Nagios in High Availability Environments

Introduction

Nagios is a versatile and functional network management tool with a GUI (graphic user interface) comparable to other commercial tools. It's free software, which makes it a perfect candidate for companies.

When Nagios is used in a production environment, its high availability component is required. However, the solutions presented in the official documentation do not make Nagios a very reliable tool in this kind of environments. This document will show a more efficient way. First, we are going to approach the already existing solutions, and afterwards a new solution.

It's expected from the reader an understanding of Nagios's existing solutions for high availability environments (http://nagios.sourceforge.net/docs/1_0/redundancy.html).

Scenario 1 – Redundant Monitoring

This solution has various problems. The most obvious one is the increase of redundant information in the network. Consequently, it wastes the available bandwidth, and increases the load on the machines that are being monitored. Another undesirable situation, when the master process recovers, is that two machines can send notifications. It happens because the master process doesn't have any knowledge of the slave process. It obliges the slave to deactivate the notifications promptly. Finally, if one of the nagios processes fails, it won't recover information gathered during the down period. This method doesn't safeguards history.

Scenario 2 - Failover Monitoring

The difference between this solution and scenario 1 is the inexistence of more than one machine making checks. In this way monitored machines will not be submitted to an extra load, and the bandwidth use is more efficient. However, the problems related with history and notifications sync remain.

Scenario 3 – Failover Sync Monitoring

Introduction

This solution guarantees consistent history synchronism, and that the only running nagios process is the most up to date. It is considered the use of existing free software tools as a starting point. Later we approach the control method that eliminates redundancy, and guarantees processes synchronism.

History Sync

A Part of the solution is to use MySQL replication, which solves part of the history synchronism. The problem is that Nagios doesn't keep history in the database, even when compiled with MySQL. To assure it, it's used PerfParse (http://perfparse.sourceforge.net) - a nagios add-on. The reader will have to consult documentation on PerfParse and MySQL (section 4.11 replication, from the official documentation) to be able to make the correct procedures.

MySQL replication is crossed, which means that each mysql server is master and slave at the same time. It guarantees information recovery in case of crash or downtime (PerfParse uses innodb tables). To do it so, it must be placed the following parameters in `my.cnf ':

```
log-bin
log-slave-updates
report-host=<host_name>
replicate-do-db=<Nagios_and_PerfParse_data_base>
server-id=<1_or_2__mysql_servers_must_have_different_numbers>
```

The order of events that should take place to set up this part of the solution is:

1. install Mysql and configure 'my.cnf' file;
2. install Nagios;
3. install Nagios plug-ins;
4. install PerfParse (probably you'll have to reconfigure and reinstall Nagios);
5. reconfigure MySQL to sync the data bases – configure masters and slaves with mysql console and start them.

If these tools are used properly, the solution that guarantees the consistent history synchronism is complete. It means that it is possible to make crossed synchronism of the Nagios+PerfParse MySQL data bases. The version and configuration chosen influences the performance of the solution in high availability environments. It is wise a thorough reading on all the tools.

Nagios Process Sync

Although history synchronism is guaranteed, there are still some issues to solve:

- to guarantee that only one nagios process is running and sending information to the database which is replicated;
- to guarantee that the only running nagios process has the most up to date data base.

