

Setup of PostgreSQL, pgAdmin and importing data

The current PostgreSQL version is 13. The tutorial uses screenshots for installing version 12, there should be only minor differences.

Wolfgang Gatterbauer

CS7240 Principles of scalable data management (sp21)

<https://northeastern-datalab.github.io/cs7240/sp21/>

1/19/2021

Overview

This document covers how to install **PostgreSQL** and **pgAdmin4**, and how to run our first queries from class.

- PostgreSQL is a popular open source database server. Unlike SQLite, PostgreSQL is a much more feature rich database management system. With PostgreSQL, you have 2 components, the server and the client. This isn't very different from your web server-browser model where the browser is your client: The web server services requests for fetching web pages whereas a database server services SQL queries on a database.
- This document will guide you through the process of setting up PostgreSQL on your machine. What this means is you will have a locally running instance of the PostgreSQL server on your machine.

Overview

- Just as a web browser helps make requests to a web server and displays the results of the request viz. a web page, similarly a database client helps you fire queries at a database server (PostgreSQL in our case), and displays the results that the database server sends over from processing those queries.
- The two most common clients that you will come across when using PostgreSQL are "psql" which is a command-line client and "pgAdmin" which is a graphical client.

Outline

1. Setup PostgreSQL for MAC
2. Setup PostgreSQL for Windows
3. Setup pgAdmin4 and run your first query
4. Setup PostgreSQL for Linux

PostgreSQL

https://postgresapp.com/downloads.html



Postgres.app

The easiest way to get started with PostgreSQL on the Mac

Introduction

Downloads

Documentation

GitHub

← 5389 Stars!

Latest Release

If you're new to Postgres, this is the file you should download. It includes everything you need to get started with PostgreSQL and PostGIS.

Postgres.app with PostgreSQL 12

Postgres.app v2.3.5 · Requires macOS 10.12 · Download Size 70MB

PostgreSQL 12.3 / PostGIS 3.0.1 / plv8 2.3.14

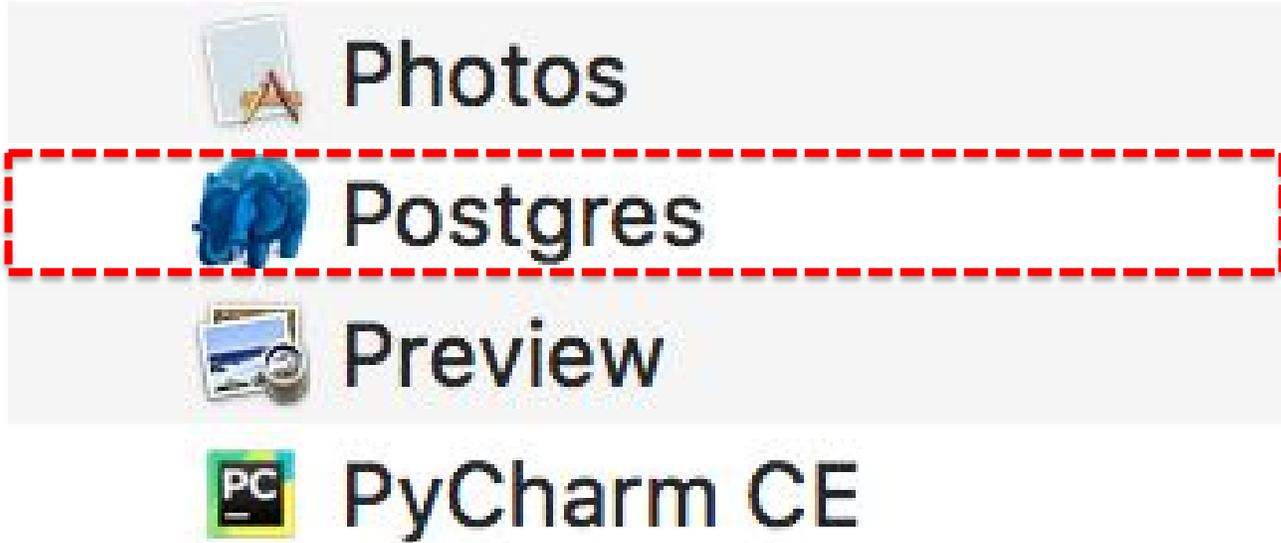
↓ Download

1) Click Here to Download

- Go to <https://postgresapp.com/downloads.html>
- Download the latest release (not a prerelease or Beta version!).



2) Extract the file you just downloaded. Typically the downloaded file should be in your downloaded folder. It will be a newer edition than in this screenshot.



3) The previous step will extract the "Postgres" application, typically in the same folder. Look for a file with a blue elephant icon.

4) Double click on this file and PostgreSQL server should be up and running.

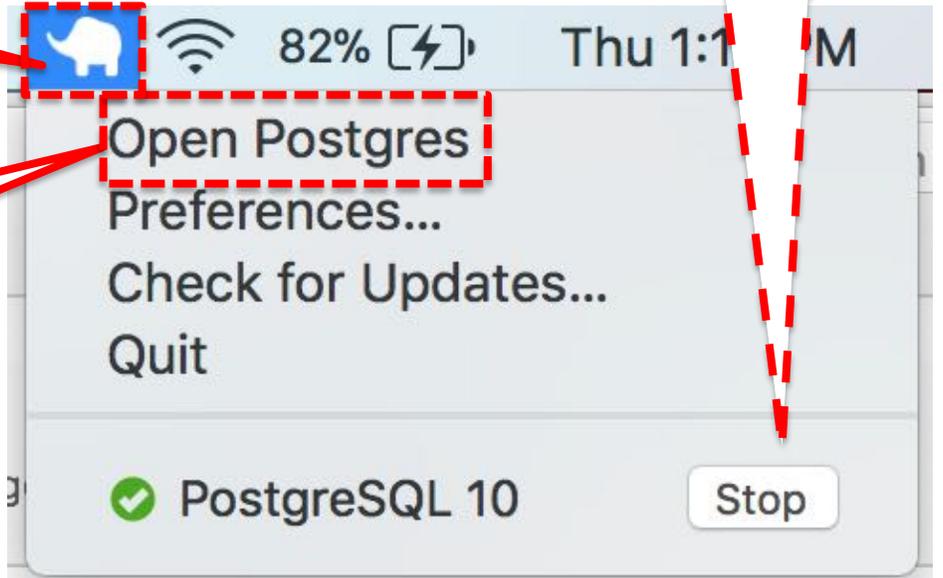


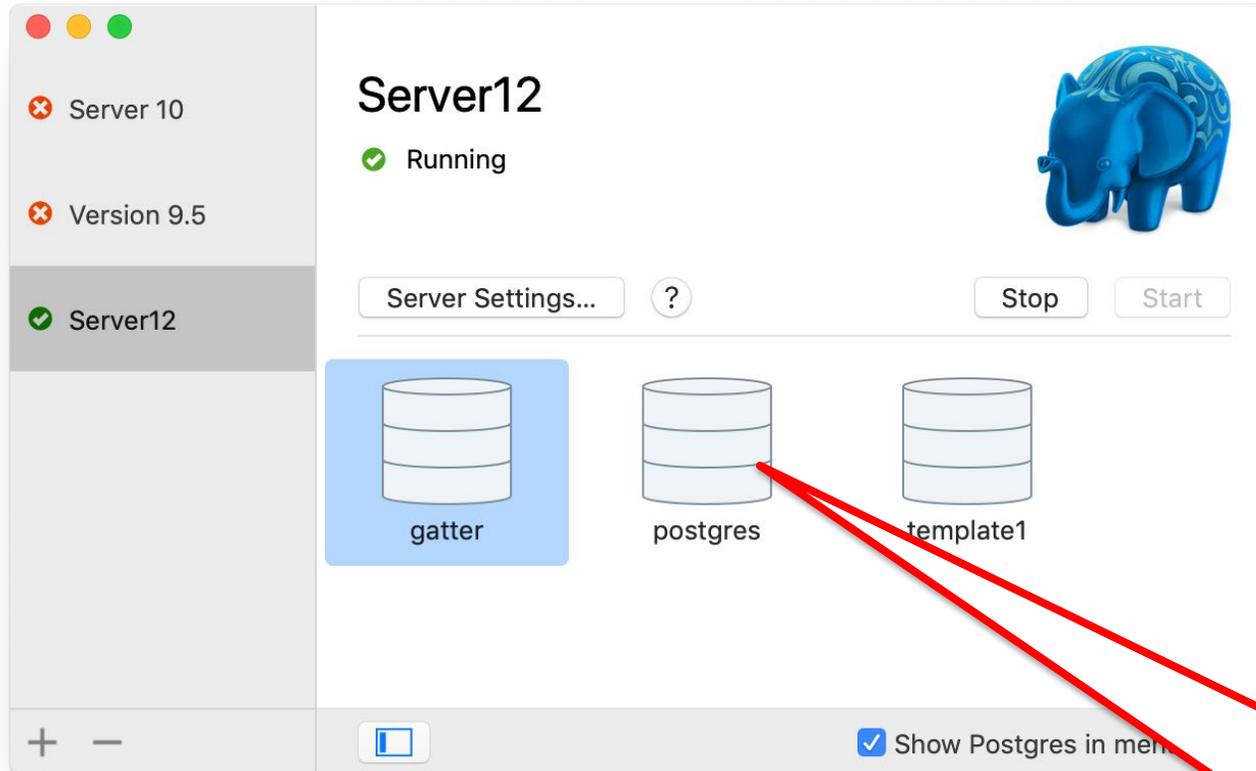
5) While the PostgreSQL server is up and running, you will see an icon show up in your menu bar at the top.

