Connecting to postgres

You should have

- a running postgresql 16 or 15 install on your machine
- pgAdmin installed

Server vs db

You connect to a postgres **server**

On that server you may have zero, one or multiple databases.

When you launch postgres on your local, you are actually launching the postgres server.

You connect and query the database on the command line using psql or in pgAdmin.

Users

The `postgres` user is the super user.

It can do anything on the server, create and manage databases, users, etc

Installing postgreSQL also creates a user with your machine user name : *username*

The *username* user is not as powerful as the *postgres* user

Attention, on Nac, there is no *postgres* user. The superuser is your *username*

Connecting to the server

You can connect in the terminal or with pgAdmin

In the terminal you use `psql` to connect

psql takes the following parameters

- -U username
- -h host (IP address or local host)
- -p port
- -d database_name

PostgreSQL port is by default 5432 and does not have to be specified each time.

so to connect to a local instance of a postgres server with user bob and database treesdb

> psql -U bob -d treesdb

to connect to a remote server IP: 123.456.789.246 same user and database

> psql -U bob -h 123.456.789.246 -d treesdb

check your connection

to see your current connection run

> psql -c "\conninfo"

if it complains of not knowing the database, add the *postgres* database as the database name

> psql -c "\conninfo" -d postgres

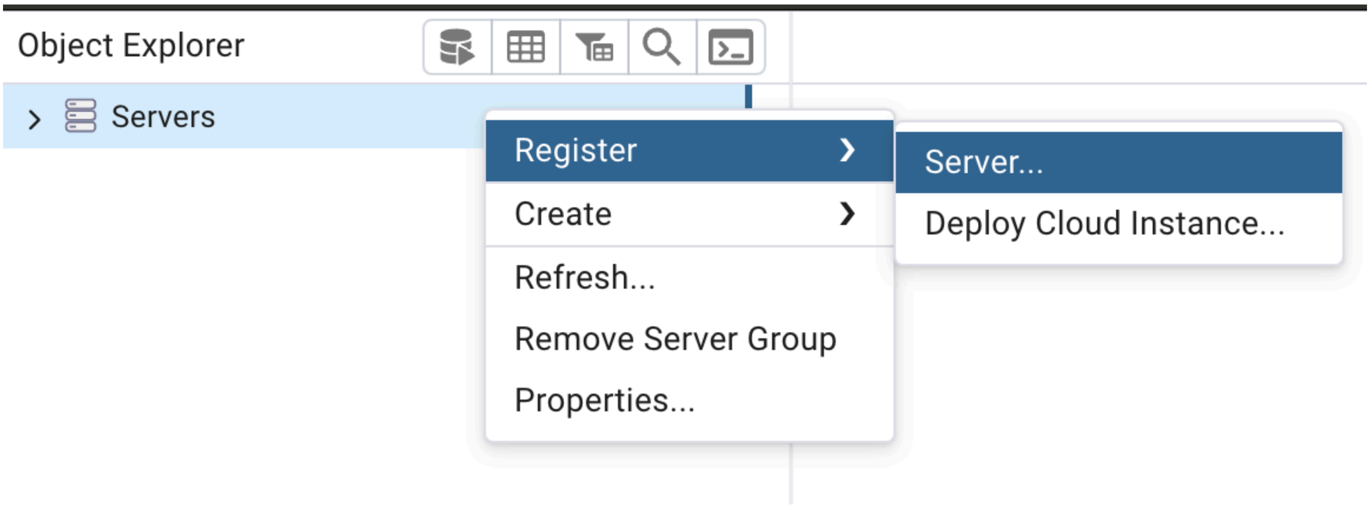
This should return:

> *You are connected to database "postgres" as user "alexis" via socket in "/tmp" at port "5432".*

in pgAdmin,

Open pgAdmin

b. Right-click on "Servers" in the left panel and choose "Create" > "Server..." c. In the "Create - Server" dialog:

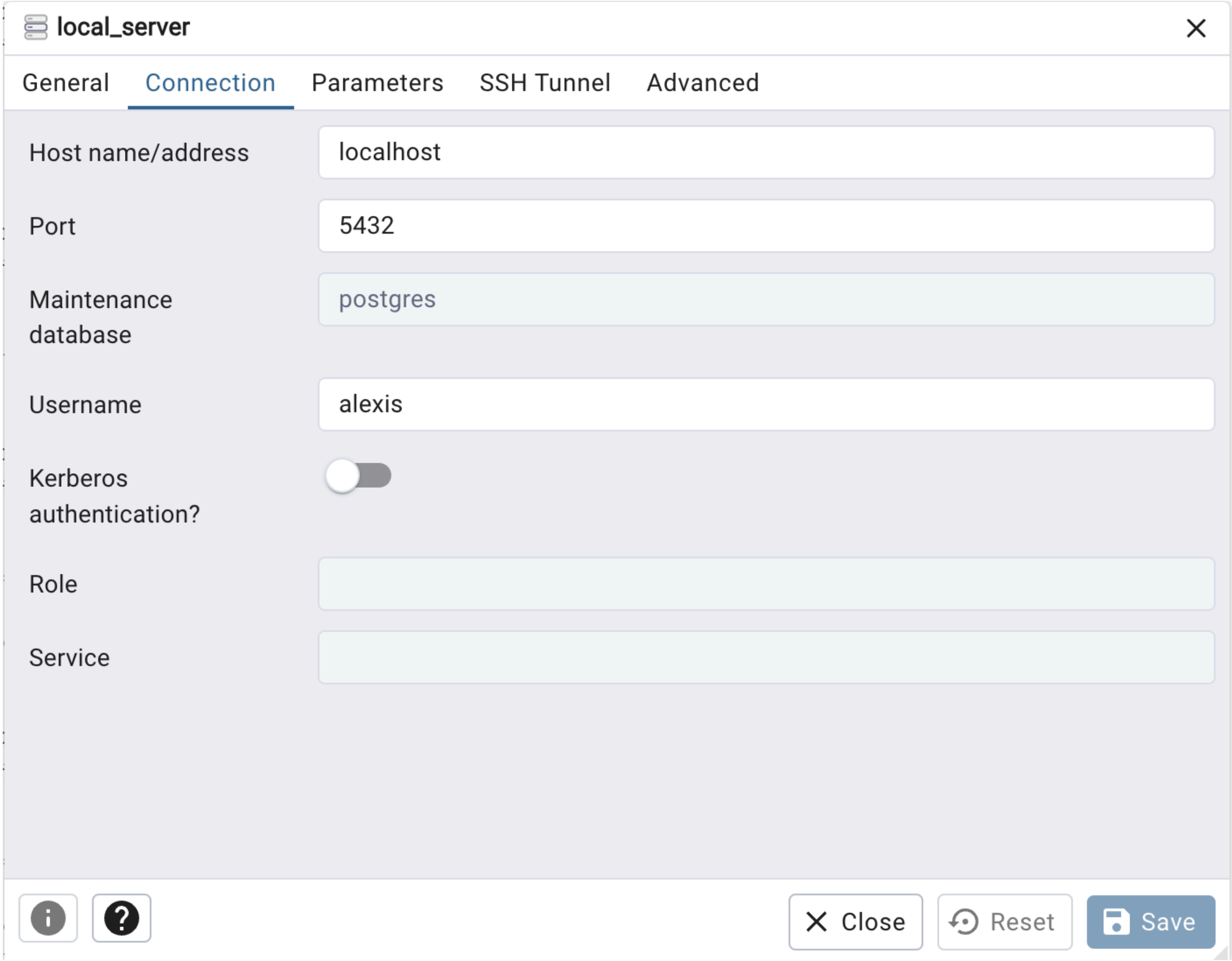


General tab:

Name: Give it a name like "Local PostgreSQL"

Connection tab:

- Host name/address: localhost
- Port: 5432 (default PostgreSQL port)
- Maintenance database: postgres
- Username: [your_username]
- Password: Leave blank if you haven't set one



Host name/address: localhost

Port: 5432

Maintenance database: postgres

Username: alexis

Kerberos authentication? ○

Role:

Service:

to connect to a remote server, just change the connection tab parameters (host, user, password)

On windows

- Install postgres and follow the instructions
- click on the psql exe file
- it will open a terminal window and prompt you with hostname
  - if you're connecting on local just press enter
  - if you're connecting to remote add the IP address

Same with the other parameters

Loading a csv dataset

The steps are

1. create the database. let's call it treesdb. you can do that either
   1. on the command line with `createdb treesdb`
   2. in psql, with CREATE DATABASE ...
2. create the table let's call it `trees`
3. upload the data from the csv file into the trees table with

\COPY ....

Restore a database from a *sql dump*

Download the treesdb_v01.sql.gz file in the github repo

go to: https://github.com/SkatAI/epitadb/tree/master/data click right on the filename and `save link as`

You can also clone the repo if you have git installed

> git clone git@github.com:SkatAI/epitadb.git

or just get the link to the file (click right on the filename and `copy link`) and use curl or wget to download the file

> wget https://github.com/SkatAI/epitadb/blob/master/data/treesdb_v01.sql.gz

Once you have the file on your local you can restore it.

