface) is a set of standards that define a common API for UNIX-like operating systems to ensure software compatibility across different platforms.
  - Uses commodity hardware
  - Can use any on-disk file system that supports extended attributes
  - Accessible using industry standard protocols like NFS and SMB
  - Provides replication, quotas, geo-replication, snapshots and **bitrot detection**
  - Allows optimization for different workloads
  - Open Source, RedHat

https://docs.gluster.org/en/latest/

Gluster LAB

PXLSystemsAdvancedII/gluster-compose-lab

Elasticsearch - Kibana LAB

PXLSystemsAdvancedII/elasticsearch-kibana-compose-lab

quorum: meer dan de helft van de nodes vormt een quorum of absolute meerderheid. De cluster is available zolang de available nodes quorum hebben.

HA-NFS LAB

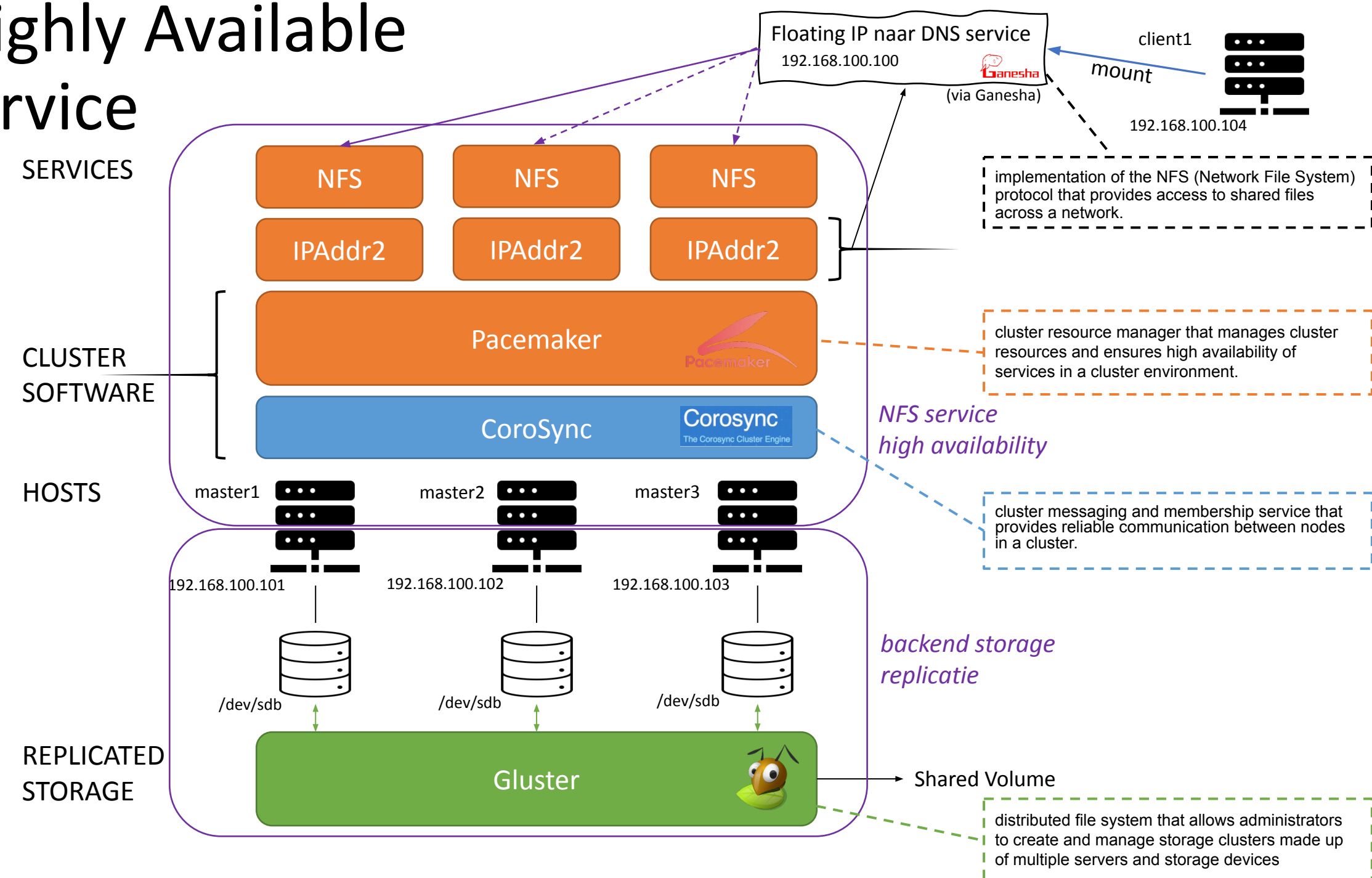
Lab: Highly Available NFS service

quorum: meer dan de helft van de nodes vormt een quorum of absolute meerderheid. De cluster is available zolang de available nodes quorum hebben.