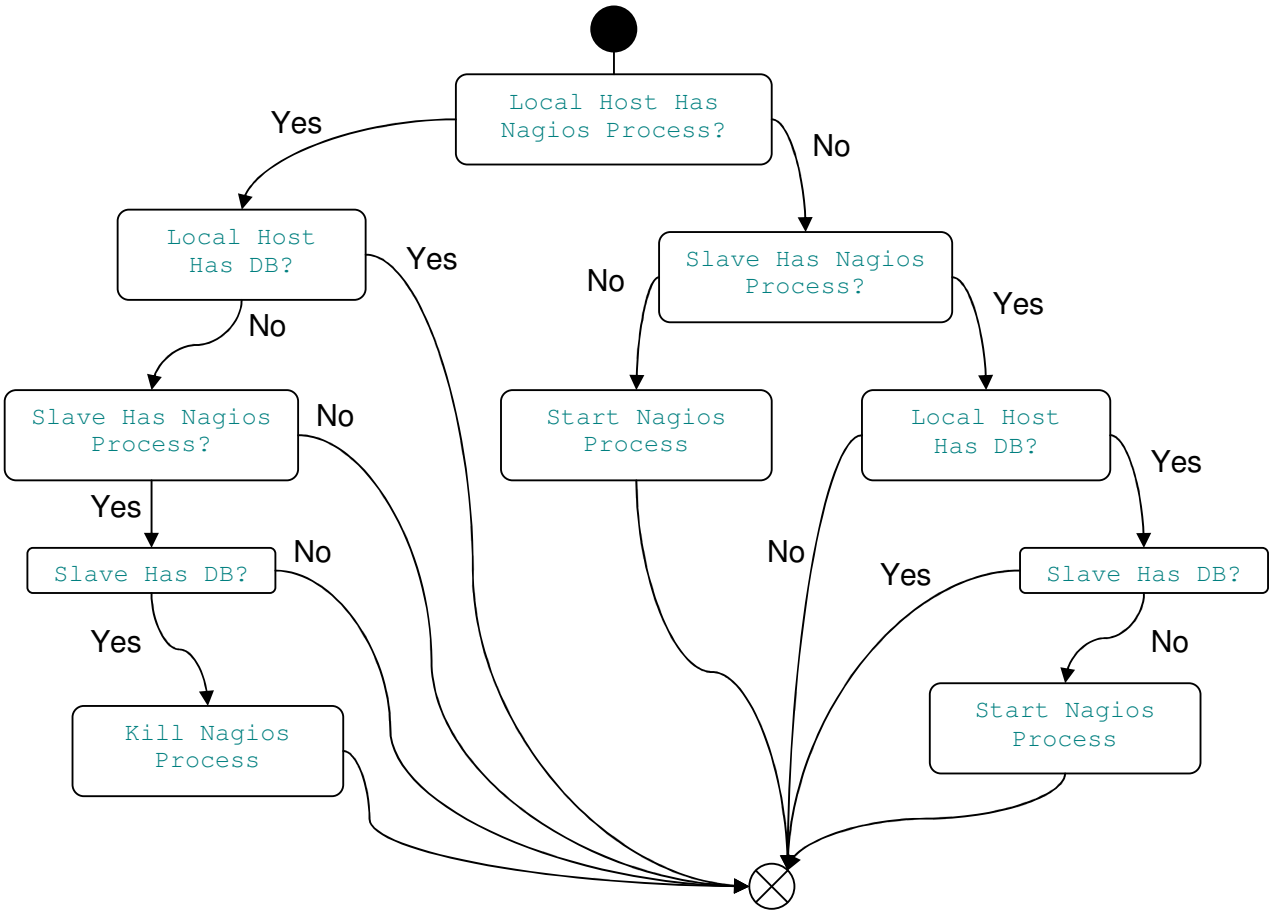
The answer is running two control scripts (`controller_master` and `controller_slave`) on different hosts. Their job is to manage the execution of nagios processes and to deal with some errors. To work properly, the `check_nrpe` plug-in and the NRPE daemon (running at boot) must be installed on both hosts. The daemon NRPE should fulfil the following checks:

