



I am going to ask you a series of questions about restoring or installing from scratch a PostgreSQL database on a Ubuntu 20.04 machine using the command line as much as possible. I have sudo in that machine. What I want from you is to not give me a series of alternatives or options on each step, just supply a single set of instructions for the most normative/canonical way to proceed and if that fails, then, and only then, you can propose other alternatives. First off let's start by this: what is the highest version of PostgreSQL that can be straightforwardly installed from official repositories for Ubuntu 20.04?

For Ubuntu 20.04, the highest PostgreSQL version available from the official Ubuntu repositories is **PostgreSQL 12**. This comes directly in the default focal repositories and is the most straightforward installation path.

### Why not 13+ on 20.04?

- PostgreSQL 13, 14, 15, etc., are available *only* from the official PostgreSQL APT repository ([apt.postgresql.org](http://apt.postgresql.org)), which requires adding a third-party repository.
- Your request specified the "most normative/canonical way" from **official Ubuntu repositories**, so that is PostgreSQL 12.

To install PostgreSQL 12 from Ubuntu's official repositories:

bash

```
sudo apt update
sudo apt install postgresql-12
```

If you're willing to use the official PostgreSQL repository instead, I can give you the straightforward instructions to install the latest stable version (currently PostgreSQL 16) on Ubuntu 20.04. Let me know if you want that instead.



So here's the thing. In the past I had installed PostgreSQL 12 on that Ubuntu 20.04 machine but I messed up (deleted) its data directory and now I think I would like to start with a clean install. I am also open to installing PostgreSQL from the official PostgreSQL APT repository. But first I think we ought to delete any traces of the existing postgresql installations on my system.

Good approach. Let's first completely remove existing PostgreSQL installations and then install the latest stable version from the official PostgreSQL APT repository.

### Step 1: Remove all existing PostgreSQL packages and data

bash

```
# Stop all PostgreSQL services
sudo systemctl stop postgresql*

# Remove all PostgreSQL packages and configuration
sudo apt purge postgresql-* -y
sudo apt autoremove -y

# Remove leftover data directories (adjust if yours was elsewhere)
sudo rm -rf /var/lib/postgresql/
sudo rm -rf /etc/postgresql/
sudo rm -rf /var/log/postgresql/
```

### Step 2: Install PostgreSQL 16 from official PostgreSQL repository

bash

```
# Add PostgreSQL repository and signing key
sudo apt install wget ca-certificates -y
sudo sh -c 'echo "deb https://apt.postgresql.org/pub/repos/apt $(lsb_release -cs)-pgdg main" > /etc/apt/sources.list.
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -'

# Install PostgreSQL 16
sudo apt update
sudo apt install postgresql-16 postgresql-client-16 -y
```

### Step 3: Verify installation

bash

```
# Check service status
sudo systemctl status postgresql@16-main

# Verify connection
sudo -u postgres psql -c "SELECT version();"
```

This gives you:

- A clean installation of PostgreSQL 16 (latest stable as of now)
- All old PostgreSQL 12 packages, config, and data completely removed
- The standard PostgreSQL directory structure at `/var/lib/postgresql/16/main/`

The `postgresql-16` package automatically creates a fresh database cluster and starts the service. You're now ready to proceed with database restoration or setup.



I finished step #1. Before doing step #2 I just wanted to let you know that I already have the following in `/etc/apt/sources.list.d/postgresql.list`:  
`deb http://apt.postgresql.org/pub/repos/apt focal-pgdg main` So I am not sure if the following: `$ sudo sh -c 'echo "deb https://apt.postgresql.org/pub/repos/apt $(lsb_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'` is really necessary

You're right. Your existing `/etc/apt/sources.list.d/pgdg.list` is correct for Ubuntu 20.04 (focal). The `focal-pgdg` is the correct suite name, not `focal-pgdg` as my command would generate.

The only difference is you have `http://` while I used `https://`, but both work. Let's proceed with Step 2 using your existing repository:

### Step 2: Install PostgreSQL 16 from official PostgreSQL repository

bash

```
# Update package list (using your existing repository)
sudo apt update

# Install PostgreSQL 16 and client
sudo apt install postgresql-16 postgresql-client-16 -y
```

### Step 3: Verify installation

bash

```
# Check service status
sudo systemctl status postgresql@16-main

# Verify connection
sudo -u postgres psql -c "SELECT version();"
```

**Note:** Since you already had the PostgreSQL repository configured, the installation will be straightforward. The system will automatically create a fresh database cluster at `/var/lib/postgresql/16/main/` and start the service.