Installatie en configuratie van een HA-NFS service op Oracle Linux 7 (==RHEL) met **Corosync**, **Pacemaker**, **Gluster** en **Ganesha**.

- NFS service hosted door 3 VMs: master1, master2, master3.
- Elk van de 3 VMs zal een gluster volume repliceren voor data redundancy en cluster tools gebruiken voor service redundancy.
- Components
  - **Corosync**: cluster messaging and membership service that provides reliable communication between nodes in a cluster.
  - **Pacemaker**: cluster resource manager that manages cluster resources and ensures high availability of services in a cluster environment.
  - **Ganesha**: implementation of the NFS (Network File System) protocol that provides access to shared files across a network.
  - **Gluster**: distributed file system that allows administrators to create and manage storage clusters made up of multiple servers and storage devices.

Lab: Highly Available NFS service

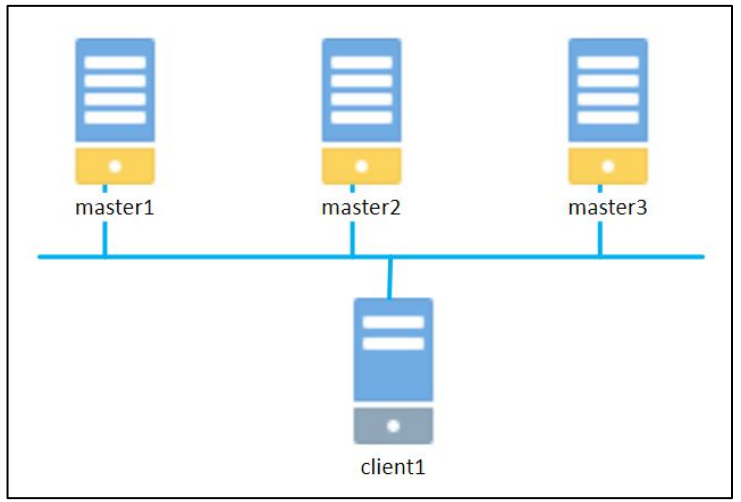


ha_nfs Lab

- Update existing environment
  - `cd ~/sysadv2-labs`
  - `git pull origin main`

- or clone new environment
  - `cd ~`
  - `git clone https://github.com/PXLSystemsAdvancedII/sysadv2-labs`

- `cd ~/sysadv2-labs/ha_nfs_lab`
- `./deploy.sh`
- maak 4 terminal windows klaar, of tabs, of via split window, om makkelijk te switchen naar master1, master2, master3 en client1
- Log in op alle masters en wordt root
  - `vagrant ssh master1`
  - `sudo -i`



Hostname	IP Address	Fully Qualified Hostname
master1	192.168.99.101	master1.vagrant.vm
master2	192.168.99.102	master2.vagrant.vm
master3	192.168.99.103	master3.vagrant.vm
client1	192.168.99.104	client1.vagrant.vm
nfs	192.168.99.100	nfs.vagrant.vm

ORACLE Linux

Stap 1: software install

Stap 1: software installatie van alle software componenten via ge-activeerde repositories.

Op alle masters:

- Install repositories
  - `sudo yum install -y oracle-gluster-release-el7`
- Enable repositories
  - `sudo yum-config-manager --enable ol7_addons ol7_latest ol7_optional_latest ol7_UEKR5`
- Install software componenten
  - `sudo yum install -y corosync glusterfs-server nfs-ganesha-gluster pacemaker pcs`

Stap 2: Maak een Gluster replicated volume

2.a filesysteem preparation

Stap 2: de extra disk van elke master prepareren, een replicated Gluster volume maken en activeren.