6) Click on the "Slonik" in the menu bar.

7) In the menu that shows up, select "Open Postgres".

If you restart your computer, the server may not be running. Thus verify that the server is running





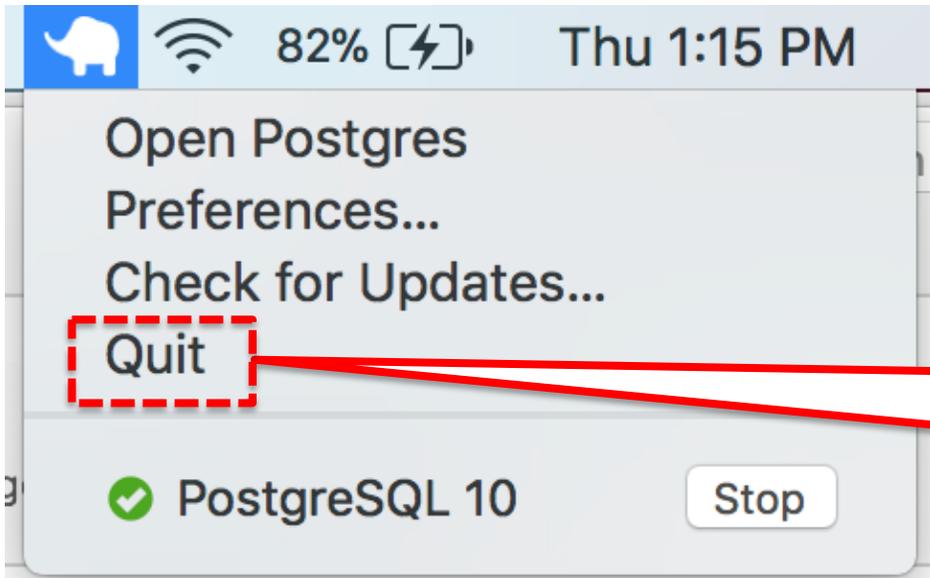
9) After you "Open Postgres", this window provides you with all your existing databases (in case you have already created some).

10) In case you have existing databases, then double-click on the database you want to work with.

```
gatter — psql -p5432 gatter — 80x24
Last login: Fri Sep 11 00:28:51 on ttys001
Wolfgangs-MacBook-Pro-new:~ gatter$ /Applications/Postgres.app/Contents/Versions
/12/bin/psql -p5432 "gatter"
psql (12.3)
Type "help" for help.
gatter=#
```

11) After clicking on the database you want to work with, the Postgres command prompt should open, that should look like <your_user_name>=#

12) This is the psql client that was mentioned earlier and where you will enter SQL commands to interact with the database server.



13) To stop the server, simply quit the application by clicking on the elephant icon in your menu bar and selecting Quit from the menu bar that shows up underneath the icon.

Command line Cheatsheet

Alternatively, start from terminal with command "psql".
You may have to add the folder containing the executable to your path. This depends on your setup. E.g.:

```
export PATH="/Applications/Postgres.app/Contents/Versions/latest/bin:$PATH"
```

See e.g. <https://stackoverflow.com/questions/36155219/psql-command-not-found-mac/36156782>

```
gatter — psql — 80x24
Last login: Fri Feb  9 22:47:42 on ttys001
[Wolfgangs-MacBook-Pro-9:~ gatter] psql
psql (9.5.5, server 10.1)
WARNING: psql major version 9.5, server major version 10.
         Some psql features might not work.
Type "help" for help.

[gatter=# \c imdb
psql (9.5.5, server 10.1)
WARNING: psql major version 9.5, server major version 10.
         Some psql features might not work.
You are now connected to database "imdb" as user "gatter".
imdb=#
```

- `\l` list existing databases
- `\c` connect to a database
- `\d` list tables in database
- `\q` disconnect from psql
- `\i` run external file

- `\d <tablename>` view details of a table
- `create database <dbname>` create DB

Example commands

```
gatter — -bash — 112x40
Last login: Sat Sep 26 16:12:49 on ttys001
Wolfgangs-MacBook-Pro-new:~ gatter$ psql
psql (12.3)
Type "help" for help.

[gatter=# create database 312;
ERROR:  syntax error at or near "312"
LINE 1: create database 312;
                        ^
[gatter=# create database "312";
CREATE DATABASE
[gatter=# \c 312;
You are now connected to database "312" as user "gatter".
[312=# set client_encoding to 'latin1';
SET
[312=# \q
Wolfgangs-MacBook-Pro-new:~ gatter$ psql -d 312 -f "/Users/gatter/Downloads/312-Personandaddress.txt"
psql:/Users/gatter/Downloads/312-Personandaddress.txt:12: NOTICE:  table "person" does not exist, skipping
DROP TABLE
psql:/Users/gatter/Downloads/312-Personandaddress.txt:13: NOTICE:  table "university" does not exist, skipping
DROP TABLE
CREATE TABLE
CREATE TABLE
INSERT 0 1
Wolfgangs-MacBook-Pro-new:~ gatter$
```

I like to use numbers for database names. But those are not common, and database names usually start with a letter. Thus I needed to use quotation marks

Example commands

```
gatter — -bash — 104x39
Last login: Mon Sep 21 17:36:05 on ttys000
Wolfgangs-MacBook-Pro-new:~ gatter$ psql
psql (12.3)
Type "help" for help.

gatter=# \l
List of databases

```

Name	Owner	Encoding	Collate	Ctype	Access privileges
302	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	
302-2	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	
gatter	gatter	UTF8	en_US.UTF-8	en_US.UTF-8	
postgres	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	
template0	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	=c/postgres + postgres=CTc/postgres
template1	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	=c/postgres + postgres=CTc/postgres

```
(6 rows)

gatter=# \c 302
You are now connected to database "302" as user "gatter".
302=# \d product
Table "public.product"

```

Column	Type	Collation	Nullable	Default
pname	character(20)		not null	
price	numeric(9,2)			
category	character(20)			
manufacturer	character(20)			

```
Indexes:
  "product_pkey" PRIMARY KEY, btree (pname)
Foreign-key constraints:
  "product_manufacturer_fkey" FOREIGN KEY (manufacturer) REFERENCES company(cname)

302=# \q
Wolfgangs-MacBook-Pro-new:~ gatter$
```

Example commands

Notice that you need the semicolon to finish your command and psql to start interpreting it.

Without the semicolon, the cursor changes from "...=#" to "...-#" indicating that it is still waiting for you to finish your code. That is useful for multiline SQL commands like the query on the right.

```
gatter — psql — 88x24
Last login: Mon Sep 21 17:36:14 on ttys000
[Wolfgangs-MacBook-Pro-new:~ gatter$ psql
psql (12.3)
Type "help" for help.

[gatter=# \c 302
You are now connected to database "302" as user "gatter".
[302=# select *
[302=# from product
[302=# ;
      pname      | price | category | manufacturer
-----+-----+-----+-----
Gizmo             | 19.99 | Gadgets   | GizmoWorks
PowerGizmo        | 29.99 | Gadgets   | GizmoWorks
SingleTouch       | 149.99 | Photography | Canon
MultiTouch        | 203.99 | Household | Hitachi
(4 rows)

302=# █
```

Problems

- One reason for starting problems with pgadmin can be that the user postgres does not have a password. In pgadmin, with the "New Server Registration" dialog form, it does not accept an empty password. In that case, perform the following steps :
 - Launch postgres command line interface using **sudo -u postgres psql**
 - Run the following command:
ALTER USER postgres WITH PASSWORD '<NewPostgresPassword>';
(FM: Remember to add the semi-colon)
 - Now setup a new server connection in pgadmin by clicking on **File -> New Server...**
 - Make sure the user is set to postgres and the password is set to **<NewPostgresPassword>**

Outline

1. Setup PostgreSQL for MAC
- 2. Setup PostgreSQL for Windows**
3. Setup pgAdmin4 and run your first query
4. Setup PostgreSQL for Linux

- Go to <http://www.postgresql.org/download/windows/>.

Windows installers

Interactive installer by EnterpriseDB

[Download the installer](#) provided by EnterpriseDB for all supported PostgreSQL versions.

This installer includes the PostgreSQL server, pgAdmin; a graphical tool for managing and developing you databases, and StackBuilder; a package manager that can be used to download and install additional PostgreSQL tools and drivers. Stackbuilder includes management, integration, migration, replication, geospatial, connectors and other tools.

