PERCONA

Distribution for PostgreSQL Documentation

11.22 (January 22, 2024)

Table of contents

1.   Percona Distribution for PostgreSQL 11 Documentation

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:

- HAProxy - a high-availability and load-balancing solution
- Patroni is an HA (High Availability) solution for PostgreSQL.
- pgAudit provides detailed session or object audit logging via the standard PostgreSQL logging facility
- pgAudit `set_user` - The `set_user` part of `pgAudit` extension provides an additional layer of logging and control when unprivileged users must escalate themselves to superuser or object owner roles in order to perform needed maintenance tasks.
- pgBackRest is a backup and restore solution for PostgreSQL
- pgBadger - a fast PostgreSQL Log Analyzer.
- PgBouncer - a lightweight connection pooler for PostgreSQL
- pg_gather - an SQL script to assess the health of PostgreSQL cluster by gathering performance and configuration data from PostgreSQL databases.
- pgpool2 - a middleware between PostgreSQL server and client for high availability, connection pooling and load balancing.
- pg_repack rebuilds PostgreSQL database objects
- pg_stat_monitor collects and aggregates statistics for PostgreSQL and provides histogram information.
- PostGIS allows storing and manipulating spacial data in PostgreSQL.
- wal2json - a PostgreSQL logical decoding JSON output plugin.
- A collection of additional PostgreSQL contrib extensions

<div>

Get started   **What's new**

</div>

> ✏️ **See also**
>
> Blog Posts
>
> - pgBackRest - A Great Backup Solution and a Wonderful Year of Growth
> - Securing PostgreSQL as an Enterprise-Grade Environment

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."

Contact Us

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

August 31, 2023

May 28, 2021

2. Release notes

2.1 Percona Distribution for PostgreSQL release notes

- Percona Distribution for PostgreSQL 11.22 Update (2024-01-22)
- Percona Distribution for PostgreSQL 11.22 (2023-12-13)
- Percona Distribution for PostgreSQL 11.21 (2023-08-31)
- Percona Distribution for PostgreSQL 11.20 (2023-06-30)
- Percona Distribution for PostgreSQL 11.19 Update (2023-05-22)
- Percona Distribution for PostgreSQL 11.19 (2023-03-31)
- Percona Distribution for PostgreSQL 11.18 (2022-12-08)
- Percona Distribution for PostgreSQL 11.17 (2022-09-08)
- Percona Distribution for PostgreSQL 11.16 (2022-06-07)
- Percona Distribution for PostgreSQL 11.15 Second Update (2022-05-05)
- Percona Distribution for PostgreSQL 11.15 Update (2022-04-14)
- Percona Distribution for PostgreSQL 11.15 (2022-04-08)
- Percona Distribution for PostgreSQL 11.14 (2021-12-20)
- Percona Distribution for PostgreSQL 11.13 Update (2021-12-07)
- Percona Distribution for PostgreSQL 11.13 (2021-09-09)
- Percona Distribution for PostgreSQL 11.12 Third Update (2021-07-15)
- Percona Distribution for PostgreSQL 11.12 Second Update (2021-07-01)
- Percona Distribution for PostgreSQL 11.12 Update (2021-06-10)
- Percona Distribution for PostgreSQL 11.12 (2021-05-24)
- Percona Distribution for PostgreSQL 11.11 Third Update (2021-06-10)
- Percona Distribution for PostgreSQL 11.11 Second Update (2021-05-10)
- Percona Distribution for PostgreSQL 11.11 Update (2021-04-12)
- Percona Distribution for PostgreSQL 11.11 (2021-03-08)
- Percona Distribution for PostgreSQL 11.10 Update (2021-06-10)
- Percona Distribution for PostgreSQL 11.10 (2020-12-15)
- Percona Distribution for PostgreSQL 11.9 (2020-09-08)
- Percona Distribution for PostgreSQL 11.8 (2020-06-11)
- Percona Distribution for PostgreSQL 11.7 (2020-04-09)
- Percona Distribution for PostgreSQL 11.6 (2020-01-23)
- Percona Distribution for PostgreSQL 11 (2019-09-17)
- Percona Distribution for PostgreSQL 11 (Beta) (2019-05-15)

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

January 22, 2024

May 28, 2021

2.2 Percona Distribution for PostgreSQL 11.22 Update (2024-01-22)

Installation

Percona Distribution for PostgreSQL is a solution with the collection of tools from PostgreSQL community that are tested to work together and serve to assist you in deploying and managing PostgreSQL. The aim of Percona Distribution for PostgreSQL is to address the operational issues like High-Availability, Disaster Recovery, Security, Spatial data handling, Observability, Performance and Scalability and others that enterprises are facing.

This update of Percona Distribution for PostgreSQL includes the new version of `pg_stat_monitor` 2.0.4 that fixes the issue with the extension causing the deadlock in the Percona Operator for PostgreSQL when executing the `pgsm_store` function.

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

January 22, 2024

January 22, 2024

2.3   Percona Distribution for PostgreSQL 11.22 (2023-12-13)

Release date:	**December 13, 2023**
Installation:	Installing Percona Distribution for PostgreSQL

Percona Distribution for PostgreSQL is a solution with the collection of tools from PostgreSQL community that are tested to work together and serve to assist you in deploying and managing PostgreSQL. The aim of Percona Distribution for PostgreSQL is to address the operational issues like High-Availability, Disaster Recovery, Security, Spatial data handling, Observability, Performance and Scalability and others that enterprises are facing.

This release is based on PostgreSQL 11.22.

2.3.1   Release Highlights

• This is the last minor release of PostgreSQL 11. Learn more about PostgreSQL 11 end of life implications.

• Docker images are now available for x86_64 architectures. Their inclusion in the distribution aims to simplify the developers' experience with the Distribution. Refer to the Docker guide for how to run Percona Distribution for PostgreSQL in Docker.

• Telemetry is now enabled in Percona Distribution for PostgreSQL to fill in the gaps in our understanding of how you use it and help us improve our products. Participation in the anonymous program is optional. You can opt-out if you prefer not to share this information. Find more information in the on Percona Distribution for PostgreSQL document.

• The `percona-postgis33` and `percona-pgaudit` packages on YUM-based operating systems are renamed `percona-postgis33_11` and `percona-pgaudit11` respectively.

The following is the list of extensions available in Percona Distribution for PostgreSQL.

Extension	Version	Description
HAProxy	2.8.3	a high-availability and load-balancing solution
Patroni	3.1.0	a HA (High Availability) solution for PostgreSQL
PgAudit	1.3.4	provides detailed session or object audit logging via the standard logging facility provided by PostgreSQL
pgAudit set_user	4.0.1	provides an additional layer of logging and control when unprivileged users must escalate themselves to superusers or object owner roles in order to perform needed maintenance tasks.
pgBackRest	2.48	a backup and restore solution for PostgreSQL
pgBadger	12.2	a fast PostgreSQL Log Analyzer.
PgBouncer	1.21.0	a lightweight connection pooler for PostgreSQL
pg_gather	v23	an SQL script for running the diagnostics of the health of PostgreSQL cluster
pgpool2	4.4.4	a middleware between PostgreSQL server and client for high availability, connection pooling and load balancing.
pg_repack	1.4.8	rebuilds PostgreSQL database objects
pg_stat_monitor	2.0.3	collects and aggregates statistics for PostgreSQL and provides histogram information.
PostGIS	3.3.4	a spatial extension for PostgreSQL.
PostgreSQL Common	256	PostgreSQL database-cluster manager. It provides a structure under which multiple versions of PostgreSQL may be installed and/or multiple clusters maintained at one time.
wal2json	2.5	a PostgreSQL logical decoding JSON output plugin

Percona Distribution for PostgreSQL also includes the following packages:

- `llvm` 12.0.1 packages for Red Hat Enterprise Linux 8 and derivatives. This fixes compatibility issues with LLVM from upstream.

- supplemental ETCD packages which can be used for setting up Patroni clusters. These packages are available for the following operating systems:

Operating System	Package	Version	Description
RHEL 7	`python3-python-etcd`	0.4.5	A Python client for ETCD
RHEL 8	`etcd`	3.3.11	A consistent, distributed key-value store
	`python3-python-etcd`	0.4.5	A Python client for ETCD

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."

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

December 13, 2023

December 20, 2021

2.4   Percona Distribution for PostgreSQL 11.21 (2023-08-31)

Release date:	**August 31, 2023**
Installation:	Installing Percona Distribution for PostgreSQL

Percona Distribution for PostgreSQL is a solution with the collection of tools from PostgreSQL community that are tested to work together and serve to assist you in deploying and managing PostgreSQL. The aim of Percona Distribution for PostgreSQL is to address the operational issues like High-Availability, Disaster Recovery, Security, Spatial data handling, Observability, Performance and Scalability and others that enterprises are facing.

This release is based on PostgreSQL 11.21.

2.4.1   Release Highlights

• Percona Distribution for PostgreSQL components now include pg_gather - the open source extension to assess the health of PostgreSQL cluster by gathering performance and configuration data from PostgreSQL databases. This tool helps you run diagnostics of your PostgreSQL cluster and is also actively used by Percona Support.

• Percona Distribution for PostgreSQL is now available on Debian 12 (bookworm).

• The support of Ubuntu 18.04 is deprecated.

The following is the list of extensions available in Percona Distribution for PostgreSQL.

Extension	Version	Description
HAProxy	2.8.1	a high-availability and load-balancing solution
Patroni	3.1.0	a HA (High Availability) solution for PostgreSQL
PgAudit	1.3.4	provides detailed session or object audit logging via the standard logging facility provided by PostgreSQL
pgAudit set_user	4.0.1	provides an additional layer of logging and control when unprivileged users must escalate themselves to superusers or object owner roles in order to perform needed maintenance tasks.
pgBackRest	2.47	a backup and restore solution for PostgreSQL
pgBadger	12.1	a fast PostgreSQL Log Analyzer.
PgBouncer	1.20.0	a lightweight connection pooler for PostgreSQL
pg_gather	v22	an SQL script for running the diagnostics of the health of PostgreSQL cluster
pgpool2	4.4.3	a middleware between PostgreSQL server and client for high availability, connection pooling and load balancing.
pg_repack	1.4.8	rebuilds PostgreSQL database objects
pg_stat_monitor	2.0.1	collects and aggregates statistics for PostgreSQL and provides histogram information.
PostGIS	3.3.4	a spatial extension for PostgreSQL.
PostgreSQL Common	252	PostgreSQL database-cluster manager. It provides a structure under which multiple versions of PostgreSQL may be installed and/or multiple clusters maintained at one time.
wal2json	2.5	a PostgreSQL logical decoding JSON output plugin

Percona Distribution for PostgreSQL also includes the following packages:

- `llvm` 12.0.1 packages for Red Hat Enterprise Linux 8 and derivatives. This fixes compatibility issues with LLVM from upstream.
- supplemental ETCD packages which can be used for setting up Patroni clusters. These packages are available for the following operating systems:

Operating System	Package	Version	Description
RHEL 7	`python3-python-etcd`	0.4.5	A Python client for ETCD
RHEL 8	`etcd`	3.3.11	A consistent, distributed key-value store
	`python3-python-etcd`	0.4.5	A Python client for ETCD

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."

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

August 31, 2023

December 20, 2021

2.5   Percona Distribution for PostgreSQL 11.20 (2023-06-30)

Release date:	**June 30, 2023**
Installation:	Installing Percona Distribution for PostgreSQL

Percona Distribution for PostgreSQL is a solution with the collection of tools from PostgreSQL community that are tested to work together and serve to assist you in deploying and managing PostgreSQL. The aim of Percona Distribution for PostgreSQL is to address the operational issues like High-Availability, Disaster Recovery, Security, Performance and Scalability and others that enterprises are facing.

This release is based on PostgreSQL 11.20.

2.5.1   Release Highlights

- Percona Distribution for PostgreSQL components now include PostGIS - the open-source extension that allows storing and manipulating spatial data in PostgreSQL.

The following is the list of extensions available in Percona Distribution for PostgreSQL.

Extension	Version	Description
HAProxy	2.6.13	a high-availability and load-balancing solution
Patroni	3.0.2	a HA (High Availability) solution for PostgreSQL
PgAudit	1.3.4	provides detailed session or object audit logging via the standard logging facility provided by PostgreSQL
pgAudit set_user	4.0.1	provides an additional layer of logging and control when unprivileged users must escalate themselves to superusers or object owner roles in order to perform needed maintenance tasks.
pgBackRest	2.44	a backup and restore solution for PostgreSQL
pgBadger	12.1	a fast PostgreSQL Log Analyzer.
PgBouncer	1.19.1	a lightweight connection pooler for PostgreSQL
pgpool2	4.4.3	a middleware between PostgreSQL server and client for high availability, connection pooling and load balancing.
pg_repack	1.4.8	rebuilds PostgreSQL database objects
pg_stat_monitor	2.0.1	collects and aggregates statistics for PostgreSQL and provides histogram information.
PostGIS	3.3.3	a spatial extension for PostgreSQL.
PostgreSQL Common	250	PostgreSQL database-cluster manager. It provides a structure under which multiple versions of PostgreSQL may be installed and/or multiple clusters maintained at one time.
wal2json	2.5	a PostgreSQL logical decoding JSON output plugin

Percona Distribution for PostgreSQL also includes the following packages:

- `llvm` 12.0.1 packages for Red Hat Enterprise Linux 8 and derivatives. This fixes compatibility issues with LLVM from upstream.

- supplemental ETCD packages which can be used for setting up Patroni clusters. These packages are available for the following operating systems:

Operating System	Package	Version	Description
RHEL 7	`python3-python-etcd`	0.4.3	A Python client for ETCD
RHEL 8	`etcd`	3.3.11	A consistent, distributed key-value store
	`python3-python-etcd`	0.4.3	A Python client for ETCD

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."

