

# Linux

## Failover-IP - Show malfunction page in case of server failure

This example assumes that a dedicated server and a virtual server are available. The dedicated server will contain all regular web pages, whether the web server is managed manually or via Plesk is irrelevant. The V-Server will contain the fault page, which will be displayed if the dedicated server fails. The V-Server is installed without Plesk, in this example the Debian operating system is used.

### Configuration of the domains

The A-records of the domain must be set to the failover IP. This is possible via the central customer management ( <https://zkm.webtropia.com> > DNS System > Edit Zone ).

### Configuration of the dedicated server

For the dedicated server, the failover IP must be configured in the system and in Plesk. If you do not use Plesk, make sure that the failover IP is configured on the web server you are using. Now set the failover IP on your dedicated server, this is possible via the Central Customer Management (CCM) > Dedicated Root Server > Failover IP Management.

### Configuration of the V-Server

On the V-Server, first install the Apache web server, which will deliver the fault page.

```
apt-get update && apt-get install apache2
```

The script also requires the bc package.

```
apt-get install bc
```

Connect to the V-Server via SSH ( or SCP ) and store the error page ( index.html ) under /var/www. If you type the IP of the V-Server into the browser, you should already see the error page.

Next, you need to install a script that will switch the IP if the dedicated server fails. The script could look like this:

```
#!/bin/bash
PATH=/bin:/usr/bin
REMOTE_IP="192.168.100.101"
API_LINK="https://zkm.fastit.net/Api/Failover/FailoverAssign?job=example123"
while [ 1 ]; do
    IS_ONLINE=$(ping -W 10 -c 1 -n $REMOTE_IP | grep "64 bytes from $REMOTE_IP:" | wc -l | bc)
    if [ $IS_ONLINE = 0 ]; then
        IS_ONLINE_DEEP=$(ping -W 10 -c 1 -n $REMOTE_IP | grep "64 bytes from $REMOTE_IP:" | wc -l | bc)
        if [ $IS_ONLINE_DEEP = 0 ]; then
            echo "Could not reach remote host. Take over IP."
            echo "$(date) [failover] Could not reach remote host. Take over IP." >> /var/log/overtake-ip.log
            wget -O /dev/null "$API_LINK"
            # With a dedicated server, the IP would still have to be booted up here.
        fi
    fi
done
exit 0
```

The script runs in an endless loop. If a ping is lost, a second ping is sent, if this is also lost, the API is triggered and the script terminates. Make sure that the IP in REMOTE\_IP is the main IP of the dedicated server and not the failover IP.

It is best to save the script to /root/overtake-ip.sh , it is sufficient to start the script via the file

# Linux

/etc/rc.local. So enter the following line in the file:

```
/root/overtake-ip.sh &
```

The & at the end starts the process in the background. Be sure to make the entry before the line "exit 0".

## How to deal with a disturbance

If the case occurs that the dedicated server is no longer reachable, the failover IP - by the script - is brought up via the Api on the V-Server, thereby the fault page is visible via the failover IP. When the failure of the dedicated server is resolved, you can route the failover IP back to the dedicated server via the central customer management (ZKM) Dedicated Root Server > Failover IP Management ). After that, you need to restart the script on the V-Server, as it was terminated when the IP was taken over. The command is:

```
/root/overtake-ip.sh &
```

Unique solution ID: #1499

Author: Bettina Brauer

Last update: 2021-04-15 08:57