Hedgehog Installation Guide

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

1. Requirements

1.1. Platform support

Hedgehog is currently only supported on Ubuntu 14.04 LTS Server.

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 not already installed, it will be installed as a dependancy of the hedgehog-database package.

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

1.3. General

1.3.1. Home directory permissions

Some users set the permissions on their home directory such that other users cannot read or execute that directory. In this case when running scripts that update the database as the *hedgehog* or *postgres* (user via 'sudo -u <user>') meaningless messages are generated from psql about being unable to cd into logged in users home directory. To avoid this spurious output change directory to one that allows other users to read/execute (e.g. /tmp) and run the scripts from there.

2. Hedgehog Installation

2.1. Hedgehog packages

The Hedgehog PPA is located at

```
sudo add-apt-repository ppa:dns-stats/presenter
```

Hedgehog comprises 3 main components which can be run on the same or different machines. Configure the additional PPA's as required and then install the main packages from the list below:

Component	Description	Main Package	Optional Package	Additional Required PPAs
Database	PostgreSQL database for Hedgehog	hedgehog-database		sudo add-apt-repository ppa:ondrej/pkg-nlnetlabs
Data Manager	Scripts for database population and management (including rssacd) Scripts for XML/DAT processing	hedgehog-data-manager		
	(Optional) Apache config for webdav based XML uploads		hedgehog-webdav-upload	
Web GUI	Web GUI front-end	hedgehog-gui		sudo add-apt-repository ppa:opencpu/rapache

Notes:

- Multiple Web GUI's can be connected to the same database.
- The upload mechanism for XML/DAT files is for the user to decide (ssh, rsync, Apache, etc.). A package for upload using Apache is provided for convenience.

For reference the other Hedgehog packages are:

Description	Required by	Package
Common package	All main hedgehog packages	hedgehog-common
RPostgresHelper	Web Gui	r-hedgehog-rpostgresqlhelper
Documentation		hedgehog-doc

Two system users are used by Hedgehog, which are created by the packages:

User	Default	Component	Note	Created by
Database owner	hedgehog	Data Manager	This is the user that will own the database created by Hedgehog and the top level datafile directories.	hedgehog-common package
Database read user	www-data	Web front-end	This defaults to the same as the default apache user	apache2 package installation

2.2. Web GUI only - Manual install of 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: (Some can repositories don't contain packages for all versions of R so if this fails quit R and re-try a different repo)

```
sudo R
install.packages(c("brew","Cairo","googleVis","R.utils","yaml","plyr"))
# The latest rcpp build is broken so hard code the last working versions
packageurl <- "https://cran.r-project.org/src/contrib/Archive/Rcpp/Rcpp_0.12.5.tar.gz"
install.packages(packageurl, repos=NULL, type="source")
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. Hedgehog requires at least the versions below of each package.

R Package	Supported Version
brew	1.0-6
Cairo	1.5-9
googleVis	0.5.10
R.utils	2.2.0
yaml	2.1.13
plyr	1.8.3

3. Hedgehog configuration

- For the Web GUI and Data Manager components, ensure the /etc/hedgehog/hedgehog.yaml configuration file contains usernames and passwords that will match those configured in the database (see next section).
- Also configure the database parameters (host, port and name) 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.
                           # [Required for Data Manager component]
  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 it is auto-generated.

Depending on you PostgreSQL configuration you may need to add passwords to this file. If you do then be aware that, by default, this file is readable by all users. You should restrict access to just the hedgehog user on a Data Manager only machine or the hedgehog and www-data user on a combined Data Manager and Web GUI.

3.1. Database configuration

3.1.1. Create a database

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

```
sudo -u postgres /usr/bin/hedgehog_database_create

#If you want to use passwords do something like this or use peer authentication
sudo -u postgres psql
alter user hedgehog password 'hedgehog123';
alter user "www-data" password 'www123';
```

Modify the PostgreSQL configuration as so:

```
sudo vi /etc/postgresql/9.3/main/postgresql.conf
# uncomment and set 'extra_float_digits = 1'
sudo service postgresql reload
```

We recommend that the 'pgtune' tool is used to obtain an optimal configuration for PostgreSQL. For example:

```
pgtune -i /etc/postgresql/9.3/main/postgresql.conf -c 200 -T DW
```

Also some queries to the database trigger DNS lookups from functions in the database. It may be optimal to run a local resolver such as Unbound.

3.2. Data Manager configuration

3.2.1. Specify the nodes/servers

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

- Edit the the */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

3.2.2. Directory permissions

The /var/lib/hedgehog/data/ is used to store incoming XML files

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

3.2.3. 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
- (Note the insertion of the GeoIP data can take some time)

```
sudo -u hedgehog /usr/bin/hedgehogctl database_init
```

Then run the following script to make sure the list of delegated TLDs in the database is up to date.

