



How to Connect to a PostgreSQL Database from Linux or Windows

February 22, 2024

LINUX POSTGRESQL

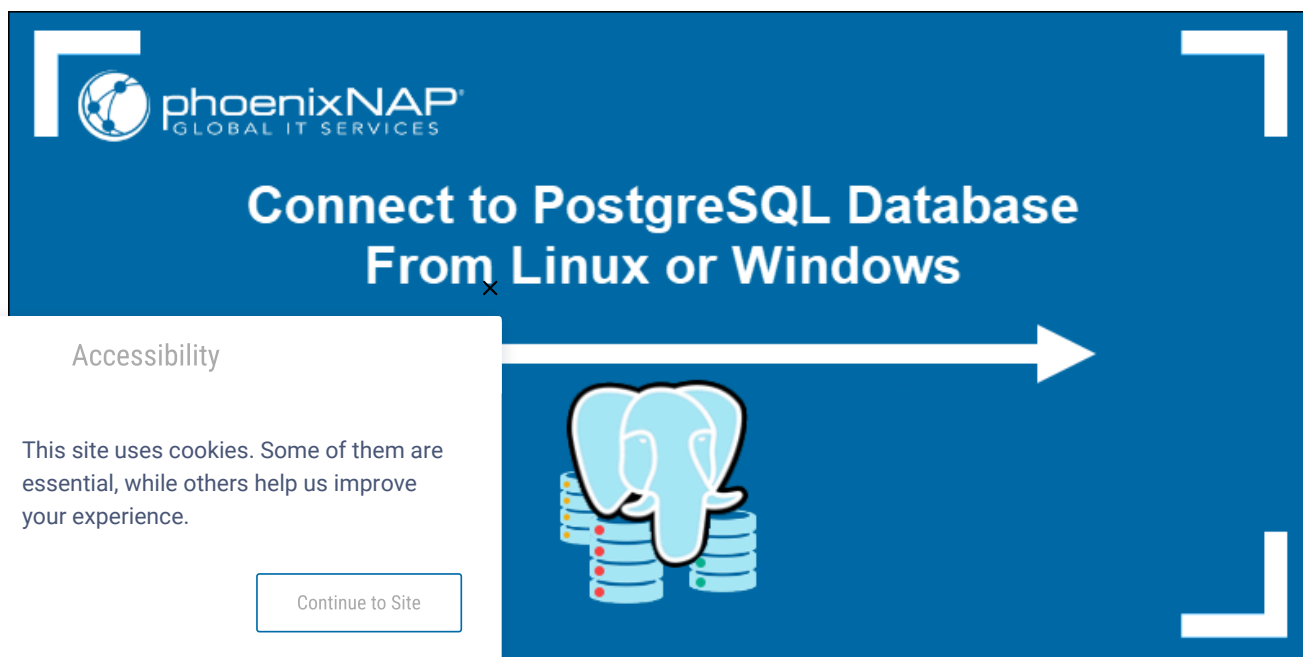
[Home](#) » [Databases](#) » How to Connect to a PostgreSQL Database from Linux or Windows

Introduction

PostgreSQL is an [open-source](#) relational [database management system](#). Users can access PostgreSQL databases via an interactive terminal program, **psql**, or a [graphical interface](#) called **pgAdmin**.

These tools enable administrators to edit, automate, and execute database queries within PostgreSQL. Both programs are compatible with [Linux](#) and Windows.

Follow the steps in this guide to learn how to connect to a PostgreSQL database and start managing your tables and data sets.



Accessibility

This site uses cookies. Some of them are essential, while others help us improve your experience.

[Continue to Site](#)

Preferences

Prerequisites

- PostgreSQL installed.
- **pgAdmin4** installed.
- Access to a [command line](#)/terminal window.
- [Sudo](#) or [root privileges](#) on Linux.

How to Access a PostgreSQL Database from Linux

PostgreSQL creates a default user account called **postgres** during the installation. Users can switch to this account to access PostgreSQL databases.

The examples in this guide are presented using [Ubuntu 22.04](#). The same commands work in other [Linux distributions](#).

Connect to PostgreSQL Database via SQL Shell (psql)

Enter the following command to open a [bash shell](#) and switch the current user context to the default **postgres** user:

```
sudo -i -u postgres
```

```
phoenixnap@phoenixnap:~$ sudo -i -u postgres
postgres@phoenixnap:~$
```

The same command applies if you have created a different user. Simply substitute the **postgres** user account name with the name of your existing user.



Note: Check out our in-depth article on different ways to [create a Postgres user](#).

Use the following command to access **psql**, a terminal-based front-end to PostgreSQL:

```
psql
```

```
postgres@phoenixnap:~$ psql
psql (14.10 (Ubuntu 14.10-0ubuntu0.22.04.1))
Type "help" for help.

postgres=#
```

To retrieve information about the current connection and user, type:

```
\conninfo
```

```
postgres=# \conninfo
You are connected to database "postgres" as user "postgres" via socket in
"/var/run/postgresql" at port "5432".
```

The output helps you determine which user and database you are currently interacting with.

PostgreSQL can support and maintain multiple databases and users simultaneously. Enter the following command to list available users and databases:

```
\l
```



```
postgres=# \l
```

Name	Owner	Encoding	Collate	Ctype	Access privileges
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 +

(3 rows)

Use the `\c` command and the database name to connect to a different database owned by the **postgres** user:

```
\c template1
```



```
postgres=# \c template1
You are now connected to database "template1" as user "postgres".
```

In this example, the name of the database is **template1**. Enter the name of the database you want to connect to.

To exit the **psql** prompt, enter:

```
\q
```



Use the **exit** command to leave the **postgres** Linux **command prompt** and return to your regular system user:

```
exit
```



```
postgres=# \q
postgres@phoenixnap:~$ exit
logout
phoenixnap@phoenixnap:~$
```

The **logout** message confirms the action.

Connect to PostgreSQL Database from Terminal

If all the components of your databases and users are correctly configured, you can bypass the intermediary bash shell and connect to PostgreSQL directly.