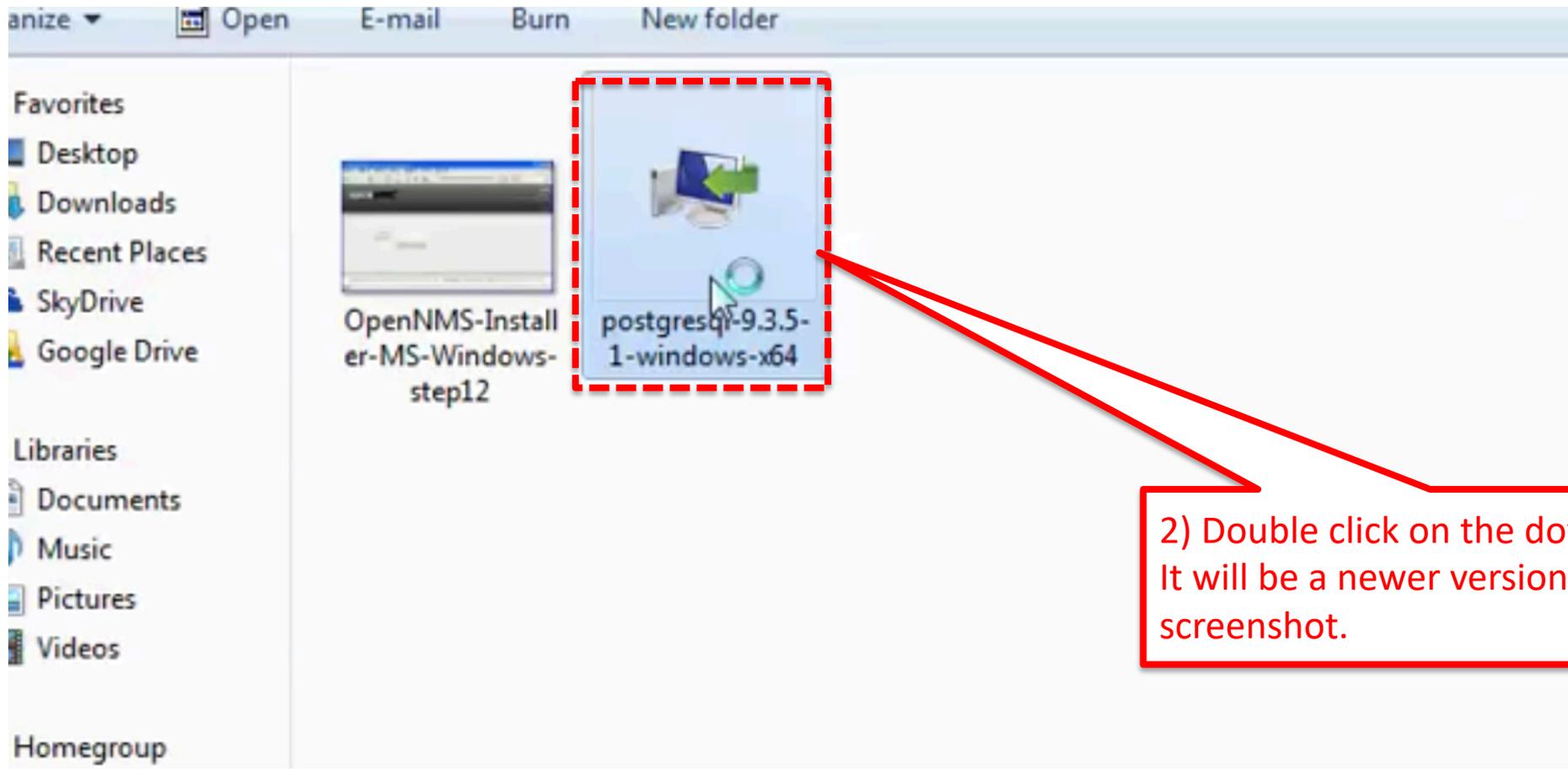
This installer can run in graphical or silent install modes.

The installer is designed to be a straightforward, fast way to get up and running with PostgreSQL on Windows.

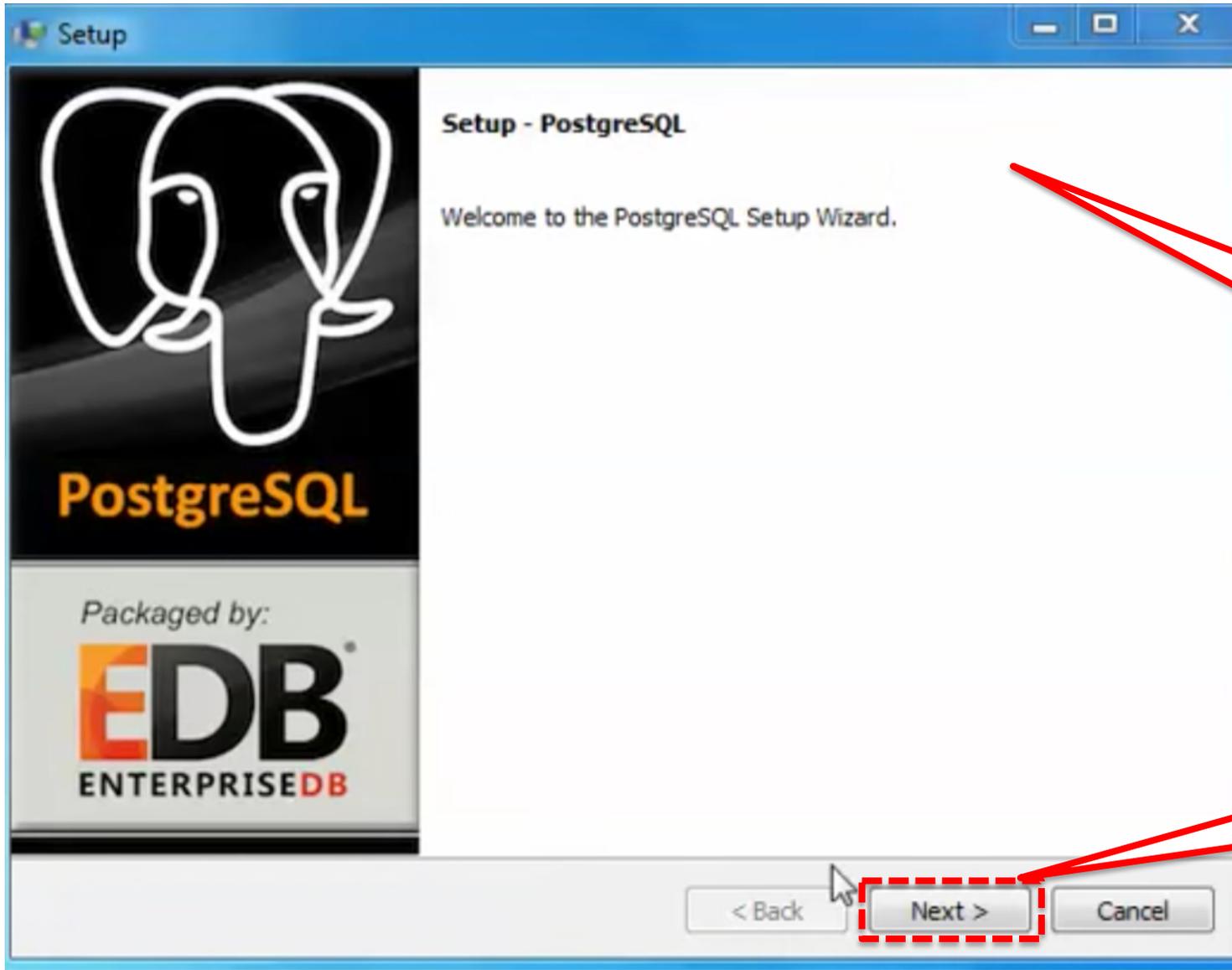
Advanced users can also download a [zip archive](#) of the binaries, without the installer. This download is intended for users who wish to include PostgreSQL as part of another application installer.

Platform support

- Double click on the downloaded file. A window will show up that will guide you through the installation.

