Hedgehog Installation Guide 2.0.0b2

The following instructions describe how to install Hedgehog 2.0.0b2 on Ubuntu 14.04 Server and set up imports of data.

The server and PostgreSQL must both be configured to use UTC!

1. Installation

1.1. Requirements

1.1.1. Platform support

Version 2.0.0b2 of Hedgehog is currently only supported on Ubuntu 14.04.1 LTS Server.

1.1.2. Database support

Hedgehog has been tested with PostgreSQL 9.3.X

1.2. Hedgehog dependencies

1.2.1. Required packages

```
sudo add-apt-repository ppa:opencpu/rapache
sudo apt-get update
sudo apt-get install r-base r-base-core r-base-dev libcairo2-dev libxt-dev \
                     r-cran-ggplot2 r-cran-dbi r-cran-cairodevice r-cran-reshape\
                     r-cran-digest postgresql postgresql-client \
                     postgresql-server-dev-9.3 postgresql-contrib pgbouncer \
                     libboost-all-dev libtool libpqxx-dev apache2-mpm-prefork\
                     apache2-prefork-dev libapache2-mod-r-base libyaml-tiny-perl
```

1.2.2. Additional R packages

Hedgehog is tested against version 3.1.1 of R

Some R packages are needed that are not available through apt, they can be installed via R.

You will be prompted to choose a mirror site, or you can specify a repo by using a command of the form

```
install.packages("name", repos='http://cran.rstudio.com/')
```

Install:

```
sudo R
install.packages(c("brew","Cairo","googleVis","RPostgreSQL","R.utils","yaml","dplyr"))
q()
```

If you are prompted to *save workspace image y/n/c*, choose *no*.

The package versions that Hedgehog has been tested against are listed below alongside each package.

R Package	Supported Version
brew	1.0-6
Cairo	1.5-6
googleVis	0.5.8
RPostgreSQL	0.4
R.utils	2.0.2
yaml	2.1.13
dplyr	0.4.1

1.3. Install Hedgehog

1.3.1. Install and build code

Download the 2.0.0b2 release from github: https://github.com/dns-stats/hedgehog/archive/2.0.0b2.tar.gz

(If you downloaded the above URL with wget the tarball will be called 2.0.0b2.tar.gz)

```
tar -xzf hedgehog-2.0.0b2.tar.gz
cd hedgehog-2.0.0b2
./autogen.sh
mkdir build
cd build
../configure
make
sudo make install
sudo make install-rpg
```

ⓘ Note that 'make install-rpg' is required in addition to make install. It installs an R package to assist with the database connection.

If no <prefix> is specified in *configure* then it defaults to */usr/local/*

- You may notice the following message during the installation step: "*Error in find.package(pkgs, lib) : there is no package called ‘RPostgreSQLHelper’ Execution halted*". This can be safely ignored.

> For large installations it may be preferable to store the data files on a separate partition to the database. See the *User guide for an overview of the Hedgehog directory structure.
>
> In this case you may wish to mount a disk here or make use of a symbolic link at this stage of the installation.

1.3.2. System users and groups

Two system users are required for Hedgehog:

- Database owner - this is the user that will own the database created by hedgehog and the top level datafile directories (default user in this example is 'hedgehog')
- Read user - this should be the apache user which requires only read access to the database (default apache user in this example is called 'www-data')

```
# create a system user and group called hedgehog
sudo addgroup --system hedgehog
sudo adduser --system --ingroup hedgehog hedgehog
# put www-data user in group hedgehog
sudo adduser www-data hedgehog
# put hedgehog user in group www-data
sudo adduser hedgehog www-data
```

1.3.3. Configure Hedgehog

- Edit the *<prefix>/etc/hedgehog/hedgehog.yaml* configuration file to make sure the user names match the users specified in the previous step.
- Also configure the database parameters (port, name etc...) as required.

```
database:
  host         : /var/run/postgresql # specify a host for the postgresql DB. If this
                                     # begins with a slash, it specifies the
                                     # directory in which the socket file is stored.
  port         : 5432      # specify port for the postgresql DB.
  name         : hedgehog  # specify dbname for the postgresql DB.
  owner        : hedgehog  # specify a user to own the postgresql DB.
                           # This user will run refile_and_grok.
  owner_pass   :           # specify a password for the owner user if needed.
  read_user    : www-data  # specify a read user for the postgresql DB.
                           # This should be the apache user.
  read_pass    :           # specify a password for the read user if needed.
```

> Do not edit the 'directories' section of this file as in 2.0 it is auto-generated.

