Hedgehog Installation Guide 2.0.0rc1

The following instructions describe how to install Hedgehog 2.0.0rc1 on Ubuntu 14.04 Server and set up imports of data.

1.1. Requirements

1.1.1. Platform support

Version 2.0.0rc1 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. The installation of this of this is not covered in detail since it may or may not be co-located with the other hedgehog components. If required, it can be installed using

```
sudo apt-get install postgresql
```

> The Hedgehog servers and PostgreSQL must both be configured to use UTC!

1.1.3. Hedgehog source code

Download the 2.0.0rc1 release from github: https://github.com/dns-stats/hedgehog/archive/2.0.0rc1.tar.gz

(If you downloaded the above URL with wget the tarball will be called 2.0.0rc1.tar.gz)

1.2. Hedgehog components

Hedgehog comprises 2 components which can be run on the same or different machines:

- Data Manager
    - Scripts for database creation and management
    - XML/DAT processing
- Web front-end

Both components can be configured to connect to a remote database server, so the database does not need to be co-located with either component.

1.3. Hedgehog dependencies

> It is necessary to only select the dependencies require for the specific components being installed.

```
# common dependancies
sudo apt-get install autoconf libtool make postgresql-client libyaml-tiny-perl
# data manager dependancies
sudo apt-get install g++ libboost-program-options-dev \
                     libboost-system-dev libboost-filesystem-dev libpqxx-dev
# web front-end dependancies
sudo add-apt-repository ppa:opencpu/rapache
sudo apt-get update
sudo apt-get install libapache2-mod-r-base r-cran-ggplot2 r-cran-dbi \
                     r-cran-cairodevice r-cran-reshape r-cran-digest \
                     r-base r-base-core libcairo2-dev libxt-dev
```

1.4. Build Hedgehog

> - By default, both components are built.
> - Use the --disable-web and/or --disable-importer configure options to select which components are actually built.
> - All installs get the hedgehog.yaml configuration file.

Building Hedgehog

```
tar -xzf hedgehog-2.0.0b2.tar.gz
cd hedgehog-2.0.0b2
./autogen.sh
mkdir build
cd build
../configure [--prefix=] [--disable-web] [--disable-importer]
make
sudo make install
```

1.5. Create a 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>/libexec/hedgehog/database_create
```

1.6. Create system users and groups

Two system users are required for Hedgehog, depending on what components are installed:

User	Default	Component	Note
Database owner	hedgehog	Data Manager	This is the user that will own the database created by hedgehog and the top level datafile directories.
Database read user	www-data	Web front-end	This defaults to the same as the default apache user

```
# create a system user and group called hedgehog
sudo addgroup --system hedgehog
sudo adduser --system --ingroup hedgehog hedgehog
```

1.7. Configure Hedgehog

- For all installs, edit the *<prefix>/etc/hedgehog/hedgehog.yaml* configuration file to make sure the user names match the users specified in the previous steps.
- Also configure the database parameters (port, name etc...) as required.
- If using passwords stored in this file, be aware that this file is readable by all users.

```
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.
                           # [Required for Database and Importer components]
                           # [Required for web-front end RSSAC reporting]
  owner_pass   :           # specify a password for the owner user if needed.
  read_user    : www-data  # specify a read user for the postgresql DB.
                           # [Required for web-front end component]
  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.

1.8. Data Manager

1.8.1. Specify the nodes/servers

> 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* file 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

1.8.2. Directory permissions

Change the ownership of the directory specified below so it is owned by the same user as the database:

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

> You may also need to alter the permissions on this directory to allow uploads via your chosen mechanism.

1.8.3. Create the database tables

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/hedgehogctl database_init
```

1.9. Web component

1.9.1. Install R packages

These instructions are for R packages that must be installed using R's built-in compilation tools (there is no Ubuntu package for them). To install interactively:

```
sudo R
install.packages(c("brew","Cairo","googleVis","RPostgreSQL","R.utils","yaml","dplyr
"))
# you will be prompted to choose a mirror site for the repo before proceeding...
install.packages("<prefix>/share/hedgehog/R_packages/RPostgreSQLHelper",
repos=NULL)
q()
# If you are prompted to save workspace image y/n/c, choose no.
```

For scripting purposes a repo can be specified by using a command of the form

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

Hedgehog is tested against version 3.1.1 of R. 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

1.9.2. Web configuration

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
```

1.9.3. Directory permissions

Change the ownership of the directory specified below so it is owned by the read user of the database:

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

1.9.4. Configure apache

> You will probably need to edit the */etc/apache2/apache2.conf* file to enable access to the Hedgehog directories by adding <*Director y*> 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
```

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
```

Uncomment the line beginning 'user.*'.

- Finally, restart apache:

```
sudo service apache2 restart
```

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

1.10. Importing 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* script is used, it is simply given different parameters:

```
> refile_and_grok -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.
```

1.10.1. Importing historical .DAT data

```
sudo -u <DB_OWNER> <prefix>/bin/refile_and_grok -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.

1.10.2. Importing real-time .XML data

1.10.2.1. Manually

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

```
sudo -u <DB_OWNER> <prefix>/bin/refile_and_grok
```

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

```
sudo -u <DB_OWNER> <prefix>/bin/hedgehogctl datafiles_create_summary
```

1.10.2.2. Automatically

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

1.11. Cron jobs

In 2.0 several cron jobs need to be configured.

1.11.1. Data Manager

Below is an example crontab for a typical data manager install (*sudo -u <DB_OWNER> crontab -e*).

> **Note that the *database_manage_partitions* script MUST be run at least once a month or the import will fail.**

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

# OPTIONAL [DATABASE]:
# Daily job to process RSSAC data. By default data is processed
# for a single day 1 week ago
0 1 * * * <prefix>/bin/hedgehogctl database_process_rssac_data -D >>
/home/hedgehog/database_process_rssac_data.log 2>&1
# OPTIONAL [IMPORTER]:
# Monthly job to tar up processed xml directories
0 2 1 * * <prefix>/bin/hedgehogctl datafiles_rm_empty_xml_dirs -D >>
/home/hedgehog/datafiles_rm_empty_xml_dirs.log 2>&1
# Monthly job to remove empty xml directories that are older than 7 days old
0 2 7 * * <prefix>/bin/hedgehogctl datafiles_tar_old_xml -D >>
/home/hedgehog/datafiles_tar_old_xml.log 2>&1
```

1.11.2. Web component

Below is an example crontab for a typical web front-end install (*sudo -u <DB_READ_USER> crontab -e*)

```
# OPTIONAL [WEB]:
# 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/hedgehogctl plotcache_generate_cached_plots -D >>
/home/hedgehog/plotcache_generate_cached_plots.log -D 2>&1
# Daily job to generate RSSAC reports. By default report is generated
# for a single day 1 week ago
0 2 * * * <prefix>/bin/hedgehogctl rssac_generate_reports -D >>
/home/hedgehog/rssac_generate_reports.log 2>&1
```