

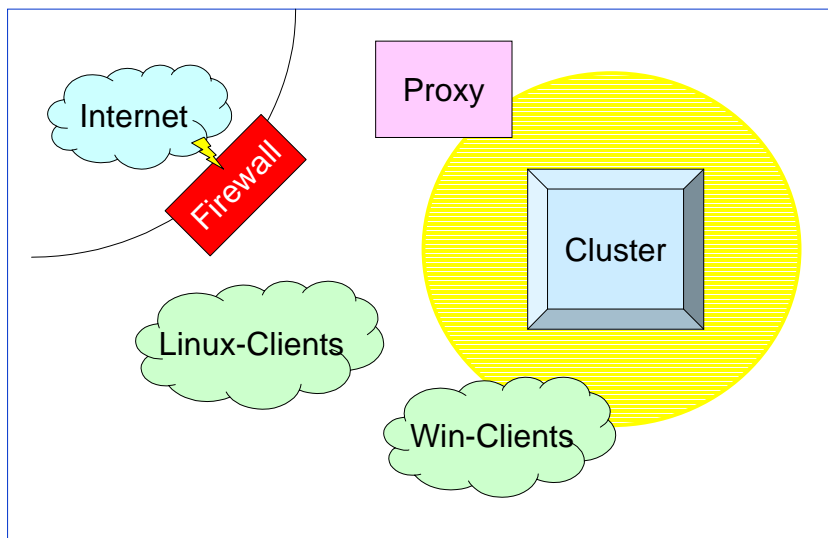
Aufbau eines hochverfügbaren Linux-Clusters