Use the following command to log into **psql** directly from the Linux terminal window:

```
sudo -u postgres psql
```



The **-u** (user) option causes **sudo** to run the specified command as a user other than root, specifically the **postgres** user.

```
phoenixnap@phoenixnap:~$ sudo -u postgres psql
psql (14.10 (Ubuntu 14.10-0ubuntu0.22.04.1))
Type "help" for help.

postgres=#
```

As with the previous method, you can now work on databases by executing queries interactively. Enter **\q** to exit the prompt.

Connect to PostgreSQL Database via pgAdmin

PgAdmin 4 is a graphical [front-end](#) tool for PostgreSQL. It provides a visual, user-friendly environment with many practical database management solutions.

1. Open a [web browser](#) and enter the pgAdmin 4 instance [URL](#). For example, if pgAdmin 4 is installed locally, type:

```
http://127.0.0.1/pgadmin4
```

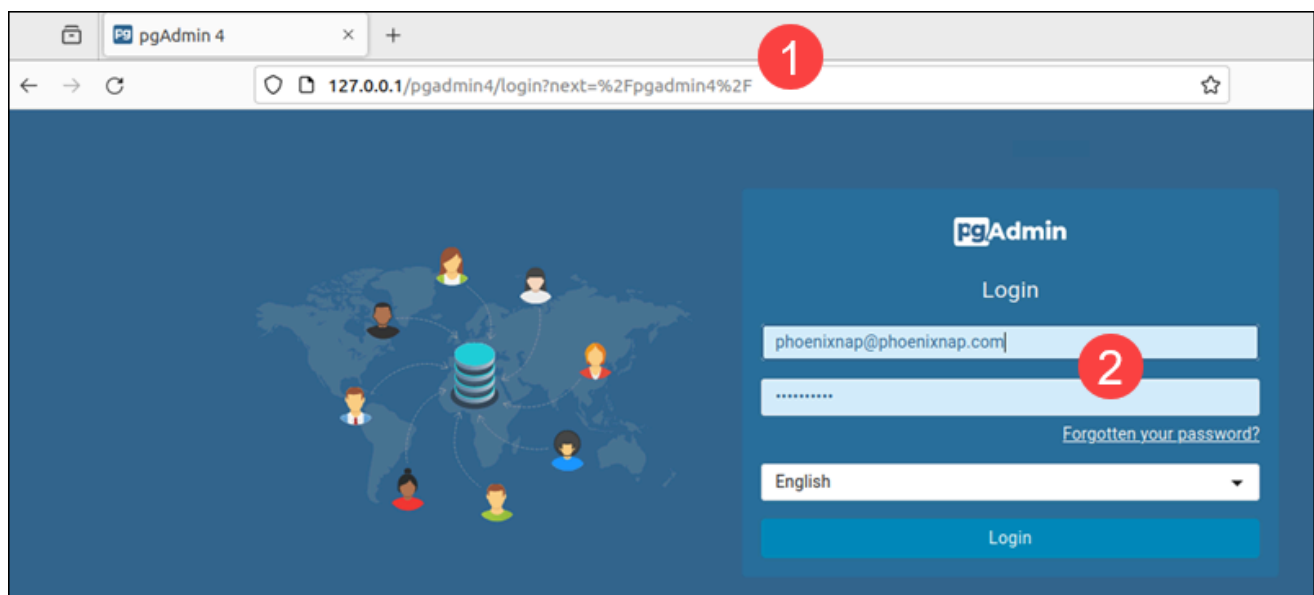


or

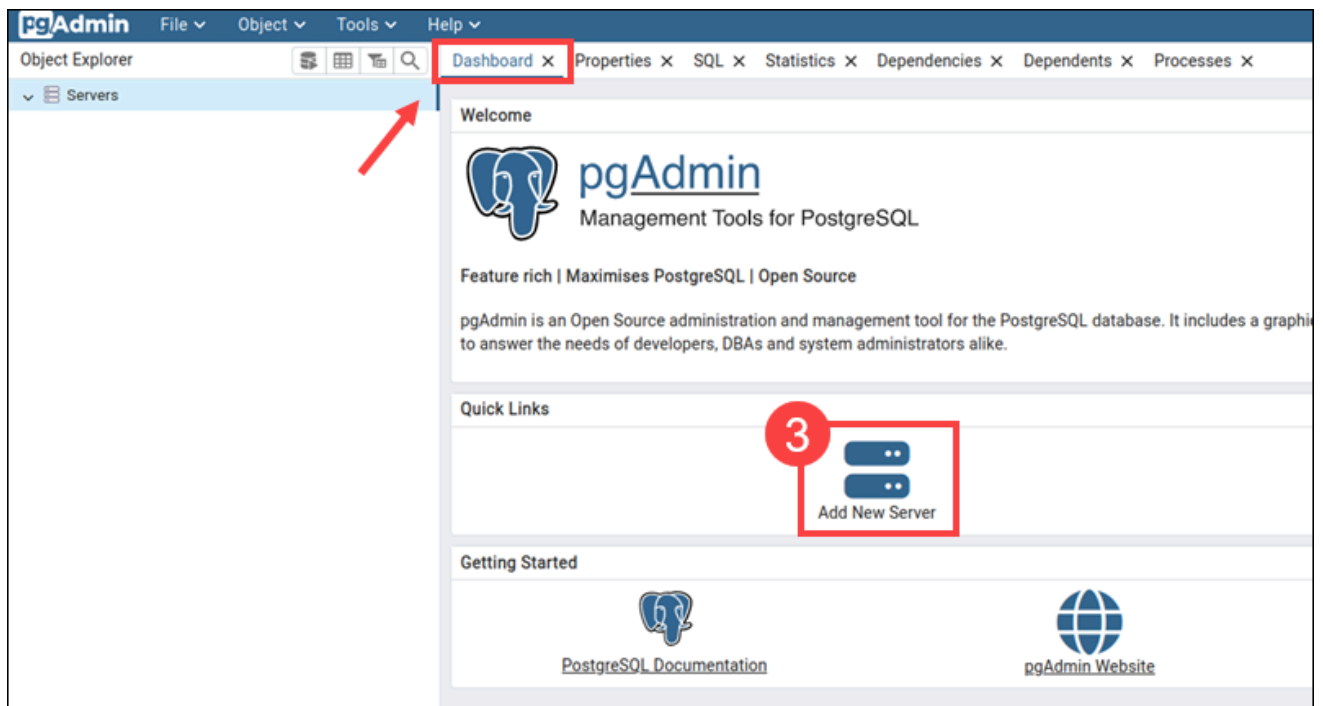
```
http://localhost/pgadmin4
```



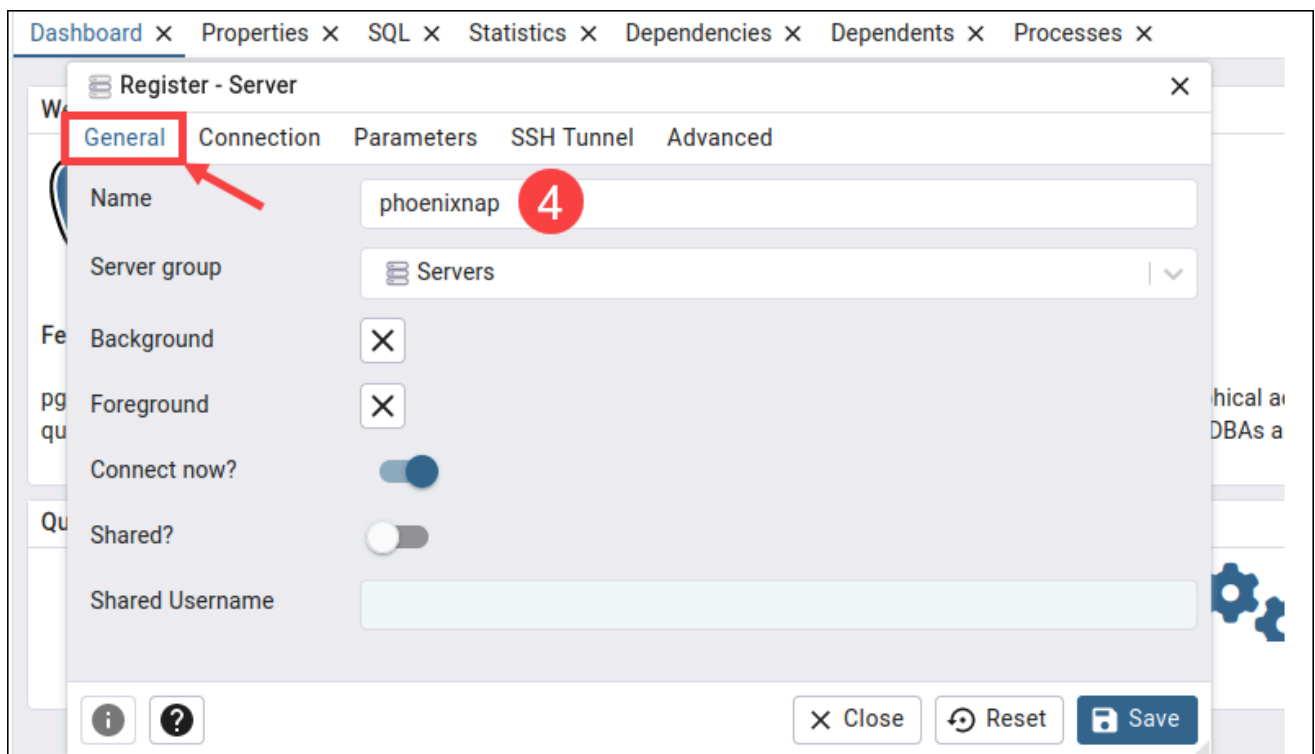
2. Enter your credentials to access the pgAdmin 4 dashboard.



