



PERCONA

www.percona.com

Percona Distribution for PostgreSQL Documentation

Release 13.0

Percona LLC and/or its affiliates 2009-2020

Oct 16, 2020

CONTENTS

I	Installation and Upgrade	2
1	Installing Percona Distribution for PostgreSQL	3
2	Upgrading Percona Distribution for PostgreSQL from 12 to 13	6
3	Minor Upgrade of Percona Distribution for PostgreSQL	12
II	Uninstall Percona Distribution for PostgreSQL	14
4	Uninstalling Percona Distribution for PostgreSQL	15
III	Release Notes	17
5	Release Notes	18

Percona Distribution for PostgreSQL is a collection of tools to assist you in managing your PostgreSQL database system: it installs PostgreSQL and complements it by a selection of extensions that enable solving essential practical tasks efficiently:

- `pg_repack` rebuilds PostgreSQL database objects
- `pgaudit` provides detailed session or object audit logging via the standard PostgreSQL logging facility
- `pgBackRest` is a backup and restore solution for PostgreSQL
- `Patroni` is an HA (High Availability) solution for PostgreSQL.
- `pg_stat_monitor` (Tech Preview Feature¹) collects and aggregates statistics for PostgreSQL and provides histogram information.
- A collection of [additional PostgreSQL contrib extensions](#)

See also:

Percona Blog Posts

- [pgBackRest - A Great Backup Solution and a Wonderful Year of Growth](#)
- [Securing PostgreSQL as an Enterprise-Grade Environment](#)
- [Announcing pg_stat_monitor Tech Preview: Get Better Insights Into Query Performance in PostgreSQL](#)

Percona Distribution for PostgreSQL is also shipped with the `libpq` library. It contains “a set of library functions that allow client programs to pass queries to the PostgreSQL backend server and to receive the results of these queries.”²

¹ Tech Preview Features are not yet ready for enterprise use and are not included in support via *SLA (Service License Agreement)*. They are included in this release so that users can provide feedback prior to the full release of the feature in a future *GA (General Availability)* release (or removal of the feature if it is deemed not useful). This functionality can change (APIs, CLIs, etc.) from tech preview to GA.

² <https://www.postgresql.org/docs/13/libpq.html>

Part I

Installation and Upgrade

INSTALLING PERCONA DISTRIBUTION FOR POSTGRESQL

Using repositories provided by Percona is the recommended way of installing Percona Distribution for PostgreSQL.

There are two repositories available for Percona Distribution for PostgreSQL. We recommend to install Percona Distribution for PostgreSQL from the *Major Release repository* (e.g. `ppg-13`) as it includes the latest version packages. Whenever a package is updated, the package manager of your operating system detects that and prompts you to update. As long as you update all Distribution packages at the same time, you can ensure that the packages you're using have been tested and verified by Percona.

The *Minor Release repository* includes a particular minor release of the database and all of the packages that were tested and verified to work with that minor release (e.g. `ppg-13.0`). You may choose to install Percona Distribution for PostgreSQL from the Minor Release repository if you have decided to standardize on a particular release which has passed rigorous testing procedures and which has been verified to work with your applications. This allows you to deploy to a new host and ensure that you'll be using the same version of all the Distribution packages, even if newer releases exist in other repositories.

The disadvantage of using a Minor Release repository is that you are locked in this particular release. When potentially critical fixes are released in a later minor version of the database, you will not be prompted for an upgrade by the package manager of your operating system. You would need to change the configured repository in order to install the upgrade.

Like many other Percona products, Percona Distribution for PostgreSQL is installed from Percona repositories by using the `percona-release` utility.

Important: Before you attempt to install Percona Distribution for PostgreSQL, update `percona-release` to its latest version. See the [documentation of percona-release](#) for details.

As soon as `percona-release` is up-to-date, *set up* the Percona Distribution for PostgreSQL product (`ppg-13`).

```
$ sudo percona-release setup ppg-13
```

Hint: The command to set up a minor version product is the following:

```
$ sudo percona-release setup ppg-13.0
```

Install Percona Distribution for PostgreSQL using the commands of your package manager (the procedure differs depending on the package manager of your operating system).