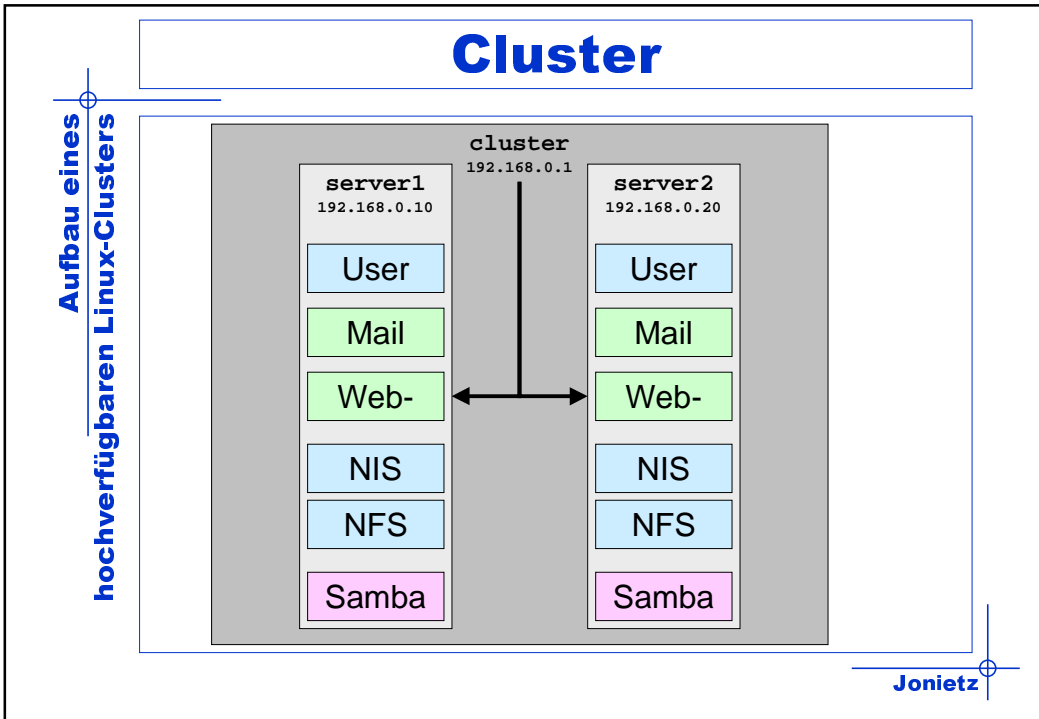
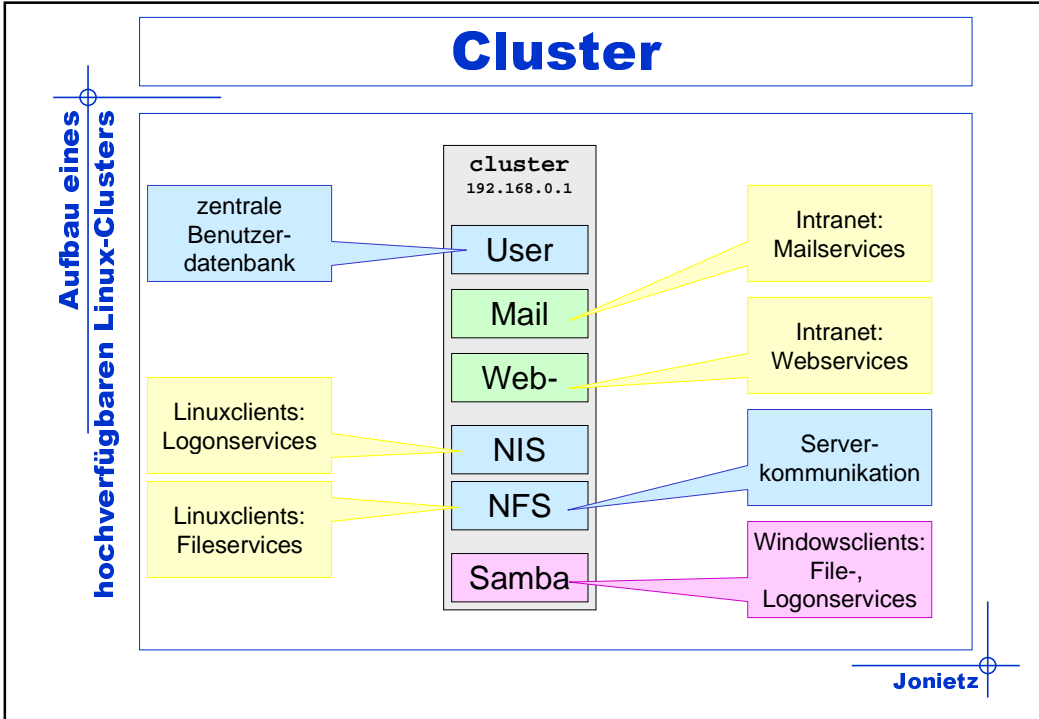
Teil II: Cluster