Op alle masters:

- Maak een XFS filesysteem op /dev/sdb met een label gluster-000
  - `sudo mkfs.xfs -f -i size=512 -L gluster-000 /dev/sdb`
- Maak een mountpoint, voeg een fstab entry toe voor een disk met label gluster-000 en mount het file systeem
  - `sudo mkdir -p /data/glusterfs/sharedvol/mybrick`
  - `echo 'LABEL=gluster-000 /data/glusterfs/sharedvol/mybrick xfs defaults  0 0' | sudo tee -a /etc/fstab`
  - `mount /data/glusterfs/sharedvol/mybrick`

Stap 2: Maak een Gluster replicated volume

2.b gluster environment

Op alle masters:

- Enable en start Gluster service
  - `sudo systemctl enable --now glusterd`

Op master1:

- Maak een Gluster environment door peers toe te voegen
  - `sudo gluster peer probe master2.vagrant.vm`
  - `sudo gluster peer probe master3.vagrant.vm`

Op alle masters:

- Check op alle peers dat ze in de Gluster environment zitten
  - `sudo gluster peer status`

Stap 2: Maak een Gluster replicated volume

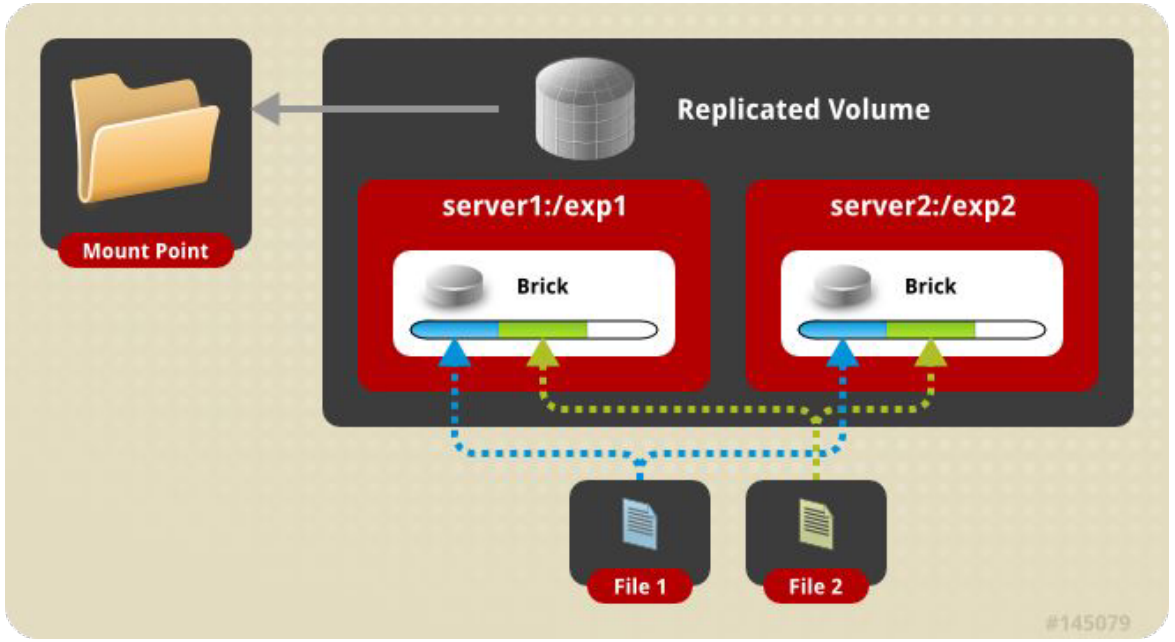
2.c gluster volume

Op master1:

- Maak een Gluster volume "sharedvol" dat replicated wordt over master1, master2 en master3
  - `sudo gluster volume create sharedvol replica 3 master{1,2,3}:/data/glusterfs/sharedvol/mybrick/brick`
- Enable het Gluster volume "sharedvol"
  - `sudo gluster volume start sharedvol`

Op alle masters:

- Check gluster volume info en status
  - `sudo gluster volume info`
  - `sudo gluster volume status`



Stap 3: Configureer Ganesha

Ganesha is de NFS server die het Gluster volume deelt. In dit voorbeeld laten we elke NFS client toe om te verbinden met onze NFS share met lees/schrijf rechten.