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

June 30, 2023

December 20, 2021

2.6   Percona Distribution for PostgreSQL 11.19 Update (2023-05-22)

Release date:	**May 22, 2023**
Installation:	Installing Percona Distribution for PostgreSQL

Percona Distribution for PostgreSQL is a solution with the collection of tools from PostgreSQL community that are tested to work together and serve to assist you in deploying and managing PostgreSQL. The aim of Percona Distribution for PostgreSQL is to address the operational issues like High-Availability, Disaster Recovery, Security, Performance and Scalability and others that enterprises are facing.

This update of Percona Distribution for PostgreSQL includes the new version of `pg_stat_monitor` 2.0.1 that fixes the issues with the database failure.

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

May 22, 2023

May 17, 2023

2.7   Percona Distribution for PostgreSQL 11.19 (2023-03-31)

Release date:	**March 31, 2023**
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 is based on PostgreSQL 11.19.

2.7.1   Release Highlights

- A new major version of `pg_stat_monitor` 2.0.0 has been released and is now generally available with Percona Distribution for PostgreSQL.
- Added a new extension `pgpool` - a middleware between PostgreSQL server and client for high availability, connection pooling and load balancing.
- Percona Distribution for PostgreSQL is now available on Red Hat Enterprise Linux 9 and compatible derivatives

The following is the list of extensions available in Percona Distribution for PostgreSQL.

Extension	Version	Description
Patroni	3.0.1	a HA (High Availability) solution for PostgreSQL
PgAudit	1.3.4	provides detailed session or object audit logging via the standard logging facility provided by PostgreSQL
pgAudit set_user	4.0.1	provides an additional layer of logging and control when unprivileged users must escalate themselves to superusers or object owner roles in order to perform needed maintenance tasks.
pgBackRest	2.43	a backup and restore solution for PostgreSQL
pgBadger	12.0	a fast PostgreSQL Log Analyzer.
PgBouncer	1.18.0	a lightweight connection pooler for PostgreSQL
pg_repack	1.4.8	rebuilds PostgreSQL database objects
pg_stat_monitor	2.0.0	collects and aggregates statistics for PostgreSQL and provides histogram information.
PostgreSQL Common	247	PostgreSQL database-cluster manager. It provides a structure under which multiple versions of PostgreSQL may be installed and/or multiple clusters maintained at one time.
wal2json	2.5	a PostgreSQL logical decoding JSON output plugin
HAProxy	2.5.11	a high-availability and load-balancing solution
pgpool	4.4.2	a middleware between PostgreSQL server and client for high availability, connection pooling and load balancing.

Percona Distribution for PostgreSQL also includes the following packages:

- `llvm` 12.0.1 packages for Red Hat Enterprise Linux 8 and derivatives. This fixes compatibility issues with LLVM from upstream.

- supplemental ETCD packages which can be used for setting up Patroni clusters. These packages are available for the following operating systems:

Operating System	Package	Version	Description
RHEL 7	`python3-python-etcd`	0.4.3	A Python client for ETCD
RHEL 8	`etcd`	3.3.11	A consistent, distributed key-value store
	`python3-python-etcd`	0.4.3	A Python client for ETCD

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."

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

April 14, 2023

December 20, 2021

2.8   Percona Distribution for PostgreSQL 11.18 (2022-12-08)

Release date:	**December 8, 2022**
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 is based on PostgreSQL 11.18.

Percona Distribution for PostgreSQL now includes the meta-packages that simplify its installation. The percona-ppg-server meta-package installs PostgreSQL and the extensions, while percona-ppg-server-ha package installs high-availability components that are recommended by Percona.

The following is the list of extensions available in Percona Distribution for PostgreSQL.

Extension	Version	Description
Patroni	2.1.4	a HA (High Availability) solution for PostgreSQL
PgAudit	1.3.4	provides detailed session or object audit logging via the standard logging facility provided by PostgreSQL
pgAudit set_user	4.0.0	provides an additional layer of logging and control when unprivileged users must escalate themselves to superusers or object owner roles in order to perform needed maintenance tasks.
pgBackRest	2.41	a backup and restore solution for PostgreSQL
pgBadger	12.0	a fast PostgreSQL Log Analyzer.
PgBouncer	1.17.0	a lightweight connection pooler for PostgreSQL
pg_repack	1.4.8	rebuilds PostgreSQL database objects
pg_stat_monitor	1.1.1	collects and aggregates statistics for PostgreSQL and provides histogram information.
PostgreSQL Common	241	PostgreSQL database-cluster manager. It provides a structure under which multiple versions of PostgreSQL may be installed and/or multiple clusters maintained at one time.
wal2json	2.5	a PostgreSQL logical decoding JSON output plugin
HAProxy	2.5.9	a high-availability and load-balancing solution

Percona Distribution for PostgreSQL also includes the following packages:

- `llvm` 12.0.1 packages for Red Hat Enterprise Linux 8 and derivatives. This fixes compatibility issues with LLVM from upstream.
- supplemental ETCD packages which can be used for setting up Patroni clusters. These packages are available for the following operating systems:

Operating System	Package	Version	Description
RHEL 7	`python3-python-etcd`	0.4.3	A Python client for ETCD
RHEL 8	`etcd`	3.3.11	A consistent, distributed key-value store
	`python3-python-etcd`	0.4.3	A Python client for ETCD

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."

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

🕐 December 8, 2022

🕐 December 20, 2021

2.9   Percona Distribution for PostgreSQL 11.17 (2022-09-08)

Release date:	**September 8, 2022**
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 is based on PostgreSQL 11.17. The set of extensions supplied with Percona Distribution for PostgreSQL now includes the HAProxy - a high-availability and load-balancing solution.

Extension	Version	Description
Patroni	2.1.4	a HA (High Availability) solution for PostgreSQL
PgAudit	1.3.4	provides detailed session or object audit logging via the standard logging facility provided by PostgreSQL
pgAudit set_user	3.0.0	provides an additional layer of logging and control when unprivileged users must escalate themselves to superusers or object owner roles in order to perform needed maintenance tasks.
pgBackRest	2.40	a backup and restore solution for PostgreSQL
pgBadger	11.8	a fast PostgreSQL Log Analyzer.
PgBouncer	1.17.0	a lightweight connection pooler for PostgreSQL
pg_repack	1.4.7	rebuilds PostgreSQL database objects
pg_stat_monitor	1.1.0	collects and aggregates statistics for PostgreSQL and provides histogram information.
PostgreSQL Common	241	PostgreSQL database-cluster manager. It provides a structure under which multiple versions of PostgreSQL may be installed and/or multiple clusters maintained at one time.
wal2json	2.4	a PostgreSQL logical decoding JSON output plugin
HAProxy	2.5.6	a high-availability and load-balancing solution

Percona Distribution for PostgreSQL also includes the following packages:

- `llvm` 12.0.1 packages for Red Hat Enterprise Linux 8 and derivatives. This fixes compatibility issues with LLVM from upstream.
- supplemental ETCD packages which can be used for setting up Patroni clusters. These packages are available for the following operating systems:

Operating System	Package	Version	Description
RHEL 7	`python3-python-etcd`	0.4.3	A Python client for ETCD
RHEL 8	`etcd`	3.3.11	A consistent, distributed key-value store
	`python3-python-etcd`	0.4.3	A Python client for ETCD

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."

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

🕐 September 8, 2022

🕐 December 20, 2021

2.10 Percona Distribution for PostgreSQL 11.16 (2022-06-07)

Date: June 7, 2022

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 is based on PostgreSQL 11.16. The set of extensions supplied with Percona Distribution for PostgreSQL now includes the HAProxy - a high-availability and load-balancing solution.

Extension	Version	Description
Patroni	2.1.3	a HA (High Availability) solution for PostgreSQL
PgAudit	1.3.4	provides detailed session or object audit logging via the standard logging facility provided by PostgreSQL
pgAudit set_user	3.0.0	provides an additional layer of logging and control when unprivileged users must escalate themselves to superusers or object owner roles in order to perform needed maintenance tasks.
pgBackRest	2.38	a backup and restore solution for PostgreSQL
pgBadger	11.8	a fast PostgreSQL Log Analyzer.
PgBouncer	1.17.0	a lightweight connection pooler for PostgreSQL
pg_repack	1.4.7	rebuilds PostgreSQL database objects
pg_stat_monitor	1.0.1	collects and aggregates statistics for PostgreSQL and provides histogram information.
PostgreSQL Common	241	PostgreSQL database-cluster manager. It provides a structure under which multiple versions of PostgreSQL may be installed and/or multiple clusters maintained at one time.
wal2json	2.4	a PostgreSQL logical decoding JSON output plugin
HAProxy	2.5.6	a high-availability and load-balancing solution

Percona Distribution for PostgreSQL also includes the following packages:

- `llvm` 12.0.1 packages for Red Hat Enterprise Linux 8 and derivatives. This fixes compatibility issues with LLVM from upstream.
- supplemental ETCD packages which can be used for setting up Patroni clusters. These packages are available for the following operating systems:

Operating System	Package	Version	Description
RHEL 7	`python3-python-etcd`	0.4.3	A Python client for ETCD
RHEL 8	`etcd`	3.3.11	A consistent, distributed key-value store
	`python3-python-etcd`	0.4.3	A Python client for ETCD
Debian 9 ('stretch')	`etcd`	3.3.11	A consistent, distributed key-value store
	`python3-etcd`	0.4.3	A Python client for ETCD

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."

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

🕐 June 7, 2022

🕐 December 20, 2021

2.11 Percona Distribution for PostgreSQL 11.15 Second Update (2022-05-05)

Date: May 5, 2022

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 update of Percona Distribution for PostgreSQL includes the general availability release of pg_stat_monitor 1.0.0 - the statistics collection tool for PostgreSQL.

We welcome your feedback on your experience with `pg_stat_monitor` on our Forum and in the public JIRA project.

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

June 7, 2022

December 7, 2021

2.12   Percona Distribution for PostgreSQL 11.15 Update (2022-04-14)

Date: April 14, 2022

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 update of Percona Distribution for PostgreSQL includes pg_stat_monitor 1.0.0-rc.2 - the new version of the statistics collection tool for PostgreSQL.

We welcome your feedback on your experience with `pg_stat_monitor` on our Forum and in the public JIRA project.

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

June 7, 2022

December 7, 2021

2.13   Percona Distribution for PostgreSQL 11.15 (2022-04-08)

Date: April 8, 2022

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 is based on PostgreSQL 11.15 and includes the extended set of extensions supplied with Percona Distribution for PostgreSQL.

Extension	Version	Description
Patroni	2.1.2	a HA (High Availability) solution for PostgreSQL
PgAudit	1.3.3	provides detailed session or object audit logging via the standard logging facility provided by PostgreSQL
pgAudit set_user	3.0.0	provides an additional layer of logging and control when unprivileged users must escalate themselves to superusers or object owner roles in order to perform needed maintenance tasks.
pgBackRest	2.37	a backup and restore solution for PostgreSQL
pgBadger	11.7	a fast PostgreSQL Log Analyzer.
PgBouncer	1.16.1	a lightweight connection pooler for PostgreSQL
pg_repack	1.4.7	rebuilds PostgreSQL database objects
pg_stat_monitor	1.0.0-rc.1	collects and aggregates statistics for PostgreSQL and provides histogram information.
wal2json	2.4	a PostgreSQL logical decoding JSON output plugin

Percona Distribution for PostgreSQL also includes the following packages:

- `llvm` 12.0.1 packages for Red Hat Enterprise Linux 8 and derivatives. This fixes compatibility issues with LLVM from upstream.

- supplemental ETCD packages which can be used for setting up Patroni clusters. These packages are available for the following operating systems:

Operating System	Package	Version	Description
RHEL 7	`python3-python-etcd`	0.4.3	A Python client for ETCD
RHEL 8	`etcd`	3.3.11	A consistent, distributed key-value store
	`python3-python-etcd`	0.4.3	A Python client for ETCD
Debian 9 ('stretch')	`etcd`	3.3.11	A consistent, distributed key-value store
	`python3-etcd`	0.4.3	A Python client for ETCD

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."

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

June 7, 2022

December 20, 2021

2.14 Percona Distribution for PostgreSQL 11.14 (2021-12-20)

Date: December 20, 2021

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 is based on PostgreSQL 11.14 and includes the extended set of extensions supplied with Percona Distribution for PostgreSQL.

Extension	Version	Description
Patroni	2.1.1	a HA (High Availability) solution for PostgreSQL
PgAudit	1.3.2	provides detailed session or object audit logging via the standard logging facility provided by PostgreSQL
pgAudit set_user	3.0.0	provides an additional layer of logging and control when unprivileged users must escalate themselves to superusers or object owner roles in order to perform needed maintenance tasks.
pgBackRest	2.36	a backup and restore solution for PostgreSQL
pgBadger	11.6	a fast PostgreSQL Log Analyzer.
PgBouncer	1.16.1	a lightweight connection pooler for PostgreSQL
pg_repack	1.4.7	rebuilds PostgreSQL database objects
pg_stat_monitor	1.0.0-rc.1	collects and aggregates statistics for PostgreSQL and provides histogram information.
wal2json	2.4	a PostgreSQL logical decoding JSON output plugin

Percona Distribution for PostgreSQL also includes the following packages:

- `llvm` 12.0.1 packages for Red Hat Enterprise Linux 8 / CentOS 8. This fixes compatibility issues with LLVM from upstream.

