HA - High Availability

- Corosync und Pacemaker
- ipvsadm
- Loadbalancing mit keepalived
- sysctl für Realserver hinter IPVS-DR
- Windows Server in LVS/DR und LVS/TUN Clustern

Corosync und Pacemaker

Hier werden Schnipsel gesammelt für eine Corosync + Pacemaker Anleitung.

Corosync Konfiguration

/etc/corosync/corosync.conf

```
# Please read the openais.conf.5 manual page

totem {
 version: 2

 # How long before declaring a token lost (ms)
 token: 3000

 # How many token retransmits before forming a new configuration
 token_retransmits_before_loss_const: 10

 # How long to wait for join messages in the membership protocol (ms)
 join: 60

 # How long to wait for consensus to be achieved before starting a new round of membership
 consensus: 1200

 # Turn off the virtual synchrony filter
 vsftype: none

 # Number of messages that may be sent by one processor on receipt of the token
 max_messages: 20

 # Limit generated nodeids to 31-bits (positive signed integers)
 clear_node_high_bit: yes

 # Disable encryption
  secauth: off

 # How many threads to use for encryption/decryption
  threads: 0

 # Optionally assign a fixed node id (integer)
 nodeid: 1234

 # This specifies the mode of redundant ring, which may be none, active, or passive.
  rrp_mode: passive

  interface {

  ringnumber: 0

  member {
   memberaddr: 10.10.222.1
  }

  member {
   memberaddr: 10.10.222.2

  }

  bindnetaddr: 10.10.222.0
  mcastport: 5405
 }

  interface {

  ringnumber: 1

  member {
   memberaddr: 10.10.108.127
```

```
  }

  member {
   memberaddr: 10.10.108.128

  }

  bindnetaddr: 10.10.108.0
  mcastport: 5406
 }

 # Use UDP unicast
 transport: udpu

}

amf {
 mode: disabled
}

service {
  # Load the Pacemaker Cluster Resource Manager
  ver:       0
  name:      pacemaker
}

aisexec {
        user:   root
        group:  root
}

logging {
        fileline: off
        to_stderr: yes
        to_logfile: yes
 logfile: /var/log/corosync/corosync.log
        to_syslog: no
 syslog_facility: daemon
        debug: off
        timestamp: on
        logger_subsys {
                subsys: AMF
                debug: off
                tags: enter|leave|trace1|trace2|trace3|trace4|trace6
        }
}
```

MySQL-HA mit DRBD

```
node dbserver1-ha1.meinserver.de
node dbserver1-ha2.meinserver.de
primitive drbd_mysql ocf:linbit:drbd \
 params drbd_resource="mysql" \
 op monitor interval="15s" role="Master" \
 op monitor interval="17s" role="Slave"
primitive fs_mysql ocf:heartbeat:Filesystem \
 params device="/dev/drbd/by-res/mysql" directory="/var/lib/mysql" fstype="xfs"
primitive ip_memcached ocf:heartbeat:IPaddr2 \
 params ip="10.10.108.136" nic="eth1" \
 op monitor interval="10s" \
 meta is-managed="true"
primitive ip_mysql ocf:heartbeat:IPaddr2 \
 params ip="10.10.108.129" nic="eth1" \
 op monitor interval="10s"
primitive memcached lsb:memcached \
 meta is-managed="true" \
 op monitor on-fail="restart" interval="20s" \
 op start interval="0" timeout="5s" \
 op stop interval="0" timeout="5s" \
 meta target-role="Started"
primitive mysqld ocf:heartbeat:mysql \
 params binary="/usr/bin/mysqld_safe" client_binary="/usr/bin/mysql" config="/etc/mysql/my.cnf"
 op monitor interval="20s" timeout="10s" \
 op start interval="0" timeout="240" \
 op stop interval="0" timeout="240" \
```

```
 meta migration-threshold="10"
primitive ping_intgw ocf:pacemaker:ping \
 params host_list="10.10.110.248 10.10.110.249" multiplier="1000" dampen="30" \
 op monitor interval="10s"
group memcachedgroup ip_memcached memcached \
 meta target-role="Started"
group mysqlgroup fs_mysql ip_mysql mysqld \
 meta target-role="Started" is-managed="true"
ms ms_drbd_mysql drbd_mysql \
 meta master-max="1" master-node-max="1" clone-max="2" clone-node-max="1" notify="true" target-r
clone ping_intgw_clone ping_intgw \
 meta globally-unique="false"
location memcachedgroup_on_connected_node memcachedgroup \
 rule $id="memcachedgroup_on_connected_node-rule" -inf: not_defined pingd or pingd lte 0
location mysqlgroup_on_connected_node mysqlgroup \
 rule $id="mysqlgroup_on_connected_node-rule" -inf: not_defined pingd or pingd lte 0
colocation mysqlgroup_on_drbd inf: mysqlgroup ms_drbd_mysql:Master
order memcached_after_ip_memcached inf: ip_memcached memcached
order mysqlgroup_after_drbd inf: ms_drbd_mysql:promote mysqlgroup:start
property $id="cib-bootstrap-options" \
 dc-version="1.1.7-ee0730e13d124c3d58f00016c3376a1de5323cff" \
 cluster-infrastructure="openais" \
 expected-quorum-votes="2" \
 stonith-enabled="false" \
 last-lrm-refresh="1355412261" \
 no-quorum-policy="ignore"
rsc_defaults $id="rsc-options" \
 resource-stickiness="100"
```