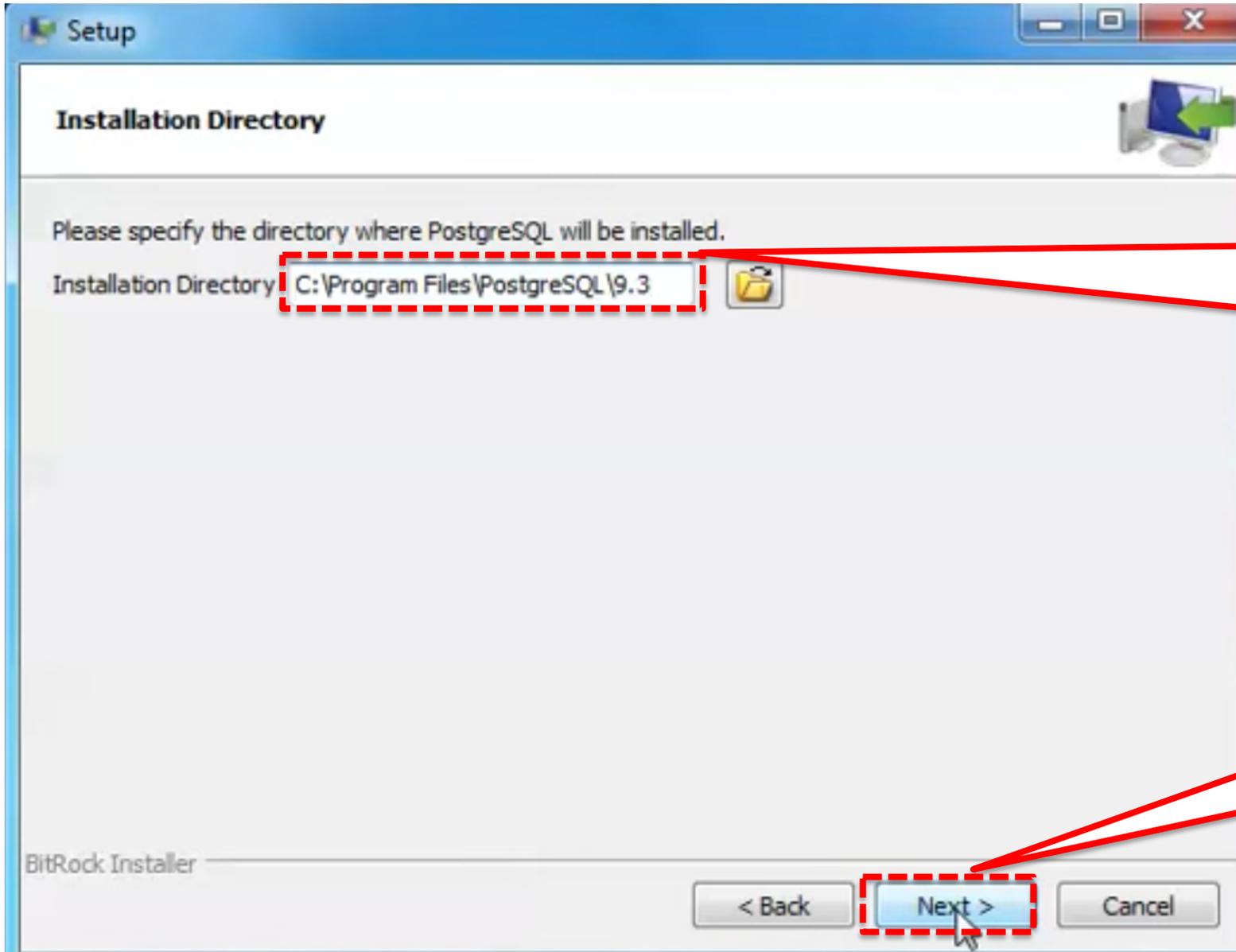


2) Double click on the downloaded file. It will be a newer version than in the screenshot.



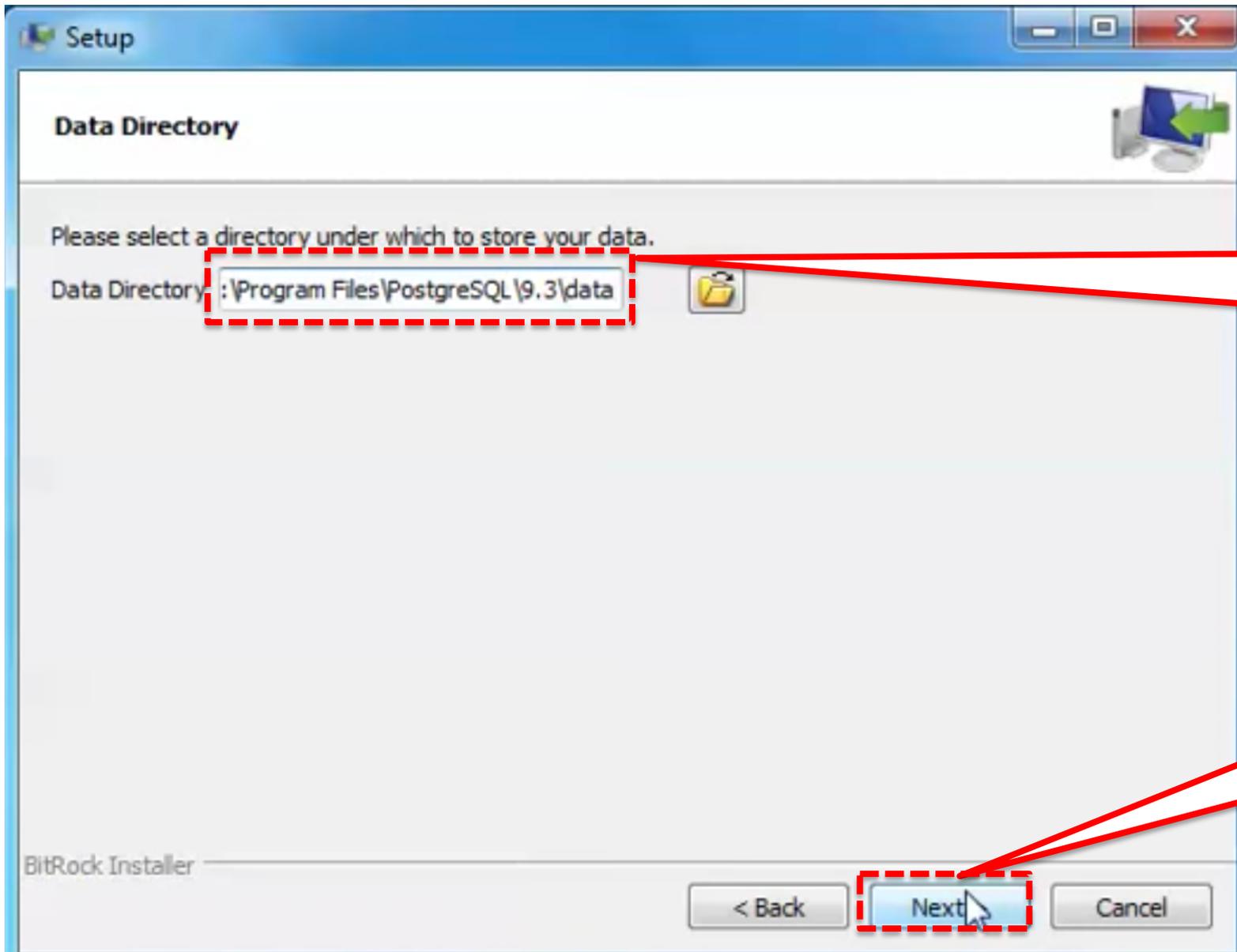
3) A window will show up that will guide you through the setup of PostgreSQL.

4) Click "Next" to continue the setup.



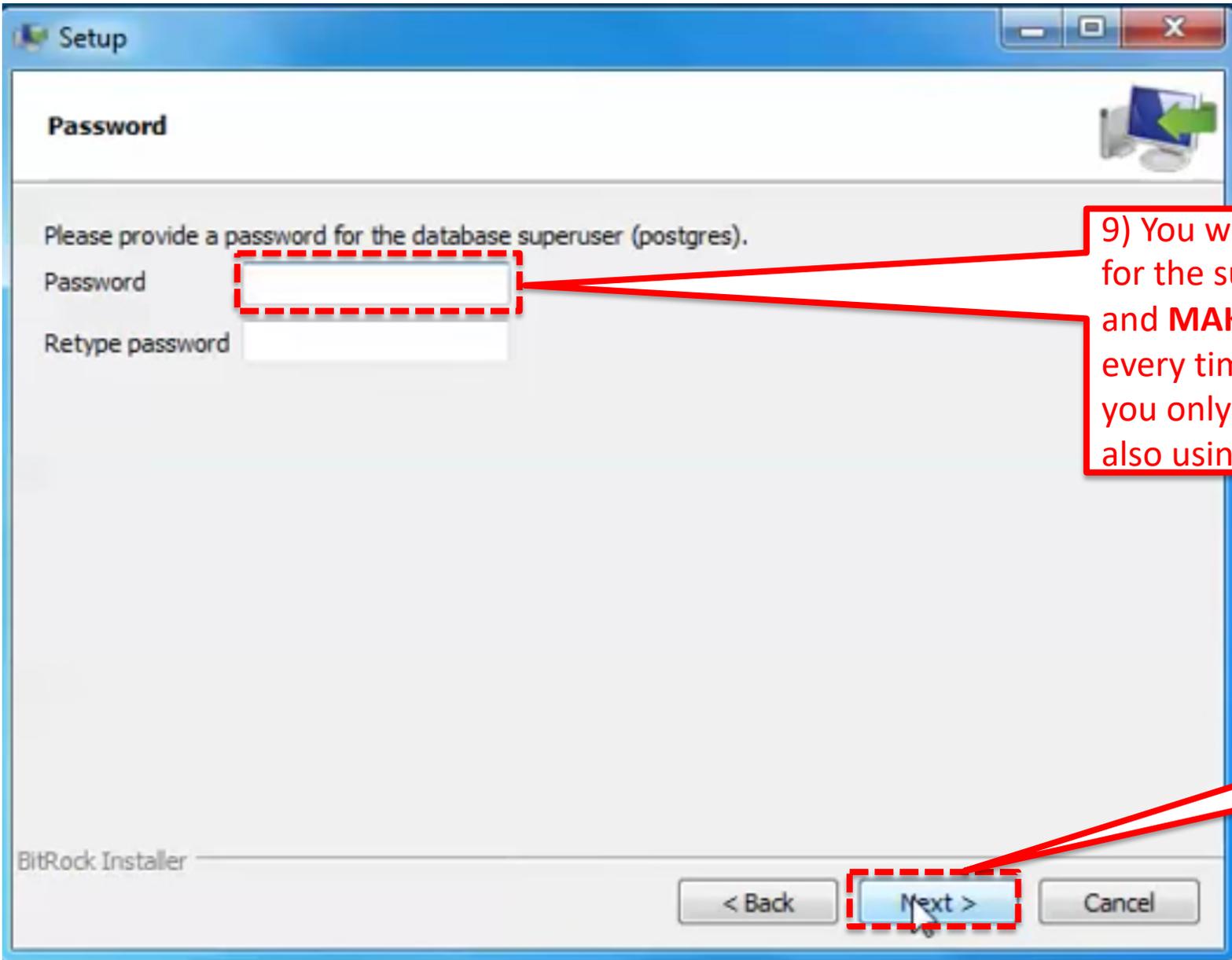
5) The installer wizard will ask you to specify a directory where PostgreSQL should be installed. It is right if you stick with the default option. So just click next.

