Getting started: Access to LAV-ESG postgres server

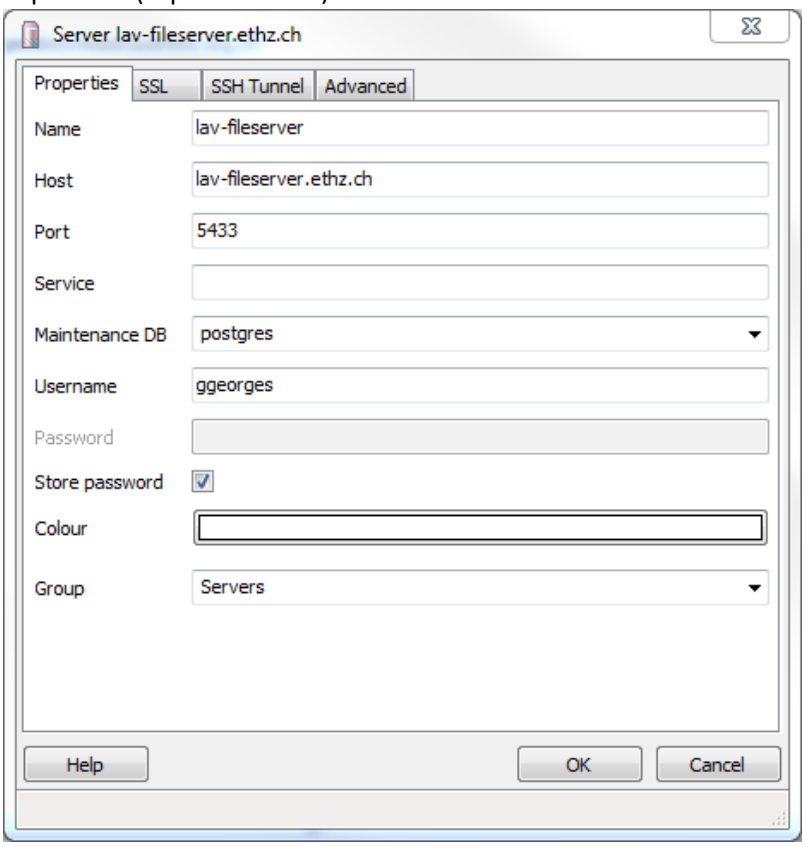
Nice that you chose to perform your project with the Energy Systems Group (ESG) of LAV. We are dealing with many different data sets that are stored on our postgres server. To access those data you need to follow the steps below.

Step 1: Getting the server certificate

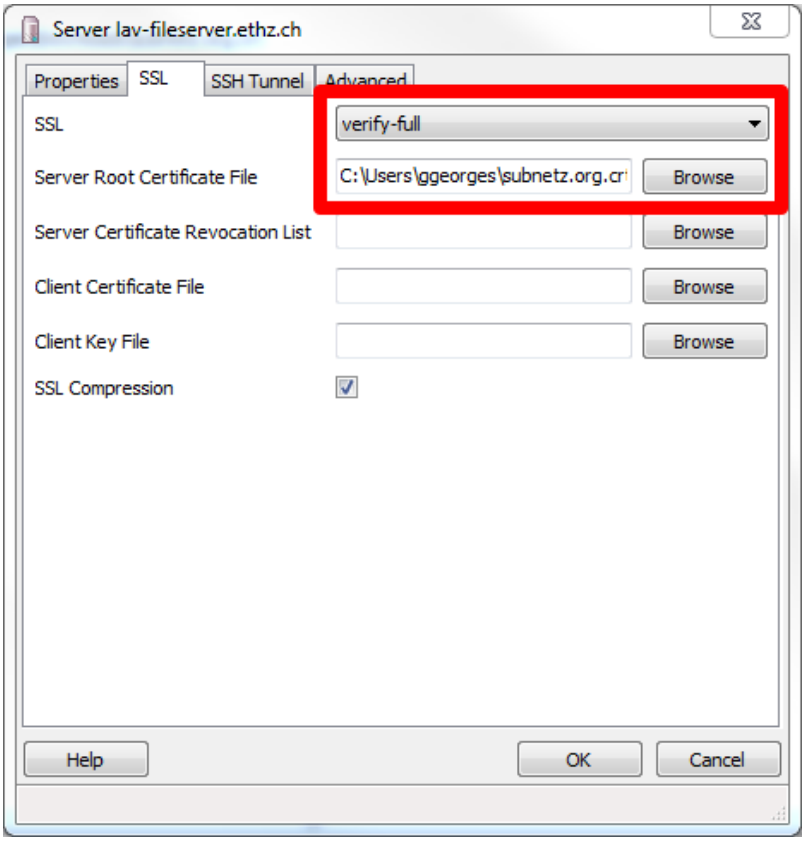
1. Download the following to a local directory (for example straight into your home directory 'C:\Users\[username]'): http://www.subnetz.org/misc/subnetz-CA.crt. Note that if you open this link in a browser, it will install the certificate locally, not download it. You must then export it from the browser:
    a. Firefox: open the 'options' dialogue. Under 'Advanced' and then 'Certificates', hit the 'View Certificates' button. Under 'Authorities', you'll find 'subnetz.org', and below it, again, 'subnetz.org'. Select it (click on it), then hit 'Export…'.
    b. Chrome: open the 'settings' page, click 'advanced settings' at the bottom (if not already displayed) and hit 'Manage certificates…' under 'HTTPS/SSL'. Find 'subnetz.org' under 'Intermediate Certification Authorities', select it (click on it) and hit the 'Export…' button.

