pgBadger is a PostgreSQL log analyzer build for speed with fully detailed reports from your PostgreSQL log file. It's a single and small script written in pure Perl language.

It uses a javascript library to draw graphs so that you don't need additional Perl modules or any other package to install. Furthermore, this library gives us more features such as zooming.

pgBadger is able to autodetect your log file format (syslog, stderr or csvlog). It is designed to parse huge log files as well as gzip compressed file.

<https://github.com/dalibo/pgbadger>

To know the new pgbadger version(now v9.1) <https://github.com/dalibo/pgbadger/blob/master/ChangeLog>

2017-01-24 - v9.1 (pgbadger newer version released date)

To download the source code with tar zip file , use below commands postgres@postgres\_host:~$ wget https://github.com/dalibo/pgbadger/archive/v9.6.tar.gz postgres@postgres\_host:~$ tar -xvzf v9.1.tar.gz

postgres@postgres\_host:~$ cd pgbadger-9.1

#Install PERL MakeMaker if not already installed:

apt-get install perl-ExtUtils-MakeMaker

sudo apt-get install libtext-csv-perl

#Compile and install: postgres@postgres\_host:/var/lib/pgsql/pgbadger-9.1$ perl Makefile.PL Checking if your kit is complete...

Looks good

Generating a Unix-style Makefile Writing Makefile for pgBadger

Writing MYMETA.yml and MYMETA.json

apt-get install make (if not install make , you have to install given command) postgres@postgres\_host:/var/lib/pgsql/pgbadger-9.1$ make

cp pgbadger blib/script/pgbadger

"/usr/bin/perl" -MExtUtils::MY -e 'MY->fixin(shift)' -- blib/script/pgbadger Manifying 1 pod document

root@postgres\_host:/var/lib/pgsql/pgbadger-9.1# make install Manifying 1 pod document

Installing /usr/local/man/man1/pgbadger.1p Installing /usr/local/bin/pgbadger

Appending installation info to /usr/local/lib/x86\_64-linux-gnu/perl/5.22.2/perllocal.pod

You have to switch the user to postgres and change the configuration file root@postgres\_host:~# su - postgres

postgres@postgres\_host:~$ . postgres.env postgres@postgres\_host:~$ cd $PGDATA postgres@postgres\_host:~$ nano $PGDATA/postgrersql.conf

#turn on logging for pgbadger by editing the line after # What to Log log\_min\_duration\_statement = 0 or

log\_min\_duration\_statement = 50 (optional 50 milli sec)

log\_line\_prefix = '%t [%p]: [%l-1] user=%u,db=%d ' log\_checkpoints = on

log\_connections = on log\_disconnections = on log\_lock\_waits = on log\_temp\_files = 0

#Postgresql needs to restart to effect the login changes parameter postgres@postgres\_host:/var/lib/pgsql$ pg\_ctl restart

**Command to generate report from a specific logfile**

ln -s /var/lib/pgsql/9.4/pgbadger-7.1/pgbadger /usr/bin/pgbadger pgbadger -f stderr -s 10 -T PGBadger-localhost.localdomain-20151202 -q -o

/var/www/html/pgbadger/report`date +\%Y-\%m-\%d`.html

/var/lib/pgsql/9.4/data/pg\_log/postgresql-2015-12-28.log

Everbridge pgbadger

ln -s /usr/local/pgsql/src/pgbadger-9.1/pgbadger /usr/local/bin/pgbadger

pgbadger -f stderr -s 10 -T PGBadger-localhost.localdomain-20151202 -q -o

/var/www/html/pgbadger/report`date +\%Y-\%m-\%d`.html

/data/pgdata/geodb/pg\_log/postgresql-2017-10-17\_133530.log

You can use a *cron* job to automatically generate pgbadger reports if logs have been created. Here is my /etc/cron.daily/pgbadger-daily\_process:

#!/bin/bash

#script to automatically generate pgbadger report according to the yesterday's PostgreSQL log

PGBADGER\_SCRIPT='/var/lib/pgsql/9.4/pgbadger-7.1/pgbadger'

PGBADGER\_OPTIONS="-f stderr -s 10 -T PGBadger-`hostname`-`date --date='yesterday'

+%Y%m%d` -q"

PGLOG\_PATH='/var/lib/pgsql/9.4/data/pg\_log' #PGLOG\_FILE\_NAME="postgresql-`date --date="yesterday" +%Y-%m-%d`.log" PGLOG\_FILE\_NAME="postgresql-`date --date= +%Y-%m-%d`.log"

PGBADGER\_RESULT\_FILE="/var/www/html/pgbadger-reports.epps-erp.com/pgbadger-`hostname`-

`date --date="yesterday" +%Y-%m-%d`".html

PGBADGER\_FULL\_COMMAND="$PGBADGER\_SCRIPT $PGBADGER\_OPTIONS -o $PGBADGER\_RESULT\_FILE

$PGLOG\_PATH/$PGLOG\_FILE\_NAME" #echo $PGBADGER\_FULL\_COMMAND

#exit

#################################################################

echo "$0 - `date` - INFO - job starts" if [ -e "$PGLOG\_PATH/$PGLOG\_FILE\_NAME" ]

then

echo "$0 - `date` - INFO - launching the creation of pgbadger report with

$PGLOG\_PATH/$PGLOG\_FILE\_NAME..."

$PGBADGER\_FULL\_COMMAND

if [ $? -eq 0 ] then

echo "$0 - `date` - OK - success to create pgbadger report" else

echo "$0 - `date` - ERROR - problem during the execution of:

$PGBADGER\_FULL\_COMMAND"

exit 1

fi else

echo "$0 - `date` - INFO - log $PGLOG\_PATH/$PGLOG\_FILE\_NAME not found" fi

echo "$0 - `date` - INFO - job done"

/var/www/html/pgbadger-reports.epps-erp.com/pgbadger-`hostname`-`date

--date="yesterday" +%Y-%m-%d`".html

postgres@postgres02:~/scripts/pgbadger$ ls -l total 8

-rwxr-xr-x 1 postgres postgres 1309 Nov 15 23:07 pgbadger.sh

-rwxr-xr-x 1 postgres postgres 1234 Nov 15 23:56 pgbadger.sh.save