The following command: \$ sudo apt install postgresql-16 postgresql-client-16 -y ... failed with the following message:  
Reading package lists... Done Building dependency tree Reading state information... Done The following additional packages will be installed: libio-pty-perl libipc-run-perl libpq5 postgresql-client-common postgresql-common postgresql-common-dev sysstat Suggested packages: postgresql-doc-16 isag The following NEW packages will be installed: libio-pty-perl libipc-run-perl postgresql-16 postgresql-client-16 postgresql-client-common postgresql-common postgresql-common-dev sysstat The following packages will be upgraded: libpq5 1 upgraded, 8 newly installed, 0 to remove and 213 not upgraded. Need to get 20.2 MB/20.7 MB of archives. After this operation, 72.9 MB of additional disk space will be used. Err:1 http://apt.postgresql.org/pub/repos/apt focal-pgdg/main amd64 postgresql-client-common all 278.pgdg20.04+1 404 Not Found [IP: 146.75.55.52 80] Err:2 http://apt.postgresql.org/pub/repos/apt focal-pgdg/main amd64 postgresql-common-dev all 278.pgdg20.04+1 404 Not Found [IP: 146.75.55.52 80] Err:3 http://apt.postgresql.org/pub/repos/apt focal-pgdg/main amd64 postgresql-client-16 amd64 16.9-1.pgdg20.04+1 404 Not Found [IP: 146.75.55.52 80] Err:4 http://apt.postgresql.org/pub/repos/apt focal-pgdg/main amd64 libpq5 amd64 17.5-1.pgdg20.04+1 404 Not Found [IP: 146.75.55.52 80] Err:5 http://apt.postgresql.org/pub/repos/apt focal-pgdg/main amd64 postgresql-client-16 amd64 16.9-1.pgdg20.04+1 404 Not Found [IP: 146.75.55.52 80] Err:6 http://apt.postgresql.org/pub/repos/apt focal-pgdg/main amd64 postgresql-common\_278.pgdg20.04+1\_all.deb 404 Not Found [IP: 146.75.55.52 80] E: Failed to fetch http://apt.postgresql.org/pub/repos/apt/pool/main/p/postgresql-common/postgresql-client-common\_278.pgdg20.04+1\_all.deb 404 Not Found [IP: 146.75.55.52 80] E: Failed to fetch http://apt.postgresql.org/pub/repos/apt/pool/main/p/postgresql-common/postgresql-common\_278.pgdg20.04+1\_all.deb 404 Not Found [IP: 146.75.55.52 80] E: Failed to fetch http://apt.postgresql.org/pub/repos/apt/pool/main/p/postgresql-common/postgresql-common\_278.pgdg20.04+1\_all.deb 404 Not Found [IP: 146.75.55.52 80] E: Failed to fetch http://apt.postgresql.org/pub/repos/apt/pool/main/p/postgresql-client-16\_16.9-1.pgdg20.04+1\_amd64.deb 404 Not Found [IP: 146.75.55.52 80] E: Failed to fetch http://apt.postgresql.org/pub/repos/apt/pool/main/p/postgresql-16/postgresql-client-16\_16.9-1.pgdg20.04+1\_amd64.deb 404 Not Found [IP: 146.75.55.52 80] E: Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?

The error shows the repository is trying to fetch packages with version pgdg20.04+1, but those packages don't exist (404 Not Found). The repository structure has changed, and your current repository entry needs updating.

Let's fix the repository configuration and install PostgreSQL 16:

## Step 2 (Revised): Update repository and install PostgreSQL 16

bash

```
# Remove the old repository file
sudo rm /etc/apt/sources.list.d/pgdg.list

# Add the correct PostgreSQL repository for Ubuntu 20.04
sudo sh -c 'echo "deb https://apt.postgresql.org/pub/repos/apt focal-pgdg main" > /etc/apt/sources.list.d/pgdg.list'

# Add the repository signing key
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -

# Update package lists
sudo apt update

# Install PostgreSQL 16
sudo apt install postgresql-16 postgresql-client-16 -y
```

### Key changes:

1. Removed the old repository file
2. Used `https://` instead of `http://`
3. Used `focal-pgdg` as the suite (not `focal-pgdg main` which would be wrong)
4. Re-added the signing key

This should resolve the 404 errors and allow installation of PostgreSQL 16.