MySQL-HA + MongoDB-HA + Memcached-HA mit DRBD

```
primitive drbd_mongodb ocf:linbit:drbd \
 params drbd_resource="mongodb0" \
 op monitor interval="29s" role="Master" \
 op monitor interval="31s" role="Slave"

primitive drbd_mysql ocf:linbit:drbd \
 params drbd_resource="mysql0" \
 op monitor interval="29s" role="Master" \
 op monitor interval="31s" role="Slave"

primitive fs_mongodb ocf:heartbeat:Filesystem \
 params device="/dev/drbd1" directory="/var/lib/mongodb" fstype="xfs"

primitive fs_mysql ocf:heartbeat:Filesystem \
 params device="/dev/drbd0" directory="/var/lib/mysql" fstype="xfs"

primitive ip_memcached ocf:heartbeat:IPaddr2 \
 params ip="10.10.149.11" cidr_netmask="32" nic="eth0" \
 op monitor interval="10s" \
 meta is-managed="true"

primitive ip_mongodb ocf:heartbeat:IPaddr2 \
 params ip="10.10.149.16" cidr_netmask="32" nic="eth0" \
 op monitor interval="10s" \
 meta is-managed="true"

primitive ip_mysql ocf:heartbeat:IPaddr2 \
 params ip="10.10.149.8" cidr_netmask="32" nic="eth0" \
 op monitor interval="10s" \
 meta is-managed="true"

primitive memcached lsb:memcached \
 meta is-managed="true" \
 op monitor on-fail="restart" interval="20s" \
 op start interval="0" timeout="15s" \
 op stop interval="0" timeout="15s"

primitive mongodb lsb:mongod \
 meta target-role="Started"

primitive mysql lsb:mysql \
 meta target-role="Started"

primitive ping_services1 ocf:pacemaker:ping \
 params host_list="10.10.148.26 10.10.148.49" multiplier="1000" dampen="30" \
```

```
 op monitor interval="10s"

group memcachedgroup ip_memcached memcached \
 meta target-role="Started"

group mongodbgroup fs_mongodb ip_mongodb mongodb \
 meta target-role="Started"

group mysqlgroup fs_mysql ip_mysql mysql \
 meta target-role="Started"

ms ms_drbd_mongodb drbd_mongodb \
 meta master-max="1" master-node-max="1" clone-max="2" clone-node-max="1" notify="true"

ms ms_drbd_mysql drbd_mysql \
 meta master-max="1" master-node-max="1" clone-max="2" clone-node-max="1" notify="true"

clone ping_services1_clone ping_services1 \
 meta globally-unique="false"

location memcachedgroup_on_connected_node memcachedgroup \
 rule $id="memcachedgroup_on_connected_node-rule" -inf: not_defined pingd or pingd lte 0

location mongodbgroup_on_connected_node mongodbgroup \
 rule $id="mongodbgroup_on_connected_node-rule" -inf: not_defined pingd or pingd lte 0

location mysqlgroup_on_connected_node mysqlgroup \
 rule $id="mysqlgroup_on_connected_node-rule" -inf: not_defined pingd or pingd lte 0

colocation mongodb_on_drbd inf: mongodbgroup ms_drbd_mongodb:Master
colocation mysql_on_drbd inf: mysqlgroup ms_drbd_mysql:Master

order memcached_after_ip_memcached inf: ip_memcached memcached
order mongodb_after_drbd inf: ms_drbd_mongodb:promote mongodbgroup:start
order mysql_after_drbd inf: ms_drbd_mysql:promote mysqlgroup:start

property $id="cib-bootstrap-options" \
 dc-version="1.1.7-ee0730e13d124c3d58f00016c3376a1de5323cff" \
 cluster-infrastructure="openais" \
 expected-quorum-votes="2" \
 stonith-enabled="false" \
 no-quorum-policy="ignore" \
 last-lrm-refresh="1434463669"

rsc_defaults $id="rsc-options" \
 resource-stickiness="100"
```

ipvsadm

ipvsadm ist ein Tool zur Steuerung des Linux-Virtual-Server Moduls

Realserver aus Loadbalancing-Verbund löschen:

```
ipvsadm -d -t 212.34.176.157/32 -r 212.34.172.54
```