Op alle masters:

- `/etc/ganesha/ganesha.conf`

```
EXPORT{
    Export_Id = 1 ;       # Unique identifier for elke EXPORT
    Path = "/sharedvol";  # Export path van onze NFS share

    FSAL {
        name = GLUSTER;          # Backing type is Gluster
        hostname = "localhost";  # Hostname van Gluster server
        volume = "sharedvol";    # Naam van ons Gluster volume
    }

    Access_type = RW;           # Export access permissions
    Squash = No_root_squash;    # Control NFS root squashing
    Disable_ACL = FALSE;        # Enable NFSv4 ACLs
    Pseudo = "/sharedvol";      # NFSv4 pseudo path for our NFS share
    Protocols = "3","4" ;       # NFS protocols supported
    Transports = "UDP","TCP" ; # Transport protocols supported
    SecType = "sys";            # NFS Security flavors supported
}
```

Stap 4: Maak een Pacemaker/Corosync cluster

4.a set authentication en enable cluster services

We gaan een Pacemaker/Corosync cluster aanmaken en starten, die bestaat uit onze drie master nodes.

Op alle masters:

- Maak een shared paswoord voor de user hacluster
  - `passwd hacluster`
- Enable de Corosync and Pacemaker services. Enable en start de pacemaker configuration system service. De Corosync en Pacemaker services zullen later gestart worden.
  - `systemctl enable corosync`
  - `systemctl enable pacemaker`
  - `systemctl enable --now pcsd`

Op master1:

- Authenticeer met alle cluster nodes via de hacluster user en het nieuwe shared paswoord van daarnet
  - `pcs cluster auth master1 master2 master3 -u hacluster -p SHAREDPASSWORD`

Stap 4: Maak een Pacemaker/Corosync cluster

4.b maak een Pacemaker cluster

Op master1:

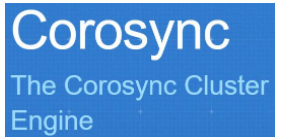
- Maak de cluster "HA-NFS"
  - `pcs cluster setup --name HA-NFS master1 master2 master3`
- Start de cluster op alle nodes
  - `pcs cluster start --all`
- Enable de cluster op alle nodes (start at boot time)
  - `pcs cluster enable --all`
- disable STONITH
  - `pcs property set stonith-enabled=false`

Op alle masters:

- De pacemaker cluster werkt
  - `pcs cluster status`

STONITH is een acroniem voor Shoot-The-Other-Node-In-The-Head, een recommended practice om een node die zich misdraagt onmiddellijk te isoleren (uitschakelen, afsnijden van gedeelde resources of anderszins immobiliseren), en wordt gewoonlijk geïmplementeerd met een remote power switch.

Stap 5: Maak Cluster Services



We gaan een resource groep aanmaken die de middelen bevat die nodig zijn om NFS services te hosten vanaf de virtuele hostnaam nfs.vagrant.vm (192.168.99.100).

Op master1:

- Maak een systemd-gebaseerde cluster resource aan om te verzekeren dat nfs-ganesha draait. Check elke 10s of nfs draait.
  - `pcs resource create nfs_server systemd:nfs-ganesha op monitor interval=10s`
- Maak een floating IP cluster resource aan dat gebruikt wordt om de NFS server aan te bieden. Check elke 10s of het adres werkt.
  - `pcs resource create nfs_ip ocf:heartbeat:IPaddr2 ip=192.168.100.100 cidr_netmask=24 op monitor interval=10s`
- Voeg de Ganesha service en IP resource samen in een groep om er zeker van te zijn dat ze altijd samen op dezelfde host blijven
  - `pcs resource group add nfs_group nfs_server nfs_ip`

Op alle masters:

- Toon de status van de pacemaker cluster en de aangeboden resources
  - `pcs status`

Stap 6: High-Availability Fail-over Test
6.a alles is top - no problems

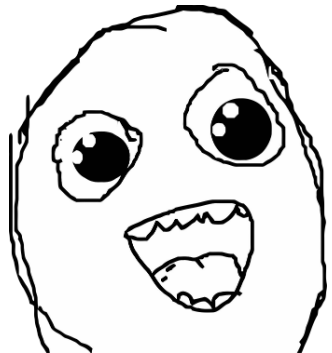
Op client1:

- Mount de NFS service van onze cluster and maak een file op de cluster.
  - `sudo -i`
  - `yum install -y nfs-utils`
  - `mkdir /sharedvol`
  - `mount -t nfs nfs.vagrant.vm:/sharedvol /sharedvol`
  - `df -h /sharedvol/`
  - `echo "All your base are belong to us" > /sharedvol/hello`

Op master2 of master3:

DIT DUURT HEEL LANG DOOR VIRTUALBOX VIRTUAL NIC BUG (+1 MIN)

- `pcs status`
- `ls /data/glusterfs/sharedvol/mybrick/brick/`



Stap 6: High-Availability Fail-over Test
6.b Boom.