This time the installation went much better, but not entirely without errors. Here's the output. The exit code was 100 as you can see below: \$ sudo apt update Get:1 https://brave-browser-apt-release.s3.brave.com stable InRelease [7,547 B] Hit:2 http://archive.canonical.com/ubuntu focal InRelease Err:1 https://brave-browser-apt-release.s3.brave.com stable InRelease The following signatures couldn't be verified because the public key is not available: NO\_PUBKEY 0686B78420038257 Hit:3 http://gr.archive.ubuntu.com/ubuntu focal InRelease Hit:4 http://gr.archive.ubuntu.com/ubuntu focal-backports InRelease Hit:6 http://dl.google.com/linux/chrome/deb stable InRelease Get:7 https://download.docker.com/linux/ubuntu focal InRelease [57.7 kB] Hit:8 http://dl.google.com/linux/earth/deb stable InRelease Hit:9 https://deb.nodesource.com/node\_14.x focal InRelease Hit:10 http://ppa.launchpad.net/kollowk/omacs/ubuntu focal InRelease Hit:11

```
http://ppa.launchpad.net/kelleyk/emacs/ubuntu focal InRelease Hit:11
http://ppa.launchpad.net/unit193/encryption/ubuntu focal InRelease Hit:12 https://packages.microsoft.com/repos/ms-
teams stable InRelease Ign:13 https://apt.postgresql.org/pub/repos/apt focal-pgdg InRelease Err:14
https://apt.postgresql.org/pub/repos/apt focal-pgdg Release 404 Not Found [IP: 146.75.55.52 443] Get:15
https://packages.cloud.google.com/apt cloud-sdk InRelease [1,620 B] Hit:16 https://cloud.r-project.org/bin/linux/ubuntu
focal-cran40/ InRelease Hit:17 https://deb.torproject.org/torproject.org noble InRelease Get:18
https://packages.cloud.google.com/apt cloud-sdk/main amd64 Packages [4,469 kB] Get:19
https://packages.cloud.google.com/apt cloud-sdk/main all Packages [1,929 kB] Hit:20
http://security.ubuntu.com/ubuntu focal-security InRelease Reading package lists... Done W: An error occurred during
the signature verification. The repository is not updated and the previous index files will be used. GPG error:
https://brave-browser-apt-release.s3.brave.com stable InRelease: The following signatures couldn't be verified
because the public key is not available: NO_PUBKEY 0686B78420038257 E: The repository
'https://apt.postgresql.org/pub/repos/apt focal-pgdg Release' no longer has a Release file. N: Updating from such a
repository can't be done securely, and is therefore disabled by default. N: See apt-secure(8) manpage for repository
creation and user configuration details. N: Skipping acquire of configured file 'main/binary-i386/Packages' as
repository 'https://deb.torproject.org/torproject.org noble InRelease' doesn't support architecture 'i386'
mperikeas@mp-ThinkStation-P320:~# $ echo $? 100
```

I see the issue. The repository structure has changed. For Ubuntu 20.04, you need to use focal-pgdg but with the older format. Let's try the correct approach:

**First, let's check what's available in the PostgreSQL repository:**

```
bash
```

```
# Remove the current repository
sudo rm /etc/apt/sources.list.d/pgdg.list

# Check what's available for focal (Ubuntu 20.04)
curl -s https://apt.postgresql.org/pub/repos/apt/dists/ | grep focal
```

Based on the common pattern, try this configuration:

```
bash
```

```
# Create the correct repository file for Ubuntu 20.04
sudo sh -c 'echo "deb [arch=amd64] https://apt.postgresql.org/pub/repos/apt/ $(lsb_release -cs)-pgdg main" > /etc/apt

# Add the signing key (if not already added)
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -

# Update and install
sudo apt update
sudo apt install postgresql-16 postgresql-client-16 -y
```

**If that still fails**, let's use the most reliable method for Ubuntu 20.04:

```
bash
```

```
# Remove any existing PostgreSQL repo
sudo rm /etc/apt/sources.list.d/pgdg.list

# Use the PostgreSQL repository setup script
sudo apt install curl ca-certificates gnupg -y
curl https://www.postgresql.org/media/keys/ACCC4CF8.asc | gpg --dearmor | sudo tee /etc/apt/trusted.gpg.d/apt.postgre
sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt $(lsb_release -cs)-pgdg main" > /etc/apt/sources.list.d

# Update and install
sudo apt update
sudo apt install postgresql-16 postgresql-client-16 -y
```

**Note:** The key issue is that PostgreSQL recently changed their repository structure. The -pgdg suffix is the correct one for Ubuntu 20.04. If none of these work, we can install from Ubuntu's official repositories (PostgreSQL 12) as a fallback, but let's try to get PostgreSQL 16 working first.



