PostgreSQL Extensions Installation Guide

Complete Step-by-Step Guide for Installing pgvector and TimescaleDB

Prerequisites

- PostgreSQL installed on your system
- Root or sudo access
- Terminal/command line access

Part 1: Installing pgvector Extension

Step 1: Install pgvector package

```bash
# Update package list
sudo apt update

# Install pgvector for PostgreSQL 16
sudo apt install postgresql-16-pgvector
```

Step 2: Restart PostgreSQL service

```bash
sudo systemctl restart postgresql
```

Step 3: Connect to PostgreSQL

```bash
sudo -u postgres psql
```

Step 4: Create pgvector extension (inside PostgreSQL)

```sql
CREATE EXTENSION vector;
```

Step 5: Verify pgvector installation

```sql
SELECT extname, extversion FROM pg_extension WHERE extname = 'vector';
```

Expected output:

```
 extname | extversion
---------+------------
 vector  | 0.8.0
(1 row)
```

Part 2: Installing TimescaleDB Extension

Step 1: Exit PostgreSQL (if still connected)

sql

```
\q
```

Step 2: Add TimescaleDB APT repository

bash

```
# Add GPG key
curl -fsSL https://packagecloud.io/timescale/timescaledb/gpgkey | sudo gpg --dearmor -o /usr/sh

# Add repository
echo "deb [signed-by=/usr/share/keyrings/timescaledb.gpg] https://packagecloud.io/timescale/tim
```

Step 3: Update package list and install TimescaleDB

bash

```
# Update package list
sudo apt update

# Install TimescaleDB for PostgreSQL 16
sudo apt install timescaledb-2-postgresql-16
```

Step 4: Run TimescaleDB tune script

bash

```
sudo timescaledb-tune
```

During the tune process, answer:

- PostgreSQL config path: y (yes)

- Update shared_preload_libraries: y (yes)
- Tune memory/parallelism/WAL: y (yes)
- Accept memory settings: y (yes)
- Accept parallelism settings: y (yes)
- Accept WAL settings: y (yes)
- Accept miscellaneous settings: y (yes)

Step 5: Restart PostgreSQL service

```bash
sudo systemctl restart postgresql
```

Step 6: Connect to PostgreSQL

```bash
sudo -u postgres psql
```

Step 7: Create TimescaleDB extension (inside PostgreSQL)

```sql
CREATE EXTENSION IF NOT EXISTS timescaledb;
```

Step 8: Verify both extensions are installed

```sql
SELECT extname, extversion FROM pg_extension WHERE extname IN ('vector', 'timescaledb');
```

Expected output:

```
  extname    | extversion
------------+------------
 vector      | 0.8.0
 timescaledb| 2.20.1
(2 rows)
```

Important Notes

Command Context

- **Terminal commands** (starting with `sudo`, `apt`, etc.) are run in your bash shell
- **SQL commands** (starting with `CREATE`, `SELECT`, etc.) are run inside PostgreSQL after connecting with `psql`

PostgreSQL Connection Commands

```bash
# Connect as postgres user
sudo -u postgres psql

# Connect to specific database
sudo -u postgres psql -d your_database_name

# Exit PostgreSQL
\q
```

Verification Commands (run inside PostgreSQL)

```sql
-- List all installed extensions
\dx

-- Check specific extensions
SELECT extname, extversion FROM pg_extension WHERE extname IN ('vector', 'timescaledb');

-- Test pgvector functionality
SELECT vector_dims(ARRAY[1,2,3]::vector);

-- Test TimescaleDB functionality
SELECT timescaledb_version();
```

Troubleshooting

Common Issues

Issue 1: Extension not available

Error: `extension "vector" is not available` **Solution:** Install the extension package first, then restart PostgreSQL

Issue 2: Permission denied

Error: `permission denied` **Solution:** Use `sudo -u postgres psql` to connect with proper privileges

Issue 3: Wrong PostgreSQL version

Error: Package not found for your PostgreSQL version **Solution:** Check your PostgreSQL version with `psql --version` and install matching extension version

Service Management Commands

```bash
# Check PostgreSQL status
sudo systemctl status postgresql

# Start PostgreSQL
sudo systemctl start postgresql

# Stop PostgreSQL
sudo systemctl stop postgresql

# Restart PostgreSQL
sudo systemctl restart postgresql

# Enable PostgreSQL to start on boot
sudo systemctl enable postgresql
```

Alternative Installation Methods

For Different Linux Distributions

CentOS/RHEL/Fedora

```bash
# pgvector
sudo dnf install pgvector_16

# TimescaleDB
sudo dnf install timescaledb-2-postgresql-16
```

Arch Linux

```bash
# pgvector
sudo pacman -S postgresql-pgvector

# TimescaleDB
yay -S timescaledb
```

Building from Source (Advanced)

```bash
# pgvector from source
git clone --branch v0.8.0 https://github.com/pgvector/pgvector.git
cd pgvector
make
sudo make install

# TimescaleDB from source
git clone https://github.com/timescale/timescaledb.git
cd timescaledb
./bootstrap
make
sudo make install
```

Summary

After following this guide, you should have:

1. ✅ PostgreSQL running with both extensions installed
2. ✅ pgvector extension available for vector operations
3. ✅ TimescaleDB extension available for time-series data
4. ✅ Optimized PostgreSQL configuration via timescaledb-tune

Both extensions are now ready to use in your PostgreSQL databases!