- `check_nagios`
- `check_mysql`

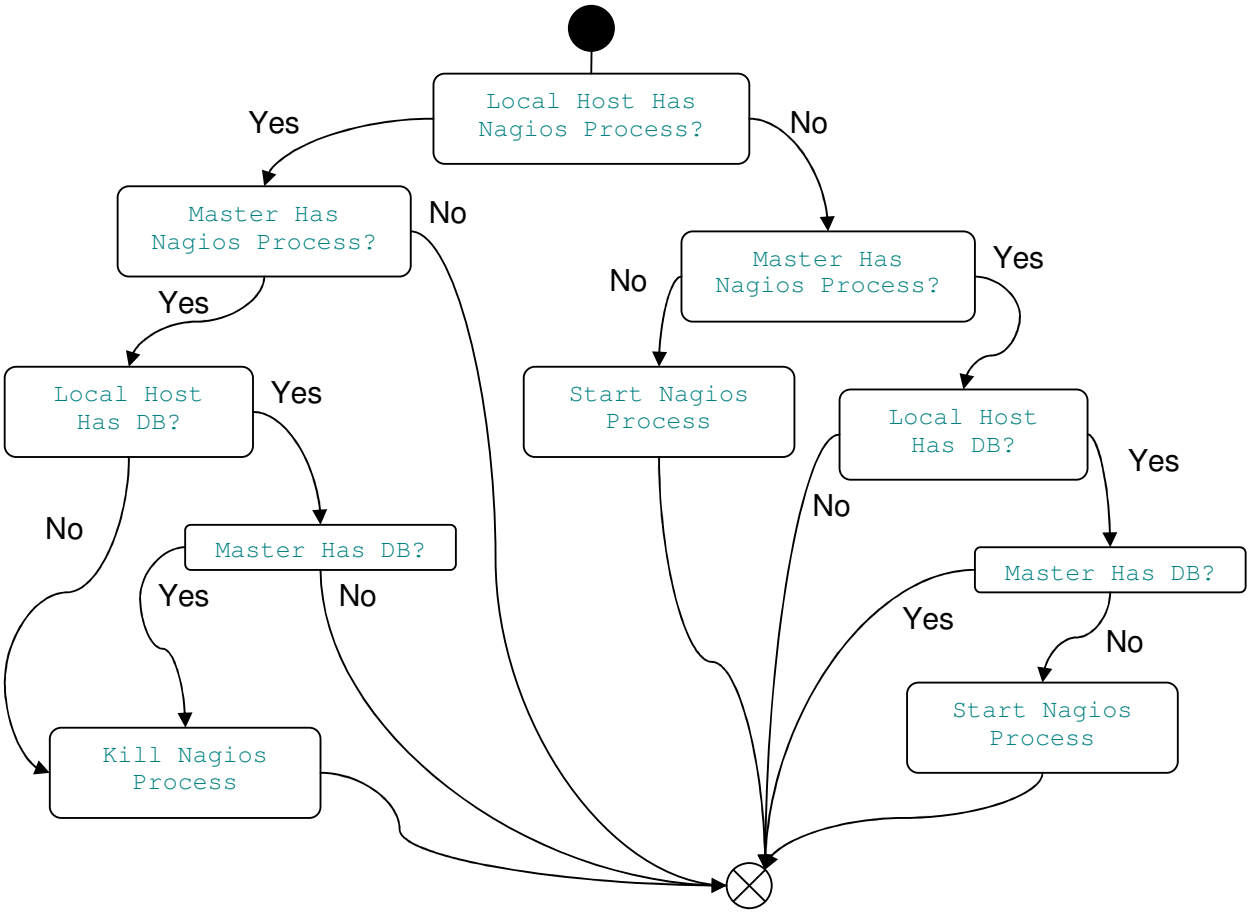
The scripts can run (every minute) in cron:

```
* * * * * /usr/local/nagios/bin/controller_master_or_slave_depending_on_the_host
```

The control scripts are optimized for execution. It's showed their activity diagrams, for the reader's understanding.



Activities Diagram 1 – controller_master script



Activities Diagram 2 – controller_slave script

Controlling Scripts Competition

The two scripts do not behave exactly like master/slave. In reality, the master script only takes-over in case it has a nagios process already running without problems; otherwise it behaves exactly like the slave script. It means that the two nagios processes can take-over alternatively when problems occur. This solution was equated because it is necessary to guarantee that the active nagios process has at all time the most up to date data base.

Complexity

The con of this solution is its complexity. The manipulation of some software tools is demanded. However, this can be an advantage for the versatility of the all solution.