Now "\$ sudo apt update" fails as follows: \$ sudo apt update Get:1 https://brave-browser-apt-release.s3.brave.com
stable InRelease [7,547 B] Hit:2 http://archive.canonical.com/ubuntu focal InRelease Err:1 https://brave-browser-apt-
release.s3.brave.com stable InRelease The following signatures couldn't be verified because the public key is not
available: NO\_PUBKEY 0686B78420038257 Hit:3 http://gr.archive.ubuntu.com/ubuntu focal InRelease Hit:4
http://gr.archive.ubuntu.com/ubuntu focal-updates InRelease Hit:5 http://gr.archive.ubuntu.com/ubuntu focal-backports
InRelease Hit:6 http://dl.google.com/linux/chrome/deb stable InRelease Get:7
https://download.docker.com/linux/ubuntu focal InRelease [57.7 kB] Hit:8 http://dl.google.com/linux/earth/deb stable
InRelease Hit:9 https://deb.nodesource.com/node\_14.x focal InRelease Hit:10
http://ppa.launchpad.net/kelleyk/emacs/ubuntu focal InRelease Hit:11
http://ppa.launchpad.net/unit193/encryption/ubuntu focal InRelease Hit:12 https://packages.microsoft.com/repos/ms-
teams stable InRelease Ign:13 https://apt.postgresql.org/pub/repos/apt focal-pgdg InRelease Err:14
https://apt.postgresql.org/pub/repos/apt focal-pgdg Release 404 Not Found [IP: 146.75.55.52 443] Get:15
https://packages.cloud.google.com/apt cloud-sdk InRelease [1,620 B] Hit:16 https://cloud.r-project.org/bin/linux/ubuntu

```
.../packages.cloud.google.com/apt/cuda/bionic InRelease [1,822 kB] Hit:17 https://deb.torproject.org/torproject.org noble InRelease Get:18  
https://packages.cloud.google.com/apt/cuda/main amd64 Packages [4,469 kB] Get:19  
https://packages.cloud.google.com/apt/cuda/main all Packages [1,929 kB] Hit:20  
http://security.ubuntu.com/ubuntu focal-security InRelease Reading package lists... Done W: An error occurred during  
the signature verification. The repository is not updated and the previous index files will be used. GPG error:  
https://brave-browser-apt-release.s3.brave.com stable InRelease: The following signatures couldn't be verified  
because the public key is not available: NO_PUBKEY 0686B78420038257 E: The repository  
'https://apt.postgresql.org/pub/repos/apt focal-pgdg Release' no longer has a Release file. N: Updating from such a  
repository can't be done securely, and is therefore disabled by default. N: See apt-secure(8) manpage for repository  
creation and user configuration details. N: Skipping acquire of configured file 'main/binary-i386/Packages' as  
repository 'https://deb.torproject.org/torproject.org noble InRelease' doesn't support architecture 'i386'  
mperdikeas@mp-ThinkStation-P320:~# $ echo $? 100
```

The PostgreSQL repository has changed its structure. For Ubuntu 20.04, the correct suite name is now simply **focal** without the -pgdg suffix. Let's fix this:

```
bash  
# Remove the current repository file  
sudo rm /etc/apt/sources.list.d/pgdg.list  
  
# Create the correct repository configuration for Ubuntu 20.04  
sudo sh -c 'echo "deb https://apt.postgresql.org/pub/repos/apt focal main" > /etc/apt/sources.list.d/pgdg.list'  
  
# Update package lists  
sudo apt update  
  
# Now install PostgreSQL 16  
sudo apt install postgresql-16 postgresql-client-16 -y
```

If that works, you'll have PostgreSQL 16 installed. If you still get a 404 error, let's check what's actually available:

```
bash  
# Check what distributions are available  
curl -s https://apt.postgresql.org/pub/repos/apt/dists/ | grep -E "focal|Name"
```

**Fallback option:** If the PostgreSQL repository continues to fail, let's install PostgreSQL 14 from Ubuntu's backports (still newer than 12).

```
bash  
# Remove PostgreSQL repository  
sudo rm /etc/apt/sources.list.d/pgdg.list  
  
# Enable backports and install PostgreSQL 14  
sudo apt update  
sudo apt install -t focal-backports postgresql-14 postgresql-client-14 -y
```

PostgreSQL 14 is available in Ubuntu 20.04 backports and is a reasonable modern version. Let me know which path you want to take.