Using the DEB Format

Important: On Debian and other systems that use the apt package manager, such as Ubuntu, components of Percona Distribution for PostgreSQL 13 can only be installed together with the server shipped by Percona (percona-postgresql-13). If you wish to use Percona Distribution for PostgreSQL, uninstall the PostgreSQL package provided by your distribution (postgresql-13) and then install the chosen components from Percona Distribution for PostgreSQL.

The following platforms are supported:

- Debian 9.0 (stretch)
- Debian 10.0 (buster)
- Ubuntu 18.04 LTS (bionic)
- Ubuntu 20.04 (Focal Fossa)

Platform Specific Notes

On Debian 9 (stretch), you need to [enable the llvm repository](#)

Install the **percona-postgresql-13** package using **apt-get install**.

```
$ sudo apt-get install percona-postgresql-13
```

Note that this package will not install the components. To install these components use the appropriate packages:

```
$ # To install pg_repack
$ sudo apt-get install percona-postgresql-13-repack
$ # To Install pgaudit
$ sudo apt-get install percona-postgresql-13-pgaudit
$ # To install pgBackRest
$ sudo apt-get install percona-pgbackrest
$ # To install Patroni
$ sudo apt-get install percona-patroni
$ # To install pg_stat_monitor
$ sudo apt-get install percona-pg-stat-monitor13
$ # To install PostgreSQL contrib extensions
$ sudo apt-get install percona-postgresql-contrib
```

Using the RPM Format

The following platforms are supported:

- CentOS 7 and Red Hat Enterprise Linux 7
- Red Hat Enterprise Linux 8

Platform Specific Notes

If you intend to install Percona Distribution for PostgreSQL on Red Hat Enterprise Linux v8, disable the *postgresql* module:

```
$ sudo dnf module disable postgresql
```

On CentOS 7, you should install the *epel-release* package:

```
$ sudo yum -y install epel-release
$ sudo yum repolist
```

Install the **percona-postgresql-13** package using **yum install**.

```
$ sudo yum install percona-postgresql13-server
```

Note that this package will not install the components. To install these components use the appropriate packages:

```
$ # To install pg_repack
$ sudo yum install percona-pg_repack13
$ # To Install pgaudit
$ sudo yum install percona-pgaudit
$ # To install pgBackRest
$ sudo yum install percona-pgbackrest
$ # To install Patroni
$ sudo yum install percona-patroni
$ # To install pg_stat_monitor
$ yum install percona-pg-stat-monitor13
$ # To install PostgreSQL contrib extensions
$ sudo yum install percona-postgresql13-contrib
```

Starting the service

Percona Distribution for PostgreSQL is not automatically started after the installation. To start Percona Distribution for PostgreSQL, initialize the cluster using the following command:

```
/usr/pgsql-13/bin/postgresql-13-setup initdb
```

Start the PostgreSQL service:

```
$ sudo systemctl start postgresql-13
```

UPGRADING PERCONA DISTRIBUTION FOR POSTGRESQL FROM 12 TO 13

This document describes the in-place upgrade of Percona Distribution for PostgreSQL using the `pg_upgrade` tool. The in-place upgrade means installing a new version without removing the old version and keeping the data files on the server.

See also:

pg_upgrade Documentation: <https://www.postgresql.org/docs/13/pgupgrade.html>

Similar to installing, we recommend you to upgrade Percona Distribution for PostgreSQL from Percona repositories.

Important: A major upgrade is a risky process because of many changes between versions and issues that might occur during or after the upgrade. Therefore, make sure to back up your data first. The backup tools are out of scope of this document. Use the backup tool of your choice.

The general in-place upgrade flow for Percona Distribution for PostgreSQL is the following:

1. Install Percona Distribution for PostgreSQL 13 packages.
2. Stop the PostgreSQL service.
3. Check the upgrade without modifying the data.
4. Upgrade Percona Distribution for PostgreSQL.
5. Start PostgreSQL service.
6. Execute the `analyze_new_cluster.sh` script to generate statistics so the system is usable.
7. Delete old packages and configuration files.

The exact steps may differ depending on the package manager of your operating system.

- *Upgrade using **DEB** format*
- *Upgrade using **RPM** format*

Upgrade using **DEB** format

Important: Run **all** commands as root or via `sudo`.