Op VirtualBox GUI:

- Poweroff van master2. Disaster. De Designated Controller is plots offline. De cluster resources nfs en floating ip werden door deze node geserved.

Op master2:

- Binnen de 10s merkt de cluster
  - Pacemaker
    - node master2 is offline
    - wij (master1 en master3) hebben quorum (meerderheid)
    - Designated Controller gaat naar een van de online nodes
      - Resources worden toegewezen
      - Corosync zorgt voor opstarten van resources
  - Gluster
    - node master2 is offline
    - replication en volume voorziening blijven actief
- Check fail-over status.
  - `pcs status`

Op client1:

Test de NFS share availability

```
ls -la /sharedvol/
cat /sharedvol/hello
```

DIT DUURT HEEL LANG DOOR VIRTUALBOX VIRTUAL NIC BUG (+1 MIN)

Op VirtualBox GUI:

Poweron van master1.

Op master1 of master2:

Check hoe de node terug online komt

```
pcs status
```

```
sudo tail -f /var/log/glusterfs/etc-glusterfs-glusterd.vol.log
```

HIGH AVAILABILITY CONCEPTS

Core Cluster Safety Concepts

Quorum
Minimum number of nodes that must agree to make safe decisions. Prevents split-brain. *E.g. etcd, Consul*

Split-Brain
A condition where two nodes believe they are primary. Leads to data corruption. *E.g. DynamoDB*

STONITH
"Shoot The Other Node In The Head", forcibly powers off failed nodes to prevent split-brain. *E.g. Kubernetes pod eviction*

Fencing *(== a form of STONITH)*
Isolation of a node (network block or eviction) to protect cluster integrity. *E.g. Kubernetes unschedulable nodes*

Leader Election
Process where cluster nodes select one leader to coordinate operations (e.g., write traffic). *E.g. etcd, Kubernetes controller* manager

Failover & Self-Healing

Failover

Automatic switchover to a standby node/service when the active one fails. *E.g. Kubernetes* Deployments

Self-Eviction / Demotion

Node voluntarily stops acting as primary if it detects loss of quorum or failure. *E.g. etcd*

Health Probes (Liveness / Readiness)

Checks used to detect unhealthy containers and remove them from service/load. *E.g. any HA service.*

Watchdog Timer

A hardware or software timer that reboots the node if it becomes unresponsive. *E.g. IOS or Android*

KERBEROS LAB

Lab: NFS File Security - sec=sys

- *zelfde set-up als nfs lab*
- server1 (as user vagrant)
  - `touch /misc/share1/file1.test`
  - `ls -la /misc`
- server0
  - `sudo chown vagrant:vagrant /share1`
- server1 (as user vagrant)
  - `touch /misc/share1/file1.test`
  - `ls -la /`
- geen unified authentication system!
  - mapped uids en gids worden gebruikt
  - op beide systemen
    - `sudo chown vagrant:vagrant /share1`

Lab: NFS File Security - Kerberos installeren

- server0
  - `sudo -i`
  - install ntp service
    - `yum -y install chrony`
    - `systemctl enable --now chronyd`
    - `chronyc tracking`
  - install kerberos service - *zie config files volgende slide*
    - `yum -y install krb5-server krb5-libs`
    - `vi /var/kerberos/krb5kdc/kadm5.acl`
      - `*/admin@PSDEMO.LOCAL *`
    - `vi /etc/krb5.conf` (*next slide*)
    - `vi /var/kerberos/krb5kdc/kdc.conf` (*next slide*)