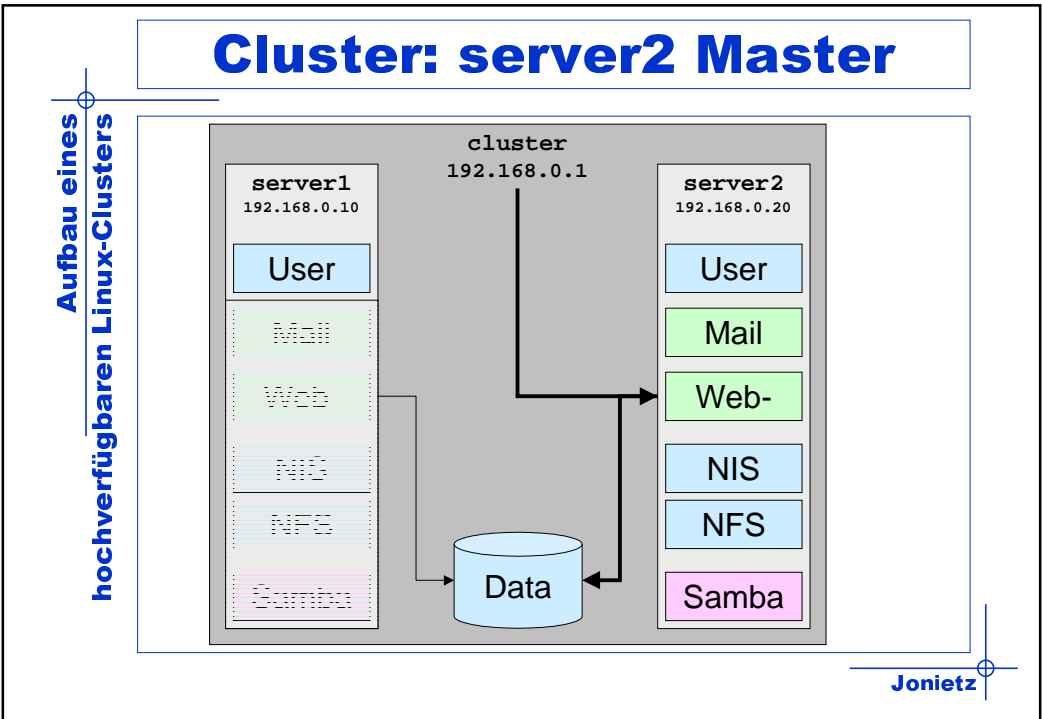
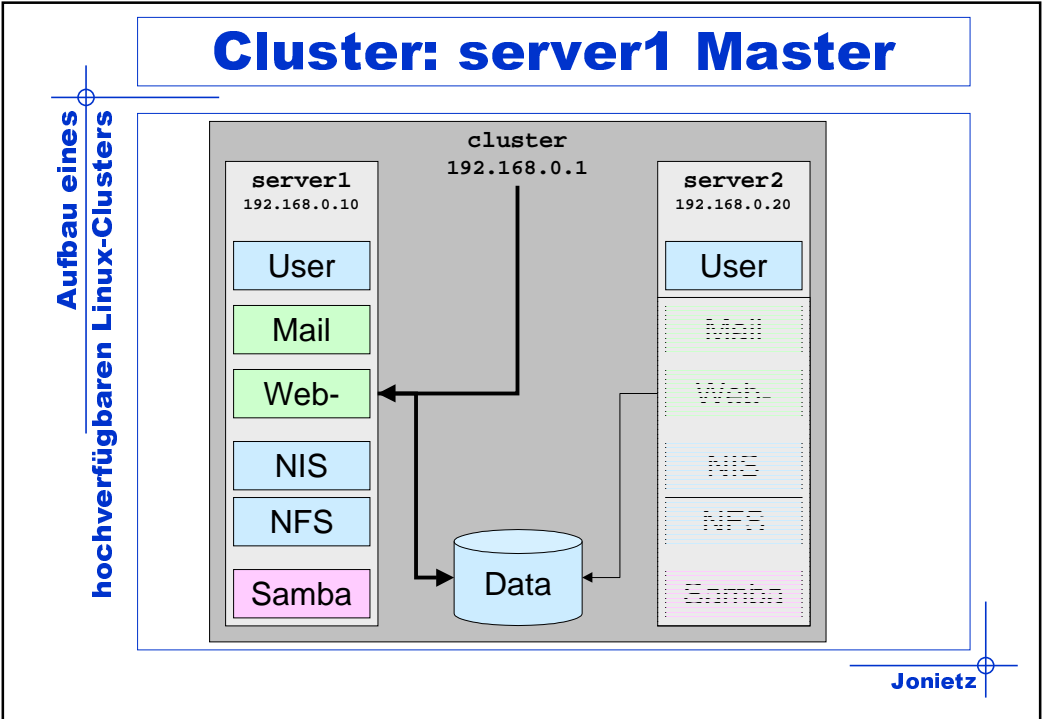
Schulinterne Fortbildung an der
BBS I Technik Kaiserslautern

