scp-on-xferlog.pl

Dieses Programm verabschiedet sich direkt nach dem Start in den Hintergrund (Dämon) und schaut dann auf Veränderungen im File /var/log/xferlog. Wird nun via FTP eine Datei hochgeladen, wird sie via SCP (passwortloses Login via Keys sollte vorher natürlich eingerichtet sein) auf die Server in @hosts kopiert. Dies ist z.B. nützlich wenn man mehrere Server in einem Loadbalancingcluster betreibt.

```
#!/usr/bin/perl
#
# 2008 Oliver Voelker <wiki(at)magenbrot.net>
#
# Dieses Script lauscht auf neue Eintraege in /var/log/xferlog und kopiert neu hochgeladene
Dateien
# mit einer Verzoegerung von max. etwa 1-5 Sekunden auf die konfigurierten Hosts.
#

use strict;
use warnings;
use File::Tail;
use File::Basename;
use POSIX qw(setsid);

my $debug = 1;                              # 1 = normal, 2 = extended logging
my $name = "/var/log/xferlog";              # watch this file
#my $logfile = "/dev/null";                 # logfile
my $logfile = "/var/log/scp-on-xferlog";    # logfile
my @hosts = ("host1.de", "host2.de", "host3.de"); # list of hosts

# don't edit anything below this line
my $line = "";

sub daemonize {
  chdir("/") or die("Can't chdir to /: $!");
  open(STDIN, "/dev/null") or die("Can't read /dev/null: $!");
  open(STDOUT, ">>$logfile") or die("Can't write to $logfile: $!");
  open(STDERR, ">>$logfile") or die("Can't write to $logfile: $!");
  defined(my $pid = fork) or die("Can't fork: $!");
  exit if($pid);
  setsid or die("Can't start a new session: $!");
  umask(0);
```

```
}

sub sharefile {
  my ($host,$file) = @_;
  my $try = 0;
  my $error = 1;
  until ($error eq "" || $try == 5) {
    print localtime(time) . " Trans:  $host" if($debug);
    system ("scp -1 \"$file\" root\@$host:$file >/dev/null 2>&1");
    if ($? != 0 && $try eq 0) {
      my $dirname = dirname($file);
      print " - Error:  Creating non-existent directory: $dirname\n" if ($debug >=2);
      system ("ssh -1 root\@$host \"mkdir -p $dirname\" >/dev/null 2>&1");
      system ("scp -1 \"$file\" root\@$host:$file >/dev/null 2>&1");
      $try++;
    } elsif ($? != 0 && $try >= 0) {
      $error=$error . " " . $host;
      print " - failed: $!\n" if ($debug);
      sleep 2;
      $try++;
    }
    else {
      print " - ok\n" if ($debug);
      $error="";
    }
  }
  return $error;
}

# flush the buffer
$| = 1;

# daemonize the program
&daemonize;

my $log = File::Tail->new(name => $name, maxinterval => 5, adjustafter => 10);
while (defined($line = $log->read)) {
  my (undef, undef, undef, undef, undef, undef, undef, undef, $file) = split(/ /, $line);
  my $count = 0;
  my $error = "1";
```

```
  until (-f $file || $count eq 5) {
    # we'll wait max 5*2 seconds for the transfer of the file to complete
    print localtime(time) . "Info:   File does not exist yet: $file\n";
    sleep 2;
    $count++;
  }

  my
($dev,$ino,$mode,$nlink,$uid,$gid,$rdev,$size,$atime,$mtime,$ctime,$blksize,$blocks)=stat($fil
e) if ($debug >= 2);
  print localtime(time) . "File:   $file Size: $size Access: " . scalar localtime($atime) . "
Modified: " . scalar localtime($mtime) . "\n" if ($debug >= 2);

  # copy files
  foreach (@hosts) {
    $error=sharefile($_,$file);
  }

  if ($error eq "") {
    print localtime(time) . " Shared: $file\n" if ($debug);
  } else {
    print localtime(time) . " Error:  $file was not copied to: $error\n";
  }
}
```

nützlich ist dann noch das Initscript, welches sich um den Start des Dämons beim Booten kümmert:

```
#!/bin/bash
#
# scp-on-xferlog        Starts scp-on-xferlog.
#
#
# chkconfig: 2345 12 88
# description: scp-on-xferlog is used to copy new transferred files (from xferlog) to other
configured hosts
#
### BEGIN INIT INFO
# Provides: $scp-on-xferlog
### END INIT INFO
```

```
# Source function library.
. /etc/init.d/functions

[ -f /usr/local/sbin/scp-on-xferlog.pl ] || exit 0

RETVAL=0

start() {
 echo -n $"Starting scp-on-xferlog: "
daemon /usr/local/sbin/scp-on-xferlog.pl
RETVAL=$?
echo
[ $RETVAL -eq 0 ] && touch /var/lock/subsys/scp-on-xferlog
return $RETVAL
}

stop() {
echo -n $"Shutting down scp-on-xferlog: "


killproc scp-on-xferlog.pl
echo
RETVAL=$?
[ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/scp-on-xferlog
return $RETVAL
}

rhstatus() {
status scp-on-xferlog.pl
}

restart() {
stop
start
}

case "$1" in
  start)
  start
```

```
;;
  stop)
  stop
;;
  status)
  rhstatus
;;
  restart)
  restart
;;
  *)
echo $"Usage: $0 {start|stop|status|restart}"
exit 1
esac

exit $?
```

und die Datei für's Logrotate:

```
/var/log/scp-on-xferlog {
    daily
    missingok
    notifempty
    postrotate
/etc/init.d/scp-on-xferlog restart
    endscript
}
```

Revision #1
Created 2021-07-27 09:54:16 UTC by magenbrot
Updated 2021-07-27 09:54:50 UTC by magenbrot