/etc/krb5.conf

```
[libdefaults]
    default_realm = PSDEMO.LOCAL
    dns_lookup_realm = false
    dns_lookup_kdc = false
    ticket_lifetime = 24h
    forwardable = true
    udp_preference_limit = 1000000

[realms]
    PSDEMO.LOCAL = {
        kdc = server0.psdemo.local:88
        admin_server = server0.psdemo.local:749
        default_domain = psdemo.local
    }

[domain_realm]
    .psdemo.local = PSDEMO.LOCAL
     psdemo.local = PSDEMO.LOCAL

[logging]
    kdc = FILE:/var/log/krb5kdc.log
    admin_server = FILE:/var/log/kadmin.log
    default = FILE:/var/log/krb5lib.log
```

/var/kerberos/krb5kdc/kdc.conf

```
default_realm = PSDEMO.LOCAL

[kdcdefaults]
    v4_mode = nopreauth
    kdc_ports = 0

[realms]
    PSDEMO.LOCAL = {
        kdc_ports = 88
        admin_keytab = /etc/kadm5.keytab
        database_name =
/var/kerberos/krb5kdc/principal
        acl_file = /var/kerberos/krb5kdc/kadm5.acl
        key_stash_file = /var/kerberos/krb5kdc/stash
        max_life = 10h 0m 0s
        max_renewable_life = 7d 0h 0m 0s
        default_principal_flags = +preauth
    }
```

Lab: NFS File Security - Kerberos server set-up

- server0
  - maak de kdc database voor de gevoelige data met een paswoord
    - `kdb5_util create -r PSDEMO.LOCAL -s`
  - maak een admin principal en steek die in de keytab
    - `kadmin.local`
      - `addprinc root/admin`
      - `ktadd -k /var/kerberos/krb5kdc/kadm5.keytab kadmin/admin`
      - `ktadd -k /var/kerberos/krb5kdc/kadm5.keytab kadmin/changepw`
      - `exit`
  - start kerberos services
    - `systemctl enable --now krb5kdc.service`
    - `systemctl enable --now kadmin.service`
  - maak een principal voor de kdc server en de nfs service en steek ze in de keytab
    - `kadmin.local`
      - `addprinc -randkey host/server0.psdemo.local`
      - `addprinc -randkey nfs/server0.psdemo.local`
      - `ktadd host/server0.psdemo.local`
      - `ktadd nfs/server0.psdemo.local`
      - `list_principals`
      - `exit`

Lab: NFS File Security - Kerberos client set-up

- server1
  - sudo yum -y install krb5-workstation
  - sudo vi /etc/krb5.conf
    - zelfde config - *zie 2 slides terug*
  - principals maken voor host en nfs service, toevoegen aan lokale keytab
    - sudo kadmin -p root/admin
      - addprinc -randkey host/server1.psdemo.local
      - addprinc -randkey nfs/server1.psdemo.local
      - ktadd host/server1.psdemo.local
      - ktadd nfs/server1.psdemo.local
      - ktadd host/server0.psdemo.local

Lab: NFS File Security - NFS kerberos5 config: server

- server0
  - pas exports aan om krb5 security model te gebruiken
    - `sudo vi /etc/exports`
      - `/share1  server1.psdemo.local(rw,sec=krb5)`
  - maak de export actief
    - `sudo exportfs -arv`
  - check runtime configuration met default options
    - `cat /var/lib/nfs/etab`

Lab: NFS File Security - NFS kerberos5 config: client

- server1
  - (re)start nfs-client
    - `sudo systemctl enable --now nfs-client.target`
  - runtime mount met sec=krb5
    - `sudo mount -t nfs -o sec=krb5 server0.psdemo.local:/share1 /mnt/`
    - `mount | grep server0`
  - verander /etc/fstab voor permanente change
    - `sudo vi /etc/fstab`
      - `server0.psdemo.local:/share1  /mnt  nfs  defaults,rw,_netdev,sec=krb5  0  0`
    - `sudo umount /mnt`
    - `sudo mount -a`
    - `mount | grep server0`
    - `ls /mnt/`

Lab: NFS File Security - NFS kerberos5 config: geef user access met ticket

- server1
  - voeg principal toe voor user vagrant
    - `sudo kadmin -p root/admin`
      - `addprinc vagrant`
      - `exit`
  - aan de kdc een kerberos ticket vragen
    - kinit
      - user password van daarnet
    - tickets opvragen
      - `klist`
  - `ls /mnt/`

EINDE KERBEROS LAB

end