3. Click **Add New Server** on the *Dashboard* tab.



4. Open the *General* tab and enter a server name in the **Name** field.

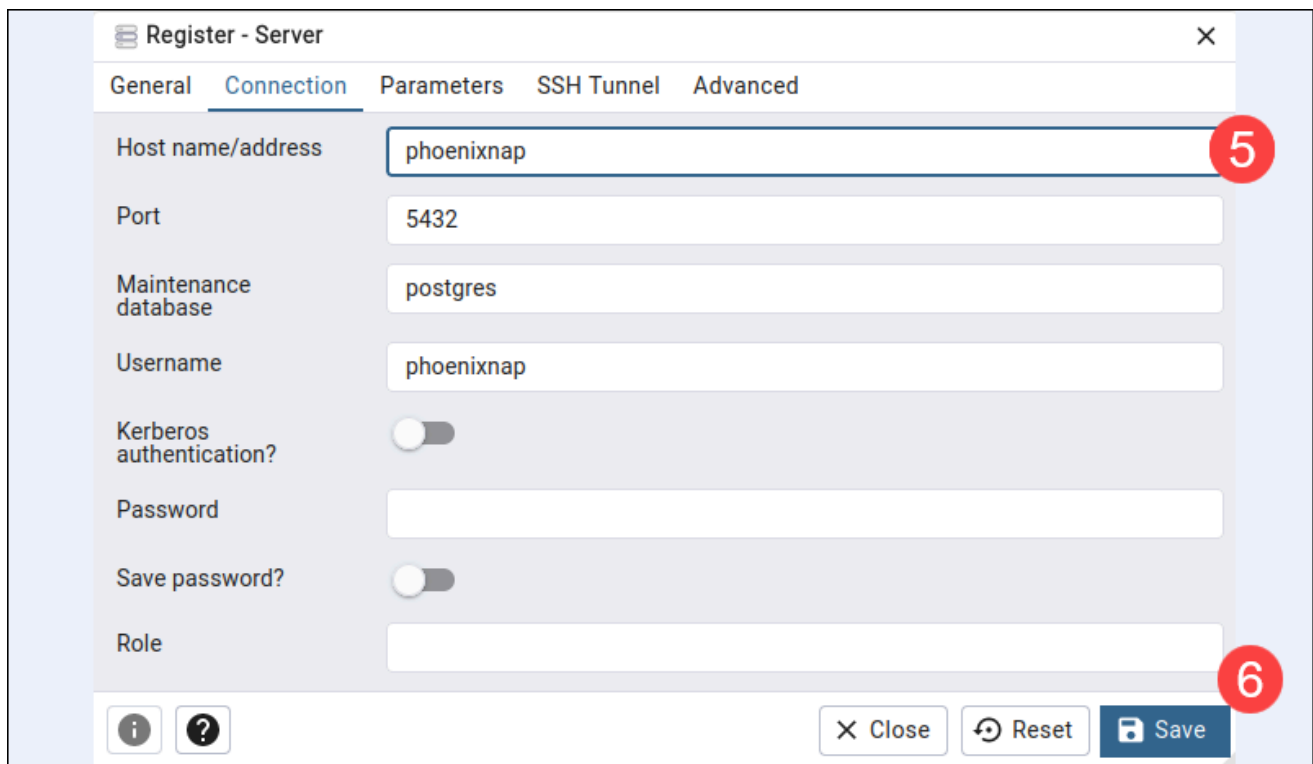


5. Access the *Connection* tab and enter the server's hostname and database user credentials.



Note: The *Host name/address* is the location of the machine where the PostgreSQL server is running.

6. Click **Save** to establish a database connection.

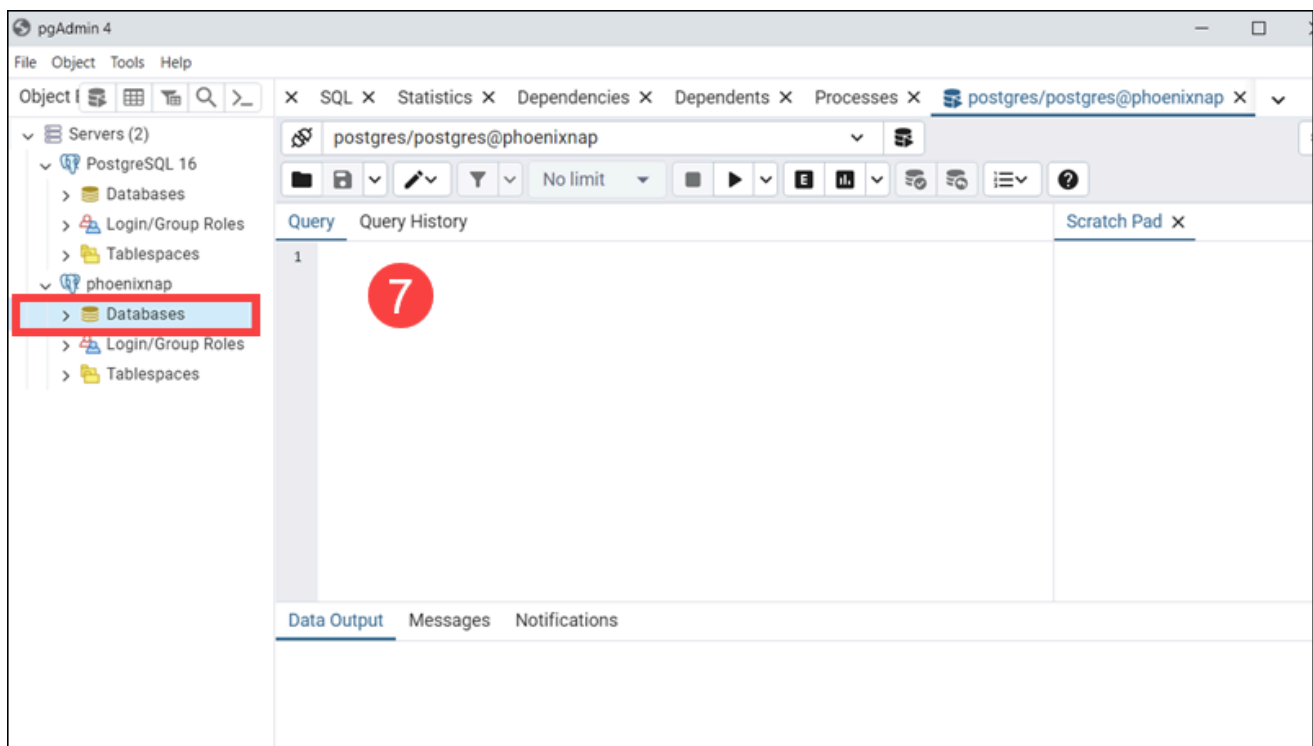


The image shows the 'Register - Server' dialog box in pgAdmin 4, with the 'Connection' tab selected. The fields are as follows:

Field	Value
Host name/address	phoenixnap
Port	5432
Maintenance database	postgres
Username	phoenixnap
Kerberos authentication?	<input type="checkbox"/>
Password	
Save password?	<input type="checkbox"/>
Role	

At the bottom right, there are buttons for 'Close', 'Reset', and 'Save'. A red circle with the number '5' is next to the 'Host name/address' field, and a red circle with the number '6' is next to the 'Save' button.

7. The interface provides an overview of the databases that your user account has access to. Press **ALT+Shift+Q** within the current database to enter and execute queries.



Note: Read our comprehensive guide if you need help [creating new databases in PostgreSQL](#).

How to Access PostgreSQL Database from Windows

On headless servers, the most common way to access PostgreSQL from Windows is to use a terminal-based solution like **psql**.

In environments that support graphical tools, users can utilize pgAdmin 4 or other GUIs like DBeaver and Navicat.