6) Click "Next" to continue the setup.



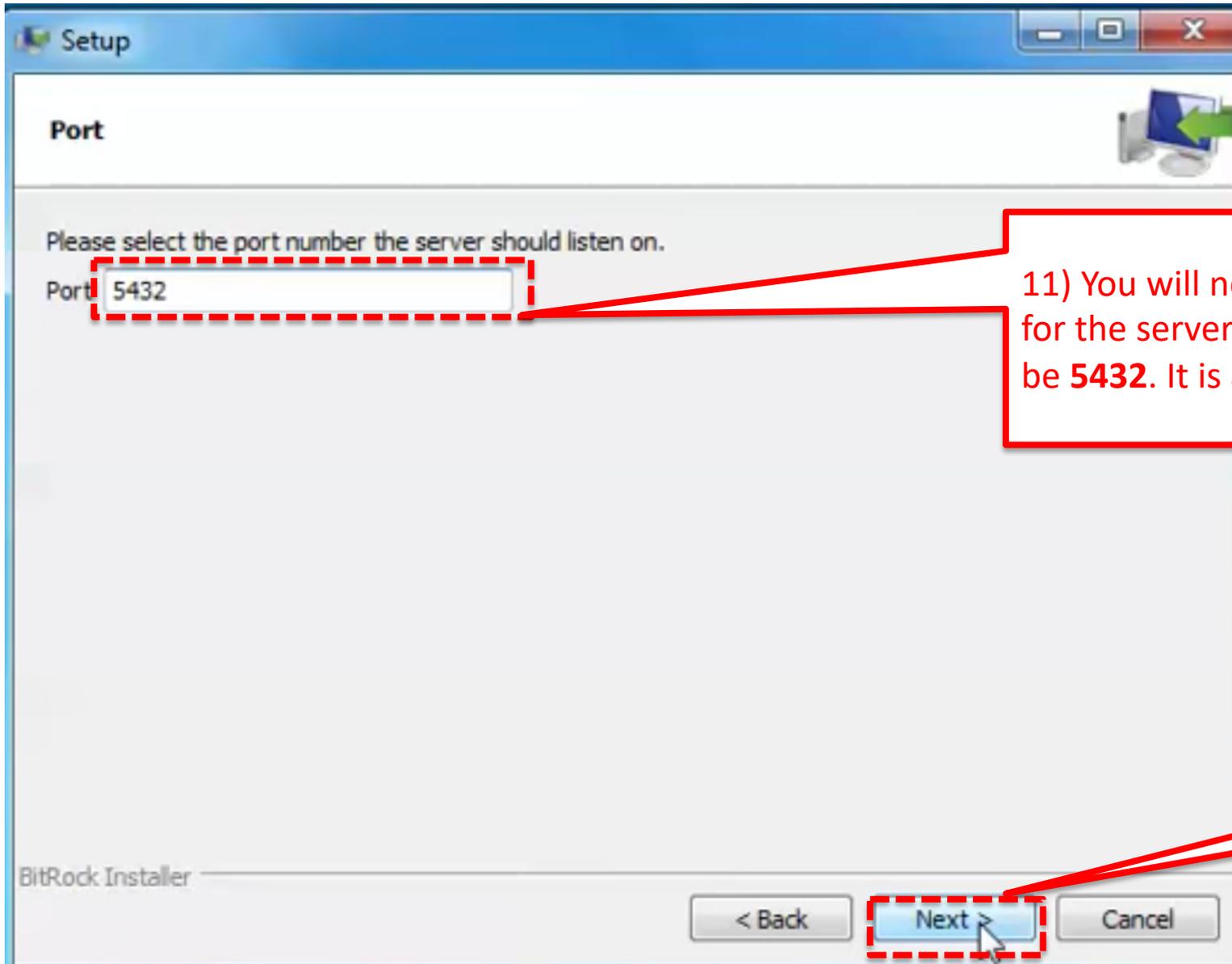
7) The installer wizard will then ask you to specify a data directory. Again, it is alright to stick with the default option.

8) Click "Next" to continue the setup.



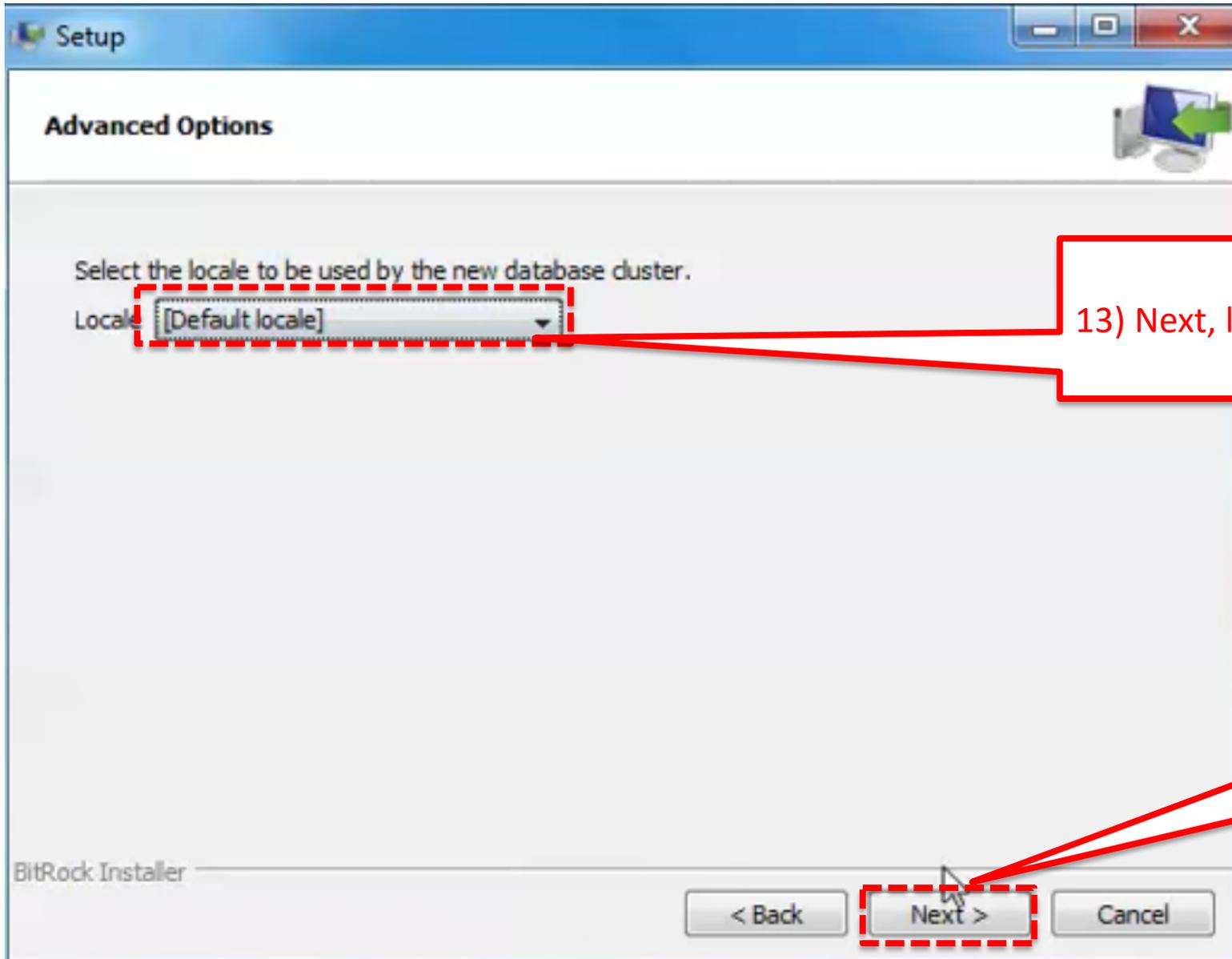
9) You will next be prompted to enter a password for the superuser "postgres". Enter the password and **MAKE A NOTE OF IT** as it will be required every time you want to work with postgres. Since you only run it locally, you can simplify your life by also using "postgres" as your pwd...