controller_master Script Code

```
#!/bin/sh

#       NAGIOS MASTER CONTROLLING SCRIPT

#       This script has the job of controlling Nagios Process in a High Availability Environment.
#       It must run integrated with other controlling systems; namely Nagios, NRPE, PerfParse, MySQL
#       with crossed replication, and the other completing control script for the Slave Replica.
#       This solution is presented by Ricardo David Martins.

#       In order to run correctly, you must give the proper values to the next list of variables.

DEBUG=3
DEBUG_FILE=/var/log/messages
NAGIOS_WAITING_KILL_TIME=1
CHECK_NAGIOS=/usr/local/nagios/libexec/check_nagios
CHECK_NRPE=/usr/local/nagios/libexec/check_nrpe
CHECK_MYSQL=/usr/local/nagios/libexec/check_mysql
NAGIOS_LOG=/usr/local/nagios/var/nagios.log
NAGIOS_LOG_AGE_LIMIT=99999999
NAGIOS_COMMAND=/usr/local/nagios/bin/nagios
NAGIOS_CONFIG=/usr/local/nagios/etc/nagios.cfg
NAGIOS_EXT_FILE=/usr/local/nagios/var/rw/nagios.cmd
REMOTE_SLAVE_HOST=???.???.???.???
NAGIOS_DB=nagios
NAGIOS_DB_USER=nagios
NAGIOS_DB_PASSWORD=*****

# DO NOT change anything below this point, unless you know what you are doing.

function time_stamp()
{
        date '+%b %e %T'
}

#DEBUG MUST BE 3 OR GREATER
PREFIX_INFO="$(time_stamp) $HOSTNAME nagios: Master Control: INFORMATION:"

#DEBUG MUST BE 2 OR GREATER
PREFIX_WARNING="$(time_stamp) $HOSTNAME nagios Master Control: WARNING:"

#DEBUG MUST BE 1 OR GREATER
PREFIX_ERROR="$(time_stamp) $HOSTNAME nagios Master Control: ERROR:"

function check_local_nagios()
{
        $CHECK_NAGIOS -F $NAGIOS_LOG -C $NAGIOS_COMMAND -e $NAGIOS_LOG_AGE_LIMIT > /dev/null 2>&1
}

function check_remote_nagios()
{
        $CHECK_NRPE -H $REMOTE_SLAVE_HOST -c check_nagios > /dev/null 2>&1
}

function check_local_mysql()
{
        $CHECK_MYSQL -d $NAGIOS_DB -u $NAGIOS_DB_USER -p $NAGIOS_DB_PASSWORD > /dev/null 2>&1
}

function check_remote_mysql()
{
        $CHECK_NRPE -H $REMOTE_SLAVE_HOST -c check_mysql > /dev/null 2>&1
}

function kill_nagios()
{
        PID=`pidof -o %PPID $NAGIOS_COMMAND`
        kill -15 $PID > /dev/null 2>&1
}

function kill_forced_nagios()
{
         PID=`pidof -o %PPID $NAGIOS_COMMAND`
         kill -9 $PID > /dev/null 2>&1
}

function launch_nagios()
{
        $NAGIOS_COMMAND -d $NAGIOS_CONFIG > /dev/null 2>&1
}
```

```
function seek_nagios_external_file()
{
       [ -e $NAGIOS_EXT_FILE ]
}

function remove_nagios_external_file()
{
       rm -f $NAGIOS_EXT_FILE > /dev/null 2>&1
}

function solution_kill_nagios()
{
       kill_nagios
       sleep $NAGIOS_WAITING_KILL_TIME
       check_local_nagios
       error1=$?
       if [ $error1 -eq 0 ]; then
              if [ $DEBUG -gt 0 ]; then
                     echo "$PREFIX_ERROR It wasn't possible to remove nagios process with SIGTERM." \
>> $DEBUG_FILE
              fi
              kill_forced_nagios
              sleep $NAGIOS_WAITING_KILL_TIME
              check_local_nagios
              error2=$?
              if [ $error2 -ne 0 -a $DEBUG -gt 1 ]; then
                     echo "$PREFIX_WARNING Nagios process was removed with SIGKILL." >> $DEBUG_FILE
              elif [ $error2 -eq 0 -a $DEBUG -gt 0 ]; then
                     echo "$PREFIX_ERROR It wasn't possible to remove nagios process with SIGKILL." \
>> $DEBUG_FILE
              fi
       elif [ $DEBUG -gt 1 ]; then
              echo "$PREFIX_WARNING Nagios process was removed." >> $DEBUG_FILE
       fi
}

function solution_launch_nagios()
{
       launch_nagios
       check_local_nagios
       error3=$?
       if [ $error3 -ne 0 ]; then
              if [ $DEBUG -gt 0 ]; then
                     echo "$PREFIX_ERROR It wasn't possible to start nagios process." \
>> $DEBUG_FILE
              fi
              seek_nagios_external_file
              error4=$?
              if [ $error4 -eq 0 ]; then
                     if [ $DEBUG -gt 1 ]; then
                            echo "$PREFIX_WARNING It's necessary to remove the nagios external \
command file." >> $DEBUG_FILE
                     fi
                     remove_nagios_external_file
                     seek_nagios_external_file
                     error5=$?
                     if [ $error5 -ne 0 ]; then
                            if [ $DEBUG -gt 1 ]; then
                                   echo "$PREFIX_WARNING Nagios external command file was deleted." \
>> $DEBUG_FILE
                            fi
                            launch_nagios
                            check_local_nagios
                            error6=$?
                            if [ $error6 -eq 0 -a $DEBUG -gt 1 ]; then
                                   echo "$PREFIX_WARNING Nagios process started." >> $DEBUG_FILE
                            elif [ $error6 -ne 0 -a $DEBUG -gt 0 ]; then

                                   echo "$PREFIX_ERROR It wasn't possible to start nagios process." \
>> $DEBUG_FILE
                            fi
                     elif [ $DEBUG -gt 0 ]; then
                            echo "$PREFIX_ERROR It wasn't possible to delete nagios external command \
file." >> $DEBUG_FILE
                     fi
              elif [ $DEBUG -gt 0 ]; then
                     echo "$PREFIX_ERROR It wasn't possible to start nagios process for an unknown \
reason." >> $DEBUG_FILE
              fi
       elif [ $DEBUG -gt 1 ]; then
              echo "$PREFIX_WARNING Nagios process was started." >> $DEBUG_FILE