1. Install Percona Distribution for PostgreSQL 13 packages.

- Enable Percona repository using the `percona-release` utility:

```
$ sudo percona-release setup ppg-13
```

- Install Percona Distribution for PostgreSQL 13 package:

```
$ sudo apt-get install percona-postgresql-13
```

- Install the components:

```
$ sudo apt-get install percona-postgresql-13-repack
$ sudo apt-get install percona-postgresql-13-pgaudit
$ sudo apt-get install percona-pgbackrest
$ sudo apt-get install percona-patroni
$ sudo apt-get install percona-pg-stat-monitor13
$ sudo apt-get install percona-postgresql-contrib
```

See also:**Percona Documentation:**

- [Percona Software Repositories Documentation](#)
- [Installing Percona Distribution for PostgreSQL](#)

2. Stop the postgresql service.

```
$ sudo systemctl stop postgresql.service
```

This stops both Percona Distribution for PostgreSQL 12 and 13.

3. Run the database upgrade.

- Log in as the postgres user.

```
$ sudo su postgres
```

- Change the current directory to the `tmp` directory where logs and some scripts will be recorded: `cd tmp/`.
- Check the ability to upgrade Percona Distribution for PostgreSQL from 12 to 13:

```
$ /usr/lib/postgresql/13/bin/pg_upgrade
--old-datadir=/var/lib/postgresql/12/main \
--new-datadir=/var/lib/postgresql/13/main \
--old-bindir=/usr/lib/postgresql/12/bin \
--new-bindir=/usr/lib/postgresql/13/bin \
--old-options '-c config_file=/etc/postgresql/12/main/postgresql.conf' \
--new-options '-c config_file=/etc/postgresql/13/main/postgresql.conf' \
--check
```

The `--check` flag here instructs `pg_upgrade` to only check the upgrade without changing any data.

Sample output

```
Performing Consistency Checks
-----
Checking cluster versions                                ok
```

```

Checking database user is the install user          ok
Checking database connection settings              ok
Checking for prepared transactions                 ok
Checking for reg* data types in user tables        ok
Checking for contrib/isn with bigint-passing mismatch ok
Checking for tables WITH OIDS                     ok
Checking for invalid "sql_identifier" user columns ok
Checking for presence of required libraries        ok
Checking database user is the install user          ok
Checking for prepared transactions                 ok

*Clusters are compatible*

```

- Upgrade the Percona Distribution for PostgreSQL

```

$ /usr/lib/postgresql/13/bin/pg_upgrade
--old-datadir=/var/lib/postgresql/12/main \
--new-datadir=/var/lib/postgresql/13/main \
--old-bindir=/usr/lib/postgresql/12/bin \
--new-bindir=/usr/lib/postgresql/13/bin \
--old-options '-c config_file=/etc/postgresql/12/main/postgresql.conf' \
--new-options '-c config_file=/etc/postgresql/13/main/postgresql.conf' \
--link

```

The `--link` flag creates hard links to the files on the old version cluster so you don't need to copy data. If you don't wish to use the `--link` option, make sure that you have enough disk space to store 2 copies of files for both old version and new version clusters.

- Go back to the regular user: **exit**
- The Percona Distribution for PostgreSQL 12 uses the 5432 port while the Percona Distribution for PostgreSQL 13 is set up to use the 5433 port by default. To start the Percona Distribution for PostgreSQL 13, swap ports in the configuration files of both versions.

```

$ sudo vim /etc/postgresql/13/main/postgresql.conf
$ port = 5433 # Change to 5432 here
$ sudo vim /etc/postgresql/12/main/postgresql.conf
$ port = 5432 # Change to 5433 here

```

4. Start the `postgresql` service.

```
$ sudo systemctl start postgresql.service
```

5. Check the `postgresql` version.

```

$ #Log in as a postgres user
$ sudo su postgres
$ #Check the database version
$ psql -c "SELECT version();"

```

6. Run the `analyze_new_cluster.sh` script

```

$ tmp/analyze_new_cluster.sh
$ #Logout
$ exit

```

7. Delete Percona Distribution for PostgreSQL 12 packages and configuration files

```
$ #Remove packages
$ sudo apt-get remove percona-postgresql-12* percona-pgbackrest percona-patroni_
↪percona-pg-stat-monitor12
$ #Remove old files
$ rm -rf /etc/postgresql/12/main
```

Upgrade using RPM format

Important: Run all commands as root or via **sudo**.

