
Subject: Monitoring Virtuals with Nagios

Posted by [Geek42](#) on Fri, 06 Mar 2009 20:51:25 GMT

[View Forum Message](#) <> [Reply to Message](#)

I currently use Nagios to monitor the (currently) 32 virtuals on my server. I found that if I was doing things the way I was supposed to, that I needed to run the nrpe server on every single one of them. I'm trying to keep things very controlled on the virtuals, so I want to run the minimal possible to do what I need on each one.

I came up with the following solution, and was wondering if anyone with better knowledge of how to do things could suggest some improvements/replacements... Of course you are also all free to use this if you think it might be useful...

Anyway, here is what I have:

1 Host Node: Server01

32 Virtuals: various names, descriptive of what they do, ie Nagios01 for my primary Nagios service.

My networking uses bridged virtual ethernet devices, mostly because they worked for me, but it should not effect this way of checking.

Nagios, running on a virtual, calls to nrpe running on Server01. If I'm just checking the local commands, no problem, they check fine. If I want to check one of the virtuals, it runs a special script via nrpe running on Server01:

```
#!/bin/bash
PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin

PROGRAMME=`basename $0`
PROGPATH=`echo $0 | sed -e 's,[\\V][^\\V][^\\V]*$,,'`
REVISION=`echo '$Revision: 0.1 $' | sed -e 's/[^0-9.]/g``

. $PROGPATH/utils.sh

VE_PATH=/usr/lib/nagios/plugins/
VZ=/usr/sbin/vzctl
SUDO=/usr/bin/sudo

print_usage() {
    echo "Usage: $PROGRAMME"
}

print_help() {
    print_revision $PROGRAMME $REVISION
    echo ""
    print_usage
    echo ""
}
```

```

echo "This plugin runs a plugin on a OpenVZ VM hosted on this system."
echo ""
support
exit 0
}
function get_veid {
#   A=`echo $1 | sed -n 's/^\([0-9]*\)\\. *^1/p`
#   B=`echo $1 | sed -n 's/^[0-9]*\\. \([0-9]*\)\\. *^1/p`
C=`echo $1 | sed -n 's/^[0-9]*\\. [0-9]*\\. \([0-9]*\)\\. *^1/p`
D=`echo $1 | sed -n 's/^[0-9]*\\. [0-9]*\\. [0-9]*\\. \([0-9]*\) $^1/p`

if [ $D -eq 250 ]; then
    C=$(( $C + 1 ))
    D=0
fi

C=$(( $C - 192 ))
ID=$(( $D + (($C * 250) )) )
}

case "$1" in
--help)
    print_help
    exit 0
    ;;
-h)
    print_help
    exit 0
    ;;
--version)
    print_revision $PROGNAME $REVISION
    exit 0
    ;;
-V)
    print_revision $PROGNAME $REVISION
    exit 0
    ;;
-t)
    echo OK $0 $1
    exit 0
    ;;
*)
    if test "$1" = "-H"; then
        get_veid $2
    else
        echo "Unknown: $1 is first paramater, should be -H"
        exit 3
    fi
fi

```

```

if test "$3" = "-c"; then
    CMD="$VE_PATH$4"
    ARGS="$5 $6 $7 $8 $9"
else
    echo "Unknown: $3 is third parameter, should be -c"
    exit 3
fi
#vzctl exec2 $ID passes back the exit code
DATA=`$SUDO $VZ exec2 $ID $CMD $ARGS 2>&1`
STATUS=$?
echo $DATA
exit $STATUS
;;
esac

```

In the above code, I just get the veid based on the ip address passed as I use a formula to get the ip address from the veid in my creation script.

Beyond that I have the following line in the /etc/sudoers file on Server01:

```
nagios ALL=NOPASSWD:/usr/sbin/vzctl
```

Which is mildly insecure but allows the nagios user to run the vzctl command. Because of this I have setup the nrpe on server01 to only allow connection from my Nagios Server, no others, otherwise you can see the problems this could cause. This is the main problem I have with this script, and would love to know a way to avoid it.

Anyway, What this does is use vzctl exec2 #veid to execute the plugins i've installed in each VZ(without any nrpe server) and then pass the results back to nagios.

In my nrpe_local.cfg on Server01 I have:

```
command[check_openvz_vm]=/usr/lib/nagios/plugins/check_openvz_vm $ARG1$ $ARG2$
$ARG3$
```

Finally on the nagios virtual I have these settings(can be in one file or multiple, nagios does not care usually):

```

define command{
    command_name    check_openvz_vm
    command_line    /usr/lib/nagios/plugins/check_nrpe -H ip.of.server.01 -c check_openvz_vm
-a "$ARG1$"
}

define service{
    use                openvz-service
    hostgroup_name     virtual-openvz
    service_description OpenVZ VE Process Count
    check_command      check_openvz_vm!-H $HOSTADDRESS$ -c check_procs -w 50
-c 75
    normal_check_interval 15
}

```

```
}  
  
define hostgroup{  
    hostgroup_name virtual-openvz  
    alias          OpenVZ Virtual Servers  
}  
  
define host{  
    use                linux-server  
    hostgroups        virtual-openvz  
    host_name         1001 Gateway  
    alias             Gateway.domain.name  
    address           10.51.196.1  
    parents           Server01  
}
```

The one real improvement I can think of is to have this actually call a command on the VE that works similar to the nrpe without constantly running.