I followed your instructions and my /etc/apt/sources.list.d/pgdg.list has now the following content: \$ sudo cat /etc/apt/sources.list.d/pgdg.list deb https://apt.postgresql.org/pub/repos/apt focal main Even with that "sudo apt update" and "sudo apt install postgresql-16 postgresql-client-16 -y" still fail with the following traces: ---%<-----\$ sudo apt update Get:1 https://brave-browser-apt-release.s3.brave.com stable InRelease [7,547 B] Hit:2 http://archive.canonical.com/ubuntu focal InRelease Err:1 https://brave-browser-apt-release.s3.brave.com stable InRelease The following signatures couldn't be verified because the public key is not available: NO\_PUBKEY 0686B78420038257 Get:3 https://download.docker.com/linux/ubuntu focal InRelease [57.7 kB] Hit:4 https://deb.nodesource.com/node\_14.x focal InRelease Hit:5 http://dl.google.com/linux/chrome/deb stable InRelease Hit:6 http://dl.google.com/linux/earth/deb stable InRelease Hit:7 https://torproject.org/bin/linux/ubuntu focal-cran40/ InRelease Hit:8 http://ppa.launchpad.net/kelleyk/emacs/ubuntu focal InRelease Hit:9 http://ppa.launchpad.net/unit193/encryption/ubuntu focal InRelease Hit:10 https://apt.postgresql.org/pub/repos/apt focal InRelease Err:11 https://apt.postgresql.org/pub/repos/apt focal Release 404 Not Found [IP: 146.75.55.52 443] Hit:12 https://packages.microsoft.com/repos/ms-teams stable InRelease Hit:13 https://packages.cloud.google.com/apt cloud-sdk InRelease Hit:14 https://deb.torproject.org/torproject.org noble InRelease Hit:15 http://security.ubuntu.com/ubuntu focal-security InRelease Hit:16 http://gr.archive.ubuntu.com/ubuntu focal InRelease Hit:17 http://gr.archive.ubuntu.com/ubuntu focal-updates InRelease Hit:18 http://gr.archive.ubuntu.com/ubuntu focal-backports InRelease Reading package lists... Done W: An error occurred during the signature verification. The repository is not updated and the previous index files will be used. GPG error: https://brave-browser-apt-release.s3.brave.com stable InRelease: The following signatures couldn't be verified because the public key is not available: NO\_PUBKEY 0686B78420038257 L: The repository 'https://apt.postgresql.org/pub/repos/apt focal Release' does not have a Release file. N: Updating from such a repository can't be done securely, and is therefore disabled by default. N: See apt-secure(8) manpage for repository creation and user configuration details. N: Skipping acquire of configured file 'main/binary-i386/Packages' as repository 'https://deb.torproject.org/torproject.org noble InRelease' doesn't support architecture 'i386' mp-ardikeas@mp-ThinkStation-P320:~# \$ sudo apt install postgresql-16 postgresql-client-16 -y Reading package lists... Done Building dependency tree Reading state information... Done E: Unable to locate package postgresql-16 L: Unable to locate package postgresql-client-16 -----%>--- So I'd say let's clean up and try to install PostgreSQL 14.

Good plan. Let's clean up and install PostgreSQL 14 from Ubuntu's backports.

### Step 1: Remove PostgreSQL repository and clean up

bash

```
# Remove PostgreSQL repository
sudo rm /etc/apt/sources.list.d/pgdg.list

# Update package lists
sudo apt update
```

### Step 2: Install PostgreSQL 14 from Ubuntu 20.04 backports

bash

```
# Install PostgreSQL 14 from backports
sudo apt install -t focal-backports postgresql-14 postgresql-client-14 -y
```

### Step 3: Verify installation

bash

```
# Check service status
sudo systemctl status postgresql@14-main

# Verify connection
sudo -u postgres psql -c "SELECT version();"
```

**Note:** PostgreSQL 14 is a well-supported version (supported until November 2026) and should work reliably on Ubuntu 20.04. The backports repository is maintained by Ubuntu and should work without the repository issues we encountered with the PostgreSQL APT repository.

If you encounter any issues with the backports installation, we have one more straightforward option: install PostgreSQL 12 from the main Ubuntu repositories (which we know works). But let's try PostgreSQL 14 first.



Well, that doesn't work either. I get: \$ sudo apt install -t focal-backports postgresql-14 postgresql-client-14 -y Reading package lists... Done Building dependency tree Reading state information... Done E: Unable to locate package postgresql-14 E: Unable to locate package postgresql-client-14 Googling I read the following: ---%<-----Ubuntu 20.04 (Focal Fossa) reached end-of-life (EOL) on April 30, 2025 for standard support, and the official PostgreSQL APT repository (apt.postgresql.org) no longer hosts packages for it as of July 1, 2025. Packages for EOL releases like Focal have been archived to apt-archive.postgresql.org. -----%>--- The following medium post suggests how to go about installing PostgreSQL 14 on Ubuntu 20.04 \*after\* December 2025. Should I go about it? <https://medium.com/@yadavarvind100100/how-to-install-postgresql-14-pgadmin4-on-ubuntu-20-04-in-december-2025-after-eol-253265e80c13>

Yes, you've identified the core issue. Since Ubuntu 20.04 reached end-of-life in April 2025, the PostgreSQL APT repository has archived packages for it. Let's follow the approach from that article.