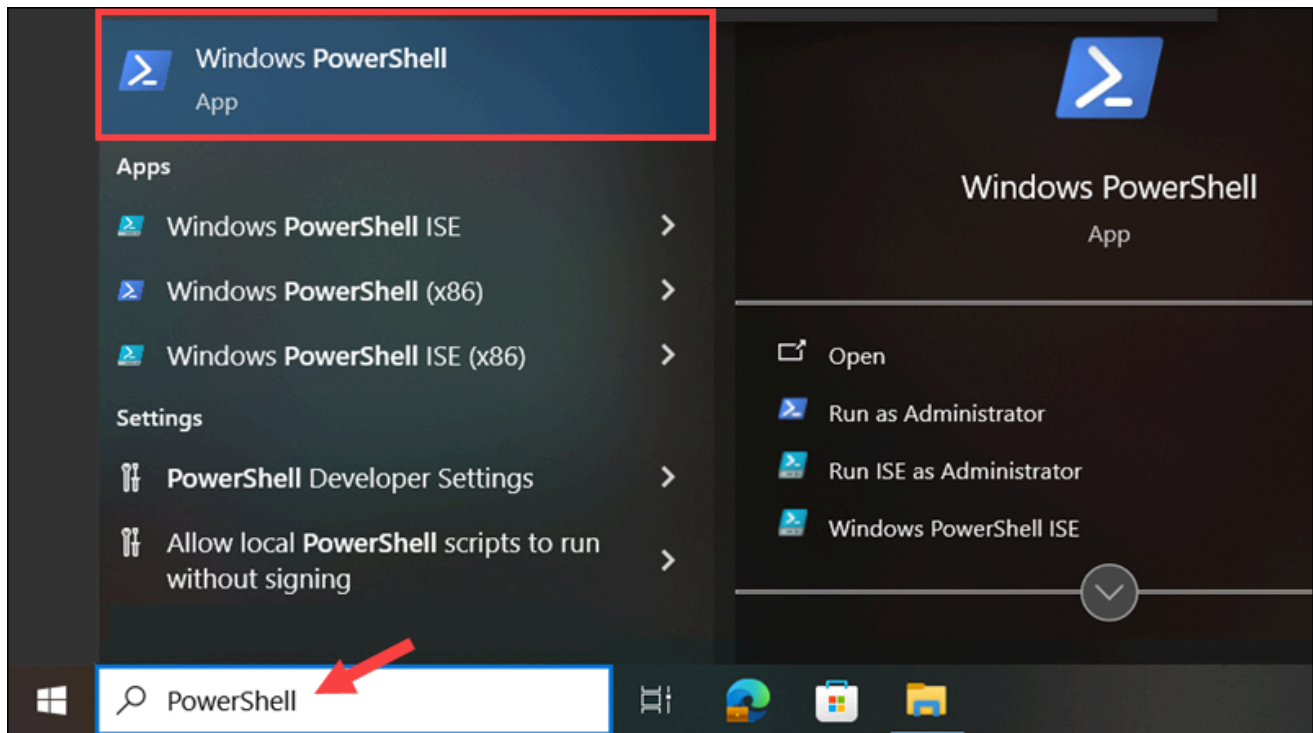


Note: Before proceeding, ensure that you have installed PostgreSQL and added PostgreSQL's `bin` directory to the [Windows PATH environment variable](#).

Connect to PostgreSQL Database via PowerShell

PowerShell is a built-in Windows shell capable of running standard **psql** commands. It also has advanced scripting capabilities that allow users to automate database management tasks.

1. Type **PowerShell** in the Windows Start menu and open the app.



2. Enter the **psql** command and specify the database name, username, and host:

```
psql -U [username] -d [database_name] -h [host]
```

Replace:

- **[username]** with your PostgreSQL database username.
- **[database_name]** with the name of the database you want to connect to.
- **[host]** with the hostname or IP address of the PostgreSQL server.

If the database is on a local machine, you can enter **localhost**. For example, to connect to a local database called **phoenixnap**, using the **postgres** user, enter:


```
psql -U postgres -d phoenixnap -h localhost
```



If the database is password protected, **psql** prompts for the password.