- supplemental ETCD packages which can be used for setting up Patroni clusters. These packages are available for the following operating systems:

Operating System	Package	Version	Description
CentOS 7	`python3-python-etcd`	0.4.3	A Python client for ETCD
CentOS 8	`etcd`	3.3.11	A consistent, distributed key-value store
	`python3-python-etcd`	0.4.3	A Python client for ETCD
Debian 9 ('stretch')	`etcd`	3.3.11	A consistent, distributed key-value store
	`python3-etcd`	0.4.3	A Python client for ETCD

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."

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

June 7, 2022

December 20, 2021

2.15   Percona Distribution for PostgreSQL 11.13 Update (2021-12-07)

Date: December 7, 2021

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 update of Percona Distribution for PostgreSQL includes the new version of pg_stat_monitor 1.0.0-RC - the statistics collection tool for PostgreSQL.

We welcome your feedback on your experience with `pg_stat_monitor` in the public JIRA project.

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

June 7, 2022

December 7, 2021

2.16 Percona Distribution for PostgreSQL 11.13 (2021-09-09)

Date: September 9, 2021

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 is based on PostgreSQL 11.13 and includes the extended set of extensions supplied with Percona Distribution for PostgreSQL.

Extension	Version	Description
Patroni	2.1.0	a HA (High Availability) solution for PostgreSQL
PgAudit	1.3.2	provides detailed session or object audit logging via the standard logging facility provided by PostgreSQL
pgAudit set_user	2.0.1	provides an additional layer of logging and control when unprivileged users must escalate themselves to superusers or object owner roles in order to perform needed maintenance tasks.
pgBackRest	2.34	a backup and restore solution for PostgreSQL
pgBadger	11.5	a fast PostgreSQL Log Analyzer.
PgBouncer	1.16.0	a lightweight connection pooler for PostgreSQL
pg_repack	1.4.6	rebuilds PostgreSQL database objects
pg_stat_monitor	0.9.2 - Beta1	collects and aggregates statistics for PostgreSQL and provides histogram information.
wal2json	2.3	a PostgreSQL logical decoding JSON output plugin

Percona Distribution for PostgreSQL also includes the ETCD packages which are used for Patroni cluster setup. These packages are available for the following operating systems:

Operating System	Package	Description
CentOS 7	`python3-python-etcd`	A Python client for ETCD
CentOS 8	`etcd`	A consistent, distributed key-value store
	`python3-python-etcd`	A Python client for ETCD
Debian 9 ('stretch')	`etcd`	A consistent, distributed key-value store
	`python3-etcd`	A Python client for ETCD

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."[1]

---

1. https://www.postgresql.org/docs/11/libpq.html ↵

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

June 7, 2022

May 28, 2021

2.17   Percona Distribution for PostgreSQL 11.12 Third Update (2021-07-15)

Date: July 15, 2021

Installation: Installing Percona Distribution for PostgreSQL

This update of Percona Distribution for PostgreSQL, includes the RPM package for `python3-python-etcd` for CentOS 7. This package is a Python client for ETCD and is used by Patroni to communicate with ETCD storage. For how to set up Patroni clusters, see Patroni documentation.

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

June 7, 2022

June 10, 2021

2.18 Percona Distribution for PostgreSQL 11.12 Second Update (2021-07-01)

Date: July 1, 2021

Installation: Installing Percona Distribution for PostgreSQL

With this update of Percona Distribution for PostgreSQL, `etcd` package is added as a `DEB` package to Percona Distribution for PostgreSQL for Debian 9 ("stretch"). This package is used to set up High Availability clusters with Patroni. For how to set up Patroni clusters, see Patroni documentation.

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

June 7, 2022

June 10, 2021

2.19   Percona Distribution for PostgreSQL 11.12 Update (2021-06-10)

Date: June 10, 2021

Installation: Installing Percona Distribution for PostgreSQL

This update of Percona Distribution for PostgreSQL includes the following fixes for Red Hat Enterprise Linux 8 / CentOS 8:

- `llvm` packages are added to the repository. This fixes compatibility issues with LLVM from upstream. To use `llvm` packages supplied by us, disable the upstream `llvm-toolset` module before the installation:

```
sudo dnf module disable llvm-toolset
```

- systemd unit file includes the correct path to Patroni configuration file.
- `etcd` and `python3-python-etcd` packages are added as `RPM` packages to Percona Distribution for PostgreSQL for Red Hat Enterprise Linux / CentOS 8. These packages are used to set up High Availability clusters with Patroni. For how to set up Patroni clusters, see Patroni documentation

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

June 7, 2022

June 10, 2021

2.20   Percona Distribution for PostgreSQL 11.12 (2021-05-24)

Date: May 24, 2021

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 is based on PostgreSQL 11.12 and includes the extended set of extensions supplied with Percona Distribution for PostgreSQL.

Extension	Version	Description
pg_repack	1.4.6	rebuilds PostgreSQL database objects
PgAudit	1.3.2	provides detailed session or object audit logging via the standard logging facility provided by PostgreSQL
pgAudit set_user	2.0.0	provides an additional layer of logging and control when unprivileged users must escalate themselves to superusers or object owner roles in order to perform needed maintenance tasks.
pgBackRest	2.33	a backup and restore solution for PostgreSQL
Patroni	2.0.2	a HA (High Availability) solution for PostgreSQL
pg_stat_monitor (Tech Preview Feature [1])		0.9.1
pgBadger	11.5	a fast PostgreSQL Log Analyzer.
PgBouncer	1.15.0	a lightweight connection pooler for PostgreSQL
wal2json	2.3	a PostgreSQL logical decoding JSON output plugin
PostgreSQL contrib extensions	11.11	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]

---

1. Tech Preview Features are not yet ready for enterprise use and are not included in support via . They are included in this release, so that users can provide feedback prior to the full release of the feature in a future 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/11/libpq.html ↵

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

June 7, 2022

May 28, 2021

2.21 Percona Distribution for PostgreSQL 11.11 Third Update (2021-06-10)

Date: June 10, 2021

Installation: Installing Percona Distribution for PostgreSQL

This update of Percona Distribution for PostgreSQL includes `llvm` packages for Red Hat Enterprise Linux 8 / CentOS 8. This fixes compatibility issues with LLVM from upstream. To use `llvm` packages supplied by us, disable the upstream `llvm-toolset` module before the installation:

```
sudo dnf module disable llvm-toolset
```

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

June 7, 2022

June 10, 2021

2.22 Percona Distribution for PostgreSQL 11.11 Second Update (2021-05-10)

Date: May 10, 2021

Installation: Installing Percona Distribution for PostgreSQL

This update of Percona Distribution for PostgreSQL includes the latest version of pg_stat_monitor 0.9.0 - the statistics collection tool for PostgreSQL. pg_stat_monitor is available as the Tech Preview Feature [1] and is supplied in the set of extensions within Percona Distribution for PostgreSQL.

We welcome your feedback on your experience with pg_stat_monitor in the public JIRA project.

---

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

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

June 7, 2022

May 28, 2021

2.23 Percona Distribution for PostgreSQL 11.11 Update (2021-04-12)

Date: April 12, 2021

Installation: Installing Percona Distribution for PostgreSQL

This update of Percona Distribution for PostgreSQL includes the latest version of pg_stat_monitor 0.8.1 - the statistics collection tool for PostgreSQL. `pg_stat_monitor` is available as the Tech Preview Feature [1] and is supplied in the set of extensions within Percona Distribution for PostgreSQL.

We welcome your feedback on your experience with `pg_stat_monitor` in the public JIRA project.

----

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

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

🕐 June 7, 2022

🕐 May 28, 2021

2.24   Percona Distribution for PostgreSQL 11.11 (2021-03-08)

Date: March 8, 2021

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.

The following is the list of extensions available in Percona Distribution for PostgreSQL.

Extension	Version	Description
pg_repack	1.4.6	rebuilds PostgreSQL database objects
Pgaudit	1.3.2	provides detailed session or object audit logging via the standard logging facility provided by PostgreSQL
pgBackRest	2.32	a backup and restore solution for PostgreSQL
Patroni	2.0.1	a HA (High Availability) solution for PostgreSQL
pg_stat_monitor (Tech Preview Feature [1])	0.6.0	
PostgreSQL contrib extensions	11.11	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 is based on PostgreSQL 11.11.

---

1. Tech Preview Features are not yet ready for enterprise use and are not included in support via SLA. They are included in this release, so that users can provide feedback prior to the full release of the feature in a future GA 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/11/libpq.html ↩

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

June 7, 2022

May 28, 2021

2.25   Percona Distribution for PostgreSQL 11.10 Update (2021-06-10)

Date: June 10, 2021

Installation: Installing Percona Distribution for PostgreSQL

This update of Percona Distribution for PostgreSQL includes `llvm` packages for Red Hat Enterprise Linux 8 / CentOS 8. This fixes compatibility issues with LLVM from upstream. To use `llvm` packages supplied by us, disable the upstream `llvm-toolset` module before the installation:

```
sudo dnf module disable llvm-toolset
```

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

June 7, 2022

June 10, 2021

2.26   Percona Distribution for PostgreSQL 11.10 (2020-12-15)

Date: December 15, 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 is based on PostgreSQL 11.10 and includes pg_stat_monitor (Tech Preview Feature) - a new statistics collection extension for PostgreSQL.

The following is the list of extensions available in Percona Distribution for PostgreSQL.

Extension	Version	Description
pg_repack	1.4.6	rebuilds PostgreSQL database objects
Pgaudit	1.3.2	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 (High Availability) solution for PostgreSQL
PostgreSQL contrib extensions	11.10	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." [1]

This release of Percona Distribution for PostgreSQL is based on PostgreSQL 11.10.

---

[1] https://www.postgresql.org/docs/11/libpq.html ↩

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

🕐 June 7, 2022

🕐 May 28, 2021

2.27   Percona Distribution for PostgreSQL 11.9 (2020-09-08)

Date: September 8, 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:

Extension	Version	Description
pg_repack	1.4.5	rebuilds PostgreSQL database objects
Pgaudit	1.4.0	provides detailed session or object audit logging via the standard logging facility provided by PostgreSQL
pgBackRest	2.29	a backup and restore solution for PostgreSQL
Patroni	2.0.0	a HA (High Availability) solution for PostgreSQL
PostgreSQL contrib extensions	11.9	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." [1]

This release of Percona Distribution for PostgreSQL is based on PostgreSQL 11.9.

---

1. https://www.postgresql.org/docs/11/libpq.html ↩

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

December 1, 2022

May 28, 2021

2.28   Percona Distribution for PostgreSQL 11.8 (2020-06-11)

Date: June 11, 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:

• Pg_repack rebuilds PostgreSQL database objects.

• Pgaudit provides detailed session or object audit logging via the standard logging facility provided by PostgreSQL

• pgBackRest - a backup and restore solution for PostgreSQL

• Patroni - an HA (High Availability) solution for PostgreSQL

• A collection of additional PostgreSQL contrib extensions

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." [1]

This release of Percona Distribution for PostgreSQL is based on PostgreSQL 11.8.

---

1. https://www.postgresql.org/docs/11/libpq.html ↩

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

🕐 December 1, 2022

🕐 May 28, 2021

2.29   Percona Distribution for PostgreSQL 11.7 (2020-04-09)

Date: April 9, 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:

- Pg_repack rebuilds PostgreSQL database objects.
- Pgaudit provides detailed session or object audit logging via the standard logging facility provided by PostgreSQL
- pgBackRest - a backup and restore solution for PostgreSQL
- Patroni - an HA (High Availability) solution for PostgreSQL
- A collection of additional PostgreSQL contrib extensions

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." [1]

This release of Percona Distribution for PostgreSQL is based on PostgreSQL 11.7.

Starting from May 15, 2020, this release of Percona Distribution for PostgreSQL includes improvements to simplify the upgrade to the major version of Percona Distribution for PostgreSQL.

---

1. https://www.postgresql.org/docs/11/libpq.html ↵

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

December 1, 2022

May 28, 2021

2.30 Percona Distribution for PostgreSQL 11.6 (2020-01-23)

Percona is happy to announce the release of Percona Distribution for PostgreSQL 11.6 on January 23, 2020. 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:

- Pg_repack rebuilds PostgreSQL database objects.
- Pgaudit provides detailed session or object audit logging via the standard logging facility provided by PostgreSQL
- pgBackRest - a backup and restore solution for PostgreSQL
- Patroni - an HA (High Availability) solution for PostgreSQL
- A collection of additional PostgreSQL contrib extensions

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." [1]

This release of Percona Distribution for PostgreSQL is based on PostgreSQL 11.6.

> 🔥 **Important**
>
> On Debian and other systems that use the `apt` package manager, such as Ubuntu, components of Percona Distribution for PostgreSQL 11 can only be installed together with the server shipped by Percona (`percona-postgresql-11`). If you wish to use Percona Distribution for PostgreSQL, uninstall the PostgreSQL package provided by your distribution (postgresql-11) and then install the chosen components from Percona Distribution for PostgreSQL.

2.30.1   Bugs Fixed

- DISTPG-34: Components shipped with Percona Distribution for PostgreSQL could not be installed alongside PostgreSQL.

---

1. https://www.postgresql.org/docs/11/libpq.html ↩

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

December 1, 2022

May 28, 2021

2.31   Percona Distribution for PostgreSQL 11 (2019-09-17)

Percona is excited to announce the GA release of Percona Distribution for PostgreSQL 11 on September 17, 2019. 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:

- Pg_repack rebuilds PostgreSQL database objects.

- Pgaudit provides detailed session or object audit logging via the standard logging facility provided by PostgreSQL

- pgBackRest - a backup and restore solution for PostgreSQL

- Patroni - an HA solution for PostgreSQL

- A collection of additional PostgreSQL contrib extensions

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." [1]

This release of Percona Distribution for PostgreSQL is based on PostgreSQL 11.