10) Click "Next" to continue the setup.



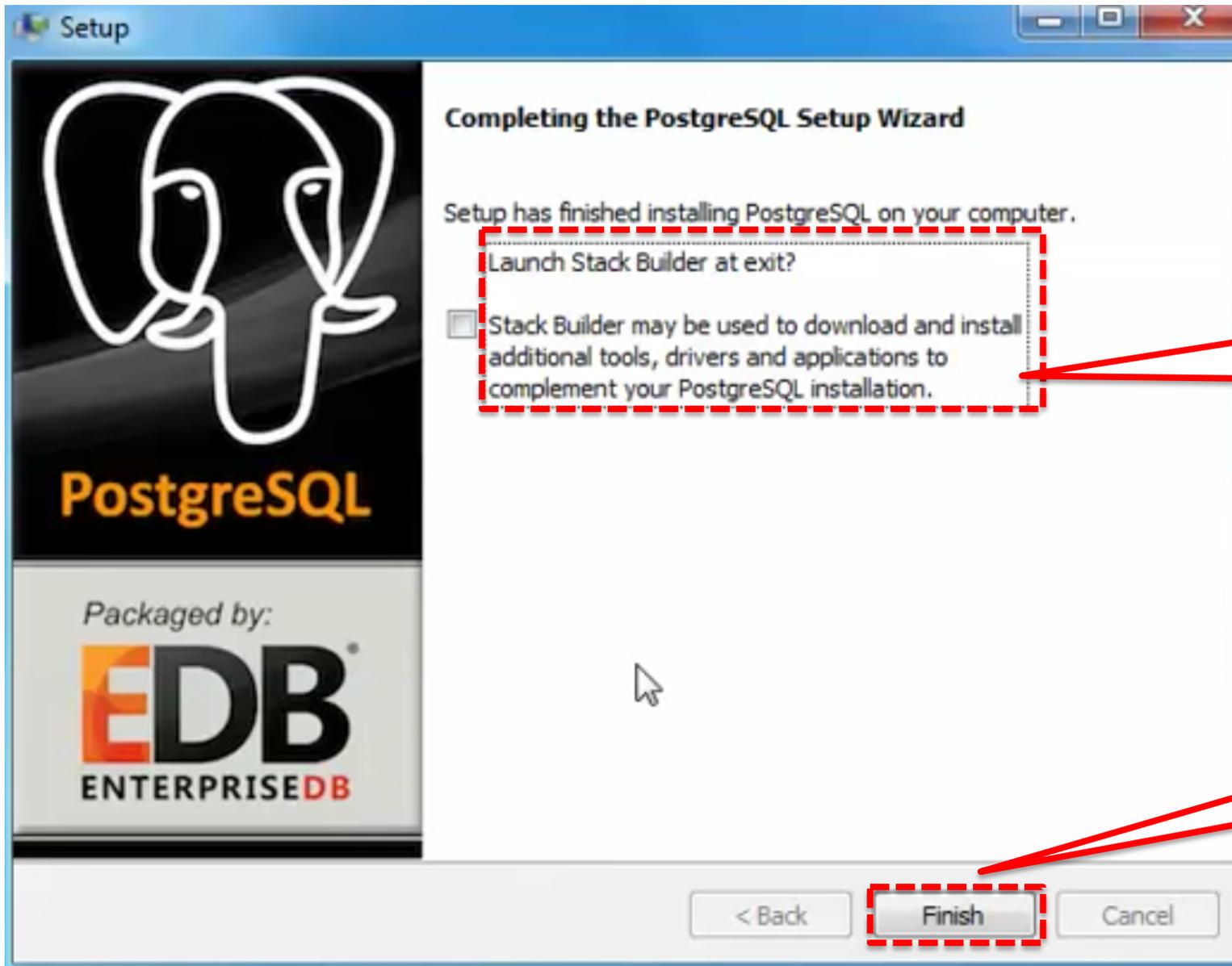
11) You will next be asked to enter a port number for the server to listen on. The default option should be **5432**. It is alright to keep it as it is.

12) Click "Next" to continue the setup.



13) Next, leave the locale selection at "Default locale"

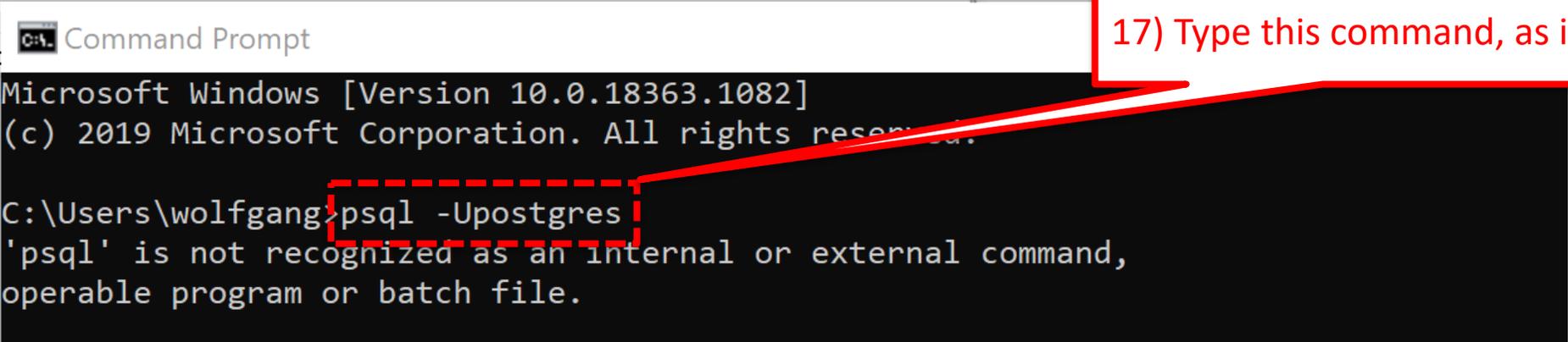
14) Click "Next" to continue the setup. It runs and may take a few minutes.



15) Next, it will ask you if you want to install Stack Builder. You don't need Stack Builder in this class. So, you uncheck the box.

14) Click "Finish" to complete the setup.

- psql is the command line access to postgres. To open it to the command prompt and type: **psql -Upostgres**

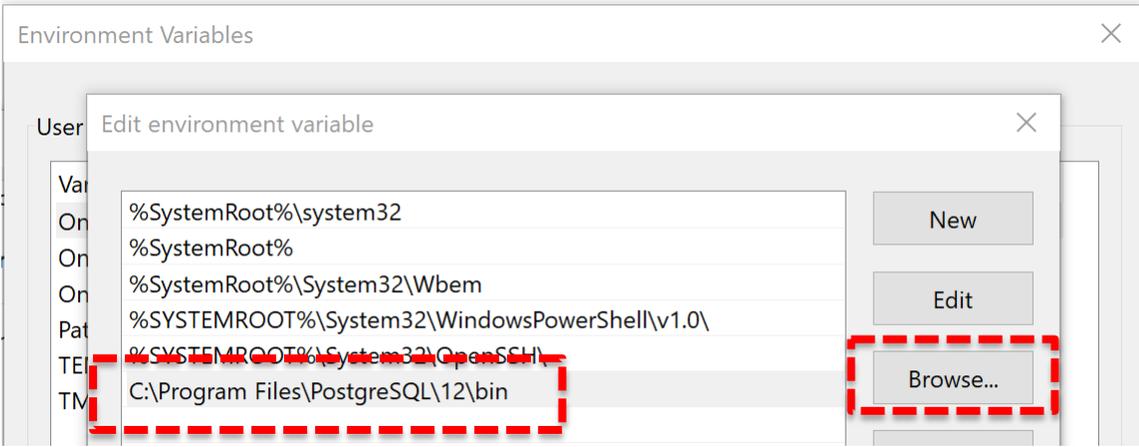


```
Command Prompt
Microsoft Windows [Version 10.0.18363.1082]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users>wolfgang>psql -Upostgres
'psql' is not recognized as an internal or external command,
operable program or batch file.
```

17) Type this command, as it is.

