

# Systems Advanced II

## Linux

NFS



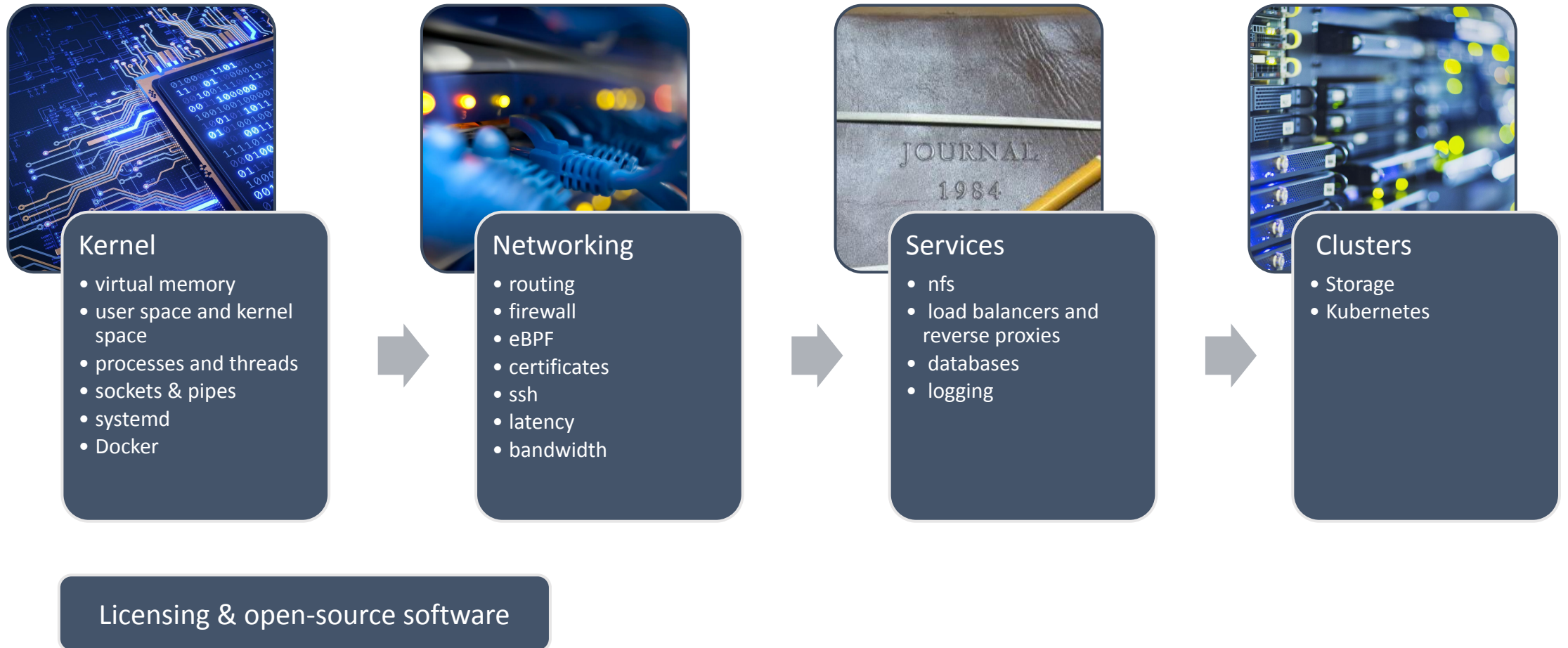
**DE HOGESCHOOL  
MET HET NETWERK**

Elfde-Liniestraat 24, 3500 Hasselt, [www.pxl.be](http://www.pxl.be)