---

[1] https://www.postgresql.org/docs/11/libpq.html ↵

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

🕒 December 1, 2022

🕒 May 28, 2021

2.32   Percona Distribution for PostgreSQL 11 (Beta) (2019-05-15)

Percona is excited to announce a beta release of Percona Distribution for PostgreSQL 11 (Beta) on August 15, 2019. 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:

- Pg_repack rebuilds PostgreSQL database objects.

- Pgaudit provides detailed session or object audit logging via the standard logging facility provided by PostgreSQL

- pgBackRest - a backup and restore solution for PostgreSQL

- Patroni - an HA (High Availability) solution for PostgreSQL

- A collection of additional PostgreSQL contrib extensions

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." [1]

This release of Percona Distribution for PostgreSQL is based on PostgreSQL 11.

---

1. https://www.postgresql.org/docs/11/libpq.html ↩

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

🕑 June 7, 2022

🕑 May 28, 2021

3. Installation and Upgrade

3.1 Install Percona Distribution for PostgreSQL

3.1.1 Install Percona Distribution for PostgreSQL

Percona Distribution for PostgreSQL is the solution with the collection of tools from PostgreSQL community that are tested to work together and serve to assist you in deploying and managing PostgreSQL. Read more.

You can select from multiple easy-to-follow installation options, but **we recommend using a Package Manager** for a convenient and quick way to try the software first.

Package manager   Docker   Kubernetes

Percona provides installation packages in `DEB` and `RPM` format for 64-bit Linux distributions. Find the full list of supported platforms and versions on the Percona Software and Platform Lifecycle page.

If you are on Debian or Ubuntu, use `apt` for installation.

If you are on Red Hat Enterprise Linux or compatible derivatives, use `yum`.

Choose your package manager below to get access to a detailed step-by-step guide.

(**Install via apt** →)   (**Install via yum** →)

Get our image from Docker Hub and spin up a cluster on a Docker container for quick evaluation.

Check below to get access to a detailed step-by-step guide.

(**Run in Docker**)

Percona Operator for Kubernetes is a controller introduced to simplify complex deployments that require meticulous and secure database expertise.

Check below to get access to a detailed step-by-step guide.

(**Get started with Percona Operator**)

Contact Us

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

December 13, 2023

May 28, 2021

3.1.2   Install Percona Distribution for PostgreSQL on Debian and Ubuntu

This document describes how to install Percona Server for PostgreSQL from Percona repositories on DEB-based distributions such as Debian and Ubuntu. Read more about Percona repositories ↗ .

Preconditions

1. Debian and other systems that use the apt package manager include the upstream PostgreSQL server package (`postgresql-11)` by default. The components of Percona Distribution for PostgreSQL 11 can only be installed together with the PostgreSQL server shipped by Percona (`percona-postgresql-11`). If you wish to use Percona Distribution for PostgreSQL, uninstall the upstream `postgresql-11` package and then install the chosen components from Percona Distribution for PostgreSQL.

2. Install `curl` for Telemetry. We use Telemetry to better understand the use of our products and improve them.

Procedure

Run all the commands in the following sections as root or using the `sudo` command:

CONFIGURE PERCONA REPOSITORY

1. Install the `percona-release` repository management tool to subscribe to Percona repositories:

   • Fetch `percona-release` packages from Percona web:

   ```
   $ wget https://repo.percona.com/apt/percona-release_latest.$(lsb_release -sc)_all.deb
   ```

   • Install the downloaded package with `dpkg` :

   ```
   $ sudo dpkg -i percona-release_latest.$(lsb_release -sc)_all.deb
   ```

   • Refresh the local cache:

   ```
   $ sudo apt update
   ```

2. Enable the repository

Percona provides two repositories for Percona Distribution for PostgreSQL. We recommend enabling the Major release repository to timely receive the latest updates.

```
$ sudo percona-release setup ppg11
```

INSTALL PACKAGES

Install using meta-package　　**Install packages individually**

The meta package enables you to install several components of the distribution in one go.

```
$ sudo apt install percona-ppg-server-11
```

1. Install the PostgreSQL server package:

```
$ sudo apt install percona-postgresql-11
```

2. Install the components:

   Install `pg_repack`:

```
$ sudo apt install percona-postgresql-11-repack
```

   Install `pgAudit`:

```
$ sudo apt install percona-postgresql-11-pgaudit
```

   Install `pgBackRest`:

```
$ sudo apt install percona-pgbackrest
```

   Install `Patroni`:

```
$ sudo apt install percona-patroni
```

   Install `pg_stat_monitor`

   Install `pgBouncer`:

```
$ sudo apt install percona-pgbouncer
```

   Install `pgAudit-set_user`:

```
$ sudo apt install percona-pgaudit11-set-user
```

   Install `pgBadger`:

```
$ sudo apt install percona-pgbadger
```

   Install `wal2json`:

```
$ sudo apt install percona-postgresql-11-wal2json
```

   Install PostgreSQL contrib extensions:

```
$ sudo apt install percona-postgresql-contrib
```

   Install HAProxy

```
$ sudo apt install percona-haproxy
```

   Install pgpool2

```
$ sudo apt install percona-pgpool2
```

   Install pg_gather

START THE SERVICE

The installation process automatically initializes and starts the default database. You can check the database status using the following command:

```
$ sudo systemctl status postgresql.service
```

CONNECT TO THE POSTGRESQL SERVER

By default, `postgres` user and `postgres` database are created in PostgreSQL upon its installation and initialization. This allows you to connect to the database as the `postgres` user.

```
$ sudo su postgres
```

Open the PostgreSQL interactive terminal:

```
$ psql
```

> 🔥 **Hint**
>
> You can connect to `psql` as the `postgres` user in one go:
>
> ```
> $ sudo su - postgres -c psql
> ```

To exit the `psql` terminal, use the following command:

```
$ \q
```

Contact Us

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

🕐 December 13, 2023

🕐 December 5, 2022

3.1.3   Install Percona Distribution for PostgreSQL on Red Hat Enterprise Linux and derivatives

This document describes how to install Percona Distribution for PostgreSQL from Percona repositories on RPM-based distributions such as Red Hat Enterprise Linux and compatible derivatives. Read more about Percona repositories.

Platform specific notes

To install Percona Distribution for PostgreSQL, do the following:

FOR PERCONA DISTRIBUTION FOR POSTGRESQL PACKAGES

> CentOS 7   RHEL8/Oracle Linux 8/Rocky Linux 8

Install the `epel-release` package:

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

Disable the `postgresql` and `llvm-toolset` modules:

```
$ sudo dnf module disable postgresql llvm-toolset
```

FOR `PERCONA-POSTGRESQL11-DEVEL` PACKAGE

You may need to install the `percona-postgresql11-devel` package when working with some extensions or creating programs that interface with PostgreSQL database. This package requires dependencies that are not part of the Distribution, but can be installed from the specific repositories:

> RHEL8   Rocky Linux 8   Oracle Linux 8   Rocky Linux 9   Oracle Linux 9

```
$ sudo yum --enablerepo=codeready-builder-for-rhel-8-rhui-rpms install perl-IPC-Run -y

$ sudo dnf install dnf-plugins-core
$ sudo dnf module enable llvm-toolset
$ sudo dnf config-manager --set-enabled powertools

$ sudo dnf config-manager --set-enabled ol8_codeready_builder install perl-IPC-Run -y

$ sudo dnf install dnf-plugins-core
$ sudo dnf module enable llvm-toolset
$ sudo dnf config-manager --set-enabled crb
$ sudo dnf install perl-IPC-Run -y

$ sudo dnf config-manager --set-enabled ol9_codeready_builder install perl-IPC-Run -y
```

FOR `PERCONA-PATRONI` PACKAGE

To install Patroni on Red Hat Enterprise Linux 9 and compatible derivatives, enable the `epel` repository

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

FOR `PGPOOL2` EXTENSION

To install `pgpool2` on Red Hat Enterprise Linux and compatible derivatives, enable the codeready builder repository first to resolve the dependencies conflict.

The following are commands for Red Hat Enterprise Linux 9 and derivatives. For Red Hat Enterprise Linux 8, replace the operating system version in the commands accordingly.

RHEL 9   **Rocky Linux 9**   **Oracle Linux 9**

```
$ sudo dnf config-manager --set-enabled codeready-builder-for-rhel-9-x86_64-rpms

$ sudo dnf config-manager --set-enabled crb

$ sudo dnf config-manager --set-enabled ol9_codeready_builder
```

FOR POSTGIS

The following commands provide instructions how to enable required repositories and modules on Red Hat Enterprise Linux 9 and derivatives.

For Red Hat Enterprise Linux 8 and derivatives, replace the operating system version in the commands accordingly.

RHEL 9   **Rocky Linux 9**   **Oracle Linux 9**   **RHEL UBI 9**

1. Install `epel` repository

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

2. Enable the `llvm-toolset dnf` module

```
$ sudo dnf module enable llvm-toolset
```

3. Enable the codeready builder repository to resolve dependencies conflict.

```
$ sudo dnf config-manager --set-enabled codeready-builder-for-rhel-9-x86_64-rpms
```

1. Install `epel` repository

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

2. Enable the `llvm-toolset dnf` module

```
$ sudo dnf module enable llvm-toolset
```

3. Enable the codeready builder repository to resolve dependencies conflict.

```
$ sudo dnf install dnf-plugins-core
$ sudo dnf config-manager --set-enabled crb
```

1. Install `epel` repository

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

2. Enable the `llvm-toolset dnf` module

```
$ sudo dnf module enable llvm-toolset
```

3. Enable the codeready builder repository to resolve dependencies conflict.

```
$ sudo dnf config-manager --set-enabled ol9_codeready_builder
```

1. Configure the Oracle-Linux repository. Create the `/etc/yum.repos.d/oracle-linux-ol9.repo` file to install the required dependencies:

/etc/yum.repos.d/oracle-linux-ol9.repo

```
[ol9_baseos_latest]
name=Oracle Linux 9 BaseOS Latest ($basearch)
baseurl=https://yum.oracle.com/repo/OracleLinux/OL9/baseos/latest/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1

[ol9_appstream]
name=Oracle Linux 9 Application Stream ($basearch)
baseurl=https://yum.oracle.com/repo/OracleLinux/OL9/appstream/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1
```

Procedure

Run all the commands in the following sections as root or using the `sudo` command.

INSTALL DEPENDENCIES

Install `curl` for Telemetry. We use it to better understand the use of our products and improve them.

```
$ sudo yum -y install curl
```

CONFIGURE THE REPOSITORY

1. Install the `percona-release` repository management tool to subscribe to Percona repositories:

```
$ sudo yum install https://repo.percona.com/yum/percona-release-latest.noarch.rpm
```

2. Enable the repository

Percona provides two repositories for Percona Distribution for PostgreSQL. We recommend enabling the Major release repository to timely receive the latest updates.

```
$ sudo percona-release setup ppg11
```

INSTALL PACKAGES

Install using meta-package   **Install packages individually**

The meta package enables you to install several components of the distribution in one go.

```
$ sudo yum install percona-ppg-server11
```

1. Install the PostgreSQL server package:

```
$ sudo yum install percona-postgresql11-server
```

2. Install the components:

   Install `pg_repack`:

```
$ sudo yum install percona-pg_repack11
```

   Install `pgaudit`:

```
$ sudo yum install percona-pgaudit11
```

   Install `pgBackRest`:

```
$ sudo yum install percona-pgbackrest
```

   Install `Patroni`:

```
$ sudo yum install percona-patroni
```

   Install `pg_stat_monitor`:

   Install `pgBouncer`:

```
$ sudo yum install percona-pgbouncer
```

   Install `pgAudit-set_user`:

```
$ sudo yum install percona-pgaudit11_set_user
```

   Install `pgBadger`:

```
$ sudo yum install percona-pgbadger
```

   Install `wal2json`:

```
$ sudo yum install percona-wal2json11
```

   Install PostgreSQL contrib extensions:

```
$ sudo yum install percona-postgresql11-contrib
```

   Install HAProxy

```
$ sudo yum install percona-haproxy
```

   Install `pg_gather`

```
$ sudo yum install percona-pg_gather
```

   Install pgpool2

Some extensions require additional setup in order to use them with Percona Distribution for PostgreSQL. For more information, refer to Enabling extensions.

START THE SERVICE

After the installation, the default database storage is not automatically initialized. To complete the installation and start Percona Distribution for PostgreSQL, initialize the database using the following command:

```
$ /usr/pgsql-11/bin/postgresql-11-setup initdb
```

Start the PostgreSQL service:

```
$ sudo systemctl start postgresql-11
```

CONNECT TO THE POSTGRESQL SERVER

By default, `postgres` user and `postgres` database are created in PostgreSQL upon its installation and initialization. This allows you to connect to the database as the `postgres` user.

```
$ sudo su postgres
```

Open the PostgreSQL interactive terminal:

```
$ psql
```

> 🔥 **Hint**
>
> You can connect to `psql` as the `postgres` user in one go:
>
> ```
> $ sudo su - postgres -c psql
> ```

To exit the `psql` terminal, use the following command:

```
$ \q
```

Contact Us

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

🕐 December 13, 2023

🕐 December 5, 2022

3.1.4   Enable Percona Distribution for PostgreSQL extensions

Some extensions require additional configuration before using them with Percona Distribution for PostgreSQL. This sections provides configuration instructions per extension.

Patroni

Patroni is the third-party high availability solution for PostgreSQL. The High Availability in PostgreSQL with Patroni chapter provides details about the solution overview and architecture deployment.

While setting up a high availability PostgreSQL cluster with Patroni, you will need the following components:

- Patroni installed on every `postresql` node.
- Distributed Configuration Store (DCS). Patroni supports such DCSs as ETCD, zookeeper, Kubernetes though ETCD is the most popular one. It is available upstream as DEB packages for Debian 10, 11 and Ubuntu 18.04, 20.04, 22.04.