If people think this is good, after any improvements of course, I'll throw this up on the wiki, otherwise I'll just keep it to myself and be quite happy with it.

Thanks

JC

Subject: Re: Monitoring Virtuals with Nagios
Posted by [givre](#) on Thu, 30 Jul 2009 13:11:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

I don't understand why this post hasn't answered
I think this script is very interesting.
Thank you for sharing this script

I will try use it on my system.

I have one question about your function "get_veid"
It doesn't work for me, and I don't see what it does really

I have made a little basic change for test.
This isn't a good solution but for some tests script it's okay.

For example:
\$1 ="192.168.0.3"

With

```
function get_veid {
#   A=`echo $1 | sed -n 's/^\([0-9]*\)\\. *^1/p`
#   B=`echo $1 | sed -n 's/^[0-9]*\\. \([0-9]*\)\\. *^1/p`
#   C=`echo $1 | sed -n 's/^[0-9]*\.[0-9]*\\. \([0-9]*\)\\. *^1/p`
#   D=`echo $1 | sed -n 's/^[0-9]*\.[0-9]*\.[0-9]*\\. \([0-9]*\) $^1/p`

#   if [ $D -eq 250 ]; then
#       C=$(( $C + 1 ))
#       D=0
#   fi

#   C=$(( $C - 192 ))
#   ID=$(( $D + (($C * 250) )) )
}
```

the result of veid isn't ok : -47997 (strange ?)

so a made:

```
function get_veid {
#   A=`echo $1 | sed -n 's/^\([0-9]*\)\\. *^1/p`
#   B=`echo $1 | sed -n 's/^[0-9]*\\. \([0-9]*\)\\. *^1/p`
#   C=`echo $a | sed -n 's/^[0-9]*\.[0-9]*\\. \([0-9]*\)\\. *^1/p`
#   D=`echo $a | sed -n 's/^[0-9]*\.[0-9]*\.[0-9]*\\. \([0-9]*\) $^1/p`

#   if [ $D -eq 250 ]; then
#       C=$(( $C + 1 ))
#       D=0
#   fi

#   C=$(( $C - 192 ))
#   ID="10$D"
}
```

On my system VM id start at 10X
Where X represent the vm number id .

But would like to understand your function to get the veid and what is exec2 ?

On my Openvz server , no problem, work fine.

On my nagios server, who isn't a VM, i get "CHECK_NRPE: Socket timeout after 10 seconds. "

I will see why.

Thanks for your answer .

Regards,

Benoit.

Subject: Re: Monitoring Virtuals with Nagios
Posted by [Geek42](#) on Thu, 30 Jul 2009 14:45:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

the get_veid function is something that you need to customize to your environment. I could not find a way to store the VEID in nagios and pass that to the script, so I use the ip address of the VE that nagios does store(some checks, like http, connect directly).

Anyway, I have a personal numbering scheme I use on my server that this translates between. IP is translated to one of 10000 possible(more really, but I ignore them) VEIDs.

My network is setup like this:

| VEID | IP |
|------|------------|
| 0001 | X.Y.0.1 |
| 0002 | X.Y.0.2 |
| ... | |
| 0250 | X.Y.0.250 |
| 0251 | X.Y.1.1 |
| 0252 | X.Y.1.2 |
| ... | |
| 0500 | X.Y.1.250 |
| 0501 | X.Y.2.1 |
| ... | |
| 1000 | X.Y.3.250 |
| ... | |
| 9999 | X.Y.39.249 |

While the function above is not exactly for that, it get's the idea across(I actually use the last large enough block of addresses in the 10.x.x.x range, so that adds some math to the process, and I collect A and B just in case I want them later).

It does waste numbers, but I find that it also makes numbering much easier to deal with. So your edit, if it works, is all you need, feel free to delete my stuff, I just put it there in case someone wants to do the same type of layout.

Anyway, look up the manual for the exec2 command, but basically it just deals with things in a better way for scripting.

The not connecting from remote is probably in your allowed ip options for nrpe on the host node.

Subject: Re: Monitoring Virtuals with Nagios
Posted by [usproblogger](#) on Sun, 20 Sep 2009 08:32:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

Wow that is some insight here, i am using hmspanel, so, I emailed your thing to them, will update on response.