

How to use Nagios monitoring for Netezza databases

Author: Adam Mateusewicz



Nagios is an open-source monitoring system that runs on the Linux operating system that monitors devices running Linux, Windows and Unix operating systems.

Most companies run Nagios to monitor such things as memory usage, disk usage, microprocessor load, the number of currently running processes and log files. The monitoring of databases will, however, generally be ignored.

In this article we will show how to create and deploy Nagios monitoring for Netezza system components:

- workload
- database usage
- data slice capacity
- number of connected users
- number of database sessions

We will assume that Nagios server is already set up and running. In our laboratory system we are running a raspberry PI 4 with ESXi installed on it. The Nagios VM is installed on Ubuntu 20.04 LTS (long term support).

Step 1 - Install NRPE and plugins on Netezza host

N.B. monitoring utilities require installation of agents on monitored systems. Netezza has the same requirement. A difficulty may be that unless you are running CP4DS version 1.0.7.8/Hammerhead Netezza systems run RedHat 6 or below. With this in mind you will not be able to install packages using YUM.

For Stripers/Makos - go to:

https://dl.fedoraproject.org/pub/archive/epel/6/x86_64/Packages/n/

For TwinFins - go to:

https://dl.fedoraproject.org/pub/archive/epel/5/x86_64/

Download the following packages: (list is for RHEL 6):

nagios-common-4.4.3-3.el6.x86_64.rpm
nagios-plugins-2.3.3-1.el6.x86_64.rpm
nagios-plugins-apt-2.3.3-1.el6.x86_64.rpm
nagios-plugins-bdii-1.0.14-1.el6.x86_64.rpm
nagios-plugins-bonding-1.4-3.el6.x86_64.rpm
nagios-plugins-breeze-2.3.3-1.el6.x86_64.rpm
nagios-plugins-cluster-2.3.3-1.el6.x86_64.rpm
nagios-plugins-dhcp-2.3.3-1.el6.x86_64.rpm
nagios-plugins-dig-2.3.3-1.el6.x86_64.rpm
nagios-plugins-disk-2.3.3-1.el6.x86_64.rpm
nagios-plugins-dns-2.3.3-1.el6.x86_64.rpm
nagios-plugins-dummy-2.3.3-1.el6.x86_64.rpm
nagios-plugins-flexlm-2.3.3-1.el6.x86_64.rpm
nagios-plugins-hpjd-2.3.3-1.el6.x86_64.rpm
nagios-plugins-http-2.3.3-1.el6.x86_64.rpm
nagios-plugins-icmp-2.3.3-1.el6.x86_64.rpm
nagios-plugins-ide_smart-2.3.3-1.el6.x86_64.rpm
nagios-plugins-ircd-2.3.3-1.el6.x86_64.rpm
nagios-plugins-ldap-2.3.3-1.el6.x86_64.rpm
nagios-plugins-load-2.3.3-1.el6.x86_64.rpm
nagios-plugins-log-2.3.3-1.el6.x86_64.rpm
nagios-plugins-mailq-2.3.3-1.el6.x86_64.rpm
nagios-plugins-mrtg-2.3.3-1.el6.x86_64.rpm
nagios-plugins-perl-2.3.3-1.el6.x86_64.rpm
nagios-plugins-smtp-2.3.3-1.el6.x86_64.rpm
nagios-plugins-snmp-2.3.3-1.el6.x86_64.rpm
nagios-plugins-ssh-2.3.3-1.el6.x86_64.rpm
nagios-plugins-swap-2.3.3-1.el6.x86_64.rpm
nagios-plugins-time-2.3.3-1.el6.x86_64.rpm
nagios-plugins-ups-2.3.3-1.el6.x86_64.rpm

```
nagios-plugins-wave-2.3.3-1.el6.x86_64.rpm
nrpe-4.0.3-1.el6.x86_64.rpm
```

Upload to each host and install with:
`rpm -ivh *.rpm`

Resolve all dependencies (if any).

Step 2 - Edit NRPE configuration:

Create initial config of NRPE:

```
service nrpe start
service nrpe stop
```

To file `/etc/nagios/nrpe.cfg` add following lines:

```
command[check_nzscratch]=/usr/lib64/nagios/plugins/check_disk -w 20% -c 10% -p /nzscratch
command[check_home]=/usr/lib64/nagios/plugins/check_disk -w 20% -c 10% -p /export/home
command[check_nz]=/usr/lib64/nagios/plugins/check_disk -w 20% -c 10% -p /nz
command[check_tmp]=/usr/lib64/nagios/plugins/check_disk -w 20% -c 10% -p /tmp
command[check_npsstate]=/usr/lib64/nagios/plugins/check_npsstate
```

Also - remember to modify permitted Nagios hosts only:

```
allowed_hosts=10.102.0.15,127.0.0.1
```

Create `/usr/lib64/nagios/plugins/check_npsstate` file:

```
#!/bin/bash
state=$(sudo -i -u nz bash -c "nzstate -terse")
if (($state == "online")); then
echo "OK - online"
exit 0
else
echo "UNKNOWN - $state"
exit 3
fi
```

Make file executable:

```
chmod +x /usr/lib64/nagios/plugins/check_npsstate
```

N.B. Nagios expects an exit code of OK or 0, otherwise it will report an error.

Because Nagios monitoring on the host is running as user 'nrpe', so test each of them as `sudo 'nrpe'` - if any commands don't work add the following lines to `/etc/sudoers`:

```
Defaults:nrpe !requiretty
nrpe ALL=(ALL) NOPASSWD: ALL
```

Alternatively, remove the commands that won't run.