# Doelstellingen

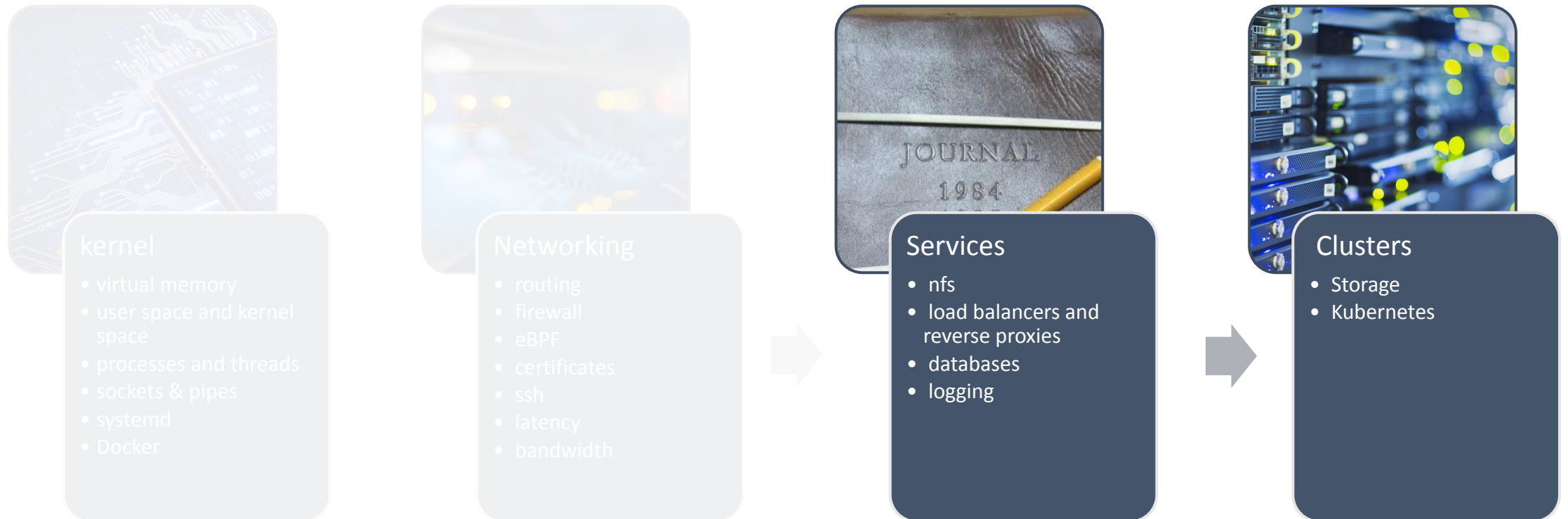
- De student:
  - De student kan Netwerk-services installeren, configureren en onderhouden.
  - De student kan microservices-infrastructuur opzetten en beheren.
  - De student kan een (eigen) cloud systeem opzetten a.d.h.v. opgelegde voorwaarden.
  - De student kan een systeem beveiligen.

