

Creazione script per gestione 3G in failover

Questo documento tenta di descrivere la procedura per creare uno script bash con il quale decidere come gestire l'attivazione o meno di un'interfaccia dati GSM/UMTS.

Premessa

Testato su alix 2d13 con modem umts e220

Obiettivo: far sì che sul net balance siano presenti minimo 2 gw attivi e funzionanti. In caso di down di una eth o di un gw , attivazione e test dell'interfaccia UMTS. Non appena le ETH ed i gw tornano UP , spegnimento interfaccia UMTS. Tutto loggato nel file checkP.

File da includere:

3g.cfg

All'interno di questo file inserire

```
#!/bin/bash
link=3
eth=2
minactive=2
maxactive=3
down2=1
IP1=`cat /var/register/system/net/nb/FO/IP1`
```

link=3 , identifica il numero di interfacce che sono usate nella nostra config dello ZS . Nel nostro esempio 2 ETH e 1 PPP0 (umts) =3

eth=2 , numero delle porte eth configurate

minactive=2 , numero minimo di GW/ETH attivi

maxactive=3 , numero max di ETH/GW attivi

down2=1 , variabile che identifica l'interfaccia UMTS accesa (=1 -> spenta)

IP1=, estrae l'ip1 di test del failover che precedentemente avete configurato nel vostro ZS, verrà usato nei test ICMP

no3g.cfg

inserire solo il numero 0 (viene fatto dallo script di post boot). Verrà usato dallo script successivamente come variabile.

Lo script:

```
#!/bin/bash
start=0
if [ -f 3g.txt ]; then
    logger -t checkP "#### check3G is running - exit"
    exit
else
    start=1
    logger -t checkP "#### check3G is starting ....."
fi
while [ $start -eq 1 ];
do
    . 3g.cfg
    . no3g.cfg
    sleep 15
    R=$RANDOM
    logger -t checkP "***$R** checkP STARTED"
    echo start > 3g.txt
    limit=`expr $eth - 1`

    for i in $(seq 0 $limit); do
        status=ping1`ping $IP1 -I ETH0$i -c 1 |grep From |sed 's/^.*=1 Destination Host //'`
        logger -t checkP "***$R** ETH0$i ---- $status"
        if [ $status == "ping1Unreachable" ]; then
            logger -t checkP "***$R** ETH$i down"
            link=`expr $link - 1`
            logger -t checkP "***$R** i=$i , down=down$i, link=$link"
        fi
    done
    PPX=pp0`ifconfig |grep pp0 |sed 's/^.*Point-to-Point //'`
    if [ $PPX == "pp0" ]; then
        link=`expr $link - 1`
        logger -t checkP "***$R** ppp0 Down"
        down2=1
    else
        link=$link
        down2=0
    fi
    logger -t checkP "***$R** link=$link , minactive=$minactive,down2=$down2"
    while [ $link -lt $minactive ];
    do
        count=0
        if [ $down2 -eq 1 ]; then
```

```

while [ $count -lt 3 ];
do
logger -t checkP "***$R** Starting link GSM/UMTS ....."
/root/kerbynet.cgi/scripts/net_updown IF,ppp0 true
sleep 10
logger -t checkP "***$R** Testing link GSM/UMTS ....."
logger -t checkP "***$R** count=$count"
PPX=pp0`ifconfig |grep pp0 |sed 's/^.*Point-to-Point //'`
logger -t checkP "***$R** PPX=$PPX"
if [ $PPX == "pp0" ]; then
count=`expr $count + 1`
else
count=50
logger -t checkP "***$R** Link GSM/UMTS Started - count=$count"
/root/kerbynet.cgi/scripts/net_updown IF,ETH00 false
logger -t checkP "***$R** NAT restarting - ETH00 down"
sleep 2
/root/kerbynet.cgi/scripts/net_updown IF,ETH00 true
logger -t checkP "***$R** NAT restarted - ETH00 Up"
no3g=0
echo $no3g > no3g.cfg
down2=0
fi
done
if [ $count -eq 3 ]; then
no3g=`expr $no3g + 1`
echo $no3g > no3g.cfg
if [ $no3g -lt 3 ]; then
./sendEmail -f zs@domain.you -t you@domain.you -m Allarme critico
interfaccia UMTS/GSM -u Interfaccia UMTS/GSM non risponde -s smtp.domain.you -o
username=you@domain.you -o password=xyzxyz
logger -t checkP "***$R** Link GSM/UMTS not available - $no3g"
/root/kerbynet.cgi/scripts/net_updown IF,ppp0 false
logger -t checkP "***$R** GSM/UMTS disable - $no3g"
down2=1
fi
if [ $no3g -gt 2 ]; then
logger -t checkP "***$R** Link GSM/UMTS not available - $no3g"
/root/kerbynet.cgi/scripts/net_updown IF,ppp0 false
logger -t checkP "***$R** GSM/UMTS disable - $no3g"
down2=1
fi
fi
link=$minactive
fi
link=$minactive
done

while [ $down2 -eq 0 ];
do

