**Purpose:**

To install on a suitable server the most-recent approved PostgreSQL database with the PostGIS spatial extension (for speedy spatial queries) on a Linux operating system. PostGIS will be used to store geospatial metadata and data. Initial access to the database is through ‘sudo –u postgres psql’.

In many Linux versions, it is **strongly** **preferred** PostgreSQL/PostGIS be installed via a Software Manager. This process follows those instructions closely**.** As part of the installation, and test table will be created and then dropped upon successful confirmation of installation.

If using Ubuntu (for instance, Linux Mint), PostgreSQL may be downloaded and installed via the single command from the shell or the graphical Software Manager:

* apt-get install postgresql-9.x (Debian, Ubuntu)
* yum install postgresql9x and postgresql9x-server (Red Hat and CentOS)

“x” is the desired version. This will be a reasonably-current version, but possibly not as current as described in the process below.

Optionally for the yum install, initialize the database and enable automatic start:

* service postgresql-9.6 initdb
* chkconfig postgresql-9.6 on
* service postgresql-9.6 start

At the conclusion of the PostgreSQL/ PostGIS installation, the application will be installed as a service and be running whenever the computer and operating system is active.

The following process closely follows the instructions on the PostgreSQL website.

| **Step** | **Major Activity** | **References, Forms and Details** |
| --- | --- | --- |
| **1** | Verify what version of Ubuntu you are running  sudo lsb\_release –a | at time of writing Ubuntu ‘xenial’ |
| **2** | Go to PostgreSQL website: | <http://www.postgresql.org/download/> |
| **3** | Select appropriate download instructions for the operating system version (xenial or otherwise) | Common Linux versions are   * CentOS 6&7 command line * Mint 18+ => Ubuntu Xenial |
| **4** | Add a line to sources.list for the repository.  e.g. for Linux Mint 18.2 command is:   * sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt xenial-pgdg main" >> /etc/apt/sources.list | * appends to existing list, if any existing |
| **5** | Import the repository signing key, and update the package lists   * wget --quiet -O - \ https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add – * sudo apt-get update |  |
| **6** | The following will install postgresql 9.6, PostGIS 2.3, PGAdmin3 and additional supplied modules including the **adminpack** extension:   * sudo apt-get install postgresql-9.6-postgis-2.3 pgadmin3 postgresql-contrib-9.6 | * A user postgres will be created, with no logon properties * /usr/bin/   + apps like pg\_... |
| **7** | To install pgRouting 2.1 for postgres 9.6:   * sudo apt-get install postgresql-9.6-pgrouting | * pgRouting extends the [PostGIS](http://www.postgis.net/) / [PostgreSQL](http://www.postgresql.org/) geospatial database to provide geospatial routing functionality * really intended for road networks, but may be useful when reconstructing vessel tracks |
| **8** | Enable the Adminpack extension:   * sudo -u postgres psql * CREATE EXTENSION adminpack; | **sudo -u postgres psql**  **is the standard default logon** |
| **9** | Test by creating a PostGIS database:   * CREATE DATABASE gisdb; * \connect gisdb; * CREATE EXTENSION postgis; * SELECT postgis\_full\_version();   \q | Should see something like:  POSTGIS="2.2.1 r14555"  GEOS="3.4.2-CAPI-1.8.2 r3921"  PROJ="Rel. 4.8.0, 6 March 2012"  GDAL="GDAL 1.10.1, released 2013/08/26"  LIBXML="2.9.1" LIBJSON="0.11.99"  RASTER  (1 row) |
| **10** | Install pgRouting to test database and examine:   * CREATE EXTENSION pgrouting; * SELECT \* FROM pgr\_version(); |  |
| **11** | Exit the psql console:   * \q |  |
| **12** | To allow database access to external clients (not recommended for GeoNetwork/GeoServer application):  * sudo vi /etc/postgresql/9.6/main/pg\_hba.conf   If you need external access, scroll to the bottom of the pg\_hba.conf file and add a line like this (which will allow all clients with md5 password encrypt authentication (right after the local rules):   * hostssl all all 0.0.0.0/0 md5   Save and exit the text editor | * “vi” text editor shown; any could be used |
| **13** | Edit the postgresql.conf file and change listen\_addresses line to   * a specific ip of the server,  **or** * '\*' to listen on all ips   If you change ip or port, you need to do a service restart.   * sudo service postgresql restart |  |
| **14** | Check location of configuration files. From the psql console(see above):   * SELECT name, setting FROM pg\_settings where category='File Locations'; | **name | setting**  **config\_file**  /etc/postgresql/9.5/main/postgresql.conf  **data\_directory**  /var/lib/postgresql/9.5/main  **external\_pid\_file**  /var/run/postgresql/9.5-main.pid  **hba\_file**  /etc/postgresql/9.5/main/pg\_hba.conf  **ident\_file**  /etc/postgresql/9.5/main/pg\_ident.conf |
| **15** | Optional. Create a new database super user to use instead of the default postgres user   * sudo su - postgres * createuser -d -E -i -l -P -r -s yourUserName   Enter your new password when prompted  Log out as postgres user:   * exit |  |
| **16** | Use PGAdmin to log into database server:   * Click the Add a connection to server button on the menu in the top-left corner | * + **Name**: nickname you want to give your connection   + **host**: localhost (until you change the connection settings)   + **port**: 5432 by default   + maintenance DB: postgres by default   + **username**: whatever you chose in the step above   + **password**: whatever you chose in the step above |
| 17 | In the operating system, set variables   * PGHOST - default host (server) * PGUSER - default user (role) * PGPORT – listen port * PGDATA - path to data directory * Set path to bin directory | As a Linux example, in ~/.bashrc include lines:  PGHOST=localhost  PGUSER=postgres  PGPORT=5432  PGDATA=/path/to/data  PATH=$PATH:/opt/PostgreSQL/bin  export PATH |