# Systems Advanced II - Linux





# Systems Advanced II - Linux



# Systems Advanced II - Linux



# Overzicht

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

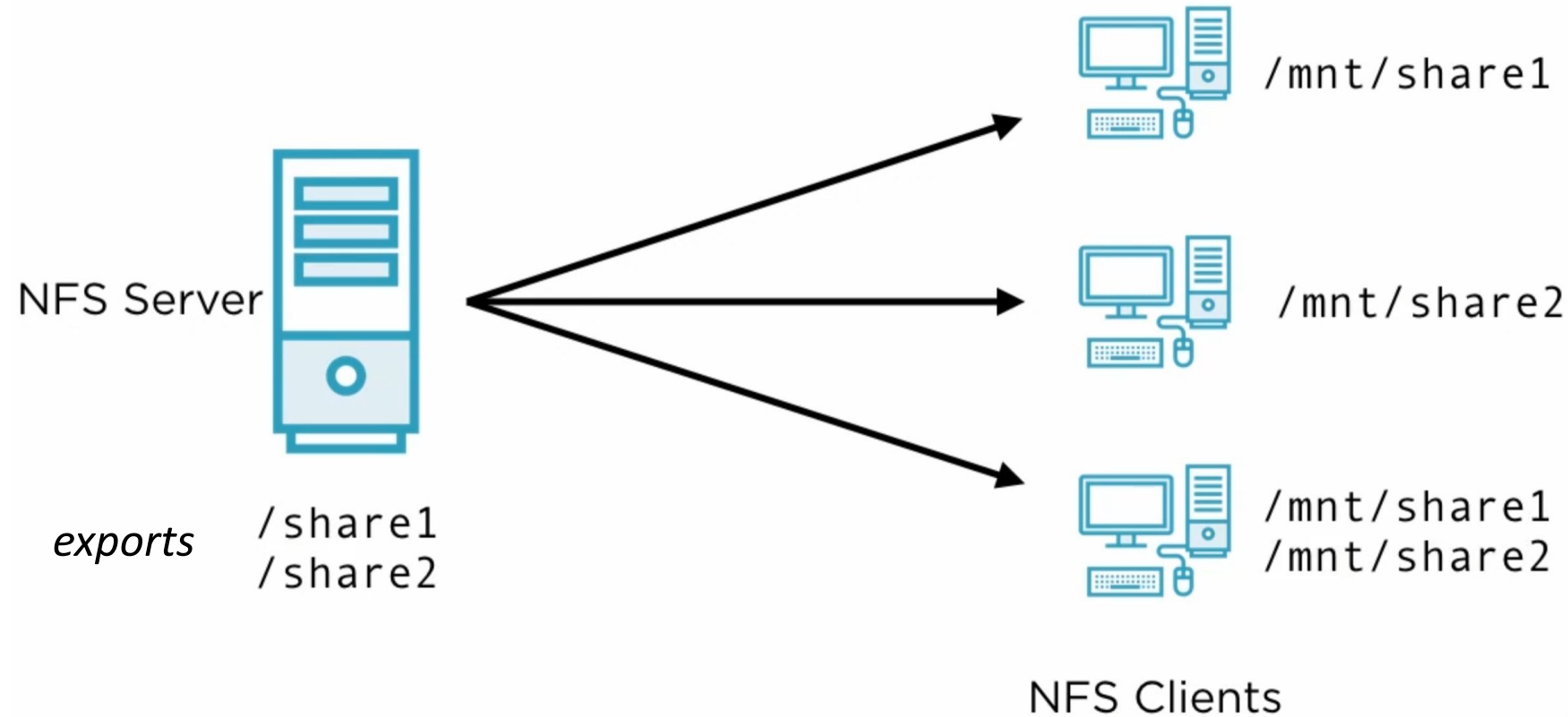
# Network File System (NFS)



- 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 \(ietf.org\)](https://www.ietf.org/rfc/rfc7530.html)
- ook ingebouwd in Windows Server