  For CentOS 8, RPM packages for ETCD is available within Percona Distribution for PostreSQL. You can install it using the following command:

  ```
  $ sudo yum install etcd python3-python-etcd
  ```

- HAProxy.

See the configuration guidelines for Debian and Ubuntu and RHEL and CentOS.

> ✏️ **See also**
>
> - Patroni documentation
> - Percona Blog:
>   - PostgreSQL HA with Patroni: Your Turn to Test Failure Scenarios

pgBadger

Enable the following options in `postgresql.conf` configuration file before starting the service:

```
log_min_duration_statement = 0
log_line_prefix = '%t [%p]: '
log_checkpoints = on
log_connections = on
log_disconnections = on
log_lock_waits = on
log_temp_files = 0
log_autovacuum_min_duration = 0
log_error_verbosity = default
```

For details about each option, see pdBadger documentation.

pgAudit set-user

Add the `set-user` to `shared_preload_libraries` in `postgresql.conf`. The recommended way is to use the ALTER SYSTEM command. Connect to psql and use the following command:

```
ALTER SYSTEM SET shared_preload_libraries = 'set-user';
```

Start / restart the server to apply the configuration.

You can fine-tune user behavior with the custom parameters supplied with the extension.

wal2json

After the installation, enable the following option in `postgresql.conf` configuration file before starting the service:

```
wal_level = logical
```

Contact Us

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

December 5, 2022

December 5, 2022

3.1.5   Repositories overview

Percona provides two repositories for Percona Distribution for PostgreSQL.

Major release repository

Major Release repository (e.g. `ppg-11`) 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.

We recommend installing Percona Distribution for PostgreSQL from the *Major Release repository*

Minor release repository

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-11.10`). 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.

Repository contents

Percona Distribution for PostgreSQL provides individual packages for its components. It also includes two meta-packages: `percona-ppg-server` and `percona-ppg-server-ha`.

Using a meta-package, you can install all components it contains in one go.

PERCONA-PPG-SERVER

Package name on Debian/Ubuntu	Package name on RHEL/derivatives
`percona-ppg-server-11`	
`percona-ppg-server11`	

The `percona-ppg-server` meta-package installs the PostgreSQL server with the following packages:

Package contents	Description
`percona-postgresql11-server`	The PostgreSQL server package.
`percona-postgresql-common`	PostgreSQL database-cluster manager. It provides a structure under which multiple versions of PostgreSQL may be installed and/or multiple clusters maintained at one time.
`percona-postgresql11-contrib`	A collection of additional PostgreSQLcontrib extensions
`percona-pg-stat-monitor11`	A Query Performance Monitoring tool for PostgreSQL.
`percona-pgaudit11`	Provides detailed session or object audit logging via the standard PostgreSQL logging facility.
`percona-pg_repack11`	rebuilds PostgreSQL database objects.
`percona-wal2json11`	a PostgreSQL logical decoding JSON output plugin.

PERCONA-PPG-SERVER-HA

Package name on Debian/Ubuntu	Package name on RHEL/derivatives
`percona-ppg-server-ha-11`	
`percona-ppg-server-11`	

The `percona-ppg-server-ha` meta-package installs high-availability components that are recommended by Percona:

Package contents	Description
`percona-patroni`	A high-availability solution for PostgreSQL.
`percona-haproxy`	A high-availability and load-balancing solution
`etcd`	A consistent, distributed key-value store
`python3-python-etcd`	A Python client for ETCD.[1]
`etcd-client`, `etcd-server`	The client/server of the distributed key-value store. [2]

1. Is included in repositories for RHEL 8 / CentOS 8 operating systems ↩
2. Are included in repositories for Debian 12 operating system ↩

Contact Us

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

December 13, 2023

September 14, 2022

3.2   Run Percona Distribution for PostgreSQL in a Docker container

Docker images of Percona Distribution for PostgreSQL are hosted publicly on Docker Hub.

For more information about using Docker, see the Docker Docs.

Make sure that you are using the latest version of Docker. The ones provided via `apt` and `yum` may be outdated and cause errors.

By default, Docker pulls the image from Docker Hub if it is not available locally.

Docker image contents

The Docker image of Percona Distribution for PostgreSQL includes the following components:

Component name	Description
`percona-postgresql11`	A metapackage that installs the latest version of PostgreSQL
`percona-postgresql11-server`	The PostgreSQL server package.
`percona-postgresql-common`	PostgreSQL database-cluster manager. It provides a structure under which multiple versions of PostgreSQL may be installed and/or multiple clusters maintained at one time.
`percona-postgresql-client-common`	The manager for multiple PostgreSQL client versions.
`percona-postgresql11-contrib`	A collection of additional PostgreSQLcontrib extensions
`percona-postgresql11-libs`	Libraries for use with PostgreSQL.
`percona-pg-stat-monitor11`	A Query Performance Monitoring tool for PostgreSQL.
`percona-pgaudit11`	Provides detailed session or object audit logging via the standard PostgreSQL logging facility.
`percona-pgaudit11_set_user`	An additional layer of logging and control when unprivileged users must escalate themselves to superuser or object owner roles in order to perform needed maintenance tasks.
`percona-pg_repack11`	rebuilds PostgreSQL database objects.
`percona-wal2json11`	a PostgreSQL logical decoding JSON output plugin.

3.2.1   Start the container

1. Start a Percona Distribution for PostgreSQL container as follows:

```
$ docker run --name container-name -e POSTGRES_PASSWORD=secret -d percona/percona-
distribution-postgresql:tag
```

Where:

- `container-name` is the name you assign to your container
- `POSTGRES_PASSWORD` is the superuser password
- `tag` is the tag specifying the version you want.

Check the full list of tags.

> 🔥 **Tip**
>
> You can secure the password by exporting it to the environment file and using that to start the container.
>
> a. Export the password to the environment file:
>
> ```
> $ echo "POSTGRES_PASSWORD=secret" > .my-pg.env
> ```
>
> b. Start the container:
>
> ```
> $ docker run --name container-name --env-file ./.my-pg.env -d percona/percona-
> distribution-postgresql:tag
> ```

2. Connect to the container's interactive terminal:

```
$ docker exec -it container-name bash
```

The `container-name` is the name of the container that you started in the previous step.

3.2.2 Connect to Percona Distribution for PostgreSQL from an application in another Docker container

This image exposes the standard PostgreSQL port (`5432`), so container linking makes the instance available to other containers. Start other containers like this in order to link it to the Percona Distribution for PostgreSQL container:

```
$ docker run --name app-container-name --network container:container-name -d app-that-uses-
postgresql
```

where:

- `app-container-name` is the name of the container where your application is running,
- `container name` is the name of your Percona Distribution for PostgreSQL container, and
- `app-that-uses-postgresql` is the name of your PostgreSQL client.

3.2.3 Connect to Percona Distribution for PostgreSQL from the `psql` command line client

The following command starts another container instance and runs the `psql` command line client against your original container, allowing you to execute SQL statements against your database:

```
$ docker run -it --network container:db-container-name --name container-name percona/
percona-distribution-postgresql:tag psql -h address -U postgres
```

Where:

- `db-container-name` is the name of your database container
- `container-name` is the name of your container that you will use to connect to the database container using the `psql` command line client
- `tag` is the tag specifying the Docker image version you want to use.
- `address` is the network address where your database container is running. Use 127.0.0.1, if the database container is running on the local machine/host.

3.2.4   Enable `pg_stat_monitor`

To enable the `pg_stat_monitor` extension after launching the container, do the following:

- connect to the server,
- select the desired database and enable the `pg_stat_monitor` view for that database:

```
create extension pg_stat_monitor;
```

- to ensure that everything is set up correctly, run:

```
\d pg_stat_monitor;
```

☰ Output

```
                        View "public.pg_stat_monitor"
        Column         |            Type          | Collation | Nullable | Default
-----------------------+--------------------------+-----------+----------+---------
 bucket                | integer                  |           |          |
 bucket_start_time     | timestamp with time zone |           |          |
 userid                | oid                      |           |          |
 dbid                  | oid                      |           |          |
 queryid               | text                     |           |          |
 query                 | text                     |           |          |
 plan_calls            | bigint                   |           |          |
 plan_total_time       | numeric                  |           |          |
 plan_min_timei        | numeric                  |           |          |
 plan_max_time         | numeric                  |           |          |
 plan_mean_time        | numeric                  |           |          |
 plan_stddev_time      | numeric                  |           |          |
 plan_rows             | bigint                   |           |          |
 calls                 | bigint                   |           |          |
 total_time            | numeric                  |           |          |
 min_time              | numeric                  |           |          |
 max_time              | numeric                  |           |          |
 mean_time             | numeric                  |           |          |
 stddev_time           | numeric                  |           |          |
 rows                  | bigint                   |           |          |
 shared_blks_hit       | bigint                   |           |          |
 shared_blks_read      | bigint                   |           |          |
 shared_blks_dirtied   | bigint                   |           |          |
 shared_blks_written   | bigint                   |           |          |
 local_blks_hit        | bigint                   |           |          |
 local_blks_read       | bigint                   |           |          |
 local_blks_dirtied    | bigint                   |           |          |
 local_blks_written    | bigint                   |           |          |
 temp_blks_read        | bigint                   |           |          |
 temp_blks_written     | bigint                   |           |          |
 blk_read_time         | double precision         |           |          |
 blk_write_time        | double precision         |           |          |
 host                  | bigint                   |           |          |
 client_ip             | inet                     |           |          |
 resp_calls            | text[]                   |           |          |
 cpu_user_time         | double precision         |           |          |
 cpu_sys_time          | double precision         |           |          |
 tables_names          | text[]                   |           |          |
 wait_event            | text                     |           |          |
 wait_event_type       | text                     |           |          |
```

Note that the `pg_stat_monitor` view is available only for the databases where you enabled it. If you create a new database, make sure to create the view for it to see its statistics data.

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

🕐 December 13, 2023

🕐 December 13, 2023

3.3   Migrate from PostgreSQL to Percona Distribution for PostgreSQL

Percona Distribution for PostgreSQL includes the PostgreSQL database and additional extensions that have been selected to cover the needs of the enterprise and are guaranteed to work together. Percona Distribution for PostgreSQL is available as a software collection that is easy to deploy.

We encourage users to migrate from their PostgreSQL deployments based on community binaries to Percona Distribution for PostgreSQL. This document provides the migration instructions.

Depending on your business requirements, you may migrate to Percona Distribution for PostgreSQL either on the same server or onto a different server.

3.3.1 Migrate on the same server

On Debian and Ubuntu Linux   **On RHEL and compatible derivatives**

To ensure that your data is safe during the migration, we recommend to make a backup of your data and all configuration files (such as `pg_hba.conf`, `postgresql.conf`, `postgresql.auto.conf`) using the tool of your choice. The backup process is out of scope of this document. You can use `pg_dumpall` or other tools of your choice.

1. Stop the `postgresql` server

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

2. Remove community packages

```
$ sudo apt-get --purge remove postgresql
```

3. Install percona-release

4. Enable the repository

```
$ sudo percona-release setup ppg11
```

5. Install Percona Distribution for PostgreSQL packages

6. (Optional) Restore the data from the backup.

7. Start the `postgresql` service. The installation process starts and initializes the default cluster automatically. You can check its status with:

```
$ sudo systemctl status postgresql
```

If `postresql` service is not started, start it manually:

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

To ensure that your data is safe during the migration, we recommend to make a backup of your data and all configuration files (such as `pg_hba.conf`, `postgresql.conf`, `postgresql.auto.conf`) using the tool of your choice. The backup process is out of scope of this document. You can use `pg_dumpall` or other tools of your choice.

1. Stop the `postgresql` server

```
$ sudo systemctl stop postgresql-11
```

2. Remove community packages

```
$ sudo yum remove postgresql
```

3. Install percona-release

4. Enable the repository

```
$ sudo percona-release setup ppg11
```

5. Install Percona Distribution for PostgreSQL packages

6. (Optional) Restore the data from the backup.

7. Start the `postgresql` service

```
$ sudo systemctl start postgresql-11
```

3.3.2   Migrate on a different server

In this scenario, we will refer to the server with PostgreSQL Community as the "source" and to the server with Percona Distribution for PostgreSQL as the "target".

To migrate from PostgreSQL Community to Percona Distribution for PostgreSQL on a different server, do the following:

On the source server:

1. Back up your data and all configuration files (such as `pg_hba.conf`, `postgresql.conf`, `postgresql.auto.conf`) using the tool of your choice.

2. Stop the `postgresql` service

   On Debian and Ubuntu   **On RHEL and derivatives**

   ```
   $ sudo systemctl stop postgresql.service

   $ sudo systemctl stop postgresql-11
   ```

3. Optionally, remove PostgreSQL Community packages

On the target server:

1. Install percona-release

2. Enable the repository

   ```
   $ sudo percona-release setup ppg11
   ```

3. Install Percona Distribution for PostgreSQL packages on the target server.

4. Restore the data from the backup

5. Start `postgresql` service

   On Debian and Ubuntu   **On RHEL and compatible derivatives**

   ```
   $ sudo systemctl start postgresql.service

   $ sudo systemctl start postgresql-11
   ```

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

December 5, 2022

July 22, 2022

3.4 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   **On Red Hat Enterprise Linux and derivatives**

   ```
   $ sudo systemctl stop postgresql.service

   $ sudo systemctl stop postgresql-11
   ```

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

   On Debian / Ubuntu   **On Red Hat Enterprise Linux and derivatives**

   ```
   $ sudo systemctl start postgresql.service

   $ sudo systemctl start postgresql-11
   ```

If you wish to upgrade Percona Distribution for PostgreSQL to the major version, refer to Upgrading Percona Distribution for PostgreSQL from 11 to 12.