1. Install Percona Distribution for PostgreSQL 13 packages

- Enable Percona repository using the **percona-release** utility:

```
$ sudo percona-release setup ppg-13
```

- Install Percona Distribution for PostgreSQL 13:

```
$ sudo yum install percona-postgresql13-server
$ #Install components
$ sudo yum install percona-pgaudit
$ sudo yum install percona-pgbackrest
$ sudo yum install percona-pg_repack13
$ sudo yum install percona-patroni
$ sudo yum install percona-pg-stat-monitor13
$ sudo yum install percona-postgresql13-contrib
```

See also:

Percona Documentation:

- [Percona Software Repositories Documentation](#)
- [Installing Percona Distribution for PostgreSQL](#)

2. Set up Percona Distribution for PostgreSQL 13 cluster

```
$ #Log in as the postgres user
$ sudo su postgres
$ #Set up locale settings
$ export LC_ALL="en_US.UTF-8"
$ export LC_CTYPE="en_US.UTF-8"
$ #Initialize cluster with the new data directory
$ /usr/pgsql-13/bin/initdb -D /var/lib/pgsql/13/data
```

3. Stop the postgresql 12 service

```
$ systemctl stop postgresql-12
```

4. Run the database upgrade.

- Log in as the postgres user

```
$ sudo su postgres
```

- Check the ability to upgrade Percona Distribution for PostgreSQL from 12 to 13:

```
$ /usr/pgsql-13/bin/pg_upgrade \
--old-bindir /usr/pgsql-12/bin \
--new-bindir /usr/pgsql-13/bin \
--old-datadir /var/lib/pgsql/12/data \
--new-datadir /var/lib/pgsql/13/data \
--link --check
```

The `--check` flag here instructs `pg_upgrade` to only check the upgrade without changing any data.

Sample output

```
Performing Consistency Checks
-----
Checking cluster versions                                ok
Checking database user is the install user              ok
Checking database connection settings                  ok
Checking for prepared transactions                    ok
Checking for reg* data types in user tables            ok
Checking for contrib/isn with bigint-passing mismatch ok
Checking for tables WITH OIDS                         ok
Checking for invalid "sql_identifier" user columns    ok
Checking for presence of required libraries           ok
Checking database user is the install user            ok
Checking for prepared transactions                    ok

*Clusters are compatible*
```

- Upgrade the Percona Distribution for PostgreSQL

```
$ /usr/pgsql-13/bin/pg_upgrade \
--old-bindir /usr/pgsql-12/bin \
--new-bindir /usr/pgsql-13/bin \
--old-datadir /var/lib/pgsql/12/data \
--new-datadir /var/lib/pgsql/13/data \
--link
```

The `--link` flag creates hard links to the files on the old version cluster so you don't need to copy data. If you don't wish to use the `--link` option, make sure that you have enough disk space to store 2 copies of files for both old version and new version clusters.

5. Start the `postgresql 13` service.

```
$ #Start postgresql service
$ systemctl start postgresql-13
$ #Check postgresql status
$ systemctl status postgresql-13
```

6. Run the `analyze_new_cluster.sh` script

```
$ #Log in as the postgres user
$ sudo su postgres
$ #Run the script
$ ./analyze_new_cluster.sh
```

7. Delete Percona Distribution for PostgreSQL 12 configuration files

```
$ ./delete_old_cluster.sh
```

8. Delete Percona Distribution for PostgreSQL 12 packages

```
$ #Remove packages  
$ sudo yum -y remove percona-postgresql12*  
$ #Remove old files  
$ rm -rf /var/lib/pgsql/12/data
```

MINOR UPGRADE OF PERCONA DISTRIBUTION FOR POSTGRESQL

Minor releases of PostgreSQL include bug fixes and feature enhancements. We recommend that you keep your Percona Distribution for PostgreSQL updated to the latest minor version.

Though minor upgrades do not change the behavior, we recommend you to back up your data first, in order to be on the safe side.

Minor upgrade of Percona Distribution for PostgreSQL includes the following steps:

1. Stopping the `postgresql` cluster;
2. Installing new version packages;
3. Restarting the `postgresql` cluster.

Note: These steps apply if you installed Percona Distribution for PostgreSQL from the Major Release repository. In this case, you are always upgraded to the latest available release.

If you installed Percona Distribution for PostgreSQL from the Minor Release repository, you will need to enable a new version repository to upgrade.

For more information about Percona repositories, refer to *Installing Percona Distribution for PostgreSQL*.

