



This kind of **Pc-Freak.net downtime** issues over the last month become too frequent (the machine was down about 5 times for 2 to 5 hours and this was too much (and weirdly enough it was not accessible from the internet even after electricity network was restored and the only solution to that was a **physical server restart (from the Power Button)**).

To decrease the number of cases in which known relatives or friends has to physically go to the server and restart it, **each time after network or electricity outage I wrote a small script to check accessibility towards Default defined Network Gateway for my server with few ICMP packages sent with good old PING command and trigger a network restart and system reboot** (in case if the network restart does fail) in a row.

1. Create reboot-if-network-is-downsh script under /usr/sbin or other dir

Here is the script itself:

```
#!/bin/sh
# Script checks with ping 5 ICMP pings 10 times to DEF GW and if so
# triggers networking restart /etc/init.d/networking restart
# Then does another 5 x 10 PINGS and if ping command returns errors,
# Reboots machine
# This script is useful if you run home router with Linux and you have
# electricity outages and machine doesn't go up if not rebooted in that case

GATEWAY_HOST='192.168.0.1';

run_ping () {
for i in $(seq 1 10); do
ping -c 5 $GATEWAY_HOST
done
}

reboot_f () {
if [ $? -eq 0 ]; then
echo "$(date +%Y-%m-%d %H:%M:%S") Ping to $GATEWAY_HOST OK" >>
/var/log/reboot.log
else
/etc/init.d/networking restart
echo "$(date +%Y-%m-%d %H:%M:%S") Restarted Network Interfaces:" >>
/tmp/rebooted.txt
for i in $(seq 1 10); do ping -c 5 $GATEWAY_HOST; done
if [ $? -eq 0 ] && [ $(cat /tmp/rebooted.txt) -lt '5' ]; then
```

```
echo "$(date "+%Y-%m-%d %H:%M:%S") Ping to $GATEWAY_HOST FAILED  
!!! REBOOTING." >> /var/log/reboot.log  
/sbin/reboot
```

```
# increment 5 times until stop  
[[ -f /tmp/rebooted.txt ]] || echo 0 > /tmp/rebooted.txt  
n=$(( echo $(( n + 1 )) > /tmp/rebooted.txt  
fi  
# if 5 times rebooted sleep 30 mins and reset counter  
if [ $(cat /tmp/rebooted.txt) -eq '5' ]; then  
sleep 1800  
cat /dev/null > /tmp/rebooted.txt  
fi  
fi  
  
}  
run_ping;  
reboot_f;
```

You can [download a copy of reboot-if-nwork-is-down.sh script here](#).

As you see in script successful runs as well as its failures are logged on server in `/var/log/reboot.log` with respective **timestamp**.

**Also a counter to 5 is kept in /tmp/rebooted.txt, incremented on each and every script run (rebooting) if, the 5 times increment is matched**

a sleep is executed for 30 minutes and the counter is being restarted.

The counter check to 5 guarantees the server will not get restarted if access to Gateway is not continuing for a long time to prevent the system is not being restarted like crazy all time.

2. Create a cron job to run `reboot-if-nwork-is-down.sh` every 15 minutes or so

I've set the script to re-run in a **scheduled (root user) cron job** every **15 minutes** with following job:

To add the script to the existing **cron rules** without rewriting my old cron jobs and without tempering to use **crontab -u root -e** (e.g. **do the cron job add in a non-interactive mode with a single bash script one liner** had to run following command:

```
{ crontab -l; echo "*/15 * * * * /usr/sbin/reboot-if-nwork-is-down.sh 2>&1 >/dev/null; } |  
crontab -
```

I know restarting a server to restore accessibility is a stupid practice but for home-use or small client servers with unguaranteed networks with a cheap **Uninterruptable Power Supply (UPS) devices** it is useful.

Summary

Time will show how efficient such a "self-healing script practice is.  
Even though I'm pretty sure that even in a Corporate businesses and large **Public / Private Hybrid Clouds** where access to remote mounted **NFS / XFS / ZFS filesystems** are failing a modifications of the script could save you a lot of nerves and troubles and unhappy customers / managers screaming at you on the phone :)

**I'll be interested to hear from others who *have a better ideas to restore ( resurrect ) access to inessible Linux server after an outage.?***