Note that there is also a sample conf file in the same directory to retrieve default settings: *<prefix>/etc/hedgehog/hedgehog.yaml.sample*.

- Change the ownership of the directory specified below so it is owned by the same user as the database, so for example in a default install:

```
sudo chown -R hedgehog:hedgehog <prefix>/var/hedgehog/
```

- Also check the parameters in the *<prefix>/etc/hedgehog/hedgehog_gui.yaml* file, which specifies parameters controlling the behaviour of the web front end. See the "Plot Caching" section in the user guide for a more detailed description of when plots are cached.

```
---
# YAML config for hedgehog GUI.
# NOTE: If this file is changed then apache must be restarted for the changes to take
effect
www:
  default_plot_type             : interactive # 'static'      -> png plots
                                              # 'interactive' -> googlevis plots
  default_interactive_plot_type : svg         # 'flash' -> plot requires flash
                                              # 'svg'   -> plot is SVG/VML and does
                                              # not require flash (but with svg plots
                                              # some legends do not wrap properly)
  use_plot_caching              : 1           # '1' -> true, use cached plots when
                                              # possible
                                              # '0' -> false, never use cached plots
  caching_delay_in_hours        : 1           # If 'use_plot_caching=1' then only
                                              # plots with an end time earlier than
                                              # this number of hours ago are cached.
                                              # More recent plots are not cached as
                                              # data may still be being imported
  presentation_delay_in_hours   : 0           # Number of hours behind now for which
                                              # the GUI will display data
```

1.3.4. Configure apache

You will probably need to edit the */etc/apache2/apache2.conf* file to enable access to the Hedgehog directories by adding *<Directory>* elements for *<prefix>/share/hedgehog>* and *<prefix>/var/hedgehog/www>*

Depending on your exact installation choices and apache configuration you may want to disable the default site using the following command:

```
sudo a2dissite 000-default.conf
```

- Add the Hedgehog configuration files to apache and enable the site (this file name can be changed if required to match any local apache policy):

```
sudo cp <prefix>/share/hedgehog/conf/hedgehog.conf /etc/apache2/sites-available/
sudo a2ensite hedgehog.conf
```

- Alter the permissions (add the line 'umask 002' to the apache envvars file):

```
> sudo vi /etc/apache2/envvars
# Enable extra permissions for Hedgehog
umask 002
```

Uploading XML via webdav using client certificates
For installs that **want to upload XML via webdav using client certificates** then:

```
sudo vi /etc/apache2/conf-available/hedgehog.conf
```

and uncomment the *"Alias /data"* line and following directory clause in this file.

Alter the apache umask so that www-data group members (i.e. hedgehog) can process the xml files :

Clearly you will also need to issue certificates to both Apache and to every collector. Help with this can be found User Guide.

apache/rapache write some of their logs to user.* so it can be useful to change the syslog config:

```
sudo vi /etc/rsyslog.d/50-default.conf
```

apache/rapache write some of their logs to user.* so it can be useful to change the syslog config:

Uncomment the line beginning 'user.*'.

- Finally, restart apache:

```
sudo service apache2 restart
```

1.3.5. Create the database

Ask your DBA to create the necessary database. This is a script to help them. It create the DATABASE, USERS and ROLES needed to run hedgehog (using default values), and can optionally accept a user specified database name and read/write user names.

```
<prefix>/sbin/hedgehog_database_create.sh
```

Ensure that the user names and passwords match those in hedgehog.yaml and match the users you created above. If required, also edit the pg_hba.conf file to allow the users access.

1.3.6. Create the database tables etc

For this version of Hedgehog the servers and nodes to be processed and displayed must be specified manually.

- Edit the the *<prefix>/etc/hedgehog/nodes.csv* to specify the servers, nodes and grouping to be used (example format is provided with entries commented out).