Loadbalancing mit keepalived

Hier eine einfach Config-Datei /etc/keepalived/keepalived.conf mit einem externen Interface eth0, auf das die virtuelle (secondary) IP 123.123.123.100 gelegt werden soll. Auf eth1 ist ein privates Netz konfiguriert, in dem die Realserver hängen. Es werden 2 virtuelle Server konfiguriert, einer für Port 80 HTTP und ein weiteres für Port 443 HTTPS Verbindungen:

```
global_defs {
   lvs_id meinLoadbalancer
}

vrrp_sync_group G1 {
  group {
        VI_1
  }
}

vrrp_instance VI_1 {
    state MASTER
    interface eth0
    virtual_router_id 51
    priority 150
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass ganzSicheresPasswort
    }
    virtual_ipaddress {
      123.123.123.100
    }
}

virtual_server 123.123.123.100 80 {
    delay_loop 6
    lb_algo wlc
    lb_kind NAT
    persistence_timeout 50
    protocol TCP

    real_server 192.168.1.20 80 {
        weight 1
        HTTP_GET {
            url {
              path /
            }
            connect_timeout 3
            nb_get_retry 3
            delay_before_retry 3
        }
    }

    real_server 192.168.1.21 80 {
        weight 1
        HTTP_GET {
            url {
              path /
            }
            connect_timeout 3
            nb_get_retry 3
            delay_before_retry 3
        }
    }
}

virtual_server 123.123.123.100 443 {
    delay_loop 6
    lb_algo wlc
    lb_kind DR
    persistence_timeout 50
    protocol TCP

    real_server 192.168.1.20 443 {
```

```
        weight 1
        SSL_GET {
            url {
              path /
            }
            connect_timeout 3
            nb_get_retry 3
            delay_before_retry 3
        }
    }

    real_server 192.168.1.21 443 {
        weight 1
        SSL_GET {
            url {
              path /
            }
            connect_timeout 3
            nb_get_retry 3
            delay_before_retry 3
        }
    }
}
```

sysctl für Realserver hinter IPVS-DR

Die `/etc/sysctl.conf` für Realserver hinter einem IPVS mit DirectRouting sollte so aussehen:

```
# Loadbalancer
net.ipv4.conf.all.arp_ignore = 1
net.ipv4.conf.eth0.arp_ignore = 1
net.ipv4.conf.all.arp_announce = 2
net.ipv4.conf.eth0.arp_announce = 2
#net.ipv4.tcp_syncookies = 1
net.core.somaxconn = 1024
```

Damit wird verhindert, dass ARP-Announcements für die Loadbalancer-IP verschickt werden (diese muss auf dem Loopback-Device konfiguriert werden).

Windows Server in LVS/DR und LVS/TUN Clustern

Wie bei Linux-Servern muss auch bei Windows die sog. Float-IP auf einem Loopback-Interface konfiguriert und ARP deaktiviert werden, wenn man Loadbalancing mit Direct-Routing oder Tunneling betreiben will.

Für diese Zwecke muss man den Microsoft Loopback Adapter installieren(z.B. bei Windows Server 2003):

1. Start ? Systemsteuerung ? Hardware ? Weiter
2. „Wurde diese Hardware bereits an den Computer angeschlossen?“ ? Ja
3. Ganz unten in der Liste „Neue Hardware hinzufügen“ auswählen ? Weiter
4. „Hardware manuell aus einer Liste wählen und installieren“ ? Weiter
5. Netzwerkadapter ? Weiter
6. Hersteller: Microsoft, Netzwerkdapter: Microsoft Loopbackadapter ? Weiter
7. Weiter

Jetzt kann man diesem Adapter wie jeder anderen Netzwerkkarte unter Windows eine IP-Adresse geben. Bei LVS sollte dies die VIP (virtuelle IP) sein, die man auch auf dem Loadbalancer konfiguriert hat.

Diese IP sollte dann idealerweise die Netzmaske 255.255.255.255 haben, um Routing-Probleme zu vermeiden (falls der Server etwa einen Server aus dem konfigurierten Netz erreichen will, könnte es sein, dass er die über das Loopback-Interface versucht). Leider läßt sich diese Netzmaske nicht via Netzwerk-Einrichtung einstellen. Hier muss mal wieder der Windows-Registry Editor bemüht werden:

In der Registry versteckt sich die Einstellung der Netzmaske hier (Windows Server 2003):

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{loopback devic
```

Die SubnetMask ist ein Hexcode, rechts im Dialogfeld können allerdings auch ASCII-Werte eingegeben werden.

Nach einem Reboot sind die neuen Einstellungen aktiv. Diese Einstellung scheint ARP zu deaktivieren und beeinflusst nicht die Routingtabelle.