```

```

ethup=0
    logger -t checkP "***$R** ## Check ETH down2=$down2 link=$link"
    for i in $(seq 0 $limit); do
        status=ping1`ping $IP1 -I ETH0$i -c 1 |grep From |sed 's/^.*=1 Destination Host //'`
        if [ $status == "ping1" ]; then
            ethup=`expr $ethup + 1`
            logger -t checkP "***$R** ETHUP=$ethup"
        fi
        if [ $ethup -eq $eth ]; then
            /root/kerbynet.cgi/scripts/net_updown IF,ppp0 false
            logger -t checkP "***$R** Link GSM/UMTS Closed - min ETH now UP"
            ./sendEmail -f zs@domain.you -t you@domain.you -u Ripristino min ETH -m
Minimo numero di interfacce ETH ripristinate. Collegamento UMTS/GSM Chiuso -s
smtp.domain.you -o username=you@domain.you -o password=xyzxyz
        fi
    done
down2=1
done

while [ $link -eq $maxactive ];
do
    logger -t checkP "***$R** Check ETH down2=$down2 link=$link"
    for i in $(seq 0 $limit); do
        status=ping1`ping $IP1 -I ETH0$i -c 1 |grep From |sed 's/^.*=1 Destination Host //'`
        logger -t checkP "***$R** ETH0$i $status"
        if [ $status == "ping1" ]; then
            check=`expr $i + 1`
            logger -t checkP "***$R** check=$check"
            if [ $check -eq $minactive ]; then
                /root/kerbynet.cgi/scripts/net_updown IF,ppp0 false
                logger -t checkP "***$R** Link GSM/UMTS Closed"
                check=0
            fi
        fi
    done
    #!!!down2=1
done
down2=1
link=$check
done
CMD="rm 3g.txt -f"
$CMD
logger -t checkP "***$R** checkP ENDED"
#start=0
done
exit

```

Descrizione:

In pratica lo script controlla che le eth siano attive facendo un ping verso \$IP1 (ip1 da voi configurato nella finestra test failover del net balancer → "Failover Ip Adresses" → IP (1)) . Nel caso una delle due sia down e minactive sia impostato a 2 , accende l'interfaccia umts. Successivamente ogni 15 secondi controlla che la eth down non sia tornata UP, in quel caso chiude l'interfaccia Umts. Se l'umts non va up , tenta 3 volte ogni 10 secondi di riavviarla. Nel caso rimanga down la disabilita e manda una mail all'amministratore. Ritenta lo script ogni 15 secondi ma manderà la mail di alert all'amministratore solo una seconda volta, le successive si limita ad aggiornare con l'errore il file di log.

Attivazione script :

creare attraverso l'interfaccia ZS un post boot inserendo:

```
# Startup Script
cd /Database/scripts
rm -f 3g.txt
echo 0 > no3g.cfg
./check3g &
```

Lo script è in loop ogni 15 secondi.

NOTE:

In caso di attivazione di check3g nel post boot , non sarà più possibile attivare manualmente l'interfaccia UMTS se non modificando ad hoc i parametri minactive nel file 3g.cfg.

Nel caso non venga fatto, ogni volta che lo script trova l'umts accesa con altri 2 gw/eth funzionanti, la chiude . Questo script è fatto per usare l'interfaccia umts come vero backup, senza la necessità di dover avere piani tariffari flat dati. Infatti l'umts viene usato esclusivamente come backup di una delle due eth.

Da notare che mettendo minactive=1 , l'umts si accenderà solo se tutte e due le eth risulteranno down, minactive=3 terrà sempre l'umts accesa.

Per la spedizione delle mail di alert ho usato lo script sendEmail scaricabile dal link:

<http://caspian.dotconf.net/menu/Software/SendEmail/sendEmail-v1.56.tar.gz>

Molto semplice da usare . Le istruzioni per l'installazione le trovate al link

<http://caspian.dotconf.net/menu/Software/SendEmail/>

Mirko Mare