



CYBERTEC PGEE

SETTING UP EXTENDED ENTERPRISE
AUDIT LOGGING



Document version: 1.1
Last change: 2025-03-18
Publisher: CYBERTEC PGEE team

TABLE OF CONTENTS

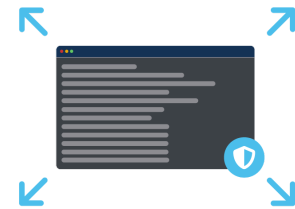
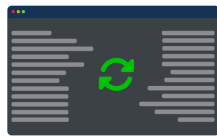
TABLE OF CONTENTS.....	2
EXTENDED POSTGRESQL AUDITING LOGGING FOR PGEE.....	3
SYSTEM ARCHITECTURE.....	4
CONFIGURATION AND SETUP GUIDE.....	5
SUPPORT AND GETTING HELP.....	9
REQUESTING HELP.....	9
VERSION HISTORY.....	11

EXTENDED POSTGRESQL AUDITING LOGGING FOR PGEE

This document describes how to utilize extended audit logging to satisfy various compliance related requirements imposed on database administrators around the world. PGEE offers a comprehensive solution and provides various important capabilities:

- **Advanced** enterprise **auditing**
- Event and change log tracking
- Integrated **compliance**
- Deep security tracking

CYBERTEC PGEE (CYBERTEC PostgreSQL Enterprise Edition) allows you to track database and audit logs in different UNIX contexts, **significantly improving security**.



PGEE Enterprise auditing is designed to **run at scale**, satisfying the needs of users and providing them with a powerful tool capable of running PostgreSQL deployments on-prem or in the cloud in the most secure way possible.

In this document, you will learn how to set up PGEE to **leverage our extended logging and auditing capabilities**.

SYSTEM ARCHITECTURE

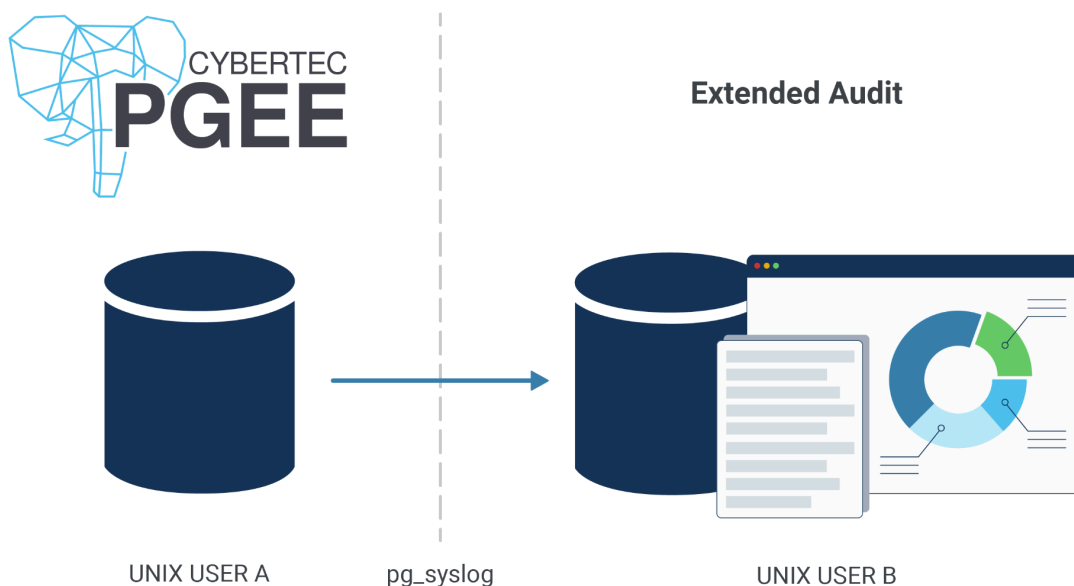
In this section, you will learn about the general architecture of CYBERTEC's advanced audit logging and understand the overall **system architecture** provided by CYBERTEC PGEE (PostgreSQL Enterprise Edition).

Extended logging is a feature of PGEE, which provides a rich set of capabilities that are important for:

- ISO compliance
- Government regulations
- Compliance related requirements

PGEE logging allows administrators to set up **custom UNIX users** to capture audit trails (for example created by PGaudit) and database logs. It provides a powerful method allowing you to operate PGEE in **highly regulated environments**.

The **key advantage** of our solution over the standard version of PostgreSQL is that audit trails can be created as different UNIX users, providing an extra layer of security by preventing the database administrator from modifying the audit trail:



This document outlines the steps needed to leverage PGEE to secure your enterprise from all kinds of threats.