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

December 5, 2022

May 28, 2021

4. Extensions

4.1 pg_stat_monitor

> ✏️ **Note**
>
> This document describes the functionality of `pg_stat_monitor` 2.0.0.

4.1.1 Overview

`pg_stat_monitor` is a Query Performance Monitoring tool for PostgreSQL. It collects various statistics data such as query statistics, query plan, SQL comments and other performance insights. The collected data is aggregated and presented in a single view. This allows you to view queries from performance, application and analysis perspectives.

`pg_stat_monitor` groups statistics data and writes it in a storage unit called *bucket*. The data is added and stored in a bucket for the defined period – the bucket lifetime. This allows you to identify performance issues and patterns based on time.

You can specify the following:

• The number of buckets. Together they form a bucket chain.

• Bucket size. This is the amount of shared memory allocated for buckets. Memory is divided equally among buckets.

• Bucket lifetime.

When a bucket lifetime expires, `pg_stat_monitor` resets all statistics and writes the data in the next bucket in the chain. When the last bucket's lifetime expires, `pg_stat_monitor` returns to the first bucket.

> 🔥 **Important**
>
> The contents of the bucket will be overwritten. In order not to lose the data, make sure to read the bucket before `pg_stat_monitor` starts writing new data to it.

Views

PG_STAT_MONITOR VIEW

The `pg_stat_monitor` view contains all the statistics collected and aggregated by the extension. This view contains one row for each distinct combination of metrics and whether it is a top-level statement or not (up to the maximum number of distinct statements that the module can track). For details about available metrics, refer to the `pg_stat_monitor` view reference.

The following are the primary keys for pg_stat_monitor:

- `bucket`

- `userid`

- `datname`

- `queryid`

- `client_ip`

- `planid`

- `application_name`

A new row is created for each key in the `pg_stat_monitor` view.

For security reasons, only superusers and members of the `pg_read_all_stats` role are allowed to see the SQL text, `client_ip` and `queryid` of queries executed by other users. Other users can see the statistics, however, if the view has been installed in their database.

PG_STAT_MONITOR_SETTINGS VIEW (DROPPED)

Starting with version 2.0.0, the `pg_stat_monitor_settings` view is deprecated and removed. All `pg_stat_monitor` configuration parameters are now available though the `pg_settings` view using the following query:

```
SELECT name, setting, unit, context, vartype, source, min_val, max_val, enumvals, boot_val,
reset_val, pending_restart FROM pg_settings WHERE name LIKE '%pg_stat_monitor%';
```

For backward compatibility, you can create the `pg_stat_monitor_settings` view using the following SQL statement:

```
CREATE VIEW pg_stat_monitor_settings

AS

SELECT *

FROM pg_settings

WHERE name like 'pg_stat_monitor.%';
```

In `pg_stat_monitor` version 1.1.1 and earlier, the `pg_stat_monitor_settings` view shows one row per `pg_stat_monitor` configuration parameter. It displays configuration parameter name, value, default value, description, minimum and maximum values, and whether a restart is required for a change in value to be effective.

To learn more, see the Changing the configuration section.

4.1.2 Installation

This section describes how to install `pg_stat_monitor` from Percona repositories. To learn about other installation methods, see the Installation section in the `pg_stat_monitor` documentation.

Preconditions:

To install `pg_stat_monitor` from Percona repositories, you need to subscribe to them. To do this, you must have the `percona-release` repository management tool up and running.

To install `pg_stat_monitor`, run the following commands:

On Debian and Ubuntu   **On Red Hat Enterprise Linux and derivatives**

1. Enable the repository

```
$ sudo percona-release setup ppg11
```

2. Update the local cache

```
$ sudo apt update
```

3. Install the package:

```
$ sudo apt-get install percona-pg-stat-monitor11
```

1. Enable the repository

```
$ sudo percona-release setup ppg11
```

2. Install the package:

```
$ sudo yum install percona-pg-stat-monitor11
```

On Debian and Ubuntu   **On Red Hat Enterprise Linux and derivatives**

4.1.3   Setup

`pg_stat_monitor` requires additional setup in order to use it with PostgreSQL. The setup steps are the following:

1. Add `pg_stat_monitor` in the `shared_preload_libraries` configuration parameter.

   The recommended way to modify PostgreSQL configuration file is using the ALTER SYSTEM command. Connect to `psql` and use the following command:

   ```
   ALTER SYSTEM SET shared_preload_libraries = 'pg_stat_monitor';
   ```

   The parameter value is written to the `postgresql.auto.conf` file which is read in addition with `postgresql.conf` file.

   > ✏️ **Note**
   >
   > To use `pg_stat_monitor` together with `pg_stat_statements`, specify both modules separated by commas for the `ALTER SYSTEM SET` command.
   >
   > The order of modules is important: `pg_stat_monitor` must be specified **after** `pg_stat_statements`:
   >
   > ```
   > ALTER SYSTEM SET shared_preload_libraries = 'pg_stat_statements, pg_stat_monitor'
   > ```

2. Start or restart the `postgresql` instance to enable `pg_stat_monitor`. Use the following command for restart:

   On Debian and Ubuntu   **On Red Hat Enterprise Linux and derivatives**

   ```
   $ sudo systemctl restart postgresql.service

   $ sudo systemctl restart postgresql-11
   ```

3. Create the extension. Connect to `psql` and use the following command:

   ```
   CREATE EXTENSION pg_stat_monitor;
   ```

   By default, the extension is created against the `postgres` database. You need to create the extension on every database where you want to collect statistics.

> 🔥 **Tip**
>
> To check the version of the extension, run the following command in the `psql` session:
>
> ```
> SELECT pg_stat_monitor_version();
> ```

4.1.4   Usage

For example, to view the IP address of the client application that made the query, run the following command:

```
SELECT DISTINCT userid::regrole, pg_stat_monitor.datname, substr(query,0, 50)
  AS query, calls, client_ip
  FROM pg_stat_monitor, pg_database
```

```
  WHERE pg_database.oid = oid;

  userid  | datname  |                        query                      | calls | client_ip
----------+----------+---------------------------------------------------+-------
 +-----------
 postgres | postgres | select bucket, bucket_start_time, query,calls fro |     1 | 127.0.0.1
 postgres | postgres | SELECT c.relchecks, c.relkind, c.relhasindex, c.r |     1 | 127.0.0.1
 postgres | postgres | SELECT  userid,  total_time, min_time, max_time,  |     1 | 127.0.0.1
```

Find more usage examples in the `pg_stat_monitor` user guide.

4.1.5   Changing the configuration

Run the following query to list available configuration parameters.

```
SELECT name, short_desc FROM pg_settings WHERE name LIKE '%pg_stat_monitor%';
```

Output

```
                name
|                                                 short_desc
------------------------------------------
 +----------------------------------------------------------------------------------
 pg_stat_monitor.pgsm_bucket_time        | Sets the time in seconds per bucket.
 pg_stat_monitor.pgsm_enable_overflow    | Enable/Disable pg_stat_monitor to grow beyond
shared memory into swap space.
 pg_stat_monitor.pgsm_enable_pgsm_query_id | Enable/disable PGSM specific query id
calculation which is very useful in comparing same query across databases and clusters..
 pg_stat_monitor.pgsm_enable_query_plan  | Enable/Disable query plan monitoring.
 pg_stat_monitor.pgsm_extract_comments   | Enable/Disable extracting comments from
queries.
 pg_stat_monitor.pgsm_histogram_buckets  | Sets the maximum number of histogram buckets.
 pg_stat_monitor.pgsm_histogram_max      | Sets the time in millisecond.
 pg_stat_monitor.pgsm_histogram_min      | Sets the time in millisecond.
 pg_stat_monitor.pgsm_max                | Sets the maximum size of shared memory in (MB)
used for statement's metadata tracked by pg_stat_monitor.
 pg_stat_monitor.pgsm_max_buckets        | Sets the maximum number of buckets.
 pg_stat_monitor.pgsm_normalized_query   | Selects whether save query in normalized
format.
 pg_stat_monitor.pgsm_overflow_target    | Sets the overflow target for pg_stat_monitor.
(Deprecated, use pgsm_enable_overflow)
 pg_stat_monitor.pgsm_query_max_len      | Sets the maximum length of query.
 pg_stat_monitor.pgsm_query_shared_buffer  | Sets the maximum size of shared memory in (MB)
used for query tracked by pg_stat_monitor.
 pg_stat_monitor.pgsm_track             | Selects which statements are tracked by
pg_stat_monitor.
 pg_stat_monitor.pgsm_track_planning     | Selects whether planning statistics are
tracked.
 pg_stat_monitor.pgsm_track_utility      | Selects whether utility commands are tracked.
```

You can change a parameter by setting a new value in the configuration file. Some parameters require server restart to apply a new value. For others, configuration reload is enough. Refer to the configuration parameters of the `pg_stat_monitor` documentation for the parameters' description, how you can change their values and if the server restart is required to apply them.

As an example, let's set the bucket lifetime from default 60 seconds to 30 seconds. Use the **ALTER SYSTEM** command:

```
ALTER SYSTEM set pg_stat_monitor.pgsm_bucket_time = 30;
```

Restart the server to apply the change:

On Debian and Ubuntu   **On Red Hat Enterprise Linux and derivatives**

```
$ sudo systemctl restart postgresql.service

$ sudo systemctl restart postgresql-11
```

Verify the updated parameter:

```
SELECT name, setting
FROM pg_settings
WHERE name = 'pg_stat_monitor.pgsm_bucket_time';

               name                 | setting
-----------------------------------+---------
   pg_stat_monitor.pgsm_bucket_time |    30
```

> ✏️ **See also**
>
> `pg_stat_monitor` Documentation
>
> Percona Blog:
>
> - pg_stat_monitor: A New Way Of Looking At PostgreSQL Metrics
> - Improve PostgreSQL Query Performance Insights with pg_stat_monitor

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

🕓 April 26, 2023

🕓 May 28, 2021

5. Solutions

5.1 High Availability in PostgreSQL with Patroni

PostgreSQL has been widely adopted as a modern, high-performance transactional database. A highly available PostgreSQL cluster can withstand failures caused by network outages, resource saturation, hardware failures, operating system crashes or unexpected reboots. Such cluster is often a critical component of the enterprise application landscape, where four nines of availability is a minimum requirement.

There are several methods to achieve high availability in PostgreSQL. This solution document provides Patroni - the open-source extension to facilitate and manage the deployment of high availability in PostgreSQL.

High availability methods

There are several native methods for achieving high availability with PostgreSQL:

- shared disk failover,
- file system replication,
- trigger-based replication,
- statement-based replication,
- logical replication,
- Write-Ahead Log (WAL) shipping,
- streaming replication.

5.1.1 Streaming replication

Streaming replication is part of Write-Ahead Log shipping, where changes to the WALs are immediately made available to standby replicas. With this approach, a standby instance is always up-to-date with changes from the primary node and can assume the role of primary in case of a failover.

Why native streaming replication is not enough

Although the native streaming replication in PostgreSQL supports failing over to the primary node, it lacks some key features expected from a truly highly-available solution. These include:

- No consensus-based promotion of a "leader" node during a failover
- No decent capability for monitoring cluster status
- No automated way to bring back the failed primary node to the cluster
- A manual or scheduled switchover is not easy to manage

To address these shortcomings, there are a multitude of third-party, open-source extensions for PostgreSQL. The challenge for a database administrator here is to select the right utility for the current scenario.

Percona Distribution for PostgreSQL solves this challenge by providing the Patroni extension for achieving PostgreSQL high availability.


5.1.2 Patroni

Patroni is a template for you to create your own customized, high-availability solution using Python and - for maximum accessibility - a distributed configuration store like ZooKeeper, etcd, Consul or Kubernetes.

Key benefits of Patroni:

- Continuous monitoring and automatic failover
- Manual/scheduled switchover with a single command
- Built-in automation for bringing back a failed node to cluster again.
- REST APIs for entire cluster configuration and further tooling.
- Provides infrastructure for transparent application failover
- Distributed consensus for every action and configuration.
- Integration with Linux watchdog for avoiding split-brain syndrome.