# NFS system

*mounts*

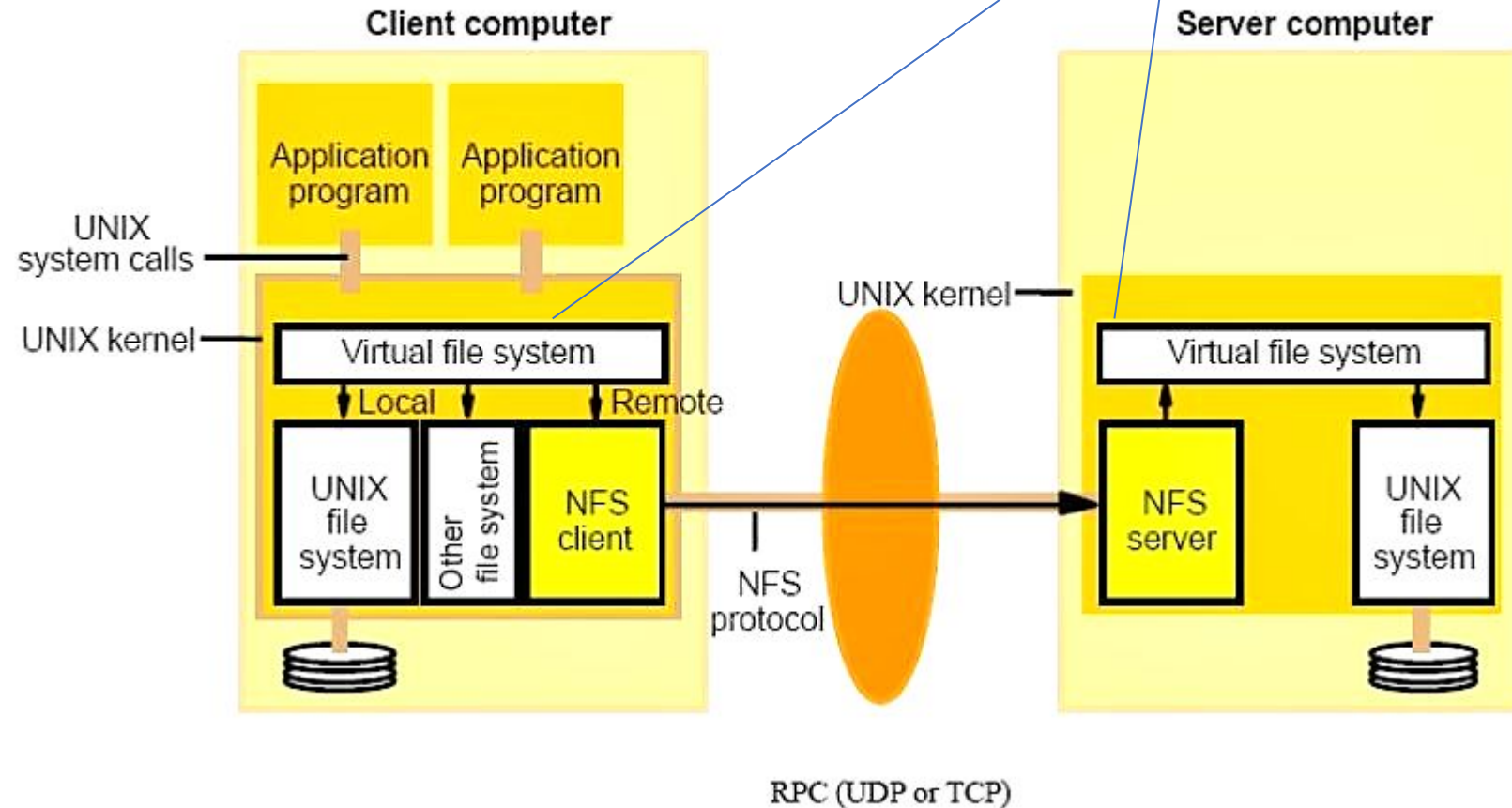




# NFS architecture

## 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
- NFSv4
  - TCP/2049
  - more reliable
  - meer security features zoals Kerberos support
  - file locking support
  - complex

# NFS Services

- Server
  - nfs-server (nfsd)
    - TCP based service (Version 4)
  - rpcbind
    - managen van RPC port reservations en connections (Version 3)
    - service discovery (Version 3 en Version 4)
- Client
  - rpcbind
    - managen van RPC port reservations en connections (Version 3)
- Versies
  - vaak zijn versie 3 en 4 allebei default geïnstalleerd in linux distributions

# 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.psdemo.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?.psdemo.local
  - [a-z]
    - server[2-9].psdemo.local
- netgroups
  - NIS netgroup (network groups defined on a Network Information Server or NIS server)



# 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 committed 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 methodes

- runtime mounting (weg na reboot)
  - `mount -t nfs -o -rw server0.psdemo.local:/share1 /mnt/share1`
- persistent mounting
  - `/etc/fstab`
    - `server0.psdemo.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 filesystem 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



# nfs-lab environment

- Update bestaande environment

- `cd ~/sysadv2-2223`
- `git pull origin main`

- OF: clone nieuwe environment

- `cd ~`
- `git clone https://github.com/PXLSystemsAdvancedII/sysadv2-2223.git`

- `cd ~/sysadv2-2223/nfs_lab`
- `vagrant up`
- `vagrant hosts list`

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

# Lab: NFS export - configure nfs server

- **server0** (`vagrant ssh server0`)
  - `sudo -i`
  - install nfs service
    - `yum -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

- **server0**

- maak een export
  - `mkdir /share1`
  - `vi /etc/exports`  
`/share1 server1.psdemo.local`
- maak de export actief
  - `exportfs -arv`
- check runtime configuration met default options
  - `cat /var/lib/nfs/etab`
- maak export r/w
  - `vi /etc/exports`  
`/share1 server1.psdemo.local(rw)`
  - `exportfs -arv`
  - `cat /var/lib/nfs/etab`