### Step 1: Remove any existing PostgreSQL repository files

bash

```
sudo rm -f /etc/apt/sources.list.d/pgdg.list
```

### Step 2: Add the PostgreSQL archive repository for Ubuntu 20.04

bash

```
# Create repository file for archived packages
echo "deb http://apt-archive.postgresql.org/pub/repos/ubuntu-pgdg main" | sudo tee /etc/apt/sources.list.d/pgdg-archive

# Add the PostgreSQL signing key (if not already added)
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -

# Update package lists
sudo apt update
```

### Step 3: Install PostgreSQL 14 (latest version available in archive)

bash

```
# Install PostgreSQL 14
sudo apt install postgresql-14 postgresql-client-14 -y
```

### Step 4: Verify installation

bash

```
# Check service status
sudo systemctl status postgresql@14-main

# Verify connection
sudo -u postgres psql -c "SELECT version();"
```

#### Important notes:

1. We're using <http://apt-archive.postgresql.org> instead of <https://apt.postgresql.org>
2. We're using `focal-pgdg` as the distribution name (the archived repository still uses this format)
3. This should give you PostgreSQL 14, which is a good modern version

If this still fails, we have one more reliable option: install PostgreSQL 12 from Ubuntu's main repositories (which should still work even after EOL since we're using local mirrors).



Cool. I managed to install PostgreSQL 14 basically following the Medium article I linked to but also your instructions. Now, under `/lib/systemd/system` I see the following files: `postgresql.service` `postgresql@.service` Which one was installed with PostgreSQL 14 and can I safely delete the other one?

The files in `/lib/systemd/system/` are **template service files** provided by the `postgresql-common` package. Both files should remain as they serve different purposes:

1. **`postgresql.service`** - This is a **global service** that manages all PostgreSQL clusters on the system. When you run `sudo systemctl start postgresql`, it starts all PostgreSQL clusters.
2. **`postgresql@.service`** - This is a **template service** for individual clusters. When you run `sudo systemctl start postgresql@14-main`, it starts only the PostgreSQL 14 cluster with data directory at `/var/lib/postgresql/14/main/`.

**Do NOT delete either file.** They work together:

- `postgresql.service` is the main service unit
- `postgresql@.service` is a template used for version-specific instances

**Check what's actually running:**

bash

```
# Check which services are active
sudo systemctl status postgresql
```

```
sudo systemctl status postgresql@14-main
# List all PostgreSQL-related services
systemctl list-units --all | grep postgresql
```

**What you likely have:**

- Both files were installed with the `postgresql-common` package (dependency of `postgresql-14`)
- Your actual running service is `postgresql@14-main.service`
- The `postgresql.service` acts as a master controller

**Leave both files intact.** If you delete one, PostgreSQL service management will break.

DEMOWATERMARK



Cool. Can you explain the output of this command: `$ systemctl list-units --all | grep postgresql postgresql.service`  
loaded active exited PostgreSQL RDBMS • `postgresql@12-main.service` loaded failed failed PostgreSQL Cluster 12-main `postgresql@14-main.service` loaded active running PostgreSQL Cluster 14-main `system-postgresql.slice` loaded active active system-postgresql.slice In particular why is the "postgresql.service" shown as "exited" and what is the "postgresql@12-main.service" ? I thought I had deleted all previous versions of PostgreSQL before installing PostgreSQL 14 in my system.

Excellent question. Let me explain each line:

## 1. `postgresql.service` - loaded active exited

This is **normal and correct**. The exited state means:

- The service **completed its job successfully** and terminated
- This is a **oneshot service** that doesn't run continuously
- Its job is to **start all individual PostgreSQL cluster services** (like `postgresql@14-main`)
- After starting them, it exits (hence `exited`)
- You can see this with: `sudo systemctl cat postgresql.service`

## 2. `postgresql@12-main.service` - loaded failed failed

This indicates a **leftover systemd unit** from your previous PostgreSQL 12 installation. Even though you removed the PostgreSQL 12 package, the systemd service file wasn't removed because:

- It's a **template service instance** (@12-main)
- Systemd still has it in its unit registry
- The service fails because PostgreSQL 12 binaries/data no longer exist

To clean this up:

bash

```
# Disable and remove the leftover service
sudo systemctl disable postgresql@12-main
sudo rm -f /etc/systemd/system/multi-user.target.wants/postgresql@12-main.service

# Reload systemd and verify
sudo systemctl daemon-reload
systemctl list-units --all | grep postgresql
```

## 3. `postgresql@14-main.service` - loaded active running

This is **your current PostgreSQL 14 cluster** - running correctly.