5.1.3   Architecture layout

The following diagram shows the architecture of a three-node PostgreSQL cluster with a single-leader node.
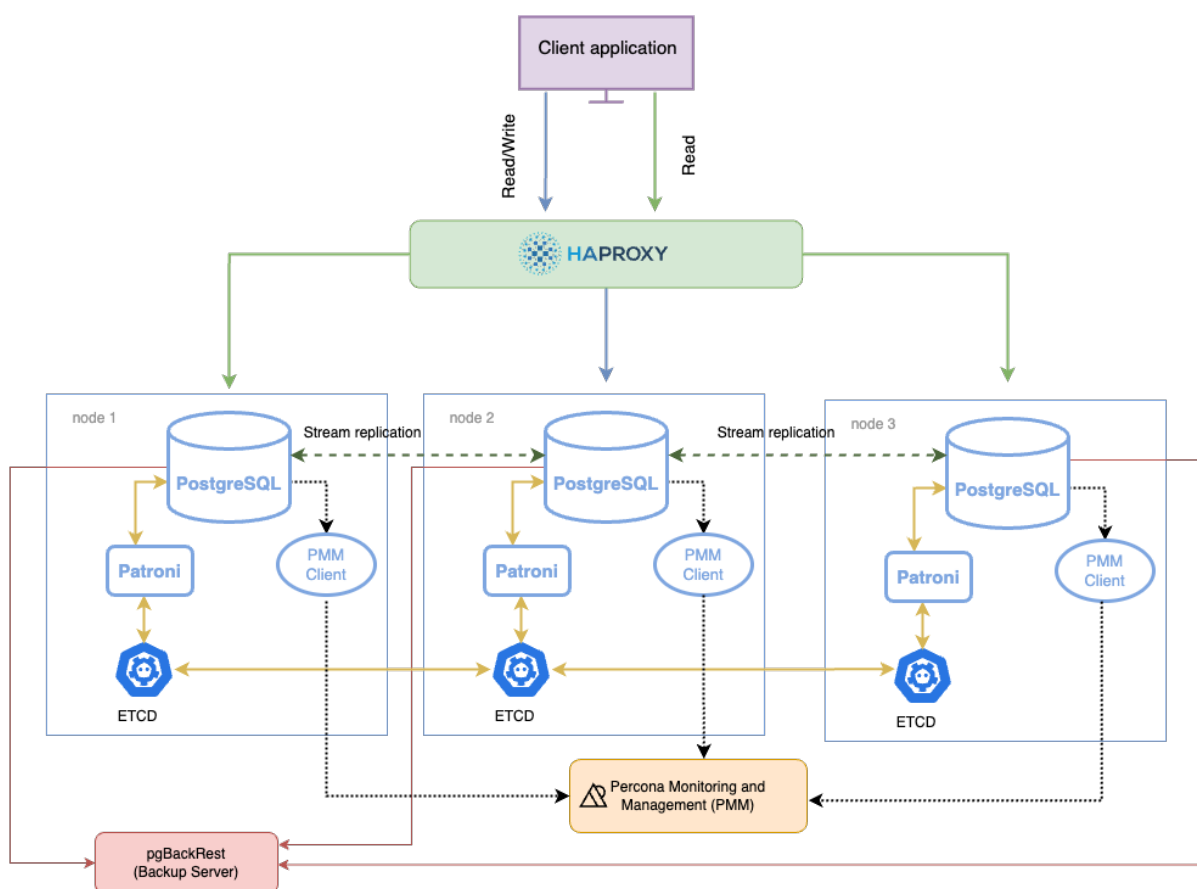
Components

The components in this architecture are:

- PostgreSQL nodes
- Patroni - a template for configuring a highly available PostgreSQL cluster.
- ETCD - a Distributed Configuration store that stores the state of the PostgreSQL cluster.
- HAProxy - the load balancer for the cluster and is the single point of entry to client applications.
- pgBackRest - the backup and restore solution for PostgreSQL
- Percona Monitoring and Management (PMM) - the solution to monitor the health of your cluster

How components work together

Each PostgreSQL instance in the cluster maintains consistency with other members through streaming replication. Each instance hosts Patroni - a cluster manager that monitors the cluster health. Patroni relies on the operational ETCD cluster to store the cluster configuration and sensitive data about the cluster health there.

Patroni periodically sends heartbeat requests with the cluster status to ETCD. ETCD writes this information to disk and sends the response back to Patroni. If the current primary fails to renew its status as leader within the specified timeout, Patroni updates the state change in ETCD, which uses this information to elect the new primary and keep the cluster up and running.

The connections to the cluster do not happen directly to the database nodes but are routed via a connection proxy like HAProxy. This proxy determines the active node by querying the Patroni REST API.

5.1.4   Next steps

Deploy on Debian or Ubuntu **Deploy on RHEL or derivatives**

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

November 1, 2023

December 15, 2021

5.2   Backup and disaster recovery in Percona Distribution for PostgreSQL

☰   **Summary**

• Overview

• Architecture

• Deployment

• Testing

5.2.1   Overview

A Disaster Recovery (DR) solution ensures that a system can be quickly restored to a normal operational state if something unexpected happens. When operating a database, you would back up the data as frequently as possible and have a mechanism to restore that data when needed. Disaster Recovery is often mistaken for high availability (HA), but they are two different concepts altogether:

• High availability ensures guaranteed service levels at all times. This solution involves configuring one or more standby systems to an active database, and the ability to switch seamlessly to that standby when the primary database becomes unavailable, for example, during a power outage or a server crash. To learn more about high-availability solutions with Percona Distribution for PostgreSQL, refer to High Availability in PostgreSQL with Patroni.

• Disaster Recovery protects the database instance against accidental or malicious data loss or data corruption. Disaster recovery can be achieved by using either the options provided by PostgreSQL, or external extensions.

PostgreSQL disaster recovery options

PostgreSQL offers multiple options for setting up database disaster recovery.

- **pg_dump** or the **pg_dumpall** utilities

This is the basic backup approach. These tools can generate the backup of one or more PostgreSQL databases (either just the structure, or both the structure and data), then restore them through the pg_restore command.

Advantages	Disadvantages
Easy to use	1. Backup of only one database at a time.
	2. No incremental backups.
	3. No point-in-time recovery since the backup is a snapshot in time.
	4. Performance degradation when the database size is large.

- **File-based backup and restore**

This method involves backing up the PostgreSQL data directory to a different location, and restoring it when needed.

Advantages	Disadvantages
Consistent snapshot of the data directory or the whole data disk volume	1. Requires stopping PostgreSQL in order to copy the files. This is not practical for most production setups.
	2. No backup of individual databases or tables.

- **PostgreSQL pg_basebackup**

This backup tool is provided by PostgreSQL. It is used to back up data when the database instance is running. `pgasebackup` makes a binary copy of the database cluster files, while making sure the system is put in and out of backup mode automatically.

Advantages	Disadvantages
1. Supports backups when the database is running.	1. No incremental backups.
2. Supports point-in-time recovery	2. No backup of individual databases or tables.

To achieve a production grade PostgreSQL disaster recovery solution, you need something that can take full or incremental database backups from a running instance, and restore from those backups at any point in time. Percona Distribution for PostgreSQL is supplied with pgBackRest: a reliable, open-source backup and recovery solution for PostgreSQL.

This document focuses on the Disaster recovery solution in Percona Distribution for PostgreSQL. The Deploying backup and disaster recovery solution in Percona Distribution for PostgreSQL tutorial provides guidelines of how to set up and test this solution.

pgBackRest

pgBackRest is an easy-to-use, open-source solution that can reliably back up even the largest of PostgreSQL databases. `pgBackRest` supports the following backup types:

- full backup - a complete copy of your entire data set.

- differential backup - includes all data that has changed since the last full backup. While this means the backup time is slightly higher, it enables a faster restore.

- incremental backup - only backs up the files that have changed since the last full or differential backup, resulting in a quick backup time. To restore to a point in time, however, you will need to restore each incremental backup in the order they were taken.

When it comes to restoring, `pgBackRest` can do a full or a delta restore. A *full* restore needs an empty PostgreSQL target directory. A *delta* restore is intelligent enough to recognize already-existing files in the PostgreSQL data directory, and update only the ones the backup contains.

`pgBackRest` supports remote repository hosting and can even use cloud-based services like AWS S3, Google Cloud Services Cloud Storage, Azure Blob Storage for saving backup files. It supports parallel backup through multi-core processing and compression. By default, backup integrity is verified through checksums, and saved files can be encrypted for enhanced security.

`pgBackRest` can restore a database to a specific point in time in the past. This is the case where a database is not inaccessible but perhaps contains corrupted data. Using the point-in-time recovery, a database administrator can restore the database to the last known good state.

Finally, `pgBackRest` also supports restoring PostgreSQL databases to a different PostgreSQL instance or a separate data directory.

5.2.2   Setup overview

This section describes the architecture of the backup and disaster recovery solution. For the configuration steps, refer to the Deploying backup and disaster recovery solution in Percona Distribution for PostgreSQL.

System architecture

As the configuration example, we will use a three server architecture where `pgBackRest` resides on a dedicated remote host. The servers communicate with each other via passwordless SSH.

> ⚠️ **Warning**
>
> Passwordless SSH may not be an ideal solution for your environment. In this case, consider using other methods, for example, TLS with client certificates.

The following diagram illustrates the architecture layout:
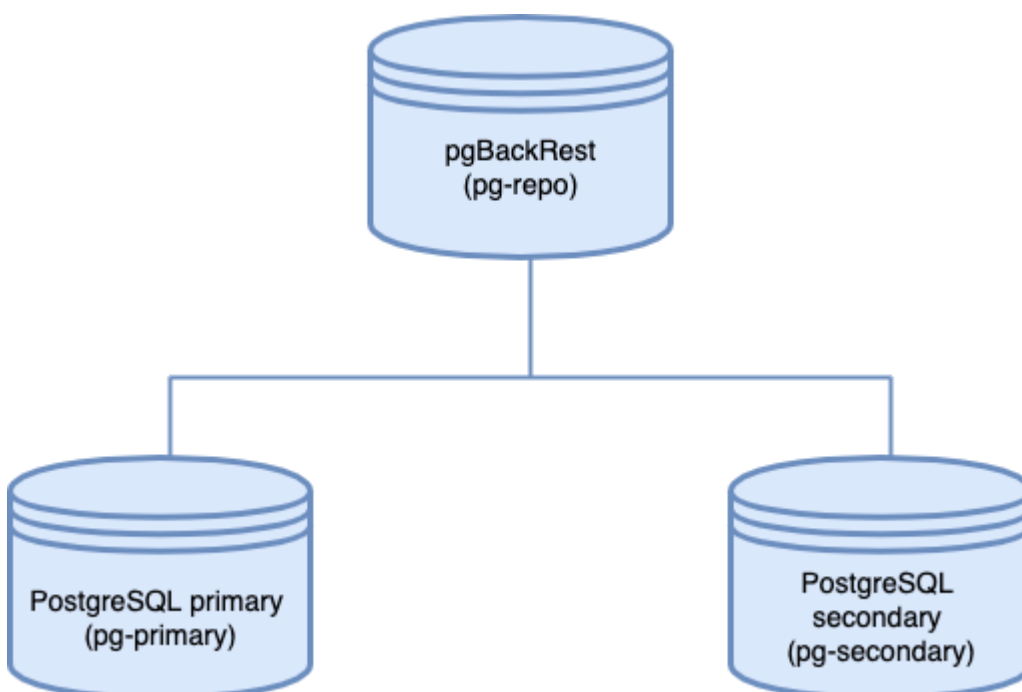
COMPONENTS:

The architecture consists of three server instances:

- `pg-primary` hosts the primary PostgreSQL server. Note that "primary" here means the main database instance and does not refer to the primary node of a PostgreSQL replication cluster or a HA setup.
- `pg-repo` is the remote backup repository and hosts `pgBackRest`. It's important to host the backup repository on a physically separate instance, to be accessed when the target goes down.
- `pg-secondary` is the *secondary* PostgreSQL node. Don't confuse it with a hot standby. "Secondary" in this context means a PostgreSQL instance that's idle. We will restore the database backup to this instance when the primary PostgreSQL instance goes down.

> ✏️ **Note**

For simplicity, we use a single-node PostgreSQL instance as the primary database server. In a production scenario, you will use some form of high-availability solution to protect the primary instance. When you are using a high-availability setup, we recommend configuring `pgBackRest` to back up the hot standby server so the primary node is not unnecessarily loaded.

Deployment

Refer to the Deploying backup and disaster recovery solution in Percona Distribution for PostgreSQL tutorial.

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

🕐 April 26, 2022

🕐 January 21, 2022

5.3   Spatial data handling

5.3.1   Spatial data manipulation

> ✏️ **Version added: 11.20**

Organizations dealing with spatial data need to store it somewhere and manipulate it. PostGIS is the open-source extension for PostgreSQL that allows doing just that. It adds support for storing the spatial data types such as:

- Geographical data like points, lines, polygons, GPS coordinates that can be mapped on a sphere.
- Geometrical data. This is also points, lines and polygons but they apply to a 2D surface.

To operate with spatial data inside SQL queries, PostGIS supports spatial functions like distance, area, union, intersection. It uses the spatial indexes like R-Tree and Quadtree for efficient processing of database operations. Read more about supported spatial functions and indexes in PostGIS documentation.

By deploying PostGIS with Percona Distribution for PostgreSQL, you receive the open source spatial database that you can use in various areas without vendor lock-in. PostGIS packages from Percona are available for all currently supported PostgreSQL versions.

When to use PostGIS

You can use PostGIS in the following cases:

- To store and manage spatial data, create and store spatial shapes, calculate areas and distances
- To build the software that visualizes spatial data on a map,
- To work with raster data, such as satellite imagery or digital elevation models.
- To integrate spatial and non-spatial data such as demographic or economic data in a database

When not to use PostGIS

Despite its power and flexibility, PostGIS may not suit your needs if:

- You need to store only a couple of map locations. Consider using the built-in geometric functions and operations of PostgreSQL
- You need real-time data analysis. While PostGIS can handle real-time spatial data, it may not be the best option for real-time data analysis on large volumes of data.
- You need complex 3D analysis or visualization.
- You need to acquire spatial data. Use other tools for this purpose and import spatial data into PostGIS to manipulate it.

Next steps:

Deployment

Contact Us

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

June 30, 2023

June 30, 2023

5.3.2   Deploy spatial data with PostgreSQL

The following document provides guidelines how to install PostGIS and how to run the basic queries.

Considerations

1. We assume that you have the basic knowledge of spatial data, GIS (Geographical Information System) and of shapefiles.

2. For uploading the spatial data and querying the database, we use the same data set as is used in PostGIS tutorial

Install PostGIS

On Debian and Ubuntu   On RHEL and derivatives

1. Enable Percona repository

   As other components of Percona Distribution for PostgreSQL, PostGIS is available from Percona repositories. Use the `percona-release` repository management tool to enable the repository.