# Lab: NFS export - mount export op client

- server<sup>1</sup>
  - `sudo yum -y install nfs-utils`
  - runtime mount
    - `sudo mount -t nfs server0.psdemo.local:/share1 /mnt/`
    - `mount | grep server0`
    - `ls /mnt/`

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

- server<sup>0</sup>

- 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
    - `firewall-cmd --permanent --zone public --add-service=rpc-bind`
    - `firewall-cmd --permanent --zone public --add-port=20048/udp`
    - `firewall-cmd --reload`

- server<sup>1</sup>

- vraag export list van server0 op
  - `showmount -e server0.psdemo.local`



# Lab: NFS export - persistent NFS mounts

- server<sup>1</sup>
  - `sudo vi /etc/fstab`
    - `server0.psdemo.local:/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<sup>1</sup>
  - 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.psdemo.local:/share1
    - `sudo systemctl restart autofs`
  - test
    - `ls /misc/`
    - `ls /misc/share1`
    - `ls /misc/`

# Lab: NFS export - NFS access

- server<sup>2</sup>
  - `sudo yum -y install nfs-utils`
  - mount export van server0
    - `sudo mount -t nfs server0.psdemo.local:/share1 /mnt/`
      - access denied
- server<sup>0</sup>
  - check exports configuration
    - `cat /etc/exports`
    - enkel server1 heeft access
  - open access in exports configuration
    - `sudo vi /etc/exports`  
/share1 server?.psdemo.local(rw)
    - `sudo exportfs -arv`
- server<sup>2</sup>
  - mount export van server0
    - `sudo mount -t nfs server0.psdemo.local:/share1 /mnt/`
  - `mount | grep server0`

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



server0

Resource with access to  
alice - UID 1001



client1

alice is UID 1001



client2

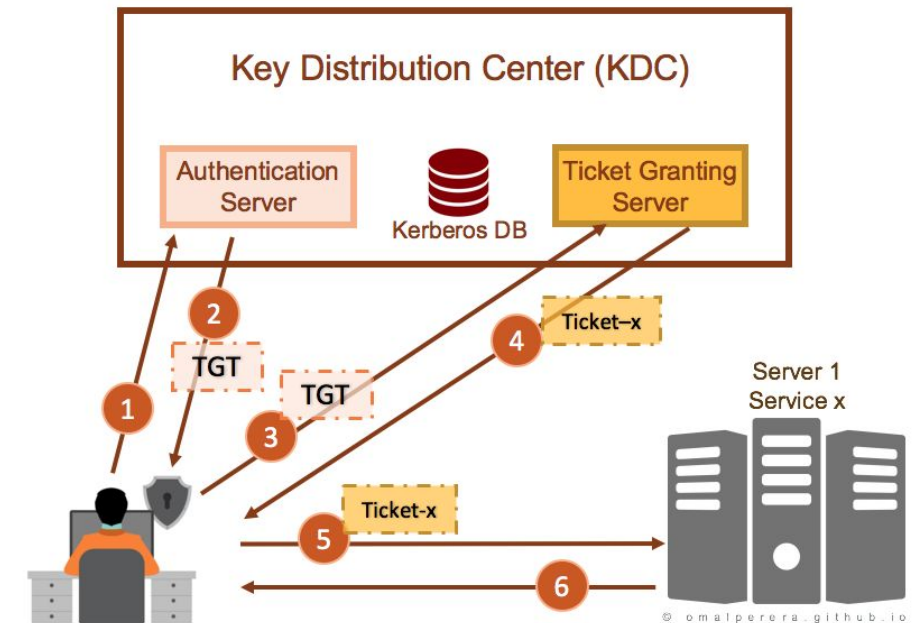
bob is UID 1001





# 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)
    - authenticceert 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