CONFIGURATION AND SETUP GUIDE

The first step to be executed is the creation of a user. Remember: The key benefit of PGEE audit logging is the ability to separate the audit trail from the database deployment on the operating system level.

Here is what we can do to create the user:

```
$ sudo adduser --system syslog
Adding system user `syslog' (UID 102) ...
Adding new user `syslog' (UID 102) with group `nogroup' ...
Not creating `/nonexistent'.
```

In the next step, we can create a directory to store the audit logs we will generate. Make sure that the directory belongs to the correct UNIX user, so that the tooling is actually able to write to this destination. Note that in the case of missing permissions your database will stop functioning correctly, as it must be able to write all audit logs to disk - otherwise, operations are prohibited to avoid any risk of a data breach.

The following listing shows how this can be achieved:

```
$ sudo mkdir /var/log/pg_log
$ sudo chown syslog: /var/log/pg_log
```

When installing PGEE, the packages will ensure the existence of an important tool: **pg_syslog**. The purpose of this tool is to attach to PGEE and extract the log so that it can be written to disk in a safe UNIX user context.

The syntax of **pg_syslog** is as follows:

```
$ /usr/lib/postgresql/17/bin/pg_syslog --help
pg_syslog is a tool to read PostgreSQL sysloger output
  from a named pipe.
```

Usage:

```
pg_syslog OPTION... FIFO
```

Options:

```
-c          configuration file
-?, --help  show this help, then exit
```

NOTE: The path to **pg_syslog** is different, depending on your Linux distribution. The examples here use Debian/Ubuntu Linux. On RedHat-based systems, the path is **/usr/pgsql-17/bin/pg_syslog**.

In the next step, we can configure **pg_syslog**. To make this process as simple as possible, the syntax is the same as what you would use in `postgresql.conf`. However, only logging-related parameters are accepted:

```
$ sudo vi /etc/postgresql/pg_syslog.conf:
log_directory = '/var/log/pg_log'
log_filename = 'postgresql-%Y-%m-%d.log'
log_rotation_age = '1d'
```

Once this is done, we have to configure systems to ensure that the service exists and starts up automatically. Let us handle the configuration first:

```
$ sudo systemctl edit --force --full pg_syslog.service
# /etc/systemd/system/pg_syslog.service
[Unit]
Description=PostgreSQL log file collector
Before=postgresql.service

[Service]
Type=simple
ExecStartPre=+/usr/bin/rm -f /run/pg_log
ExecStartPre=+/usr/bin/mkfifo --mode 660 /run/pg_log
ExecStartPre=+/usr/bin/chown syslog:postgres /run/pg_log
ExecStart=+/usr/lib/postgresql/17/bin/pg_syslog -c
/etc/postgresql/pg_syslog.conf /run/pg_log
User=syslog

[Install]
WantedBy=multi-user.target
```

Finally, we can enable the service:

```
$ sudo systemctl enable --now pg_syslog
Created symlink
/etc/systemd/system/multi-user.target.wants/pg_syslog.service
→ /etc/systemd/system/pg_syslog.service.
```

There is now a FIFO owned by the new "syslog" user that postgres has write access to, and the first file has appeared in `/var/log/pg_log`:

```
$ ls -al /run/pg_log
prw-rw---- 1 syslog postgres 0 Feb 21 16:27 /run/pg_log|

$ ls -al /var/log/pg_log/
total 12
drwxr-xr-x  2 syslog nogroup 4096 Feb 21 16:16 ./
drwxr-xr-x 12 root    root    4096 Feb 21 16:03 ../
```

```
-rw----- 1 syslog nogroup 0 Feb 21 16:27 postgresql-2025-02-21.log
```

Now that the tooling has been wired up, we can check the existence of the service to ensure that all processes are running successfully:

```
$ systemctl status pg_syslog
• pg_syslog.service - PostgreSQL log file collector
  Loaded: loaded (/etc/systemd/system/pg_syslog.service;
         enabled; preset: enabled)
  Active: active (running) since
         Fri 2025-02-21 16:27:47 CET; 7min ago
  Main PID: 3882645 (pg_syslog)
  Tasks: 1 (limit: 19135)
  Memory: 196.0K
  CPU: 23ms
  CGroup: /system.slice/pg_syslog.service
          └─3882645 /usr/lib/postgresql/17/bin/pg_syslog -c
          /etc/postgresql/pg_syslog.conf /run/pg_log
```

```
Feb 21 16:27:47 jenkins systemd[1]: Starting pg_syslog.service - PostgreSQL
log file collector...
```

```
Feb 21 16:27:47 jenkins systemd[1]: Started pg_syslog.service - PostgreSQL
log file collector.
```

Finally, we can again turn our attention to PGEE and make **adjustments to postgresql.conf**.

