

# Installing pgAdmin4 in an Ubuntu LXC Container

## Introduction

**pgAdmin4** is a popular open-source web application for the graphical administration of PostgreSQL databases. It offers a user-friendly interface for managing databases, querying data, creating tables, and many other administrative tasks.

This guide describes the installation of pgAdmin4 in an LXC container running Ubuntu, as well as the connection to a PostgreSQL server using SSL encryption.

## Prerequisites

- An LXC container running Ubuntu 22.04 or 24.04 (e.g., CTID 130, Hostname: **pgadmin**, IP: **192.168.137.130**)
- Internet access within the container
- Access to a PostgreSQL server with the IP **192.168.137.160**
- User **padl** created in the database

## Step 1: Update System

```
apt update && apt upgrade -y
```

## Step 2: Install Necessary Packages

```
apt install -y curl gnupg lsb-release wget ca-certificates
```

### Why is this step necessary?

These packages are often not pre-installed on a fresh Ubuntu LXC container but are absolutely required for the following steps:

Package	Purpose
<b>curl</b>	For downloading the GPG key for the pgAdmin4 repository
<b>gnupg</b>	To process GPG keys and verify repositories
<b>lsb-release</b>	Provides, for example, the current Ubuntu version ( <b>jammy</b> , <b>focal</b> ) for the repository entry
<b>wget</b>	Optional tool for downloading files – helpful for troubleshooting

Package	Purpose
ca-certificates	Provides valid root certificates so that HTTPS connections (e.g., to pgadmin.org) function
<b>Note</b> ⚠ Without these packages, later commands might fail (e.g., "command not found" or "Repository cannot be verified").	

You can check if a package is already installed with:

```
dpkg -l | grep <package name>
# Example:
dpkg -l | grep curl
```

## Step 3: Add PostgreSQL Repository and pgAdmin4 Key

```
curl https://www.pgadmin.org/static/packages_pgadmin_org.pub | gpg --dearmor -o
/usr/share/keyrings/pgadmin-keyring.gpg
echo "deb [signed-by=/usr/share/keyrings/pgadmin-keyring.gpg]
https://ftp.postgresql.org/pub/pgadmin/pgadmin4/apt/$(lsb_release -cs) pgadmin4
main" > /etc/apt/sources.list.d/pgadmin4.list

apt update
```

### Goal of Step 3:

- The trusted GPG key for pgAdmin4 is imported → so apt classifies the source as secure.
- The pgAdmin4 repository is added to your `sources.list.d` → so `apt install pgadmin4-web` works.
- An `apt update` is subsequently performed → so apt finds the new packages.

### What happens in the commands?

```
curl https://www.pgadmin.org/static/packages_pgadmin_org.pub | gpg --dearmor -o
/usr/share/keyrings/pgadmin-keyring.gpg
```

- *Downloads the public GPG key from the official pgAdmin website.*
- *Converts it to the format apt understands and securely stores it under `/usr/share/keyrings`.*

```
echo "deb [signed-by=/usr/share/keyrings/pgadmin-keyring.gpg]
https://ftp.postgresql.org/pub/pgadmin/pgadmin4/apt/$(lsb_release -cs) pgadmin4
main" > /etc/apt/sources.list.d/pgadmin4.list
```

- *Creates a new file (`pgadmin4.list`) in `/etc/apt/sources.list.d/`.*
- *Adds the pgAdmin4 repository – adapted to the Ubuntu version `($(lsb_release -cs))` → e.g., `jammy`*

```
apt update
```

- Loads the package information, including the new pgAdmin4 repository.

### 💡 Why is this necessary?

**Without this step:**

- **The system would not find pgAdmin4 (as it is not included in the official Ubuntu repositories).**
- **Or you would receive a security warning if you attempt to install an unverified package.**

## Step 4: Install pgAdmin4 Web Version

```
apt install -y pgadmin4-web
```

### ☑ Goal of Step 4:

- Installation of the pgAdmin4 web application, including web server integration (Apache2)
- Preparation for the initial configuration in the next step

### 🔍 What exactly happens?

The command `apt install -y pgadmin4-web`:

- Installs pgAdmin4 (web version)
- Automatically installs necessary dependencies such as:
  - Apache2 (as web server)
  - Python packages (e.g., Flask, Gunicorn)
  - pgAdmin4-specific files (e.g., web interface, static content, scripts)

### 📦 Included Components after Installation:

Component	Function
/usr/pgadmin4/	Application code and configuration
/usr/pgadmin4/bin/setup-web.sh	Script for initial configuration of the web interface
apache2	Web server on which the pgAdmin interface is hosted

### 📋 Prerequisite:

**The previous step (`apt update` + pgAdmin4 repository) must be completed so that the pgAdmin4-web package is available.**

## Step 5: Initial Configuration of pgAdmin4

```
/usr/pgadmin4/bin/setup-web.sh
```

The following information is requested:

- E-mail address for admin access (e.g., test@test.de)
- Password
- Web server mode (choose: Yes for Apache2 integration)

#### Goal of Step 5:

- Create an admin user for the web interface
- Set up the Apache2 web server configuration
- Create the necessary configuration files and paths
- Activate pgAdmin4 as a web application under `/pgadmin4`

#### Command:

```
/usr/pgadmin4/bin/setup-web.sh
```

#### What exactly happens during execution?

- User information query:
  - Admin e-mail address
  - Admin password
  - → These credentials are saved in an SQLite database and are required for logging into the web interface.
- Create configuration:
  - `config_local.py` is generated with user-specific settings
  - `pgadmin4.db` (user database) is created
- Apache2 integration:
  - Apache2 is automatically configured (via WSGI module)
  - A new site configuration for `/pgadmin4` is activated
  - Apache2 is restarted so the web interface works immediately

#### Result:

After successful setup, pgAdmin4 is reachable at:

```
http://<container-ip>/pgadmin4
```

Example:

```
http://192.168.137.130/pgadmin4
```

#### Files that are created/adjusted:

Path	Description
/var/lib/pgadmin/pgadmin4.db	SQLite database with user accounts
/var/lib/pgadmin/	Working directory for pgAdmin
/etc/apache2/conf-enabled/pgadmin4.conf	Apache configuration for the web interface
/usr/pgadmin4/web/config_local.py	Local configuration (e.g., mail server, paths, access)

## Step 6: Firewall & Test Access

Ensure that the LXC container is reachable via the browser (e.g., <http://192.168.137.130/pgadmin4>). If UFW is active:

```
ufw allow 80 //for HTTP protocol
ufw allow 443 //for HTTPS protocol
```

## Step 7: Connect PostgreSQL Server with SSL Connection in pgAdmin4

After successfully setting up SSL encryption on the PostgreSQL server (see documentation [[2600 postgresql.md]]), encrypted access can be established via pgAdmin4.

### Create Server Connection in pgAdmin4

1. Open the pgAdmin4 web interface in the browser:

<http://192.168.137.130/pgadmin4>

2. Log in with the previously created admin account.

3. Right-click on "**Servers**" in the left panel and select "**Create**" → "**Server...**"

### General Tab

Field	Input
Name	PostgreSQL-SSL (can be chosen freely)

### Connection Tab

Field	Input
Host name/address	192.168.137.160
Port	5432
Maintenance database	postgres (Standard)
Username	e.g., pdal

Field	Input
<b>Password</b>	[Enter password]
<b>Save Password?</b>	<input checked="" type="checkbox"/> activate

**⚠ Note:** For the connection to work, the PostgreSQL server must allow the pgAdmin4 container's IP (see [pg\\_hba.conf](#)) and be configured for SSL ([postgresql.conf](#)).

## SSL Tab

Field	Input
<b>SSL mode</b>	<code>require</code> , <code>verify-ca</code> , or <code>verify-full</code> (depending on security requirements and server configuration)
<b>Root Certificate</b>	Only for <code>verify-ca</code> or <code>verify-full</code> : e.g., <code>/usr/local/share/ca-certificates/postgres.crt</code>
<b>Client Certificate</b>	<i>optional</i> , only if mTLS (Client Certificates) is active
<b>Client Certificate Key</b>	<i>optional</i>

**⚠** If no Root Certificate is specified, `require` can be used to establish an encrypted but unverified connection.

## Test and Save Connection

- Click "**Save**"
- The connection now appears on the left under "Servers".
- If configuration is successful, the database structure is loaded.

### **⚠ Note**

If the connection fails:

- **Check error message:** e.g., certificate error or "Wrong authentication".
- **Temporarily set SSL Mode to disable** to check if the issue is SSL-related.
- **View log files on the PostgreSQL server** (`/var/log/postgresql/`) to analyze connection attempts and errors.

## The pgAdmin4 Interface

On the left side, under the assigned name, you will find "Databases". The database you have already created should appear here. Under "Schemas" -> "Tables", the example database should be visible. Here, you can easily create databases or tables. Testing queries is also very simple here.

Under "Login/Group Roles", the roles we created should be visible. You can also create additional roles here.

In the upper right corner, under the email address, you will find the user administration. Here, you should create an admin user for pgAdmin4. All other users can also be created here.

---

## Sources

- "pgAdmin4 — pgAdmin4 9.5 documentation". Accessed: July 22, 2025. [Online]. Available at: [pgadmin4 Doc](#)
  - "18.9. Secure TCP/IP Connections with SSL", PostgreSQL Documentation. Accessed: July 22, 2025. [Online]. Available at: [PostgreSQL Doc](#)
- 

## License

This work is licensed under the **Creative Commons Attribution-ShareAlike 4.0 International License**.

[To the license text on the Creative Commons website](#)