       fi
}
```

```
#*********************************************************
#Test if nagios process is running OK on the local machine*
#*********************************************************
check_local_nagios
error7=$?
if [ $error7 -eq 0 ]; then
        if [ $DEBUG -gt 2 ]; then
                echo "$PREFIX_INFO The local host is running nagios." >> $DEBUG_FILE
        fi

        #*****************************************************************
        #Test if nagios mysql data base is running OK on the local machine*
        #*****************************************************************
        check_local_mysql
        error8=$?
        if [ $error8 -ne 0 ]; then
                if [ $DEBUG -gt 0 ]; then
                        echo "$PREFIX_ERROR The local data base isn't running." >> $DEBUG_FILE
                fi

                #*********************************************************
                #Test if nagios process is running OK on the remote host*
                #*********************************************************
                check_remote_nagios
                error9=$?
                if [ $error9 -eq 0 ]; then
                        if [ $DEBUG -gt 2 ]; then
                                echo "$PREFIX_INFO The remote host is running nagios." >> $DEBUG_FILE
                        fi

                        #*****************************************************************
                        #Test if nagios mysql data base is running OK on the remote host*
                        #*****************************************************************
                        check_remote_mysql
                        error10=$?
                        if [ $error10 -eq 0 ]; then
                                if [ $DEBUG -gt 2 ]; then
                                        echo "$PREFIX_INFO The remote data base is running." >> $DEBUG_FILE
                                fi
                                solution_kill_nagios
                        elif [ $DEBUG -gt 0 ]; then
                                echo "$PREFIX_ERROR The remote data base isn't running." >> $DEBUG_FILE
                        fi

                elif [ $DEBUG -gt 2 ]; then
                        echo "$PREFIX_INFO The remote host isn't running nagios." >> $DEBUG_FILE
                fi
        elif [ $DEBUG -gt 2 ]; then
                echo "$PREFIX_INFO The local data base is running." >> $DEBUG_FILE
        fi
else
        if [ $DEBUG -gt 2 ]; then
                echo "$PREFIX_INFO The local host isn't running nagios." >> $DEBUG_FILE
        fi

        #*********************************************************
        #Test if nagios process is running OK on the remote host*
        #*********************************************************
        check_remote_nagios
        error11=$?
        if [ $error11 -eq 0 ]; then
                if [ $DEBUG -gt 2 ]; then
                        echo "$PREFIX_INFO The remote host is running nagios." >> $DEBUG_FILE
                fi

                #*****************************************************************
                #Test if nagios mysql data base is running OK on the local machine*
                #*****************************************************************
                check_local_mysql
                error12=$?
                if [ $error12 -eq 0 ]; then
                        if [ $DEBUG -gt 2 ]; then
                                echo "$PREFIX_INFO The local data base is running." >> $DEBUG_FILE
                        fi

                        #*****************************************************************
                        #Test if nagios mysql data base is running OK on the remote host*
                        #*****************************************************************
                        check_remote_mysql
                        error13=$?
                        if [ $error13 -ne 0 ]; then
                                if [ $DEBUG -gt 0 ]; then
                                        echo "$PREFIX_ERROR The remote data base isn't running." >> \
```

```
$DEBUG_FILE
                                fi
                                solution_launch_nagios
                        elif [ $DEBUG -gt 2 ]; then
                                echo "$PREFIX_INFO The remote data base is running." >> $DEBUG_FILE
                        fi
                elif [ $DEBUG -gt 0 ]; then
                        echo "$PREFIX_ERROR The local data base isn't running." >> $DEBUG_FILE
                fi
        else
                if [ $DEBUG -gt 2 ]; then
                        echo "$PREFIX_INFO The remote host isn't running nagios." >> $DEBUG_FILE
                fi
                solution_launch_nagios
        fi
fi

#*****************************************************
#Test if nrpe is running on the local and remote hosts*
#*****************************************************
if [ $DEBUG -gt 0 ]; then
        $CHECK_NRPE -H localhost
        error14=$?
        if [ $error14 -ne 0 ]; then
                echo "$PREFIX_ERROR The local host isn't running the NRPE daemon." >> $DEBUG_FILE
        fi
        $CHECK_NRPE -H $REMOTE_SLAVE_HOST
        error15=$?
        if [ $error15 -ne 0 ]; then
                echo "$PREFIX_ERROR The remote host isn't running the NRPE daemon." >> $DEBUG_FILE
        fi
fi
```