NOTE: If your IT security department won't allow you to do this, restrict the commands you run to:

Start NRPE:

```
service nrpe start
```

On Nagios server check if NRPE is running:

```
root@nagios:~# /usr/lib/nagios/plugins/check_nrpe -H 10.102.0.73 -c check_npsstat
```

OK = online

3. Setup Nagios server:

The preferred method is to create a separate directory for all the custom scripts and utilities.

Add following to `/usr/local/nagios/etc/nagios.cfg`:

```
cfg_dir=/usr/local/nagios/etc/NETEZZA
```

In this directory create all your monitoring scripts and add servers:

```
root@nagios:/usr/local/nagios/etc/NETEZZA# cat netezza_hosts.cfg | grep .
```

```
#####  
#####
```

```
REMOTE HOST DEFINITION
```

```
#####  
#####
```

```
define hostgroup {  
  hostgroup_name netezza  
  members NetezzaDev3MAJ  
}  
define host {  
  use netezza  
  host_name NetezzaDev3MAJ  
  alias NetezzaDev3MAJ  
  address 10.102.0.73  
}
```

```
root@nagios:/usr/local/nagios/etc/NETEZZA# cat netezza_template.cfg
```

```
define host{  
  name netezza  
  notifications_enabled 1  
  event_handler_enabled 1  
  flap_detection_enabled 1  
  process_perf_data 1  
  retain_status_information 1  
  retain_nonstatus_information 1  
  check_command check-host-alive  
  check_interval 5  
  max_check_attempts 2  
  notification_interval 0  
  notification_period 24x7  
  notification_options d,u,r
```

```
register 0
}
```

```
root@nagios:/usr/local/nagios/etc/NETEZZA# cat netezza_service.cfg
```

```
#####  
#####
```

```
Netezza disks usage definition
```

```
#####  
#####
```

```
define command {  
command_name check_nzscratch  
command_line /usr/lib/nagios/plugins/check_nrpe -H $HOSTADDRESS$ -c check_nzscratch  
}  
define command {  
command_name check_nz  
command_line /usr/lib/nagios/plugins/check_nrpe -H $HOSTADDRESS$ -c check_nz  
}  
define command {  
command_name check_tmp  
command_line /usr/lib/nagios/plugins/check_nrpe -H $HOSTADDRESS$ -c check_tmp  
}  
define command {  
command_name check_npsstate  
command_line /usr/lib/nagios/plugins/check_nrpe -H $HOSTADDRESS$ -c check_npsstate  
}  
define service {  
hostgroup_name netezza  
service_description check_nzscratch  
check_command check_nzscratch  
max_check_attempts 2  
check_interval 2  
retry_interval 2  
check_period 24x7  
check_freshness 1  
notification_interval 2  
notification_period 24x7  
notifications_enabled 1  
register 1  
}  
define service {  
hostgroup_name netezza  
service_description check_nz  
check_command check_nz  
max_check_attempts 2
```

```
check_interval 2
retry_interval 2
check_period 24x7
check_freshness 1
notification_interval 2
notification_period 24x7
notifications_enabled 1
register 1
}
define service {
hostgroup_name netezza
service_description check_tmp
check_command check_tmp
max_check_attempts 2
check_interval 2
retry_interval 2
check_period 24x7
check_freshness 1
notification_interval 2
notification_period 24x7
notifications_enabled 1
register 1
}
define service {
hostgroup_name netezza
service_description check_npsstate
check_command check_npsstate
max_check_attempts 2
check_interval 2
retry_interval 2
check_period 24x7
check_freshness 1
notification_interval 2
notification_period 24x7
notifications_enabled 1
register 1
}
```

Verify Nagios configuration:

```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

If no errors - restart Nagios:

```
systemctl restart nagios
```

Now you have your database monitoring installed:

The screenshot shows the Nagios web interface. On the left is a navigation menu with options like 'General', 'Home', 'Documentation', 'Current Status', 'Tactical Overview', 'Map (Legacy)', 'Hosts', 'Services', 'Host Groups', 'Service Groups', and 'Summary'. The main content area is divided into several sections: 'Current Network Status' (last updated Thu May 5 07:52:09 UTC 2022), 'Host Status Totals' (Up: 2, Down: 0, Unreachable: 0, Pending: 0), and 'Service Status Totals' (Ok: 12, Warning: 0, Unknown: 0, Critical: 0, Pending: 0). Below these is a table titled 'Service Status Details For All Hosts' with columns for Host, Service, Status, Last Check, Duration, Attempt, and Status Information. The table shows services like 'check_npsstate', 'check_nz', 'check_nzscratch', and 'check_tmpr' all in 'OK' status.

All that needs to be done now for development of custom monitoring is:

1. On Netezza host add:

```
command[<CUSTOM_COMMAND_NAME>]=<PATH>/<SCRIPT_NAME>
```

and restart NRPE (note that script needs to exist)

2. Write your script on Netezza host <PATH>/<SCRIPT_NAME>

3. During testing - you need to run NRPE on remote machine to verify it working:

```
/usr/lib/nagios/plugins/check_nrpe -H <NETEZZA_HOST_IP> -c  
CUSTOM_COMMAND_NAME
```

We hope you find this useful - if you want any assistance with your Netezza problems why not reach out to us [here](#).



Smart Associates (Aotearoa) Ltd	Smart Associates Ltd	Smart Associates ApS
203/11 Vinegar Lane Grey Lynne Auckland 1021 New Zealand	Valley View, The Old Quarry Haslemere Surrey GU27 3SS United Kingdom	% 360 Law Firm Lautrupsgade 7, 3. tv DK-2100 København Ø Copenhagen, Denmark
T: +64 (9) 415-8120	T: +44 (208) 133-6008	T: +45 36 98 71 11