Step 2: Access with pgAdmin

1. Install pgAdmin: http://www.pgadmin.org (under 'Download')
   (if you are performing this step in the 'Studentenraum' or if you don't have admin rights, install the portable version: https://sourceforge.net/projects/pgadminportable/)
2. Open pgAdmin and hit the 'Add Server' button (the one with the plug). Under the 'properties' tab, indicate the following. Use the username and password provided your supervisor (separate sheet).



3. In the ‘SSL’ tab, do the following adjustments (pointing ‘Server Root Certificate File’ to wherever you chose in ‘Step 1’)



4. After you hit OK, double-click the newly created icon in the ‘Object browser’. Double-clicking a database gives you access to its contents.

You are now able to see the structure and data contained in the fileserver. You can execute SQL queries directly with pgAdmin or use python (shown below).

Step3: Accessing the server from within python

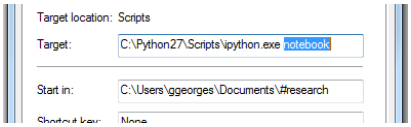
1. Install ‘psycopg2’
    a. If using the Anaconda distribution (enter in prompt):
    `conda install psycopg2`
    b. If (a) doesn’t work, or if you use legacy python (from python.org): get the binaries from ‘http://initd.org/psycopg/download’ – they have been known to cause problems in the past, especially the 64-bit version
    c. If (b) doesn’t work: get the ‘whl’ file matching your python interpreter from www.lfd.uci.edu/~gohlke/pythonlibs (‘Ctrl+F’ for ‘psycopg’). Then install the downloaded file using:
    `pip install [name of the file you just downloaded]`

With ‘psycopg2’ you basically installed all required packages to connect to the LAV fileserver. There is an additional package available, which can be downloaded from a github repository. It is an in-house package, customized for working with the ESG data sets. We highly recommend you to install and use it (especially in combination with jupyter notebooks).

It is up to you which interpreter you’ll use – we recommend ‘jupyter notebook’ since it provides a nice form of documentation. A potential alternative is ‘Spyder’ which is quite similar to Matlab. If you like to give ‘jupyter notebook’ a try, please continue with number 2 and 3 below. Otherwise you can continue with 4 directly.

2. Install LAV-ESG package
    a. Download latest 'whl' from:
    https://github.com/LAV-ESG/ipython_psql_extension/releases/latest
    b. Install the downloaded file using:
    `pip install [name of the file you just downloaded]`
    (make sure you are in the correct directory in the command prompt)

3. Getting started with 'jupyter notebook'
    a. Create shortcut to 'jupyter.exe' (should be in the 'Scripts' directory of your Python installation on windows); in the shortcut's "Properties" dialog, append the word "notebook" to the "Target" field and set the 'Start In' field to whatever directory you work in; finally, change the Icon to 'Scripts\ipython.ico'



    b. Download the gettingStarted notebook from:
    https://github.com/LAV-ESG/getting-started/releases/latest
    and store it somewhere in the same directory as the *start in* specified in a.
    c. Double click on shortcut (should open a new tab in your browser), select the notebook in the browser, open it and go through it. If you decide to work with jupyter notebooks and the LAV-ESG extension you can skip point 4.

4. Connection example code:
    Below you find an example how to connect to the server using python. If you are using the in-house package, you can directly use the predefined functions.

```
In [11]: import psycopg2

In [12]: def connect_to_postgres(dbname='mobility', host="lav-fileserver", port=5433, user="lav_db", password=None, sslcert=None):

             if password is None:
                 import getpass
                 password = getpass.getpass("password:")

             dsn = ["dbname='{}'".format(str(dbname)),
                    "host='{}'".format(str(host)),
                    "port='{}'".format(str(port)),
                    "user='{}'".format(str(user)),
                    "password='{}'".format(str(password))]
             if sslcert is not None:
                 dsn.append("sslcert='{}'".format(str(sslcert)))
             conn = psycopg2.connect(" ".join(dsn))
             return conn

In [*]: conn = connect_to_postgres(sslcert=r"C:\Users\ggeorges\subnetz-CA.crt")
        password: ••••••••••••••
```

NOTE:

Next to the host, username and password, your DSN will have to explicitly specify the non-standard port 5433 (port=5433), the path to the server root certificate file using the (sslcert='[path_to_subnetz.crt]'), and the database you want to use via 'dbname'. The psycopg documentation http://initd.org/psycopg/docs/ provides a good starting point (start with the 'Basic module usage' section).

IMPORTANT:

For security reasons, generally try to use 'getpass' to retrieve the password from the user at runtime whenever possible (instead of providing the password in plaintext in the code). Also rely on the 'read access' credentials whenever possible.

Links to learn python:

- http://www.codecademy.com/en/tracks/python (kind of fun, fully web-based)
- https://docs.python.org/2/tutorial/index.html (the official tutorial; somewhat broader)
- https://www.youtube.com/watch?v=-p4CVtPZoPo (difference w.r.t. “normal” Python)
- https://www.youtube.com/watch?v=UWmZAAfXds4 (bit dry, but instructive video)
- http://scipy-lectures.github.io/index.html (great resource, but dry as bone)