But first you must create the database either in pgAdmin (click right on serrver name and create > database; check the encoding is UTF8) or with the query `sql CREATE DATABASE treesdb_v01 WITH OWNER = alexis ENCODING = 'UTF8' LOCALE_PROVIDER = 'libc' CONNECTION LIMIT = -1 IS_TEMPLATE =`

replace the owner name (alexis) with your username

The restore the database with

In the terminal

> pg*restore --username "your*username" --no-password --dbname "treesdb*v01" --section=pre-data --section=data --section=post-data --verbose "your path to/treesdb*v01.sql.gz"

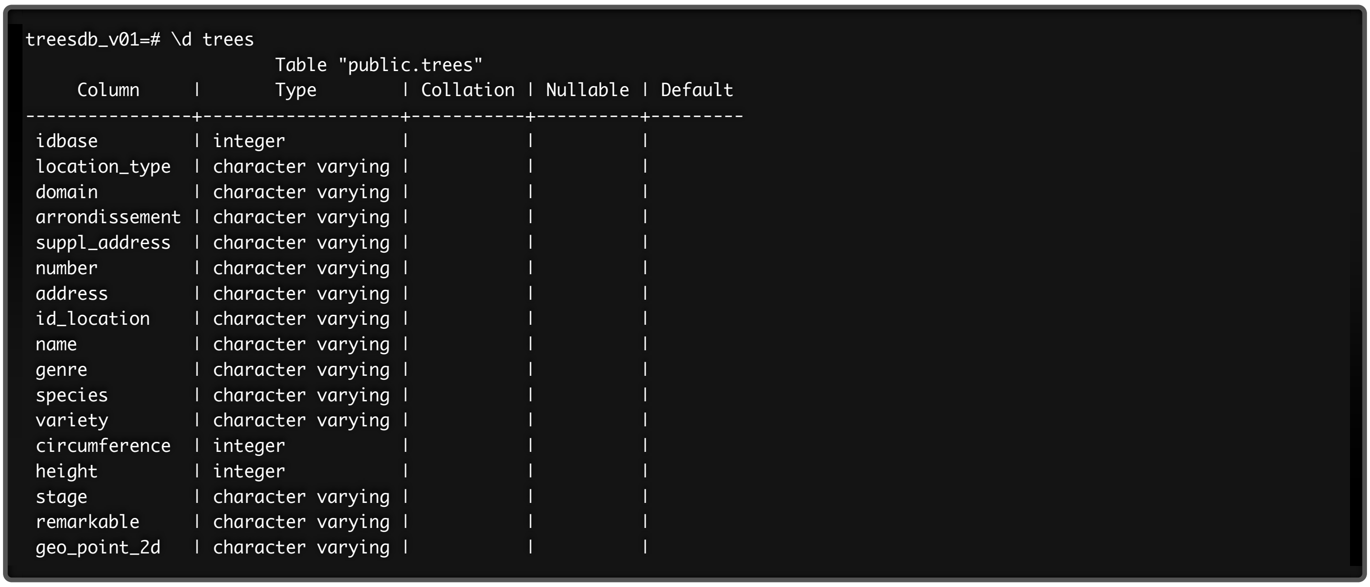
or in pgAdmin

In the restore dialog:

- Set "Format" to "Custom or tar"
- Browse and select your dump file (treesdb_v01.sql.gz).
- In the "Sections" tab, make sure "Pre-data", "Data", and "Post-data" are all checked. (not sure that's even required)

Check the data

You should have a single table called trees with the columns we saw last week.



```
treesdb_v01=# \d trees
                        Table "public.trees"
     Column     |       Type        | Collation | Nullable | Default
----------------+-------------------+-----------+----------+---------
 idbase         | integer           |           |          |
 location_type  | character varying |           |          |
 domain         | character varying |           |          |
 arrondissement | character varying |           |          |
 suppl_address  | character varying |           |          |
 number         | character varying |           |          |
 address        | character varying |           |          |
 id_location    | character varying |           |          |
 name           | character varying |           |          |
 genre          | character varying |           |          |
 species        | character varying |           |          |
 variety        | character varying |           |          |
 circumference  | integer           |           |          |
 height         | integer           |           |          |
 stage          | character varying |           |          |
 remarkable     | character varying |           |          |
 geo_point_2d   | character varying |           |          |
```

Accents and the Encoding

We need the database to be UTF8 encoded for the Paris trees data.

if you notice that the accent in some columns are not properly encoded for instance é is displayed as Ã©, à as Ã ...