controller_slave Script Code

```
#!/bin/sh

#       NAGIOS SLAVE CONTROLLING SCRIPT

#       This script has the job of controlling Nagios Process in a High Availability Environment.
#       It must run integrated with other controlling systems; namely Nagios, NRPE, PerfParse, MySQL
#       with crossed replication, and the other completing control script for the Master Replica.
#       This solution is presented by Ricardo David Martins.

#       In order to run correctly, you must give the proper values to the next list of variables.

DEBUG=3
DEBUG_FILE=/var/log/messages
NAGIOS_WAITING_KILL_TIME=1
CHECK_NAGIOS=/usr/local/nagios/libexec/check_nagios
CHECK_NRPE=/usr/local/nagios/libexec/check_nrpe
CHECK_MYSQL=/usr/local/nagios/libexec/check_mysql
NAGIOS_LOG=/usr/local/nagios/var/nagios.log
NAGIOS_LOG_AGE_LIMIT=99999999
NAGIOS_COMMAND=/usr/local/nagios/bin/nagios
NAGIOS_CONFIG=/usr/local/nagios/etc/nagios.cfg
NAGIOS_EXT_FILE=/usr/local/nagios/var/rw/nagios.cmd
REMOTE_MASTER_HOST=???.???.???.???
NAGIOS_DB=nagios
NAGIOS_DB_USER=nagios
NAGIOS_DB_PASSWORD=*****

# DO NOT change anything below this point, unless you know what you are doing.

function time_stamp()
{
        date '+%b %e %T'
}

#DEBUG MUST BE 3 OR GREATER
PREFIX_INFO="$(time_stamp) $HOSTNAME nagios: Slave Control: INFORMATION:"

#DEBUG MUST BE 2 OR GREATER
PREFIX_WARNING="$(time_stamp) $HOSTNAME nagios: Slave Control: WARNING:"

#DEBUG MUST BE 1 OR GREATER
PREFIX_ERROR="$(time_stamp) $HOSTNAME nagios: Slave Control: ERROR:"

function check_local_nagios()
{
        $CHECK_NAGIOS -F $NAGIOS_LOG -C $NAGIOS_COMMAND -e $NAGIOS_LOG_AGE_LIMIT > /dev/null 2>&1
}

function check_remote_nagios()
{
        $CHECK_NRPE -H $REMOTE_MASTER_HOST -c check_nagios > /dev/null 2>&1
}

function check_local_mysql()
{
        $CHECK_MYSQL -d $NAGIOS_DB -u $NAGIOS_DB_USER -p $NAGIOS_DB_PASSWORD > /dev/null 2>&1
}

function check_remote_mysql()
{
        $CHECK_NRPE -H $REMOTE_MASTER_HOST -c check_mysql > /dev/null 2>&1
}

function kill_nagios()
{
        PID=`pidof -o %PPID $NAGIOS_COMMAND`
        kill -15 $PID > /dev/null 2>&1
}

function kill_forced_nagios()
{
         PID=`pidof -o %PPID $NAGIOS_COMMAND`
         kill -9 $PID > /dev/null 2>&1
}

function launch_nagios()
{
        $NAGIOS_COMMAND -d $NAGIOS_CONFIG > /dev/null 2>&1
}
```

```
function seek_nagios_external_file()
{
        [ -e $NAGIOS_EXT_FILE ]
}

function remove_nagios_external_file()
{
        rm -f $NAGIOS_EXT_FILE > /dev/null 2>&1
}

function solution_kill_nagios()
{
        kill_nagios
        sleep $NAGIOS_WAITING_KILL_TIME
        check_local_nagios
        error1=$?
        if [ $error1 -eq 0 ]; then
                if [ $DEBUG -gt 0 ]; then
                        echo "$PREFIX_ERROR It wasn't possible to remove nagios process with SIGTERM." \
>> $DEBUG_FILE
                fi
                kill_forced_nagios
                sleep $NAGIOS_WAITING_KILL_TIME
                check_local_nagios
                error2=$?
                if [ $error2 -ne 0 -a $DEBUG -gt 1 ]; then
                        echo "$PREFIX_WARNING Nagios process was removed with SIGKILL." >> $DEBUG_FILE
                elif [ $error2 -eq 0 -a $DEBUG -gt 0 ]; then
                        echo "$PREFIX_ERROR It wasn't possible to remove nagios process with SIGKILL." \
>> $DEBUG_FILE
                fi
        elif [ $DEBUG -gt 1 ]; then
                echo "$PREFIX_WARNING Nagios process was removed." >> $DEBUG_FILE
        fi
}

function solution_launch_nagios()
{
        launch_nagios
        check_local_nagios
        error3=$?
        if [ $error3 -ne 0 ]; then
                if [ $DEBUG -gt 0 ]; then