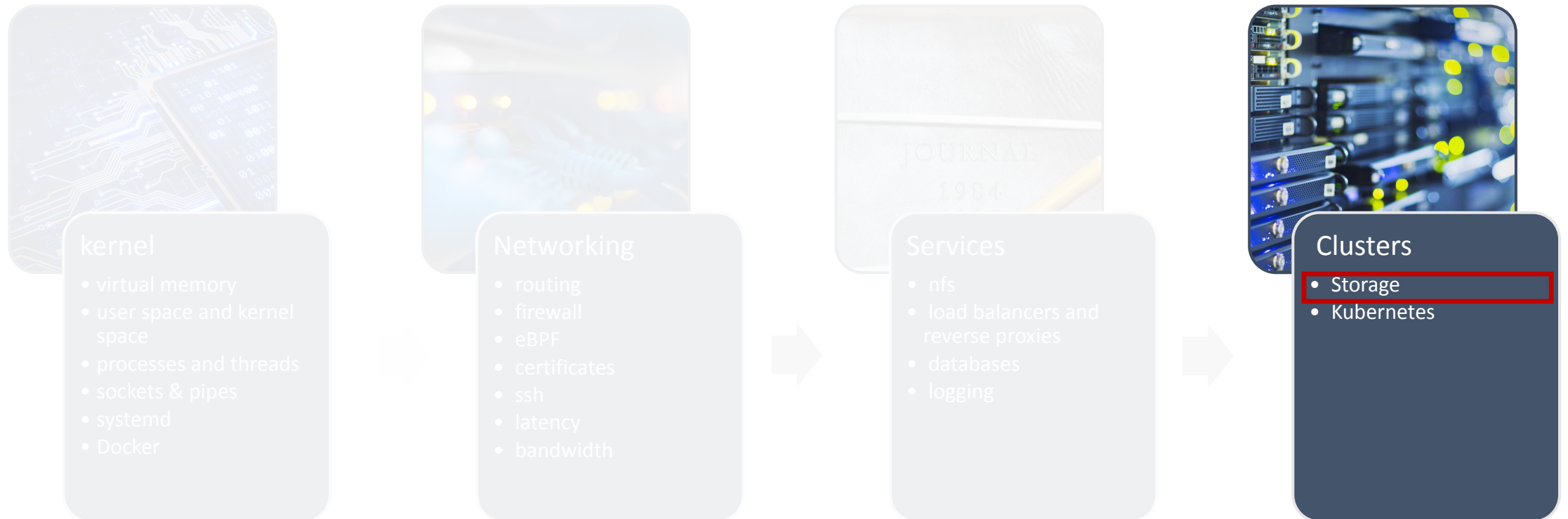


# Kerberos Authentication

- 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



# 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

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.

