POSTGRES 1st

PGF Postgres — Installation Guide (Linux)

1) Supported platforms & scope

- Distros: Debian/Ubuntu, RHEL/Rocky/Alma/Amazon Linux, openSUSE/SLES (runtime via the installer), Arch (runtime via the installer).
- Packages provided: **.deb** (Debian/Ubuntu) and **.rpm** (RHEL family).
  For SUSE/Arch you'll still **install PostgreSQL via the bundled installer** after installing the package (the installer handles those distros).
- PostgreSQL majors: **13, 14, 15, 16, 17**.

2) Pre-requisites

System & permissions

- `root` or `sudo` privileges.
- `systemd` available and enabled.
- Outbound HTTPS access if using PGDG repositories:
  - `https://apt.postgresql.org` (Debian/Ubuntu)
  - `https://download.postgresql.org` (RHEL/systems using yum/dnf/zypper)
- Time sync (NTP/chrony) recommended.

Package build tools (only needed if you're building the .deb/.rpm from the kits)

- **Debian/Ubuntu**: `build-essential debhelper devscripts debconf`
- **RHEL/Rocky/Alma/Amazon Linux**: `rpm-build`

Runtime dependencies (auto-pulled by the packages)

- `bash`, `systemd`, `curl`, `ca-certificates`, `gnupg`, `lsb-release` (or `gnupg2` on RPM).

Network & security

- Default DB port: **5432** (change with `--port`).
- Firewalls:
  - Ubuntu: `sudo ufw allow 5432/tcp` (if you open access).
  - RHEL: `sudo firewall-cmd --add-port=5432/tcp --permanent && sudo firewall-cmd --reload`.
- SELinux (RHEL/SLES):

  - If you use a **non-default port** (e.g., 5433):
    ```
    sudo semanage port -a -t postgresql_port_t -p tcp 5433
    ```
    (Install policycoreutils-python-utils if semanage is missing.)

Storage & sizing (guidance)

- Minimum free disk at install: **10 GB** (for binaries + initial cluster); size up for your data directory.
- RAM: **≥ 2 GB** minimum; **≥ 4 GB** recommended for general workloads.
- Data directory defaults to distro paths; override with --data-dir.

Offline/proxy environments

- If PGDG repos are blocked:
  - Use --use-distro-repo to install from OS repos, **or**
  - Mirror/snapshot PGDG internally and ensure the target hosts trust your mirror's GPG keys.

3) What the packages install

All variants install these branded helpers:

- /usr/bin/pgf-postgres-setup — runs the **multi-distro installer** with your defaults from /etc/pgf-postgres/config.
- /usr/bin/pgf-postgres — service helper: start|stop|restart|status|logs.
- /usr/lib/pgf-postgres/pgf-postgres-installer.sh — the installer script that:
  - Adds PGDG repo (unless --use-distro-repo).
  - Installs postgresql-<major> + client.
  - Initializes and enables the service.
  - Applies port and optional pg_hba rules (--allow).
- /etc/pgf-postgres/config — conffile for defaults (major, port, allows).

4) Method 1 : Single meta-package (pgf-postgres) — Install

4.1 Build the package (if starting from the kit)

Debian/Ubuntu

```
sudo apt-get update
sudo apt-get install -y build-essential debhelper devscripts debconf
cd pgf-postgres        # from the meta-kit
debuild -us -uc
# Result: ../pgf-postgres_1.0.0-1_all.deb
```

RHEL/Rocky/Alma/Amazon Linux

```
sudo dnf -y install rpm-build
cd pgf-postgres        # from the meta-kit
tar czf /tmp/pgf-postgres-1.0.0.tar.gz --transform 's,^,pgf-postgres-1.0.0/,' installer scripts etc debian rpm LICENSE README.md
rpmbuild -ba rpm/pgf-postgres.spec --define "_sourcedir /tmp" --define "_version 1.0.0"
# Result: ~/rpmbuild/RPMS/noarch/pgf-postgres-1.0.0-1.noarch.rpm
```

4.2 Install the package

Debian/Ubuntu

```
sudo apt-get install -y ./pgf-postgres_1.0.0-1_all.deb
# Debconf will prompt for default major (13..17). You can change later:
sudo dpkg-reconfigure pgf-postgres
```

RHEL family

```
sudo dnf -y install ~/rpmbuild/RPMS/noarch/pgf-postgres-1.0.0-1.noarch.rpm
# Default major is 16; to change:
sudo pgf-postgres-configure 15
```

4.3 Provision PostgreSQL

Pick one of the following:

- Use defaults from /etc/pgf-postgres/config:

```
sudo pgf-postgres-setup -y
```

- Override on the fly (examples):

```
# Install PG 14, open to a VPC CIDR, and use non-default port
sudo pgf-postgres-setup --version 14 --allow 10.0.0.0/8 --port 5433 -y

# Use distro repo instead of PGDG (offline/strict envs)
sudo pgf-postgres-setup --use-distro-repo -y
```

4.4 Verify & operate

```
pgf-postgres status
sudo -u postgres psql -c "select version();"
# Optional: open firewall if you added --allow rules
# Ubuntu: sudo ufw allow 5432/tcp
# RHEL:   sudo firewall-cmd --add-port=5432/tcp --permanent && sudo firewall-cmd --reload
```

4.5 Uninstall (keeps data unless you purge via the installer)

```
# Remove the pgf-postgres package:
# Debian: sudo apt-get remove pgf-postgres
# RHEL:   sudo dnf remove -y pgf-postgres

# To remove the PostgreSQL software:
sudo pgf-postgres-setup --uninstall -y

# To purge data directories as well:
sudo pgf-postgres-setup --uninstall --purge -y
```

5) Method 2: Individual per-version packages (pgf-postgres13|14|15|16|17)

Each kit defaults the installer to that specific major (you can still override with --version if you want).