   ```
   $ sudo percona-release setup ppg11
   ```

2. Install PostGIS packages

   ```
   $ sudo apt install percona-postgis
   ```

3. The command in the previous step installs the set of PostGIS extensions. To check what extensions are available, run the following query from the `psql` terminal:

   ```
   SELECT name, default_version,installed_version
   FROM pg_available_extensions WHERE name LIKE 'postgis%' or name LIKE address%';
   ```

   > ✏️ **Note**
   >
   > To enable the `postgis_sfcgal-3` extension on Ubuntu 18.04, you need to manually install the required dependency:
   >
   > ```
   > $ sudo apt-get install libsfcgal1
   > ```

For Red Hat Enterprise Linux 8 and derivatives, replace the operating system version in the following commands accordingly.

1. Check the Platform specific notes and enable required repositories and modules for the dependencies relevant to your operating system.

2. Enable Percona repository

   As other components of Percona Distribution for PostgreSQL, PostGIS is available from Percona repositories. Use the `percona-release` repository management tool to enable the repository.

   ```
   $ sudo percona-release setup ppg11
   ```

3. Install the extension

   ```
   $ sudo yum install percona-postgis33_11 percona-postgis33_11-client
   ```

This installs the set of PostGIS extensions. To check what extensions are available, run the following query from the `psql` terminal:

```
SELECT name, default_version,installed_version
FROM pg_available_extensions WHERE name LIKE 'postgis%' or name LIKE 'address%';
```

Enable PostGIS extension

1. Create a database and a schema for this database to store your data. A schema is a container that logically segments objects (tables, functions, views, and so on) for better management. Run the following commands from the `psql` terminal:

```
CREATE DATABASE nyc;
\c nyc;
CREATE SCHEMA gis;
```

2. To make PostGIS functions and operations work, you need to enable the `postgis` extension. Make sure you are connected to the database you created earlier and run the following command:

```
\c nyc;
CREATE EXTENSION postgis;
```

3. Check that the extension is enabled:

```
SELECT postgis_full_version();
```

The output should be similar to the following:

```
postgis_full_version
---------------------------------------------------------------------------------------
 POSTGIS="3.3.3" [EXTENSION] PGSQL="140" GEOS="3.10.2-CAPI-1.16.0" PROJ="8.2.1"
LIBXML="2.9.13" LIBJSON="0.15" LIBPROTOBUF="1.3.3" WAGYU="0.5.0 (Internal)"
```

Upload spatial data to PostgreSQL

PostGIS provides the `shp2pgsql` command line utility that converts the binary data from shapefiles into the series of SQL commands and loads them into the database.

1. For testing purposes, download the sample data set:

```
$ curl -LO https://s3.amazonaws.com/s3.cleverelephant.ca/postgis-workshop-2020.zip
```

2. Unzip the archive. From the folder where the `.shp` files are located, execute the following command and replace the `dbname` value with the name of your database:

```
shp2pgsql \
  -D \
  -I \
  -s 26918 \
  nyc_streets.shp \
  nyc_streets \
  | psql -U postgres dbname=nyc
```

The command does the following:

- `-D` flag instructs the command to generate the dump format
- `-I` flag instructs to create the spatial index on the table upon the data load
- `-s` indicates the spatial reference identifier of the data. The data we load is in the Projected coordinate system for North America and has the value 26918.
- `nyc_streets.shp` is the source shapefile
- `nyc_streets` is the table name to create in the database
- `dbname=nyc` is the database name

3. Check the uploaded data

```
\d nyc_streets;
                            Table "public.nyc_streets"
 Column |              Type               | Collation | Nullable |
Default
--------+---------------------------------+-----------+----------
+----------------------------------------
 gid    | integer                         |           | not null |
nextval('nyc_streets_gid_seq'::regclass)
 id     | double precision                |           |          |
 name   | character varying(200)          |           |          |
 oneway | character varying(10)           |           |          |
 type   | character varying(50)           |           |          |
 geom   | geometry(MultiLineString,26918) |           |          |
Indexes:
    "nyc_streets_pkey" PRIMARY KEY, btree (gid)
    "nyc_streets_geom_idx" gist (geom)
```

4. Repeat the command to upload other shapefiles in the data set: `nyc_census_blocks`, `nyc_neighborhoods`, `nyc_subway_stations`

Contact Us

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

December 13, 2023

June 30, 2023

5.3.3   Query spatial data

After you installed and set up PostGIS, let's find answers to the following questions by querying the database:

What is the population of the New York City?

```sql
SELECT Sum(popn_total) AS population
  FROM nyc_census_blocks;
```

Output:

```
population
------------
    8175032
(1 row)
```

What is the area of Central Park?

To get the answer we will use the `ST_Area` function that returns the areas of polygons.

```sql
SELECT ST_Area(geom) / 1000000
  FROM nyc_neighborhoods
  WHERE name = 'Central Park';
```

Output:

```
      st_area
-------------------
 3.5198365965413294
(1 row)
```

By default, the output is given in square meters. To get the value in square kilometers, divide it by 1 000 000.

How long is Columbus Circle?

```sql
SELECT ST_Length(geom)
  FROM nyc_streets
  WHERE name = 'Columbus Cir';
```

Output:

```
     st_length
------------------
 308.3419936909855
(1 row)
```

Contact Us

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

June 30, 2023

June 30, 2023

5.3.4   Spatial database upgrade

When using PostgreSQL and PostGIS for some time, you may eventually come to the decision to upgrade your spatial database. There can be different reasons for that: to receive improvements and/or bug fixes that come with a minor version of the database/extension, reaching the end of life of the currently used software and others.

The spatial database upgrade consists of two steps:

• upgrade of PostgreSQL, and

• upgrade of the PostGIS extension.

> 🔥 **Important**
>
> Before the upgrade, backup your data.

Upgrade PostGIS

Each version of PostGIS is compatible with several versions of PostgreSQL and vise versa. The best practice is to first upgrade the PostGIS extension on the source cluster to match the compatible version on the target cluster and then upgrade PostgreSQL. Please see the PostGIS Support matrix for version compatibility.

PostGIS is enabled on the database level. This means that the upgrade is also done on the database level.

PostGIS 3 and above   **PostGIS 2.5**

Connect to the database where it is enabled and run the `PostGIS_Extensions_Upgrade()` function:

```
SELECT postgis_extensions_upgrade();
```

Repeat these steps to upgrade PostGIS on every database where it is enabled.

Connect to the database with the enabled extension and run the following commands:

```
ALTER EXTENSION postgis UPDATE;
SELECT postgis_extensions_upgrade();
```

Starting with version 3, vector and raster functionalities have been separated in two individual extensions. Thus, to upgrade those, you need to run the `postgis_extensions_upgrade();` twice.

```
SELECT postgis_extensions_upgrade();
```

TIP: If you don't need the raster functionality, you can drop the `postgis_raster` extension after the upgrade.

Repeat these steps to upgrade PostGIS on every database where it is enabled.

Upgrade PostgreSQL

Upgrade PostgreSQL either to the latest minor or to the major version.

If you are using long deprecated views and functions and / or need the expertise in upgrading your spatial database, contact Percona Managed Services for an individual upgrade scenario development.

Contact Us

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

June 30, 2023

June 30, 2023

5.4   LDAP Authentication

When a client application or a user that runs the client application connects to the database, it must identify themselves. The process of validating the client's identity and determining whether this client is permitted to access the database it has requested is called **authentication**.

Percona Distribution for PortgreSQL supports several authentication methods, including the LDAP authentication. The use of LDAP is to provide a central place for authentication – meaning the LDAP server stores usernames and passwords and their resource permissions.

The LDAP authentication in Percona Distribution for PortgreSQL is implemented the same way as in upstream PostgreSQL.

CONTACT US

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

June 7, 2022

June 7, 2022

6. Telemetry on Percona Distribution for PostgreSQL

Percona telemetry fills in the gaps in our understanding of how you use Percona Distribution for PostgreSQL to improve our products. Participation in this anonymous program is optional. You can opt-out if you prefer to not share this information.

6.1 What information is collected

Currently, telemetry is added only to the Percona packages and Docker images. It collects only information about the installation environment. Future releases may add additional telemetry metrics.

Be assured that access to this raw data is rigorously controlled. Percona does not collect personal data. All data is anonymous and cannot be traced to a specific user. To learn more about our privacy practices, read the Percona Privacy statement.

The following is an example of the collected data:

```
[{"id" : "c416c3ee-48cd-471c-9733-37c2886f8231",
"product_family" : "PRODUCT_FAMILY_POSTGRESQL",
"instanceId" : "6aef422e-56a7-4530-af9d-94cc02198343",
"createTime" : "2023-11-01T10:46:23Z",
"metrics":
[{"key" : "deployment","value" : "PACKAGE"},
{"key" : "pillar_version","value" : "11.22"},
{"key" : "OS","value" : "Oracle Linux Server 8.8"},
{"key" : "hardware_arch","value" : "x86_64 x86_64"}]}]
```

6.2 Disable telemetry

Starting with Percona Distribution for PostgreSQL 11.22, telemetry is enabled by default. If you decide not to send usage data to Percona, you can set the `PERCONA_TELEMETRY_DISABLE=1` environment variable for either the root user or in the operating system prior to the installation process.

Debian-derived distribution   **Red Hat-derived distribution**   **DOCKER**

Add the environment variable before the install process.

```
$ sudo PERCONA_TELEMETRY_DISABLE=1 apt install percona-postgresql-11
```

Add the environment variable before the install process.

```
$ sudo PERCONA_TELEMETRY_DISABLE=1 yum install percona-postgresql11-server
```

Add the environment variable when running a command in a new container.

```
$ docker run --name container-name -e POSTGRES_PASSWORD=secret -e
PERCONA_TELEMETRY_DISABLE=1 -d percona/percona-distribution-postgresql:tag
```

Contact Us

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

December 13, 2023

December 13, 2023

7. 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.

On Debian and Ubuntu using `apt`   **On Red Hat Enterprise Linux and derivatives using** `yum`

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 PostgreSQL service.

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

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

   ```
   $ sudo apt remove percona-postgresql-11* percona-pgbackrest percona-patroni percona-
   pgbadger percona-pgbouncer
   ```

3. Remove configuration and data files.

   ```
   $ rm -rf /etc/postgresql/11/main
   ```

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 PostgreSQL service.

   ```
   $ sudo systemctl stop postgresql-11
   ```

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

   ```
   $ sudo yum remove percona-postgresql11* percona-pgbadger
   ```

3. Remove configuration and data files

   ```
   $ rm -rf /var/lib/pgsql/11/data
   ```

Contact Us

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

December 5, 2022

May 28, 2021

8. Copyright and licensing information

Percona Distribution for PostgreSQL is licensed under the PostgreSQL license and licenses of all components included in the Distribution.

8.1 Documentation licensing

Percona Distribution for PostgreSQL documentation is (C)2016-2023 Percona LLC and/or its affiliates and is distributed under the Creative Commons Attribution 4.0 International Public License license.

Contact Us

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

June 28, 2023

May 28, 2021

9.   Trademark Policy

This Trademark Policy is to ensure that users of Percona-branded products or services know that what they receive has really been developed, approved, tested and maintained by Percona. Trademarks help to prevent confusion in the marketplace, by distinguishing one company's or person's products and services from another's.

Percona owns a number of marks, including but not limited to Percona, XtraDB, Percona XtraDB, XtraBackup, Percona XtraBackup, Percona Server, and Percona Live, plus the distinctive visual icons and logos associated with these marks. Both the unregistered and registered marks of Percona are protected.

Use of any Percona trademark in the name, URL, or other identifying characteristic of any product, service, website, or other use is not permitted without Percona's written permission with the following three limited exceptions.

First, you may use the appropriate Percona mark when making a nominative fair use reference to a bona fide Percona product.

Second, when Percona has released a product under a version of the GNU General Public License ("GPL"), you may use the appropriate Percona mark when distributing a verbatim copy of that product in accordance with the terms and conditions of the GPL.

Third, you may use the appropriate Percona mark to refer to a distribution of GPL-released Percona software that has been modified with minor changes for the sole purpose of allowing the software to operate on an operating system or hardware platform for which Percona has not yet released the software, provided that those third party changes do not affect the behavior, functionality, features, design or performance of the software. Users who acquire this Percona-branded software receive substantially exact implementations of the Percona software.

Percona reserves the right to revoke this authorization at any time in its sole discretion. For example, if Percona believes that your modification is beyond the scope of the limited license granted in this Policy or that your use of the Percona mark is detrimental to Percona, Percona will revoke this authorization. Upon revocation, you must immediately cease using the applicable Percona mark. If you do not immediately cease using the Percona mark upon revocation, Percona may take action to protect its rights and interests in the Percona mark. Percona does not grant any license to use any Percona mark for any other modified versions of Percona software; such use will require our prior written permission.

Neither trademark law nor any of the exceptions set forth in this Trademark Policy permit you to truncate, modify or otherwise use any Percona mark as part of your own brand. For example, if XYZ creates a modified version of the Percona Server, XYZ may not brand that modification as "XYZ Percona Server" or "Percona XYZ Server", even if that modification otherwise complies with the third exception noted above.

In all cases, you must comply with applicable law, the underlying license, and this Trademark Policy, as amended from time to time. For instance, any mention of Percona trademarks should include the full trademarked name, with proper spelling and capitalization, along with attribution of ownership to Percona Inc. For example, the full proper name for XtraBackup is Percona XtraBackup. However, it is acceptable to omit the word "Percona" for brevity on the second and subsequent uses, where such omission does not cause confusion.

In the event of doubt as to any of the conditions or exceptions outlined in this Trademark Policy, please contact trademarks@percona.com for assistance and we will do our very best to be helpful.

Contact Us

For free technical help, visit the Percona Community Forum.

To report bugs or submit feature requests, open a JIRA ticket.

For paid support and managed or consulting services , contact Percona Sales.

June 28, 2023

June 28, 2023