```
sudo -u hedgehog /usr/bin/hedgehogctl database_update_tlds_from_zone 2>/dev/null
```

3.2.4. (Optional) Webdav upload

If using webdav to upload XML files then add the following to the /etc/apache2/envvars file:

```
umask 002
```

Then enable the dependancies:

```
sudo a2enmod dav
sudo a2enmod ssl
suod a2enmod dav_fs
```

And then enable the hedgehog webdav site:

```
sudo a2ensite hedgehog-webdav-upload
```

3.3. Web GUI configuration

Check the parameters in the/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)
  default_node_grouping         : instance    # choose from 'none', 'instance', 'city' or
                                              # 'country'
  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 importing
  presentation_delay_in_hours   : 0           # Number of hours behind now for which the
                                              # GUI will display data
  support_url                   :             # configurable target for "Support" external
                                              # link on Homepage. The default dns-stats.org
                                              # issue tracker used if left blank
  default_server                :             # Optionally specify the default server to
                                              # use in the server drop-down
                                              # (default is first alphabetically)
```

3.3.1. Apache configuration

Depending on your exact installation choices and apache configuration you may need 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 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, reload apache:

```
sudo service apache2 reload
```

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

4. 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 */usr/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 working directory and processes to output format


-w Working directory to search for input files (default: )
-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 data. Default is to process all data.
-R Reserved processors. Number of CPUS processors to exclude from import (default 0).
-a Append output to the refile_and_grok.stdout file (default is overwrite)
-h Show this help.
```

4.1. Importing historical .DAT data

```
sudo -u <DB_OWNER> /usr/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.

4.2. Importing real-time .XML data

4.2.1. 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> /usr/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> /usr/bin/hedgehogctl datafiles_create_summary
```

4.2.2. Automatically

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

4.3. Importing zone-size and load-time data for RSSAC

To do this run the *rssacd* demon specifying the server of interest. This listens for NOTIFY messages and after receiving one:

- obtains the zone size from the notifying server by performing an XFR and
- probes each node listed for the configured server with an IP address to calculate the zone load time

> This requires that management IP addresses are configured for the nodes via the nodes.csv file (also note that *rssacd* needs restarting if new nodes are added).

rssacd can be run from the command line for testing, or using an init script.

From the command line:

```
/usr/sbin/rssacd --log ~/var/log/hedgehog/rssacd.log -s <server-name> -z
<fully_qualified_zone_name>
```

From an init script:

- Configure the server name, TSIG key, etc. in the /etc/hedgehog/rssac.conf file

Then run:

```
update-rc.d rssacd defaults
```

5. Cron jobs

In 2.1.1 several cron jobs need to be configured.

5.1. Data Manager component

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 to create the tables for next month or the import will fail.**

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

# OPTIONAL:
# Daily jobs to process RSSAC data. By default data is processed
# for a single day 1 week ago. Must be run before the rssac_generate_reports script (see Web GUI
Component cron jobs)
0 1 * * * /usr/bin/hedgehogctl database_process_rssac_data -D >>
/var/log/hedgehog/database_process_rssac_data.log 2>&1
# Weekly job to update the delegated TLDs from the IANA database.
0 2 * * 0 <prefix>/bin/hedgehogctl database_update_tlds_from_zone >>
/var/log/hedgehog/database_update_tlds_from_zone.log 2>&1
# Monthly job to update the geoIP database.
0 3 1 * * <prefix>/bin/hedgehogctl database_update_geoip >>
/var/log/hedgehog/database_update_geoip.log 2>&1
# Monthly job to remove empty xml directories that are older than 7 days old
0 2 1 * * /usr/bin/hedgehogctl datafiles_rm_empty_xml_dirs -D >>
/var/log/hedgehog/datafiles_rm_empty_xml_dirs.log 2>&1
# Monthly job to tar up processed xml directories
0 2 7 * * /usr/bin/hedgehogctl datafiles_tar_old_xml -D >>
/var/log/hedgehog/datafiles_tar_old_xml.log 2>&1
```

5.2. Web GUI component

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

```
# 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 * * * /usr/bin/hedgehogctl plotcache_generate_cached_plots -D >>
/var/log/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. Must be run after the database_process_rssac_data script (see
Data Manager cron jobs)
0 3 * * * /usr/bin/hedgehogctl rssac_generate_reports >>
/var/log/hedgehog/rssac_generate_reports.log 2>&1
```