In case psql is not recognized, you need to set update PATH variables. See following link to see how to set the PATH: <https://sqlbackupandftp.com/blog/setting-windows-path-for-postgres-tools>



- psql is the command line access to postgres. To open it to the command prompt and type: **psql -Upostgres**

```
Command Prompt - psql -Upostgres
Microsoft Windows [Version 10.0.18363.1082]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\wolfgang>psql -Upostgres
Password for user postgres:
psql (12.4)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

postgres=#
```

17) Type this command, as it is.

18) It will then ask for the password you set during the PostgreSQL setup. Note: that the password you enter will not be visible to you, so just keep typing it CORRECTLY!

19) On executing the previous command, the prompt should change and should now look like "postgres=#"

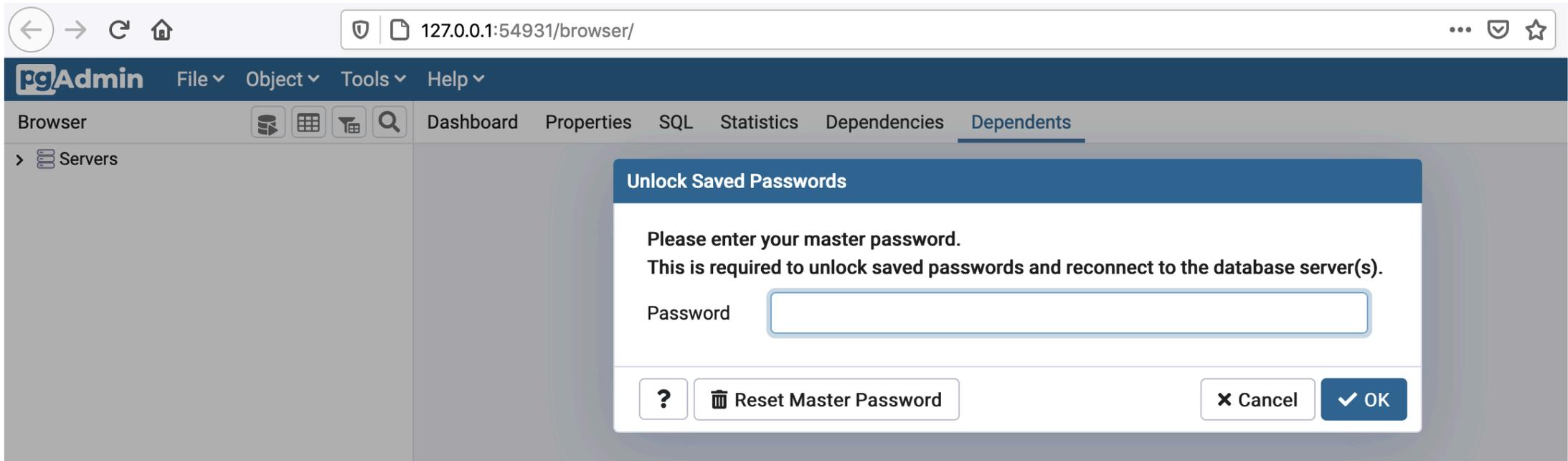
- You are now in the psql program where you can enter queries to interact with the PostgreSQL server.
- Entering “\q” at this prompt should exit the psql program.

Example: postgres=# \q

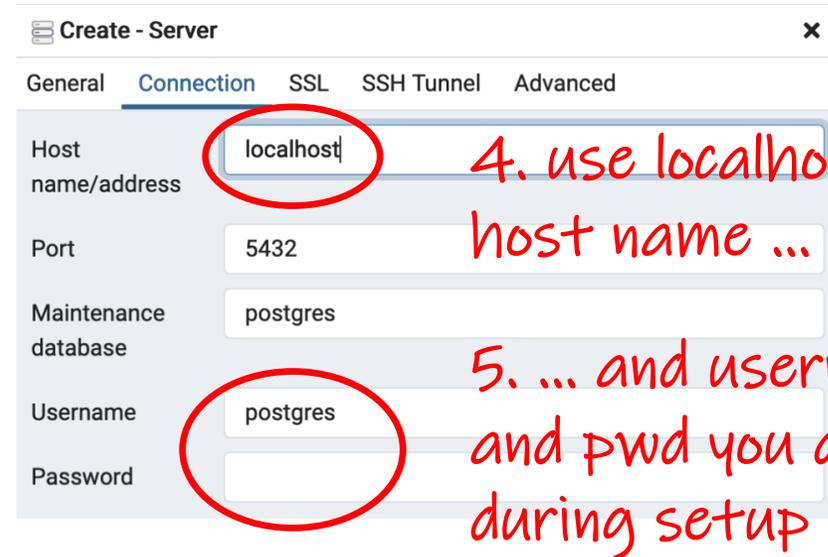
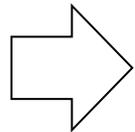
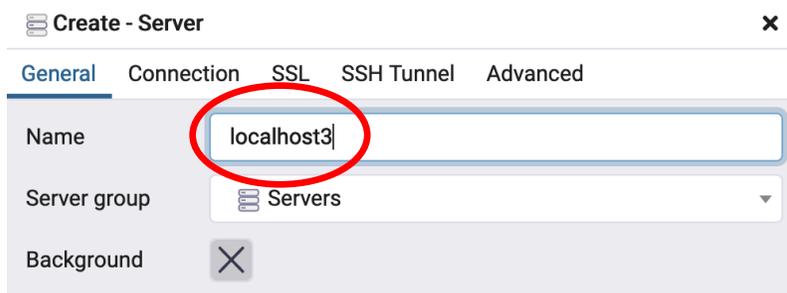
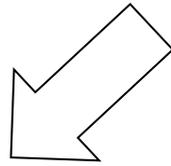
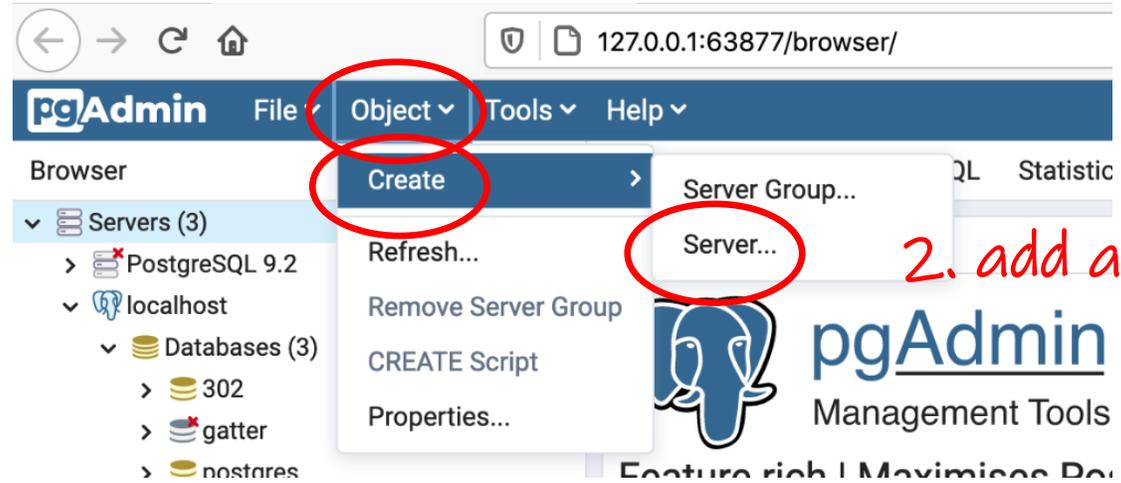
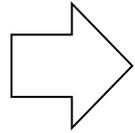
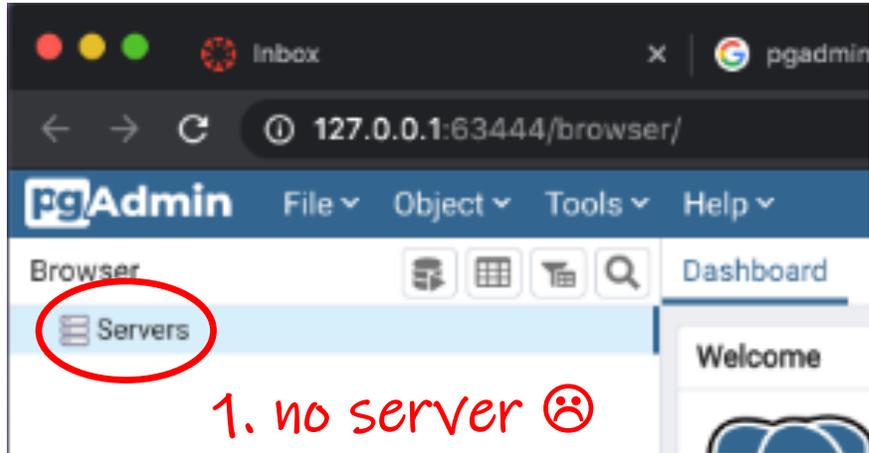
Outline