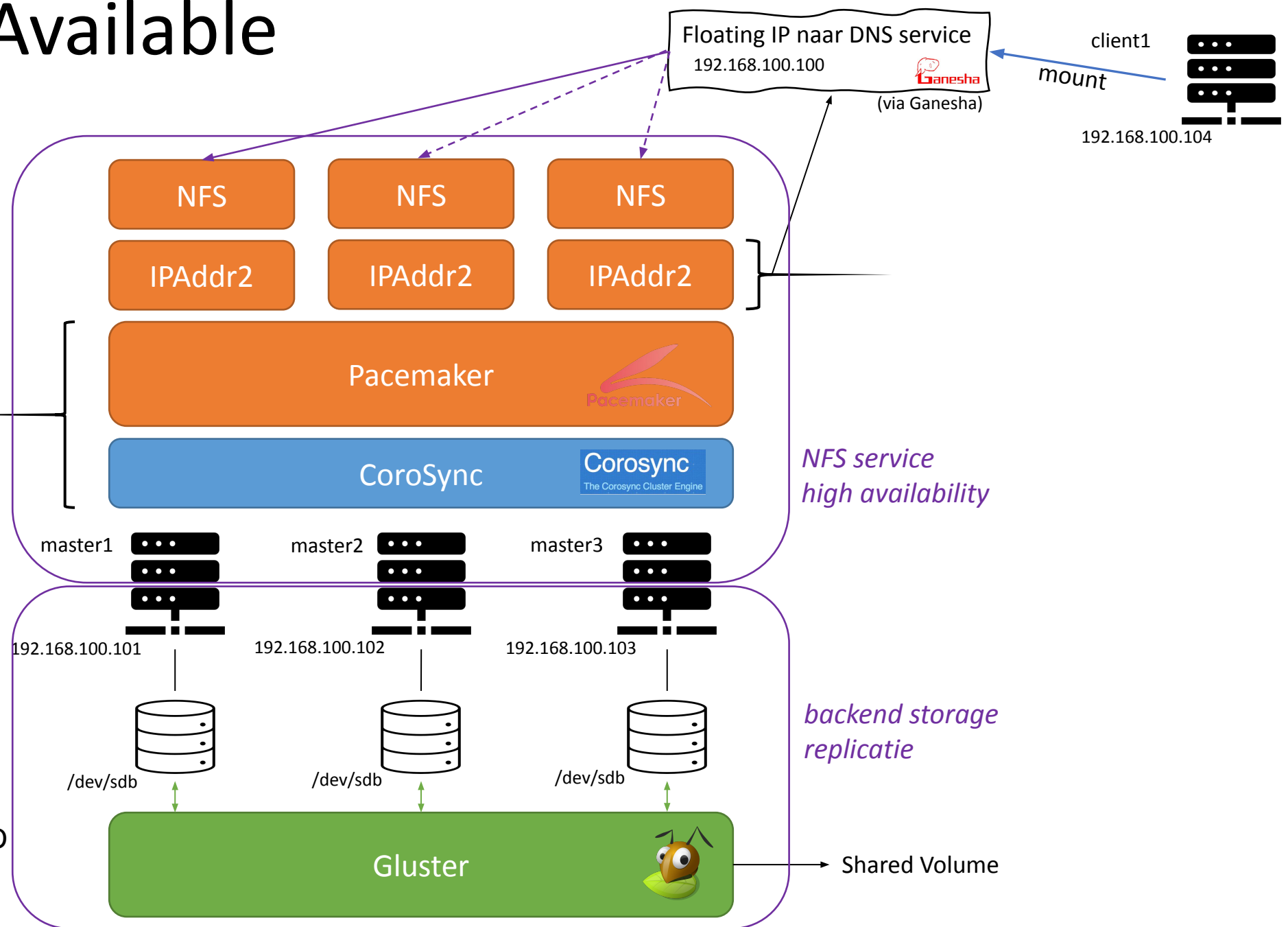
# Lab: Highly Available NFS service

SERVICES

CLUSTER SOFTWARE

HOSTS

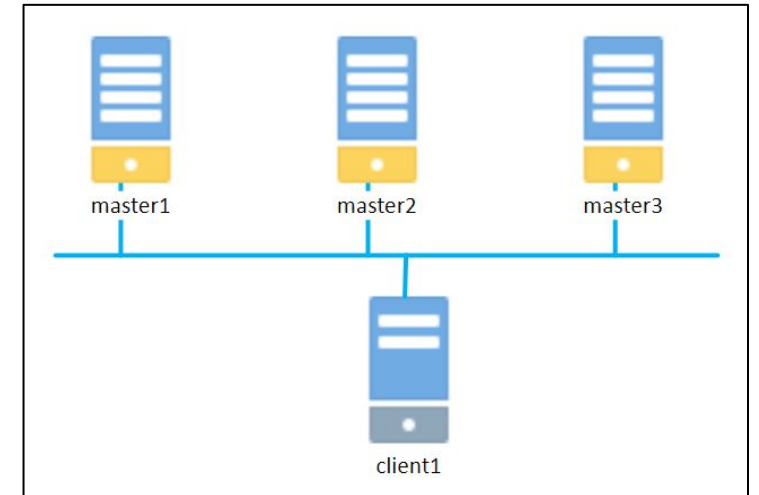
REPLICATED STORAGE





# ha\_nfs-lab environment

- Update bestaande environment
  - `cd ~\sysadv2-2223`
  - `git pull origin main`
- OF: clone nieuwe environment
  - `cd ~`
  - `git clone https://github.com/PXLSystemsAdvancedII/sysadv2-2223.git`
- `cd ~/sysadv2-2223/ha_nfs_lab`
- `vagrant up`
- 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