## 4. `system-postgresql.slice` - loaded active active

This is a **systemd slice** for resource management (CPU, memory limits) of PostgreSQL processes. It's automatically created and is normal.

## Summary:

- `postgresql.service` (exited): ✓ **Normal** - started your cluster and exited
- `postgresql@12-main.service` (failed): ⚠ **Leftover** - needs cleanup
- `postgresql@14-main.service` (running): ✓ **Your working PostgreSQL 14**
- `system-postgresql.slice` (active): ✓ **Normal systemd resource group**

The leftover `postgresql@12-main.service` is just a systemd unit file remnant - no PostgreSQL 12 binaries or data exist on your system anymore. Clean it up with the commands above.



The following command: `$ sudo systemctl disable postgresql@12-main` Succeeded and exited with status code 0

The following command, `sudo systemctl disable postgresql@12-main` succeeded and exited with status code 0. However, the following file does not exist: `/etc/systemd/system/multi-user.target.wants/postgresql@12-main.service`. Only this file exists: `/etc/systemd/system/multi-user.target.wants/postgresql.service`

That's expected and good! Since the file doesn't exist, you've already successfully disabled the `postgresql@12-main` service. The `disable` command removed the symlink if it existed.

**Now just reload systemd and check:**

bash

```
sudo systemctl daemon-reload
systemctl list-units --all | grep postgresql
```

You should now see only:

- `postgresql.service` (exited) - normal
- `postgresql@14-main.service` (running) - your PostgreSQL 14
- `system-postgresql.slice` (active) - normal

The `postgresql@12-main.service` should no longer appear in the list (or appear as "not-found" if you use `--all`).

**Note:** The `postgresql.service` symlink in `/etc/systemd/system/multi-user.target.wants/` is correct - it's the main PostgreSQL service that manages all clusters.

**Everything is now clean:** You have a working PostgreSQL 14 installation with no leftover PostgreSQL 12 services.



The "sudo systemctl daemon-reload" exited with exit code 0, however "postgresql@12-main.service" still appears on the output of: `$ sudo systemctl list-units --all | grep -i postgres`

That's because the unit is still loaded in systemd's memory. Let's completely remove it:

bash

```
# First, mask the service to prevent it from being started
sudo systemctl mask postgresql@12-main

# Then reset the failed state
sudo systemctl reset-failed postgresql@12-main

# Finally, reload systemd and check
sudo systemctl daemon-reload
systemctl list-units --all | grep -i postgres
```

If it still appears, try:

bash

```
# Check where the unit file actually is
sudo systemctl status postgresql@12-main

# Look for any leftover unit files
sudo find /etc/systemd/system -name "*postgresql*12*" -ls
sudo find /lib/systemd/system -name "*postgresql*12*" -ls

# If you find any, remove them and reload
sudo systemctl daemon-reload
```

**But honestly**, if it shows as failed but inactive, it's harmless. The important thing is that PostgreSQL 14 is running and PostgreSQL 12 is not. The failed service is just a systemd artifact that will disappear on next reboot.



Wonderful. You first batch of three (3) instructions, namely: \$ sudo systemctl mask postgresql@12-main \$ sudo systemctl reset-failed postgresql@12-main \$ sudo systemctl daemon-reload ... successfully cleared the output of systemctl list-units that now appears as follows: \$ systemctl list-units --all | grep -i postgres postgresql.service loaded active exited PostgreSQL RDBMS postgresql@14-main.service loaded active running PostgreSQL Cluster 14-main system-postgresql.slice loaded active active system-postgresql.slice

Perfect! Your PostgreSQL installation is now clean and properly configured.

## Current Status:

1. **✓ postgresql.service (exited)** - Master service that started your cluster
2. **✓ postgresql@14-main.service (running)** - Your active PostgreSQL 14 database cluster
3. **✓ system-postgresql.slice (active)** - Resource management slice
4. **✓ No PostgreSQL 12 remnants** - Successfully cleaned up

## Verification Commands:

bash

```
# Verify PostgreSQL 14 is running
sudo systemctl status postgresql@14-main

# Check PostgreSQL version
sudo -u postgres psql -c "SELECT version();"

# Check listening on (default: 5432)
sudo ss -tlnp | grep postgres

# Check data directory
sudo ls -la /var/lib/postgresql/14/main/
```

## Key Directories:

- **Data:** /var/lib/postgresql/14/main/
- **Config:** /etc/postgresql/14/main/
- **Logs:** /var/log/postgresql/postgresql-14-main.log

## Next Steps:

You can now:

1. Restore your database from backup using pg\_restore or psql
2. Configure PostgreSQL settings in /etc/postgresql/14/main/postgresql.conf
3. Set up authentication in /etc/postgresql/14/main/pg\_hba.conf

Your PostgreSQL 14 installation on Ubuntu 20.04 is complete and ready for use!



