

# Connecting with Python

This guide explains how to establish a connection between a **Python** application and a **PostgreSQL** database using the `psycopg2-binary` package. It walks through the necessary setup, configuration, and execution of a simple SQL query.

## Variables

To connect to a PostgreSQL database, you only need one environment variable — the **connection URI**. This URI contains all the necessary information like username, password, host, port, and database name.

Variable	Description	Purpose
<b>POSTGRESQL_URI</b>	Full PostgreSQL connection string (from the Elestio service overview page)	Provides all necessary credentials and endpoint details in a single URI format.

The URI will look like this:

```
postgresql://<USER>:<PASSWORD>@<HOST>:<PORT>/<DATABASE>
```

You can find the details needed in the URI from the **Elestio service overview** details. Copy and replace the variables carefully in the URI example provided above.



postgresql-2p7j1

PostgreSQL

Running

Open terminal

Delete service

Clone this service

Overview

Tools

Backups

Metrics

Monitoring

Logs

Audit

Security

Alerts

Termination protection

Disabled. VM can be powered off and terminated.

Protection deactivated



Database Admin

Display your database credentials

Hide DB Credentials

Host postgresql-2p7j1-u7774.vm.elestio.app



Port 25432



User postgres



Password \*\*\*\*\*

Show password



CLI PGPASSWORD=\*\*\*\*\* psql --host=postgresql-2p7j1-u7774.vm.elestio.app --port=25432 --username=postgres

Show password



# Prerequisites

## Install Python

Check if Python is installed by running:

```
python --version
```

If not installed, download it from [python.org](https://python.org) and install it.

## Install `psycopg2-binary` Package

The `psycopg2-binary` package enables Python applications to interact with PostgreSQL. Install it using:

```
pip install psycopg2-binary
```

## Code

Once all prerequisites are set up, create a new file named `pg.py` and add the following code and replace the `POSTGRESQL_URI` with actual link or in environment setup as you wish:

```
import psycopg2

def get_db_version():
    try:
        db_connection = psycopg2.connect('POSTGRESQL_URI')
        db_cursor = db_connection.cursor()
        db_cursor.execute('SELECT VERSION()')
        db_version = db_cursor.fetchone()[0]
        return db_version

    except Exception as e:
        print(f"Database connection error: {e}")
        return None

    finally:
        if 'db_cursor' in locals():
            db_cursor.close()
        if 'db_connection' in locals():
            db_connection.close()

def display_version():
    version = get_db_version()
    if version:
        print(f"Connected to PostgreSQL: {version}")

if __name__ == "__main__":
    display_version()
```

To execute the script, open the terminal or command prompt and navigate to the directory where `pg.js`. Once in the correct directory, run the script with the command

```
python pg.py
```

If the connection is successful, the terminal will display output similar to:

```
Connected to PostgreSQL: PostgreSQL 16.8 (Debian 16.8-1.pgdg120+1) on x86_64-pc-linux-gnu, compiled by
gcc (Debian 12.2.0-14) 12.2.0, 64-bit
```



---

Revision #4

Created 31 March 2025 07:09:39 by kaiwalya

Updated 31 March 2025 08:55:32 by kaiwalya