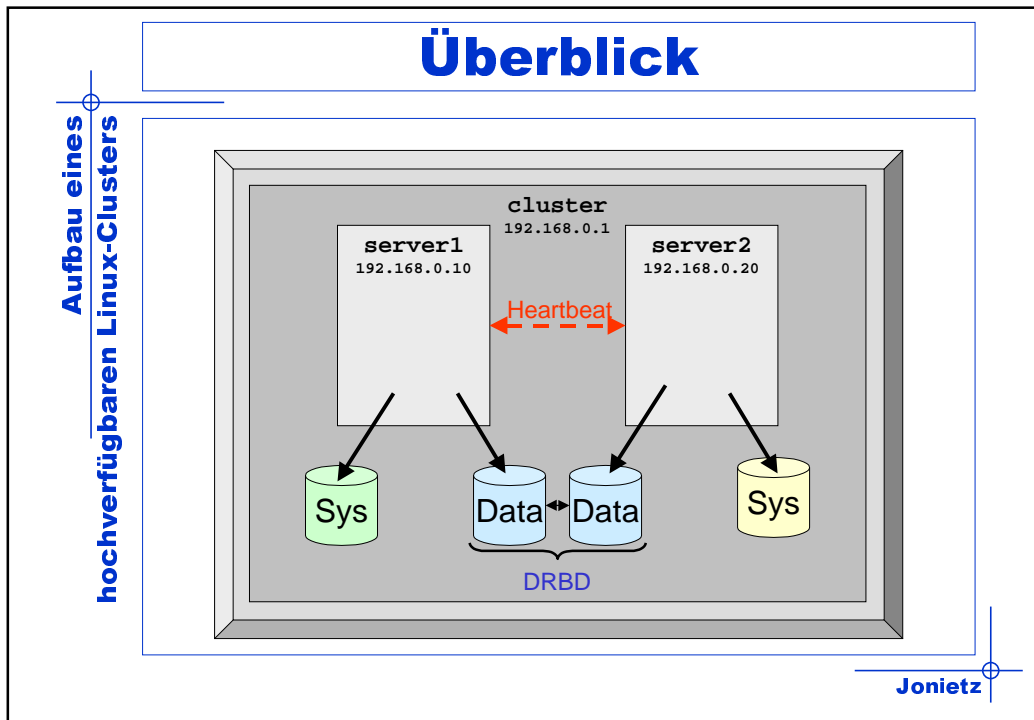
IFB 2002

Umfeld









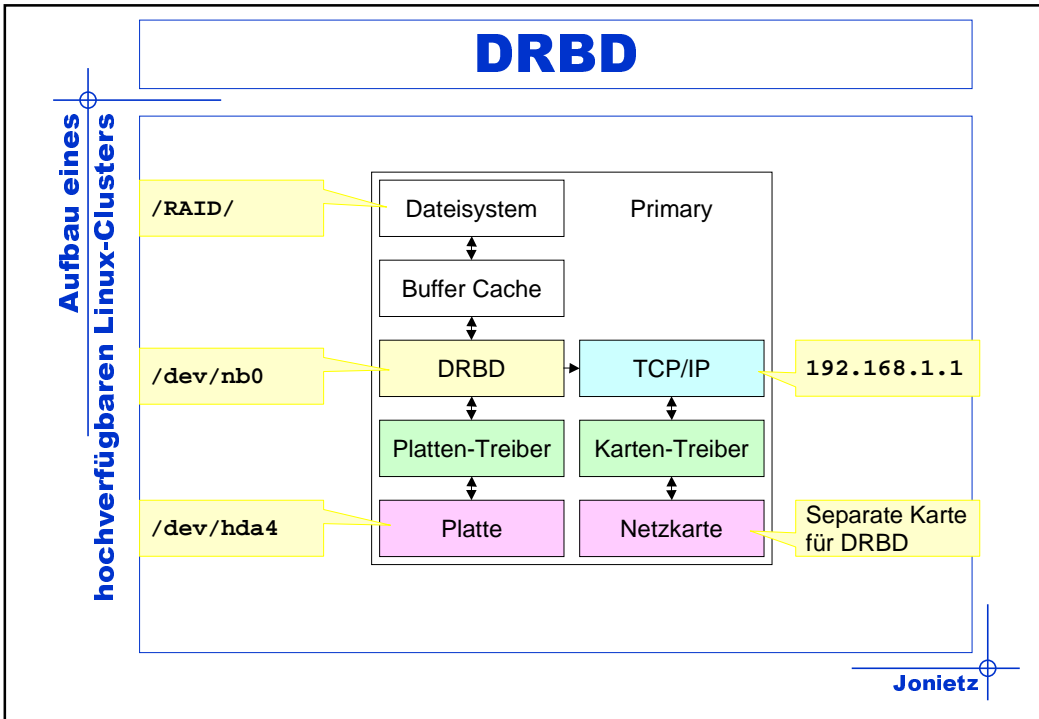
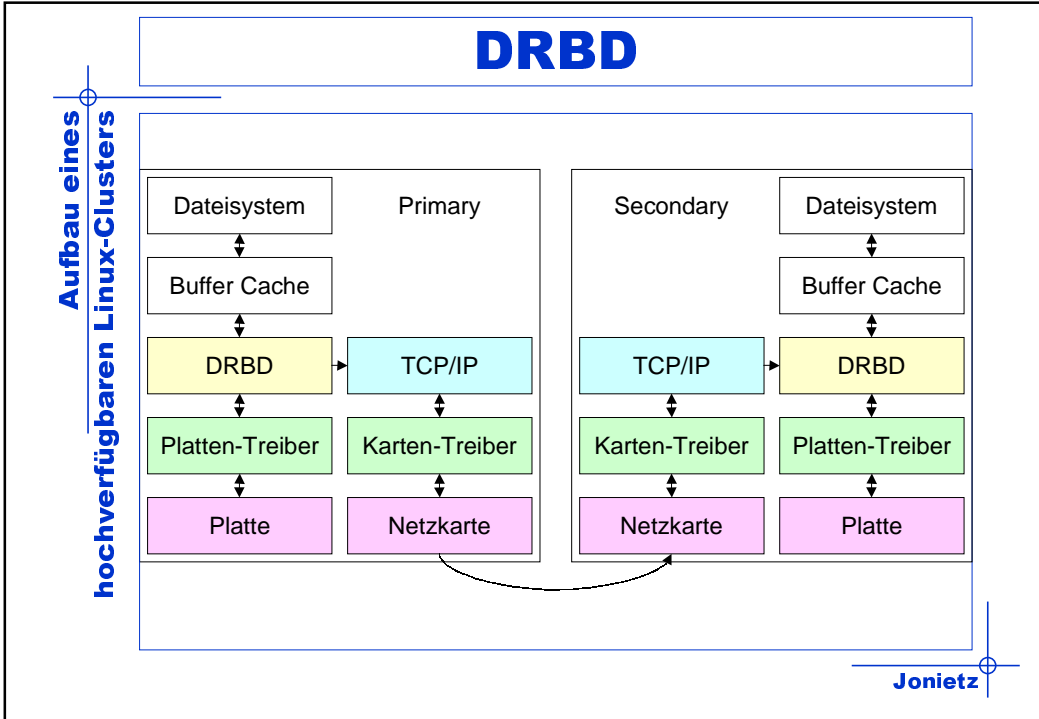
- ## Cluster
- Aufbau eines hochverfügbaren Linux-Clusters
- ▶ Der Cluster besteht aus zwei Rechnern
 - die sich prinzipiell unterscheiden können
 - jeweils vollständig konfiguriert sind (eigenes System)
 - ▶ Daten
 - liegen auf jedem Rechner und werden durch eine Art rechnerübergreifendes RAID synchron gehalten
 - ▶ Konfigurationsdateien
 - werden manuell / automatisch abgeglichen
- Jonietz