**Windows Terminal** split window:

**[ALT] [SHIFT] [D]**

**[ALT] [SHIFT] [-]**

**[ALT] [SHIFT] [=]**

**Windows Terminal** resize window:

**[ALT] [SHIFT] [↑] [←] [↓] [→]**

# 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 filesystem preparation

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

### Op **alle** masters:

- Maak een XFS filesystem 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 master**1**:

- 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

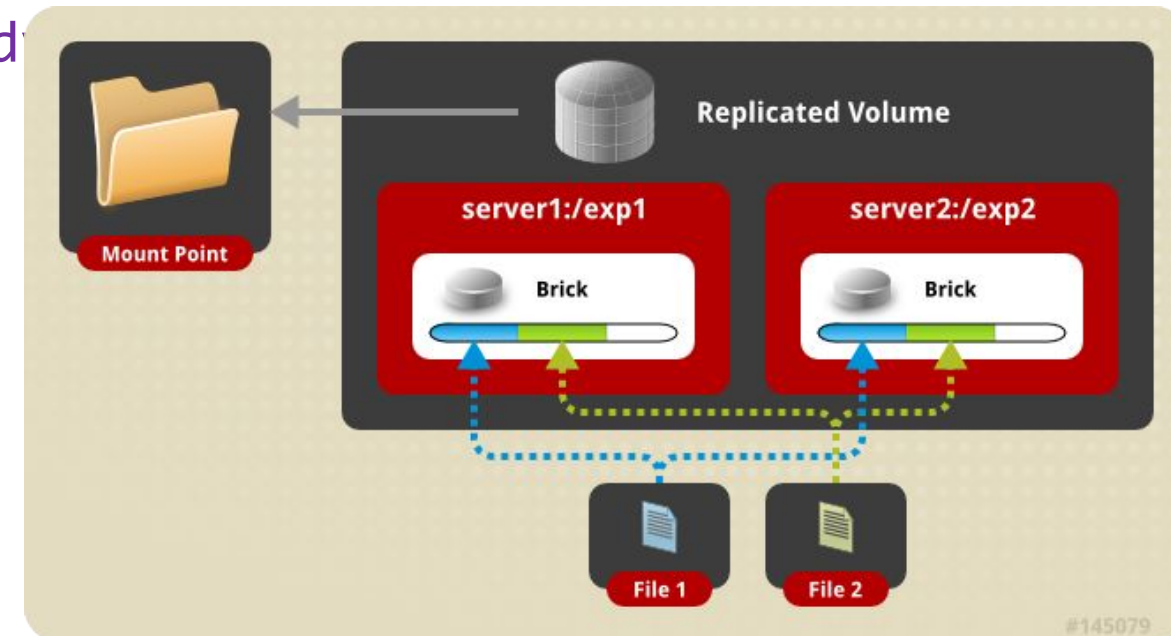
## 2.c gluster volume

### Op master<sup>1</sup>:

- 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 <sup>alle</sup> 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 master<sup>1</sup>:

- 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

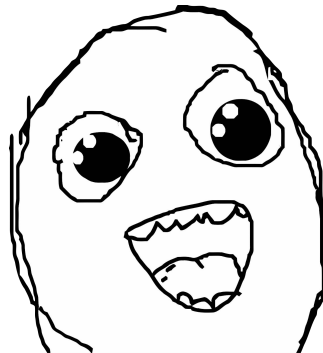
### Op client1:

- 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:

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

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



# Stap 6: High-Availability Fail-over Test

## 6.b Boom.

### Op VirtualBox GUI:

- Poweroff van master<sup>2</sup>. 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
```



KERBEROS LAB

# Lab: NFS File Security - sec=sys

- *zelfde set-up als nfs lab*
- server<sup>1</sup> (as user vagrant)
  - `touch /misc/share1/file1.test`
  - `ls -la /misc`
- server<sup>0</sup>
  - `sudo chown vagrant:vagrant /share1`
- server<sup>1</sup> (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

- server<sup>1</sup>
  - `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

- server<sup>1</sup>
  - (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

- server<sup>1</sup>
  - 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**

# Overzicht

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

end