    - Note that the current GUI layout is optimised for nodes with short names (<6 characters) of the same length
- Then run the command below noting the following:
    - If you have historic data to import then use the -m flag to specify the month of the oldest data that will need importing. Otherwise the database tables will be created to hold data from this month onwards.
    - Note that this script will also create the directory structure for all the specified servers and nodes under the data directory if it does not exist

```
sudo -u <DB_OWNER> <prefix>/bin/hedgehog_database_init.sh
```

1.3.7. Test the Hedgehog front end

At this point you should be able to see the servers and nodes in the web front end at the URL http://<server-name>/hedgehog

2. Running Hedgehog

2.1. Import data

Hedgehog can process data in the following 3 ways:

Source format	Output format	
XML	Database	For real time uploads
DAT	Database	For import of historic data
XML	DAT	For backwards compatibility with DSC

In each case the *<prefix>/bin/refile_and_grok.sh* script is used, it is simply given different parameters:

```
> refile_and_grok.sh -h

refile_and_grok - finds all input files in the working directory and processes to
output format

-w Working directory to search for input files (default: <prefix>/var/hedgehog/data)
-i Input file format <XML|DAT> (default: XML)
-o Output file format <DAT|DB> (default: DB)
-c Non-interactive mode - use this flag when being run by a cron job
-s Start date from which to process incoming data (XML input only)
-r Disable processing of rssac datasets. Default is to process all datasets.
-R Reserved processors. Number of CPUS processors to exclude from import (default 0).
-h Show this help.
```

2.1.1. Importing historical .DAT data

```
sudo -u hedgehog <prefix>/bin/refile_and_grok.sh -i DAT
```

Be aware that this can take a long time if there is a significant amount of historic data and it may be advisable to run this in stages.

2.1.2. Importing real-time .XML data

2.1.2.1. Manually

- This can be done manually by running the *refile_and_grok.sh* script (consider running this nohup as it may take a while depending on how much data there is to process).

```
sudo -u hedgehog <prefix>/bin/refile_and_grok.sh
```

- A snapshot of the progress of the data import can be generated by running the command below:

```
sudo -u hedgehog <prefix>/bin/hedgehog_import_create_summary.sh
```

2.1.2.2. Automatically

- Configure a regular cron job for *refile_and_grok.sh* as shown below

2.2. Maintenance

2.2.1. Cron jobs

In 2.0 several cron jobs need to be configured. They should run as the 'hedgehog' system user.

```
sudo -u hedgehog crontab -e
```

Below is an example crontab for a typical system.

> Note that the *hedgehog_manage_partitions_cron.sh* script **MUST** be configured to run at least once a month.

```
# REQUIRED:
# Import XML data every 15 mins
00,15,30,45 * * * * <prefix>/bin/refile_and_grok.sh -c >>
/home/hedgehog/refile_and_grok_xml_to_db.sh.log 2>&1
# Twice monthly job to make sure the DB tables for next month are created
# ahead of time
0 6 15,28 * * <prefix>/bin/hedgehog_manage_partitions.sh >>
/home/hedgehog/hedgehog_manage_partitions.sh.log 2>&1

# OPTIONAL:
# Daily job to create cached plots for the previous day to make loading common plots
# quicker. Run a few hours after midnight so all data is uploaded.
0 4 * * * <prefix>/bin/hedgehog_plotcache_generate_cached_plots.sh -D >>
/home/hedgehog/hedgehog_plotcache_generate_cached_plots.sh.log -D 2>&1
# Daily job to generate RSSAC reports. By default report is generated
# for a single day 1 week ago
0 1 * * * <prefix>/bin/hedgehog_rssace_generate_reports.sh -D >>
/home/hedgehog/hedgehog_rssac_generate_reports.sh.log 2>&1
# Monthly job to tar up processed xml directories
0 2 1 * * <prefix>/bin/hedgehog_datafiles_rm_empty_xml_dirs.sh -D >>
/home/hedgehog/hedgehog_datafiles_rm_empty_xml_dirs.sh.log 2>&1
# Monthly job to remove empty xml directories that are older than 7 days old
0 2 7 * * <prefix>/bin/hedgehog_datafiles_tar_old_xml.sh -D >>
/home/hedgehog/hedgehog_datafiles_tar_old_xml.sh.log 2>&1
```