Before the upgrade, update the `percona-release` utility to the latest version. This is required to install the new version packages of Percona Distribution for PostgreSQL. Refer to [Percona Software Repositories Documentation](#) for update instructions.

Important: Run all commands as root or via `sudo`.

1. Stop the `postgresql` service.

- On Debian / Ubuntu:

```
$ sudo systemctl stop postgresql.service
```

- On Red Hat Enterprise Linux / CentOS:

```
$ sudo systemctl stop postgresql-13
```

2. Install new version packages. See *Installing Percona Distribution for PostgreSQL*.
3. Restart the `postgresql` service.

- On Debian / Ubuntu:

```
$ sudo systemctl start postgresql.service
```

- On Red Hat Enterprise Linux / CentOS:

```
$ sudo systemctl start postgresql-13
```

If you wish to upgrade Percona Distribution for PostgreSQL to the major version, refer to *Upgrading Percona Distribution for PostgreSQL from 12 to 13*.

Part II

Uninstall Percona Distribution for PostgreSQL

UNINSTALLING PERCONA DISTRIBUTION FOR POSTGRESQL

To uninstall Percona Distribution for PostgreSQL, remove all the installed packages and data / configuration files.

Note: Should you need the data files later, back up your data before uninstalling Percona Distribution for PostgreSQL.

Using the DEB format

To uninstall Percona Distribution for PostgreSQL on platforms that use **apt** package manager such as Debian or Ubuntu, complete the following steps.

Run all commands as root or via **sudo**.

1. Stop the Percona Distribution for PostgreSQL service.

```
$ sudo systemctl stop postgresql.service
```

2. Remove the **percona-postgresql** packages.

```
$ sudo apt-get remove percona-postgresql-13* percona-pgbackrest percona-patroni_
↳percona-pg-stat-monitor13
```

3. Remove configuration and data files.

```
$ rm -rf /etc/postgresql/13/main
```

Using the RPM format

To uninstall Percona Distribution for PostgreSQL on platforms that use **yum** package manager such as Red Hat Enterprise Linux or CentOS, complete the following steps.

Run all commands as root or via **sudo**.

1. Stop the Percona Distribution for PostgreSQL service.

```
$ sudo systemctl stop postgresql-13
```

2. Remove the **percona-postgresql** packages

```
$ sudo yum remove percona-postgresql13*
```

3. Remove configuration and data files

```
$ rm -rf /var/lib/pgsql/13/data
```

Part III

Release Notes

RELEASE NOTES

Percona Distribution for PostgreSQL 13.0

Date October 16, 2020

Installation *Installing Percona Distribution for PostgreSQL*

Percona Distribution for PostgreSQL is a collection of tools to assist you in managing PostgreSQL. Percona Distribution for PostgreSQL installs PostgreSQL and complements it by a selection of extensions that enable solving essential practical tasks efficiently.

This release of Percona Distribution for PostgreSQL is based on the latest major version of [PostgreSQL 13.0](#). It also includes [pg_stat_monitor](#) (Tech Preview Feature¹) - a new statistics collection extension for PostgreSQL.

Extension	Version	Description
pg_repack	1.4.6	rebuilds PostgreSQL database objects
Pgaudit	1.4.1	provides detailed session or object audit logging via the standard logging facility provided by PostgreSQL
pgBackRest	2.30	a backup and restore solution for PostgreSQL
Patroni	2.0.1	a HA solution for PostgreSQL
pg_stat_monitor (Tech Preview Feature)	0.6.0	collects and aggregates statistics for PostgreSQL and provides histogram information.
PostgreSQL contrib extensions	13.0	a collection of additional extensions for PostgreSQL

Percona Distribution for PostgreSQL is also shipped with the [libpq](#) library. It contains “a set of library functions that allow client programs to pass queries to the PostgreSQL backend server and to receive the results of these queries.”²

This release of Percona Distribution for PostgreSQL is based on PostgreSQL 13.0.

¹ Tech Preview Features are not yet ready for enterprise use and are not included in support via *SLA (Service License Agreement)*. They are included in this release so that users can provide feedback prior to the full release of the feature in a future *GA (General Availability)* release (or removal of the feature if it is deemed not useful). This functionality can change (APIs, CLIs, etc.) from tech preview to GA.

² <https://www.postgresql.org/docs/13/libpq.html>