> UPDATE trees SET column*name = convert*from(convert*to(column*name, 'LATIN1'), 'UTF8')

do that for the columns : name, genre, species, variety, address ...

you can also check that server, client and table are UTF8 encoded with

> SHOW server*encoding; SHOW client*encoding; SELECT pg*encoding*to*char(encoding) FROM pg*database WHERE datname = 'treesdb';

This is a version of the database with the proper encoding.

create the new database

- create a new database called treesdb_02
- set the user to your usename (should appear in the dropdown)
- set the encoding to UTF8

The SQL tab should show

```SQL
CREATE DATABASE treesdb_02
    WITH
    OWNER = alexis
    ENCODING = 'UTF8'
    LOCALE_PROVIDER = 'libc'
    CONNECTION LIMIT = -1
    IS_TEMPLATE = False;
```

restore the database

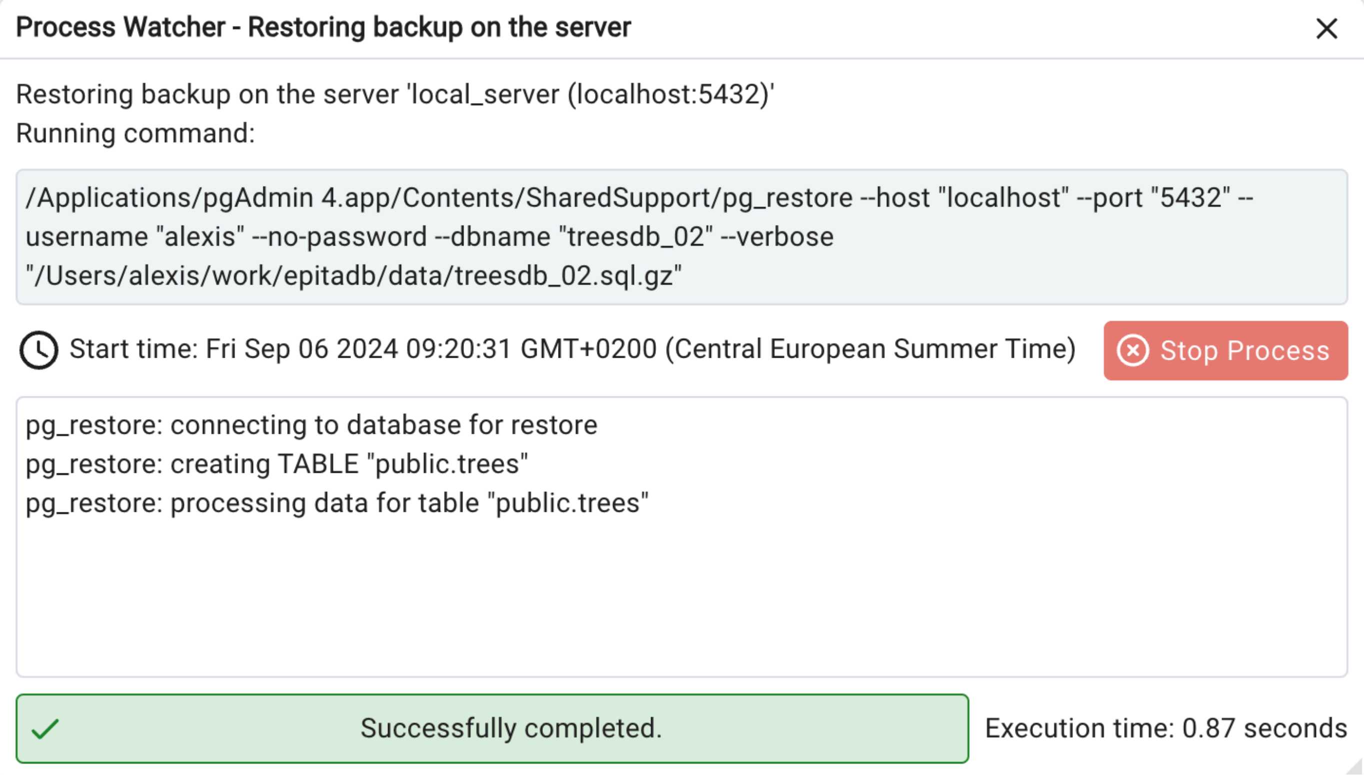
Then click right on the treesdb_02 database and click on restore

- select the filename treesdb_v02.sql.gz,
- select format custom or tar
- click restore

The command line equivalent is

```Bash
pg_restore --host "localhost" \
--port "5432" \
--username "alexis" \
--no-password \
--dbname "treesdb_02" \
"/Users/alexis/work/epitadb/data/treesdb_02.sql.gz"
```

while the db is restored, look at the process



Process Watcher - Restoring backup on the server

Restoring backup on the server 'local_server (localhost:5432)'
Running command:

```
/Applications/pgAdmin 4.app/Contents/SharedSupport/pg_restore --host "localhost" --port "5432" --username "alexis" --no-password --dbname "treesdb_02" --verbose "/Users/alexis/work/epitadb/data/treesdb_02.sql.gz"
```

Start time: Fri Sep 06 2024 09:20:31 GMT+0200 (Central European Summer Time)

```
pg_restore: connecting to database for restore
pg_restore: creating TABLE "public.trees"
pg_restore: processing data for table "public.trees"
```

✓ Successfully completed.

Execution time: 0.87 seconds

Then check that the tree table has been created in the public schema of the treesdb_02 database

- select random rows
- \d trees