This being said, I just noticed what also appears to be a relic of PostgreSQL 12. Can I also safely remove this file too?  
\$ sudo ls /etc/systemd/system | grep -i postgres postgresql@12-main.service

**Yes, absolutely remove it.** That file is a leftover symlink from PostgreSQL 12 that should have been removed when you purged PostgreSQL 12.

bash

```
# Remove the leftover symlink
sudo rm /etc/systemd/system/postgresql@12-main.service

# Reload systemd to clean up
sudo systemctl daemon-reload

# Verify it's gone
sudo ls /etc/systemd/system | grep -i postgres
```

This file is a **systemd symlink** that was created when PostgreSQL 12 was enabled. Since you already:

1. Purged PostgreSQL 12 packages
2. Disabled the service
3. Masked it

This symlink is just a leftover artifact. Removing it cleans up the last trace of PostgreSQL 12 from your system.

**Note:** This is in `/etc/systemd/system/` (custom/system symlinks), not `/lib/systemd/system/` (package-provided template files). The template file `postgresql@.service` in `/lib/systemd/system/` should remain.

 Cool, now I want to add the postgis (geospatial) extension to it. When I am on a psql shell and I do: `CREATE EXTENSION IF NOT EXISTS postgis;` I get the following error: `ERROR: could not open extension control file "/usr/share/postgresql/14/extension/postgis.control": No such file or directory`

You need to install the PostGIS package for PostgreSQL 14. Here's how:

```
bash
# Install PostGIS for PostgreSQL 14
sudo apt install postgresql-14-postgis-3 -y

# Also install the -scripts package for additional functions
sudo apt install postgis -y

# Now create the extension as postgres user
sudo -u postgres psql -c "CREATE EXTENSION IF NOT EXISTS postgis;"

# Verify installation
sudo -u postgres psql -c "SELECT PostGIS_Version();"
```

**Alternative:** If you want the full PostGIS suite including topology, raster, etc.:

```
bash
# Install the complete PostGIS bundle
sudo apt install postgresql-14-postgis-3-scripts -y
```

#### Explanation:

- `postgresql-14-postgis-3` - Core PostGIS extension for PostgreSQL 14
- `postgis` - Command-line tools and utilities
- `postgresql-14-postgis-3-scripts` - Additional scripts and extensions

After installation, the control file will be at `/usr/share/postgresql/14/extension/postgis.control` and you can create the extension.

**Note:** Make sure you're connected to the database where you want PostGIS enabled:

```
bash
# Connect to your specific database
sudo -u postgres psql -d your_database_name -c "CREATE EXTENSION postgis;"
```



Cool, that succeeded, now I want to create the following extension: CREATE EXTENSION IF NOT EXISTS pgaudit; but that also fails on plsql with: ERROR: could not open extension control file "/usr/share/postgresql/14/extension/pgaudit.control": No such file or directory

You need to install the pgaudit extension package for PostgreSQL 14:

bash

```
# Install pgaudit for PostgreSQL 14
sudo apt install postgresql-14-pgaudit -y

# Now create the extension
sudo -u postgres psql -c "CREATE EXTENSION IF NOT EXISTS pgaudit;

# Verify installation
sudo -u postgres psql -c "SELECT * FROM pg_available_extensions WHERE name = 'pgaudit';"
```

**Important note about pgaudit:** Unlike PostGIS, pgaudit needs additional configuration. After installing, you must:

1. **Edit PostgreSQL configuration:**

bash

```
sudo nano /etc/postgresql/14/main/postgresql.conf
```

Add or modify these lines:

ini

```
shared_preload_libraries = 'pgaudit'
pgaudit.log = 'all,vis'
pgaudit.log_catalog = 'off'
pgaudit.log_parameters = 'on'
pgaudit.log_relations = 'on'
```

2. **Restart PostgreSQL:**

bash

```
sudo systemctl restart postgresql@14-main
```

3. **Now create the extension:**

bash

```
sudo -u postgres psql -c "CREATE EXTENSION IF NOT EXISTS pgaudit;"
```

**Note:** The shared\_preload\_libraries change requires a restart, not just a reload. Also, pgaudit is session-level, so you might need to create it in each database where you want auditing.