

6 - Installation - Postgres Plugin (CentOS 6.4)

- [References](#)
- [Prerequisites](#)
 - [Postgres](#)
 - [CentOS and Perl packages](#)
- [Download and install the Postgres Plugin](#)
 - [Download](#)
 - [Set up the Postgres user "nagios"](#)
 - [Set up OS user "nagios"](#)
 - [Test locally](#)
 - [Amend nrpe.cfg](#)
 - [Test on Nagios server](#)

References

Nagios exchange: http://exchange.nagios.org/directory/Plugins/Databases/Oracle/check_oracle_health/details

Postgres Plugin: http://bucardo.org/wiki/Check_postgres

Prerequisites

Postgres

The Postgres database server must be up and running.

In `/var/lib/pgsql/data/pg_hba.conf`, the authentication method for "local" should be "ident":

```
# "local" is for Unix domain socket connections only
local all all ident
```

CentOS and Perl packages

Perl must be installed (this is the default).

To install additional Perl modules, CPAN should be installed, too:

Perl Setup

```
yum -y install cpan
cpan
install YAML
install Time::HiRes
install Net::SMTP
exit
```

Download and install the Postgres Plugin

Download

Check for the latest version at http://bucardo.org/wiki/Check_postgres, scroll down to "Download".

Download and install

```
cd /tmp
wget http://bucardo.org/downloads/check_postgres-2.21.0.tar.gz
tar -xzf check_postgres*.tar.gz
cd check_postgres-*
cp check_postgres.pl /usr/local/nagios/libexec/
```

Set up the Postgres user "nagios"

Create Postgres User

```
su - postgres
psql
CREATE USER nagios WITH PASSWORD 'some_password';
\q
exit # back to root
```

Set up OS user "nagios"

To keep the Postgres password from prying eyes, a password file is required in the home directory of OS user "nagios":

Password file

```
vi ~nagios/.pgpass
# insert:
localhost:5432:templatel:nagios:some_password
# insert end (EOF)
# save and exit vi
chown nagios:nagios ~nagios/.pgpass
chmod 600 ~nagios/.pgpass
```

Test locally

Test

```
su - nagios
cd /usr/local/nagios/libexec/
./check_postgres.pl -H localhost -db templatel -u nagios --action connection
./check_postgres.pl -H localhost -db templatel -u nagios --action dbstats
./check_postgres.pl -H localhost -db templatel -u nagios --action bloat
# in the next statement, you should set the limits as appropriate
./check_postgres.pl -H localhost -db templatel -u nagios --action=database_size --warning='30 GB' --
critical='35 GB'
exit # back to root
```

Amend nrpe.cfg

Add the following lines to /usr/local/nagios/etc/nrpe.cfg

nrpe.cfg

```
vi /usr/local/nagios/etc/nrpe.cfg
...
command[check_pg_connection]=/usr/local/nagios/libexec/check_postgres.pl -H localhost -db template1 -u nagios --
action connection
command[check_pg_dbstats]=/usr/local/nagios/libexec/check_postgres.pl -H localhost -db template1 -u nagios --
action dbstats
command[check_pg_bloat]=/usr/local/nagios/libexec/check_postgres.pl -H localhost -db template1 -u nagios --
action bloat
# in the next line, you should set the limits as appropriate
command[check_pg_database_size]=/usr/local/nagios/libexec/check_postgres.pl -H localhost -db template1 -u
nagios --action database_size --warning='30 GB' --critical='35 GB'
```

Test on Nagios server

Replace the "yourserver.yourdomain.tld" with the IP address or IP name of your MySQL server:

```
/usr/local/nagios/libexec/check_nrpe -H yourserver.yourdomain.tld -c check_pg_connection
```

If the system returns "ERROR: Could not find a suitable psql executable", your Postgres installation may not use the default location. Normally this should be solved by setting the environment variable "PGBINDIR" to the path to psql, and adding this path to the PATH variable (e.g. in /etc/profile.d/), but for some reasons this doesn't work. You have to modify "check_postgres.pl"

```
vi /usr/local/nagios/libexec/check_postgres.pl
... locate the line starting with "$PGBINDIR ="
... and replace it with
$PGBINDIR = '/usr/local/pgsql/bin';
... or wherever your psql binary is installed.
```