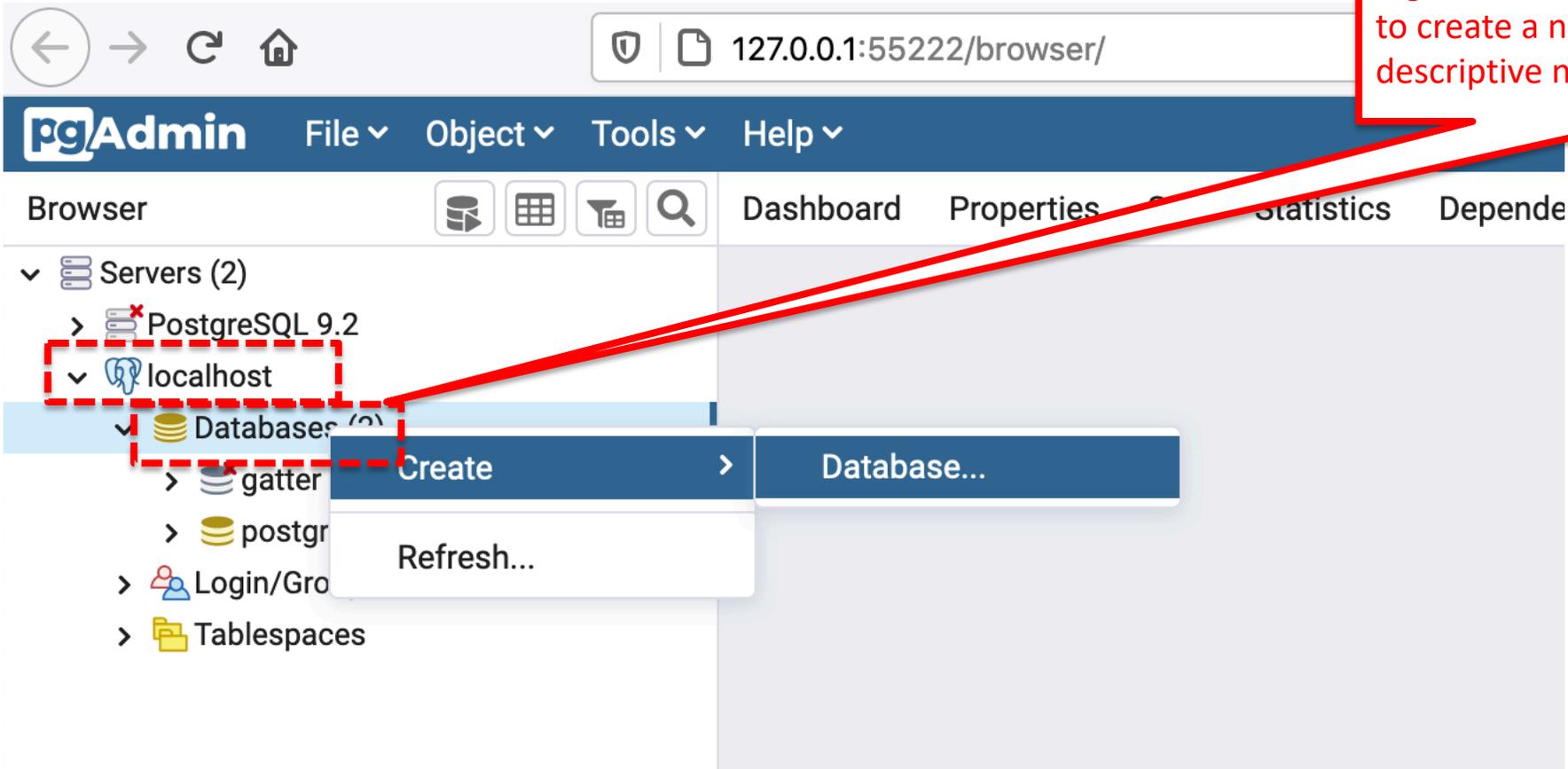
1. Setup PostgreSQL for MAC
2. Setup PostgreSQL for Windows
3. Setup pgAdmin4 and run your first query
4. Setup PostgreSQL for Linux

- pgAdmin is a visual client for your postgres database. Download and install version 4 from: <https://www.pgadmin.org/download/>
- After installation, run "pgAdmin 4"  pgAdmin 4.app
- You may have to reset the master password (e.g. using "postgres")



- In case you don't see a server, you may have to manually add one





Right click to "Databases" under localhost to create a new empty database. Choose a descriptive name to remember later.

The screenshot shows the pgAdmin web interface. The browser address bar displays `127.0.0.1:55222/browser/#`. The navigation menu includes **pgAdmin**, **File**, **Object**, **Tools**, and **Help**. The left sidebar shows a tree view of the database structure: **Servers (2)** (PostgreSQL 9.2), **localhost** (Databases (3): **302**, gatter, postgres), **Login/Group Roles**, and **Tablespaces**. The **302** database is selected and highlighted in blue. A red dashed box highlights the **302** database in the sidebar and the **Query Tool** icon in the top toolbar. The main panel shows the SQL editor with the following code:

```
1 -- Database 302
2
3 -- DROP DATABASE "302";
4
5 CREATE DATABASE "302"
6   WITH
7   OWNER = postgres
8   ENCODING = 'UTF8'
9   LC_COLLATE = 'en_US.UTF-8'
10  LC_CTYPE = 'en_US.UTF-8'
11  TABLESPACE = pg_default
12  CONNECTION LIMIT = -1;
```

I named the new database "302" as I intend here to create our example database "302" from class.

Navigate to the SQL file folder on Canvas and open the text file "302". Notice that just opening in the Canvas native read-only view does not work, really open the file with your local text editor, then copy and paste the commands into the pgadmin interface

CS3200 17751 Database	
▶ folder Lecture slides	
▶ folder media	
▼ folder SQL files	
▶ folder Tutorials	

Name	
300-SmallIMDB.txt	
302-Simpleproducts.txt	
304-Worker.txt	
305-Conceptualevaluationstrat...	

127.0.0.1:55222/browser/#

pgAdmin File Object Tools Help

Browser Dashboard Properties SQL Statistics Depen

Query Tool

- Servers (2)
 - PostgreSQL 9.2
 - localhost
 - Databases (3)
 - 302
 - gatter
 - postgres
 - Login/Group Roles
 - Tablespaces

```
1 -- Database: 302
2
3 -- DROP DATABASE "302";
4
5 CREATE DATABASE "302"
6 WITH
7 OWNER = postgres
8 ENCODING = 'UTF8'
9 LC_COLLATE = 'en_US.UTF-8'
10 LC_CTYPE = 'en_US.UTF-8'
11 TABLESPACE = pg_default
12 CONNECTION LIMIT = -1;
```

Paste into the query tool

- Servers (2)
 - PostgreSQL 9.2
 - localhost
 - Databases (3)
 - 302
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Schemas (1)

302/postgres@localhost

Query Editor Query History

```
1 select * from company
```

Data Output Explain Messages Notifications

	cname [PK] character (20)	stockprice integer	country character (20)
1	GizmoWorks	25	USA
2	Canon	65	Japan
3	Hitachi	15	Japan

Your first query ...

... and the result!

Outline

1. Setup PostgreSQL for MAC
2. Setup PostgreSQL for Windows
3. Setup pgAdmin4 and run your first query
- 4. Setup PostgreSQL for Linux**

Step 1: Install PostgreSQL

Here are the installation steps on Ubuntu (this installation will also work on any Debian-based distribution):

1. Open a terminal window.
2. Issue the command *sudo apt-get install postgresql*.
3. Type the sudo password necessary to give you admin rights and hit Enter.
4. Allow apt to pick up any necessary dependencies.

Step 2: Change the default user password

If you don't follow this step, you will not be able to add databases and administer PostgreSQL, and the database will not be secure.

- Here's how to change the password for the default user. The user in question is postgres, and the password is changed like so:
 1. Open a terminal window.
 2. Issue the command *sudo passwd postgres*.
 3. Type (and confirm) the password to be used for this user.
- The postgres user will be the only user on your system that can open the PostgreSQL prompt without defining a database, which means postgres is the only user who can administer PostgreSQL.
- To test this, change to the postgres user with the command *su - postgres* and then enter the command *psql*. You should now be at the Postgres prompt, which looks like:

postgres=#

- All other users gain access to the prompt like so:

psql DB_NAME

Where, DB_NAME is the name of an existing database.

Step 3: Change the Postgres admin password

The administrator password must be set; otherwise, external applications will not be able to communicate with the database.

To change the admin password for Postgres, follow these steps:

1. Open a terminal window.
2. Change to the postgres user.
3. Log in to the postgres prompt.
4. Issue the command `\password postgres`.
5. Enter (and verify) the new password.
6. Exit the prompt with the command `\q`.

Step 4: Create your first database

This is where it gets exciting. Let's create a new database called testdb. To do this, follow these steps:

1. Open a terminal window.
2. Change to the postgres user.
3. Log in to the postgres prompt.
4. Issue the command *CREATE DATABASE ACTOR;* .