                        echo "$PREFIX_ERROR It wasn't possible to start nagios process." \
>> $DEBUG_FILE
                fi
                seek_nagios_external_file
                error4=$?
                if [ $error4 -eq 0 ]; then
                        if [ $DEBUG -gt 1 ]; then
                                echo "$PREFIX_WARNING It's necessary to remove the nagios external \
command file." >> $DEBUG_FILE
                        fi
                        remove_nagios_external_file
                        seek_nagios_external_file
                        error5=$?
                        if [ $error5 -ne 0 ]; then
                                if [ $DEBUG -gt 1 ]; then
                                        echo "$PREFIX_WARNING Nagios external command file was deleted." \
>> $DEBUG_FILE
                                fi
                                launch_nagios
                                check_local_nagios
                                error6=$?
                                if [ $error6 -eq 0 -a $DEBUG -gt 1 ]; then
                                        echo "$PREFIX_WARNING Nagios process started." >> $DEBUG_FILE
                                elif [ $error6 -ne 0 -a $DEBUG -gt 0 ]; then

                                        echo "$PREFIX_ERROR It wasn't possible to start nagios process." \
>> $DEBUG_FILE
                                fi
                        elif [ $DEBUG -gt 0 ]; then
                                echo "$PREFIX_ERROR It wasn't possible to delete nagios external command \
file." >> $DEBUG_FILE
                        fi
                elif [ $DEBUG -gt 0 ]; then
                        echo "$PREFIX_ERROR It wasn't possible to start nagios process for an unknown \
reason." >> $DEBUG_FILE
                fi
        elif [ $DEBUG -gt 1 ]; then
                echo "$PREFIX_WARNING Nagios process was started." >> $DEBUG_FILE
        fi
}
```

```
#*********************************************************
#Test if nagios process is running OK on the local machine*
#*********************************************************
check_local_nagios
error7=$?
if [ $error7 -eq 0 ]; then
        if [ $DEBUG -gt 2 ]; then
                echo "$PREFIX_INFO The local host is running nagios." >> $DEBUG_FILE
        fi

        #*********************************************************
        #Test if nagios process is running OK on the remote host*
        #*********************************************************
        check_remote_nagios
        error9=$?
        if [ $error9 -eq 0 ]; then
                if [ $DEBUG -gt 2 ]; then
                        echo "$PREFIX_INFO The remote host is running nagios." >> $DEBUG_FILE
                fi

                #*****************************************************************
                #Test if nagios mysql data base is running OK on the local machine*
                #*****************************************************************
                check_local_mysql
                error8=$?
                if [ $error8 -ne 0 ]; then
                        if [ $DEBUG -gt 0 ]; then
                                echo "$PREFIX_ERROR The local data base isn't running." >> $DEBUG_FILE
                        fi
                        solution_kill_nagios
                else
                        if [ $DEBUG -gt 2 ]; then
                                echo "$PREFIX_INFO The local data base is running." >> $DEBUG_FILE
                        fi

                        #*****************************************************************
                        #Test if nagios mysql data base is running OK on the remote host*
                        #*****************************************************************