Überwachung

- ▶ Es handelt sich um ein **Standby-Cluster**
 - ein Rechner ist aktiv (Master)
 - die anderen Rechner überwachen den Master durch Versand von **Heartbeats**
 - fällt der Master aus, so übernimmt ein Standby-Rechner den Cluster
- ▶ **Übernahme** bedeutet:
 - alle Dienste werden auf altem Master gestoppt (sofern möglich) und auf neuem Master gestartet
 - der neue Master übernimmt die Adresse des Clusters

DRBD

- ▶ **Distributed Replicated Block Device**
- ▶ im Prinzip ein Geräte-Treiber
- ▶ Art rechnerübergreifendes Software-Raid mit Semantik eines Shared Device
- ▶ läuft auf Standard-Hardware
- ▶ Primary-Knoten Lese/Schreib-Zugriff
- ▶ Secondary-Knoten nur Lese-Zugriff
 - Aber (bei uns) problematisch...



DRBD - Protokolle

- ▶ Schreiboperation wird als beendet angesehen, sobald:
 - ▶ Protokoll A
 - an anderen Rechner losgeschickt wurden
 - ▶ Protokoll B
 - Bestätigung über Ankunft auf anderem Rechner erhalten wird
 - ▶ Protokoll C
 - Bestätigung über erfolgreiches Schreiben auf anderem Rechner erhalten wird

DRBD - Konfiguration

```

▶ resource drbd0 {
    protocol=B
    fsckcmd=fsck -p -y

    disk {
    }

    net {
        sync-rate=100000           maximale Bandbreite
        tl-size=8000               Transfer-Log-Größe
        timeout=60                 Wann wird abgebrochen?
        connect-int=10             Wann wieder Verbindung aufnehmen?

        ping-int=10                Alle 10s keep-alive-Paket senden
    }

    # Konfiguration der Knoten ausgelassen
}
    
```

DRBD - Konfiguration

```
▶ on stan {  
    device=/dev/nb0           Welches Gerät soll erscheinen?  
    disk=/dev/hda4           Welches ist das untergeordnete Gerät?  
    address=192.168.1.1      IP des lokalen Netzinterfaces  
    port=7788  
}  
  
on olli {  
    device=/dev/nb0  
    disk=/dev/hda4  
    address=192.168.1.2  
    port=7788  
}
```

DRBD - Lasttest

```
▶ #!/bin/bash  
$zaehl  
$datei  
cd /RAID/tests  
while true; do  
    zaehl=500  
    datei=1  
    rm *  
    while [ $zaehl -le 5000000 ]; do  
        echo $datei  
        echo $zaehl  
        dd if=/dev/zero of=datei$datei bs=1k count=$zaehl  
        sleep 1  
        zaehl=`expr $zaehl \* 2`  
        datei=`expr $datei + 1`  
    done  
    sleep 30  
done
```

DRBD - Log

- ▶ DRBD kann auf der Konsole beobachtet werden:

```
#!/bin/bash
# show.sh
while true; do
    cat /proc/drbd > /dev/tty12
    sleep 2
done
```

Heartbeat

- ▶ Heartbeat sorgt für den eigentlichen Cluster-Charakter
 - eine Cluster-Adresse
 - zwei Cluster-Knoten, von denen immer einer aktiv (Master) ist und die Cluster-Adresse innehat
 - Knoten überwachen sich durch den Versand von Heartbeats, möglichst über mehrere Leitungen
 - Wird eine Übernahme notwendig, werden die Dienste auf dem abgebenden Rechner gestoppt, die Cluster-Adresse freigegeben, die Dienste auf dem anderen Rechner gestartet und die Cluster-Adresse übernommen.