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\WINDOWS\system32> psql -U postgres -d phoenixnap -h localhost
Password for user postgres:
psql (16.2)
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=#
```

You can now use standard **psql** commands to manage the database.



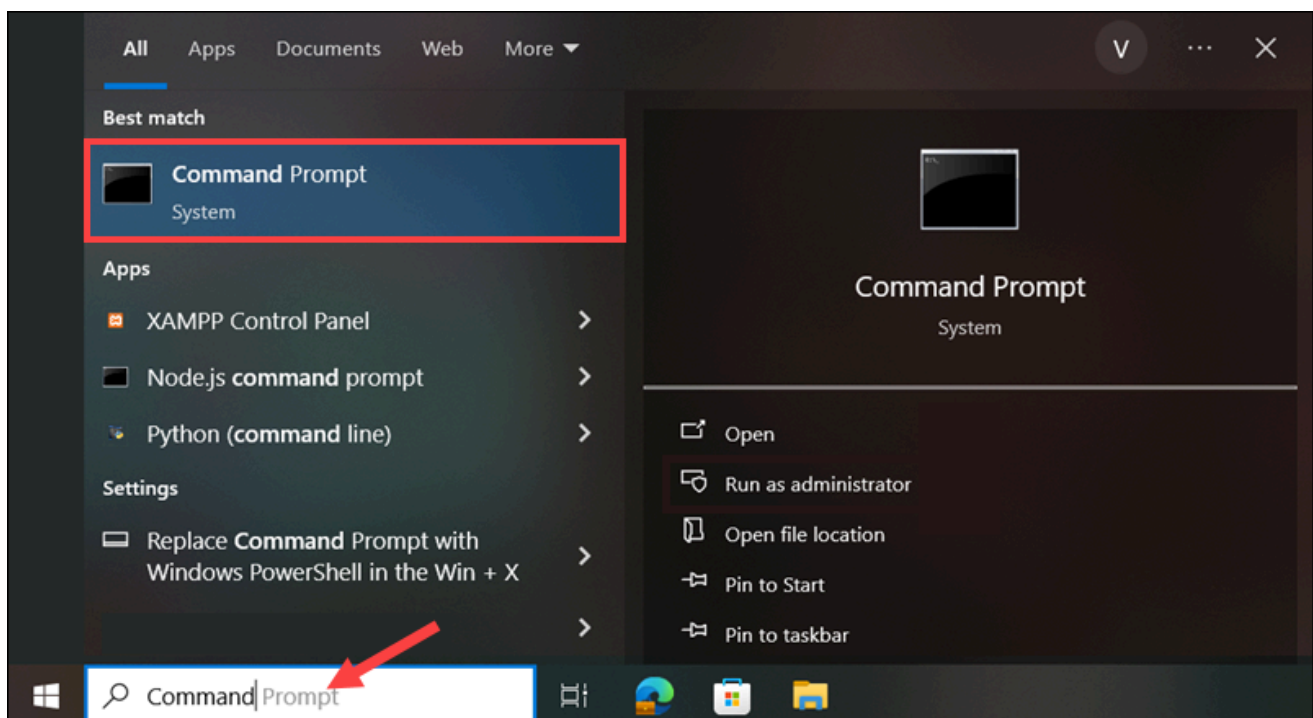
Note: The PostgreSQL server uses **port 5432** by default. If your server is configured to use a custom port, add the **-p [port]** option to the psql command to specify the **port number**.

Type **\q** to close the connection and exit the **psql** session.

Connect to PostgreSQL Database via CMD

To connect to a PostgreSQL database from the Windows Command Prompt (CMD):

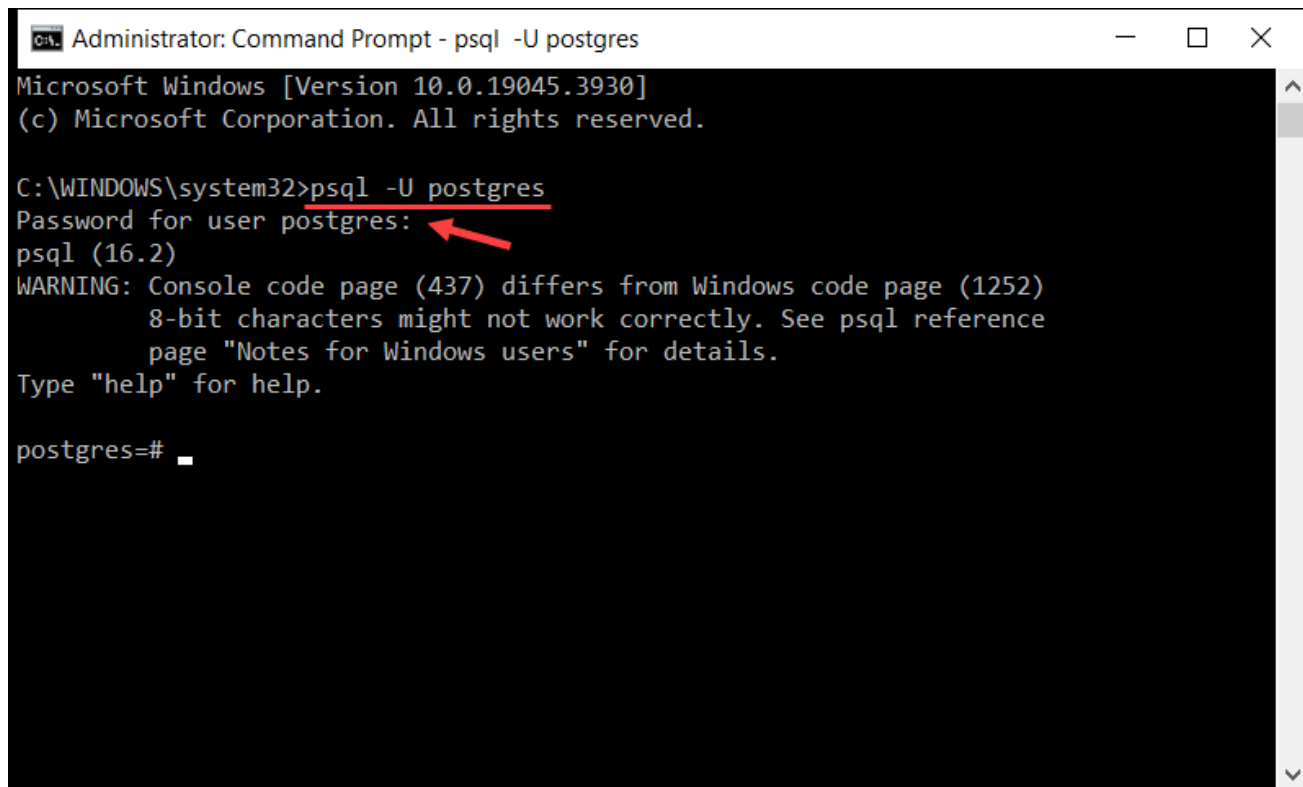
1. Type **Command Prompt** in the Windows search bar and launch the app.



2. Enter the following command to initiate a session as the **postgres** user:

```
psql -U postgres
```

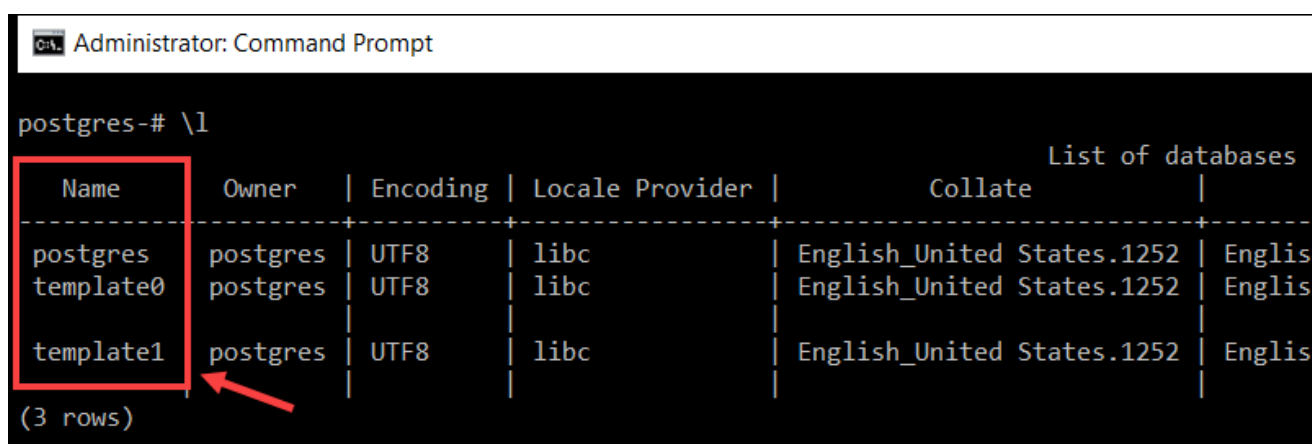
The system prompts you to enter the password for the **postgres** user. If you already created a different user, replace **postgres** in the command above with your PostgreSQL username.



A screenshot of a Windows Command Prompt window titled "Administrator: Command Prompt - psql -U postgres". The window shows the following text: "Microsoft Windows [Version 10.0.19045.3930] (c) Microsoft Corporation. All rights reserved. C:\WINDOWS\system32>psql -U postgres Password for user postgres: psql (16.2) 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=#". A red arrow points to the password prompt.

3. To list existing databases, use the **\l** meta-command:

```
\l
```



A screenshot of a Windows Command Prompt window titled "Administrator: Command Prompt". The window shows the following text: "postgres=# \l List of databases". Below this is a table with 6 columns: Name, Owner, Encoding, Locale Provider, Collate, and another column. The table contains 3 rows of data. A red box highlights the first column (Name) and a red arrow points to the third row (template1).

Name	Owner	Encoding	Locale Provider	Collate	
postgres	postgres	UTF8	libc	English_United States.1252	Englis
template0	postgres	UTF8	libc	English_United States.1252	Englis
template1	postgres	UTF8	libc	English_United States.1252	Englis

(3 rows)

4. To switch to a different database, type **\c** followed by the database name. For example, to connect to the **template1** database, enter:

```
\c template1
```

```
Administrator: Command Prompt

postgres-# \c template1
You are now connected to database "template1" as user "postgres". ←
template1-#
```

Replace **template1** with the name of the database you wish to connect to.

Use the **\q** command to exit **psql** and return to the main CMD interface.

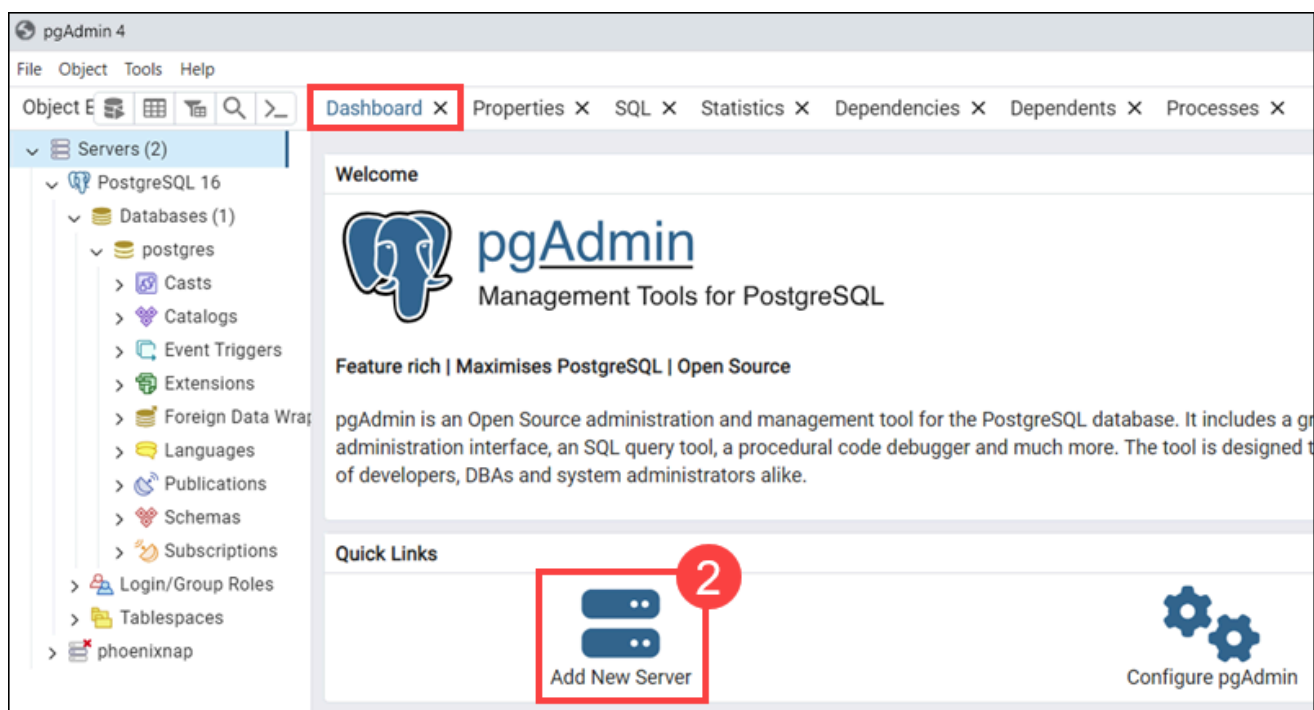
Access PostgreSQL Database via pgAdmin

When [installing PostgreSQL on Windows](#), pgAdmin 4 is often included in the installation bundle.

This graphical interface provides an easy and user-friendly way to log in, administer, and shape databases to fit your requirements.

To access a database using pgAdmin 4:

1. Launch **pgAdmin 4** from the Start Menu or double-click the desktop shortcut.
2. Select **Add New Server** in the *Dashboard* tab.



3. Enter the server name in the **Name** field in the *General* tab.

Register - Server

General | Connection | Parameters | SSH Tunnel | Advanced

Name: 3

Server group:

Background: ☐

Foreground: ☐

Connect now?: ☒

❗ Either Host name or Service must be specified.

Close Reset Save

4. Add the necessary server information and credentials to the *Connection* tab. These include the PostgreSQL hostname/address, port, maintenance database, username, and password.

5. Click **Save** to establish a connection.

Register - Server

General | **Connection** | Parameters | SSH Tunnel | Advanced

Host name/address:

Port:

Maintenance database: 4

Username:

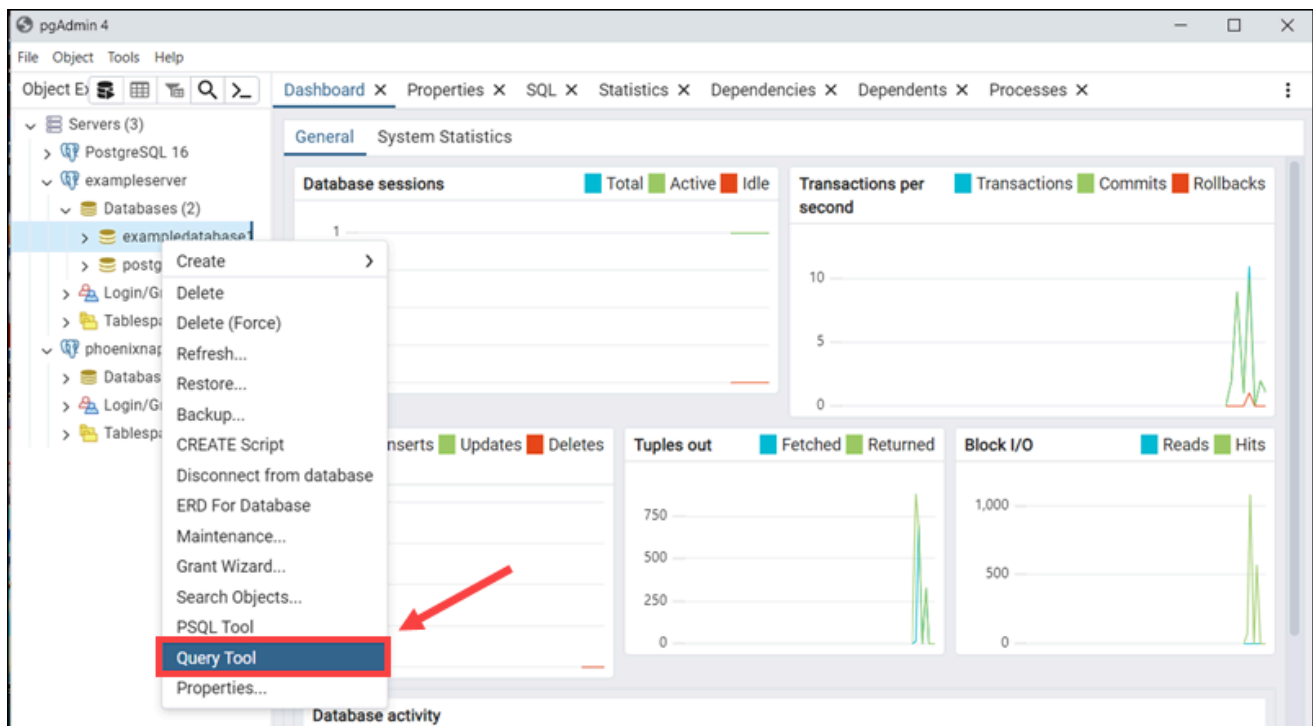
Kerberos authentication?: ☐

Password:

Save password?: ☐

Close Reset Save 5

A list of servers and databases appear in the left navigation panel. To start querying a database, right-click a database and select the **Query Tool** option.



The central field can be used to enter the SQL query.

You can also press **ALT+Shift+Q** to start writing queries for the selected server.

Conclusion

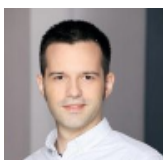
This guide showed you how to connect to your PostgreSQL database on both Linux and Windows servers.

Establishing a connection is an essential first step. Next, explore querying and viewing data within databases using the [PostgreSQL SELECT statement](#).

Was this article helpful?

Yes

No



Vladimir Kaplarevic

Vladimir is a resident Tech Writer at phoenixNAP. He has more than 7 years of experience in implementing e-commerce and online payment solutions with various global IT services providers. His articles aim to instill a passion for innovative technologies in others by providing practical advice and using an engaging writing style.

Next you should read

[MySQL](#)

PostgreSQL Vs MySQL

March 30, 2023

PostgreSQL and MySQL are both excellent database solutions, and well suited for most workloads. Explore the differences between the two most widely used database management systems.

READ MORE

[Databases](#)

PostgreSQL Data Types

July 29, 2021

As a relational database management system, PostgreSQL offers a variety of data types for users to work with. This tutorial covers the different built-in data types available in PostgreSQL.

READ MORE

[Databases, Web Servers](#)

How to Run PostgreSQL on

Docker

November 20, 2023

Run PostgreSQL on Docker and set up the database service for simplified application development. This article shows you two ways to run PostgreSQL on Docker as well as how to connect to your database deployed inside the container.

READ MORE

Databases, SysAdmin How to Install SQL Workbench for PostgreSQL

March 13, 2020

Save time and effort by managing different database systems with a single tool. Find out how to set up SQL Workbench to connect to a PostgreSQL database with four (4) easy steps.

READ MORE

 Live Chat  Get a Quote  Support | 1-855-330-1509  Sales | 1-877-588-5918

[Privacy Center](#) [Do not sell or share my personal information](#)

- [Contact Us](#)
- [Legal](#)
- [Privacy Policy](#)
- [Terms of Use](#)
- [DMCA](#)
- [GDPR](#)
- [Sitemap](#)

©2024 Copyright phoenixNAP | Global IT Services. All Rights Reserved.