Configure postgresql.conf:

```
# This is used when logging to stderr:
logging_collector = on
    # Enable capturing of stderr, jsonlog,
    # and csvlog into log files. Required
    # to be on for csvlogs and jsonlogs.
    # (change requires restart)

# These are only used if logging_collector is on:
log_directory = '/run'
    # directory where log files are written,
    # can be absolute or relative to PGDATA
log_filename = 'pg_log'
    # log file name pattern,
    # can include strftime() escapes
```

Note that the filename must not have any extension or % placeholders; these are configured in /etc/postgresql/pg_syslog.conf.

Finally, we can enable those settings and restart PGEE as shown in the next listing:

```
$ sudo systemctl restart postgresql
```

As we can see, the log is now written to `/var/log/pg_log/`. The original log file only contains a hint that the log is now being written elsewhere. (The directory mentioned is the FIFO location, not the final log directory.)

```
$ sudo tail /var/log/postgresql/postgresql-17-main.log
2025-02-21 16:27:54.791 CET [3882669] LOG:  redirecting log output to
logging collector process
2025-02-21 16:27:54.791 CET [3882669] HINT:  Future log output will appear
in directory "/run".
```

```
$ sudo cat /var/log/pg_log/postgresql-2025-02-21.log
2025-02-21 16:27:54.791 CET [3882669] LOG:  PostgreSQL 17.4 EE 1.4.1 (Debian
17.4ee1.4.1-1.pgee12+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian
12.2.0-1
4) 12.2.0, 64-bit starting
```

Congratulations. Your deployment is now successfully providing **extended audit logs** in a **UNIX user context of your choice**.

SUPPORT AND GETTING HELP

REQUESTING HELP

Thank you for using CYBERTEC PGEE and **thank you for being our customer.**
Your feedback is important to us and we are looking forward to hearing from you.
If you are facing any issues or technical questions, please reach out to our technical team and make use of our 24x7 support and ticketing system.

CYBERTEC Support Portal

Our consultants are eager to help you with any technical and business related issues.



Our consultants are eager to help you with any technical and business related issues.

AUSTRIA (HQ)

CYBERTEC POSTGRESQL
INTERNATIONAL (HQ)

SWITZERLAND

CYBERTEC POSTGRESQL
SWITZERLAND

URUGUAY

CYBERTEC POSTGRESQL
SOUTH AMERICA

ESTONIA

CYBERTEC POSTGRESQL
NORDIC

POLAND

CYBERTEC POSTGRESQL
POLAND

SOUTH AFRICA

CYBERTEC POSTGRESQL
SOUTH AFRICA

CYBERTEC PostgreSQL International (HQ)

Römerstraße 19
2752 Wöllersdorf
Austria
Phone: +43 (0)2622 93022-0
E-Mail: office@cybertec.at

CYBERTEC PostgreSQL Switzerland

Bahnhofstraße 10
8001 Zürich
Switzerland
Phone: +41 43 456 2684
E-Mail:
swiss@cybertec-postgresql.com

CYBERTEC PostgreSQL Nordic

Fahle Office
Tartu mnt 84a-M302
10112 Tallinn
Estonia
Phone: +372 712 3013
E-Mail:
nordic@cybertec-postgresql.com

CYBERTEC PostgreSQL Poland

Pl. Inwalidów 10
01-552 Warsaw
Poland
E-Mail:
poland@cybertec-postgresql.com

CYBERTEC PG Database Services South America S.A.

Misiones 1486, Piso 3
11000 Montevideo
Uruguay
E-Mail:
latam@cybertec-postgresql.com

CYBERTEC PostgreSQL South Africa

No. 26, Cambridge Office Park
5 Bauhinia Street, Highveld Techno
Park
0046 Centurion
South Africa
Phone: +27(0)012 881 1911
E-Mail:
africa@cybertec-postgresql.com



If you need further information

For more information, or if you have any questions about our range of products, tools and services, contact us. There's no obligation—send us an inquiry via email or give us a call.



Contact

 **CYBERTEC PostgreSQL International GmbH**
Römerstraße 19
2752 Wöllersdorf
AUSTRIA

 + 43 (0) 2622 93022-0
 sales@cybertec-postgresql.com

VERSION HISTORY

Version	Effective Date	Description	Author	Reviewed By	Approved By
1.0	2025-03-07	Initial version	Hans-Jürgen Schönig	Christoph Berg	
1.1	2025-03-18	Proofreading and Design Review	Sarah Gruber	Scarlett Riggs	Andrea Schantl-Weiß