Heartbeat - Konfiguration

▶ 3 Dateien:

- **ha.cf** Hauptkonfiguration

```
logfile /var/log/ha.log
udp eth0      udp-heartbeats über eth0
udp eth1      udp-heartbeats über eth1
keepalive 2   alle 2 Sekunden ein Heartbeat
deadtime 10  10s kein HB dann Knoten tot
udpport 694   verwende Port 694
node server1  server1 ist ein Knoten
node server2  server2 ist auch ein Knoten
```
- **haresources** Ressourcen (Dienste)

```
server1 192.168.0.1 datadisk::drbd0 quota apache smb
```
- **authkeys** Authentikation

```
auth 1
1 crc
```

HA - Ressourcen

- ▶ Beim Start des Masters werden die Dienste in einer festzulegenden Reihenfolge gestartet, die über die Cluster-Adresse verfügbar sein sollen:

```
server1 192.168.0.1 datadisk::drbd0 quota apache smb
```

- Erst **datadisk** mit dem Parameter **drbd0**, damit auf die Platte zugegriffen werden kann
- dann **quota** zur aktivierung der Benutzerquotas
- dann **Apache** und **Samba**
- ▶ Beim Stoppen ist die Reihenfolge genau umgekehrt.
- ▶ Alle diese Aufrufe rufen Skripte in **resource.d/**

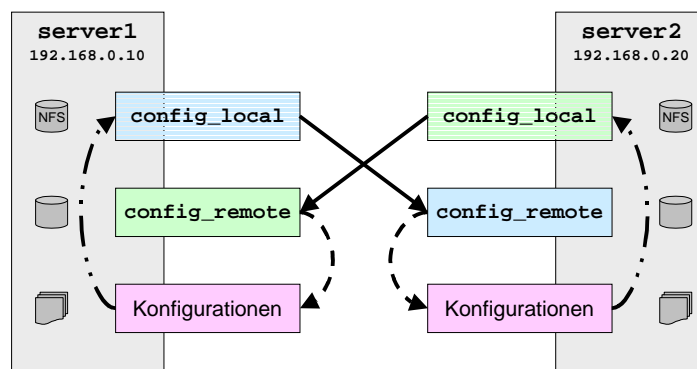
Dienst-Skripte

▶ Beispiel-Skript: Quota-Dienst

```
#!/bin/bash
/usr/sbin/rcquota $1
echo "quota rueckgabe:" $? > /dev/tty11
/usr/sbin/rpc.rquotad
echo "rpc.rquotad rueckgabe:" $? > /dev/tty11
```

- ▶ Startet bzw. stoppt Quota über weitere Skripte, gibt Informationen auf eine Konsole

Konfigurationen



- NFS-Schreiben (instantan)
- - → ucopy (periodisch / bei Übernahme)
- · → pcopy (periodisch / bei Änderungen)

Distribution der Konfs.

▶ ucopy

- Master vergleicht aktuelle Konfigurationen mit denen in `config_remote`, sucht neueste und verwendet diese
- Aufruf: periodisch, bzw. spätestens wenn der Rechner Master wird

▶ pcopy

- sucht die einzelnen Konfigurationsdateien, legt Archiv an und kopiert beides nach `config_local` (damit landet es - sofern NFS verfügbar - auf dem anderen Rechner) und nach `/etc`
- Aufruf: periodisch bzw. nach Änderungen

NFS

- ▶ Per NFS werden die Verzeichnisse zum Abgleich der Konfigurationen verteilt. Freigaben in `/etc/exports`:

```
/tmp 192.168.0.0/24(rw,root_squash)
```

Auf `/tmp` dürfen alle Rechner aus dem Netz zugreifen, lesen und schreiben, allerdings verliert root seine Privilegien.

```
/config_remote
```

```
192.168.0.10(rw,no_root_squash,sync)
```

Auf `/config_remote` darf der andere Rechner lesend und schreiben zugreifen, root behält seine Privilegien, die Schreibzugriffe werden gleich geschrieben.

NFS

- ▶ Auf dem jeweils anderen Rechner wird das Verzeichnis importiert:

- Hier: `/etc/fstab` von `server2`.
Auf `server1` entsprechend vertauscht.

```
server1:/config_remote /config_local nfs defaults 0 0
```

- Hänge das Verzeichnis `/config_remote` des Rechners `server1` in den Mountpoint `/config_local` ein.