                        check_remote_mysql
                        error10=$?
                        if [ $error10 -eq 0 ]; then
                                if [ $DEBUG -gt 2 ]; then
                                        echo "$PREFIX_INFO The remote data base is running." >> $DEBUG_FILE
                                fi
                                solution_kill_nagios
                        elif [ $DEBUG -gt 0 ]; then
                                echo "$PREFIX_ERROR The remote data base isn't running." >> $DEBUG_FILE
                        fi
                fi
        elif [ $DEBUG -gt 2 ]; then
                echo "$PREFIX_INFO The remote host isn't running nagios." >> $DEBUG_FILE
        fi
else
        if [ $DEBUG -gt 2 ]; then
                echo "$PREFIX_INFO The local host isn't running nagios." >> $DEBUG_FILE
        fi

        #*********************************************************
        #Test if nagios process is running OK on the remote host*
        #*********************************************************
        check_remote_nagios
        error11=$?
         if [ $error11 -eq 0 ]; then
                if [ $DEBUG -gt 2 ]; then
                        echo "$PREFIX_INFO The remote host is running nagios." >> $DEBUG_FILE
                fi

                #*****************************************************************
                #Test if nagios mysql data base is running OK on the local machine*
                #*****************************************************************
                check_local_mysql
                error12=$?
                if [ $error12 -eq 0 ]; then
                        if [ $DEBUG -gt 2 ]; then
                                echo "$PREFIX_INFO The local data base is running." >> $DEBUG_FILE
                        fi

                        #*****************************************************************
                        #Test if nagios mysql data base is running OK on the remote host*
                        #*****************************************************************
                        check_remote_mysql
                        error13=$?
                        if [ $error13 -ne 0 ]; then
```

```
                                if [ $DEBUG -gt 0 ]; then
                                        echo "$PREFIX_ERROR The remote data base isn't running." >> \
$DEBUG_FILE
                                fi
                                solution_launch_nagios
                        elif [ $DEBUG -gt 2 ]; then
                                echo "$PREFIX_INFO The remote data base is running." >> $DEBUG_FILE
                        fi
                elif [ $DEBUG -gt 0 ]; then
                        echo "$PREFIX_ERROR The local data base isn't running." >> $DEBUG_FILE
                fi
        else
                if [ $DEBUG -gt 2 ]; then
                        echo "$PREFIX_INFO The remote host isn't running nagios." >> $DEBUG_FILE
                fi
                solution_launch_nagios
        fi
fi

#*****************************************************
#Test if nrpe is running on the local and remote hosts*
#*****************************************************
if [ $DEBUG -gt 0 ]; then
        $CHECK_NRPE -H localhost
        error14=$?
        if [ $error14 -ne 0 ]; then
                echo "$PREFIX_ERROR The local host isn't running the NRPE daemon." >> $DEBUG_FILE
        fi
        $CHECK_NRPE -H $REMOTE_MASTER_HOST
        error15=$?
        if [ $error15 -ne 0 ]; then
                echo "$PREFIX_ERROR The remote host isn't running the NRPE daemon." >> $DEBUG_FILE
        fi
fi
```