5.1 Build the package (if starting from the kit)

Debian/Ubuntu (example for PG15)

```
sudo apt-get update
sudo apt-get install -y build-essential debhelper devscripts
cd pgf-postgres15
debuild -us -uc
# Result: ../pgf-postgres15_1.0.0-1_all.deb
```

RHEL family (example for PG15)

```
sudo dnf -y install rpm-build
cd pgf-postgres15
tar czf /tmp/pgf-postgres15-1.0.0.tar.gz --transform 's,^,pgf-postgres15-1.0.0/,' installer scripts etc debian rpm LICENSE README.md
rpmbuild -ba rpm/pgf-postgres15.spec --define "_sourcedir /tmp" --define "_version 1.0.0"
# Result: ~/rpmbuild/RPMS/noarch/pgf-postgres15-1.0.0-1.noarch.rpm
```

5.2 Install the package

Debian/Ubuntu

```
sudo apt-get install -y ./pgf-postgres15_1.0.0-1_all.deb
```

RHEL family

```
sudo dnf -y install ~/rpmbuild/RPMS/noarch/pgf-postgres15-1.0.0-1.noarch.rpm
```

Packages **conflict** with one another to avoid file overlap. Install only **one** per host (or use the meta-package instead).

5.3 Provision PostgreSQL (fixed default)

```
# Will default to its own major (e.g., 15 here)
sudo pgf-postgres-setup -y

# You can still override explicitly:
sudo pgf-postgres-setup --version 15 --allow 10.0.0.0/8 -y
```

5.4 Verify & operate

```
pgf-postgres status
sudo -u postgres psql -c "select version();"
```

5.5 Uninstall / purge

```
# Remove PGF package (keeps data):
# Debian: sudo apt-get remove pgf-postgres15
# RHEL:   sudo dnf remove -y pgf-postgres15

# Remove PostgreSQL software:
sudo pgf-postgres-setup --uninstall -y

# Purge data directories as well:
sudo pgf-postgres-setup --uninstall --purge -y
```

6) Common options & examples

- **Set a custom data directory** (will migrate/init appropriately per distro):
- sudo pgf-postgres-setup --data-dir /var/lib/postgresql/data-pgf -y
- **Open access for specific networks** (adds to pg_hba.conf and sets listen_addresses='*'):
- sudo pgf-postgres-setup --allow 10.0.0.0/8 --allow 192.168.0.0/16 -y
- **Change port**:
- sudo pgf-postgres-setup --port 5433 -y
- **Use distro repos instead of PGDG**:
- sudo pgf-postgres-setup --use-distro-repo -y

7) Distro notes & troubleshooting

- **Debian/Ubuntu**
  If you see NO_PUBKEY errors for PGDG: the installer writes a key to /etc/apt/keyrings/postgresql.gpg. Ensure your APT can read that file and that /etc/apt/sources.list.d/pgdg.list exists with the correct codename (e.g., jammy-pgdg).
- **RHEL/Rocky/Alma**
  If you see AppStream conflicts, ensure the Red Hat module is disabled before PGDG:
- sudo dnf -qy module disable postgresql

  Then re-run pgf-postgres-setup.

- **Amazon Linux**
  If the PGDG repo package doesn't match, either:
  - run with --use-distro-repo, or
  - install the appropriate PGDG repo release that matches your AL version before running the setup.
- **SUSE/SLES**
  The installer can add a PGDG zypper repo best-effort. If unavailable for your exact version, use the distro repo (--use-distro-repo).
- **Service unit name**
  The helper auto-detects postgresql, postgresql-17, postgresql-16, etc. If detection fails, list units with:
- systemctl list-unit-files | grep -i postgres

8) Security & compliance reminders

- The installer changes listen_addresses to '*' **only** if you pass one or more --allow rules; otherwise it stays on localhost.
- You are responsible for firewall rules and network ACLs (the installer does not open ports automatically).
- Rotate credentials and configure TLS as per your security policy (TLS enablement is out of scope for this installer but can be layered on top).

9) Verification checklist

- PostgreSQL is running:
- pgf-postgres status
- Version matches your intended major:
- sudo -u postgres psql -c "select version();"
- Port & access are as expected:
    - grep -E '^port' /etc/postgresql/<ver>/main/postgresql.conf (Deb/Ubuntu)
    - grep -E '^port' /var/lib/pgsql/<ver>/data/postgresql.conf (RHEL)
    - grep -E 'host\s+all\s+all' .../pg_hba.conf shows your --allow CIDRs.