EDB Pgpool-II

Version 4

1   EDB Pgpool-II

EDB Pgpool-II is an enhanced version of the community Pgpool product, intended to support additional EDB Postgres Advanced Server capabilities. You can use it with PostgreSQL and EDB Postgres Advanced Server. EDB Pgpool-II acts as middleware between client applications and a Postgres database server.

Using EDB Pgpool-II adds the following benefits to your application connection infrastructure:

- Transparently reroutes and ensures load balancing of read-only transactions to standby database servers
- Reuses connections to prevent reconnects
- Brings down Postgres connections by queuing stale connections
- Integrates with failover managers to follow the primary

EDB supports the following Pgpool-II functionality:

- Load balancing
- Connection pooling
- Replication
- High availability
- Connection limits
- Watchdog
- Limiting exceeding connections
- In-memory query cache

EDB Pgpool-II also provides extensions to EDB Postgres Advanced Server functionality. These extensions are packaged as modules in an extensions directory, and they provide additional features for EDB Postgres Advanced Server. These modules are generally not included in the core database. After they are loaded in a database, the extensions can function like built-in features. They allow you to use simple `SELECT` commands to use PCP remotely.

> **Note**
>
> EDB Pgpool-II extensions are delivered only for supported combinations of EDB Postgres Advanced Server versions and operating systems.

2   Release notes

The EDB Pgpool-II documentation describes the latest version of EDB Pgpool-II, including minor releases and patches. The release notes provide information on what was new in each release. For new functionality introduced in a minor or patch release, the content also indicates the release that introduced the feature.

Version	Release Date	Upstream merges
4.4.5	17 May 2024	Upstream 4.4.5
4.4.4	26 Feb 2024	Upstream 4.4.4
4.4.3	09 Nov 2023	Upstream 4.4.3
4.4.2	14 Feb 2023	Upstream 4.4.2
4.3.2, 4.2.9, 4.1.12, 4.0.19, 3.7.24	05 Jul 2022	Upstream 4.3.2, 4.2.9, 4.1.12, 4.0.19, 3.7.24
4.3.0	01 Feb 2022	Upstream 4.3.0
4.2.6	01 Dec 2021	Upstream 4.2.6
4.2.5	10 Jun 2021	Upstream 4.2.5 and 4.2.4

2.1   EDB PgPool-II 4.4.5 release notes

Released: 17 May 2024

EDB Pgpool-II 4.4.5 includes the following upstream merge:

Type	Description
Upstream merge	Merged with community Pgpool-II 4.4.5. See the community Release Notes for details.

2.2   EDB PgPool-II 4.4.4 release notes

Released: 26 Feb 2024

EDB Pgpool-II 4.4.4 includes the following upstream merge:

Type	Description
Upstream merge	Merged with community Pgpool-II 4.4.4. See the community Release Notes for details.
Bug fix	Fixed an issue for JDBC application, where it was throwing an error while doing prepare for call() statement with load balancing feature enabled.

2.3   EDB PgPool-II 4.4.3 release notes

Released: 09 Nov 2023

EDB Pgpool-II 4.4.3 includes the following upstream merge:

Type	Description
Upstream merge	Merged with community Pgpool-II 4.4.3. See the community Release Notes for details.
Enhancement	Added support for EDB Postgres Advanced Server 16.

2.4   EDB PgPool-II 4.4.2 release notes

Released: 14 Feb 2023

EDB Pgpool-II 4.4.2 includes the following upstream merge:

Type	Description

Type	Description
Upstream merge	Merged with community Pgpool-II 4.4.2. See the community Release Notes for details.
Enhancement	Support for EDB Postgres Advanced Server 15.

2.5   EDB PgPool-II 4.3.2 release notes

Released: 05 Jul 2022

EDB Pgpool-II 4.3.2 includes the following upstream merge:

Type	Description
Upstream merge	Merged with community Pgpool-II 4.3.2. See the community Release Notes for details.

2.6   EDB PgPool-II 4.3.0 release notes

Released: 01 Feb 2022

EDB Pgpool-II 4.3.0 includes the following upstream merge and updates:

Type	Description
Upstream merge	Merged with community Pgpool-II 4.3.0. See the community Release Notes for details.
Enhancement	Added EDB Postgres Advanced Server and PostgreSQL v14 support.
Enhancement	Removed the deprecated configuration parameter `pool_conn_dbname` .

2.7   EDB PgPool-II 4.2.9 release notes

Released: 05 Jul 2022

EDB Pgpool-II 4.2.9 includes the following upstream merge:

Type	Description
Upstream merge	Merged with community Pgpool-II 4.2.9. See the community Release Notes for details.

2.8   EDB PgPool-II 4.2.6 release notes

Released: 01 Dec 2021

EDB Pgpool-II 4.2.6 includes the following upstream merge and security fix:

Type	Description
Upstream merge	Merged with community Pgpool-II 4.2.6. See the community Release Notes for details.
Security fix	Reject extraneous data after SSL encryption handshake. In the server-side implementation of SSL negotiation, it was possible for a man-in-the-middle attacker to inject arbitrary SQL commands if it was configured to use cert authentication or hostssl + trust. This addresses PostgreSQL's CVE-2021-23214. In the client-side implementation of SSL negotiation, it was possible for a man-in-the-middle attacker to inject arbitrary responses if the database server is using trust authentication with a clientcert requirement. It is not possible with cert authentication because Pgpool-II does not implement the cert authentication between Pgpool-II and PostgreSQL. This addresses PostgreSQL's CVE-2021-23222.

> **Note**
>
> This security fix is also available in EDB Pgpool-II 4.1.9, 4.0.16, 3.7.21, and 3.6.28.

2.9   EDB PgPool-II 4.2.5 release notes

Released: 10 Jun 2021

EDB Pgpool-II 4.2.5 includes the following upstream merges and bug fixes:

Type	Description
Upstream merge	Merged with community Pgpool-II 4.2.5. See the community Release Notes for details.
Upstream merge	Merged with community Pgpool-II 4.2.4. See the community Release Notes for details.
Bug fix	Fixed maximum length of hostnames, including domain name. Now the correct value is 256 instead of 128.
Bug fix	Fixed query cache so that it does not cache SQLValueFunctions.
Bug fix	Implemented follow_primary command-locking over the watchdog channel. This is a supplementary fix for a race condition between detach_false_primary and follow_primary_command.
Bug fix	Fixed the issue wherein the client-side hangs when a NoData response follows a Describe statement.
Bug fix	Fixed an issue that sent an incorrect length error query (to abort transactions running on the non-main nodes), throwing an "invalid string in message" error at the backend in SI mode.
Bug fix	Fixed an issue wherein child processes created by follow_primary command were orphaned when pgpool was shut down.
Bug fix	Fixed an issue wherein, after executing the follow_primary command, the status of a standby node was set as down using the pcp_detach_node.
Bug fix	Fixed the watchdog communication race condition wherein the watchdog notified the new message to the main process, which took significant time to process the message.
Bug fix	Fixed an issue wherein the watchdog node status was not updating after rebooting one of the cluster nodes.
Bug fix	Fixed an issue wherein the follow_primary_command failed when watchdog was enabled.
Bug fix	Fixed a bug related to PGPOOL SHOW heartbeat and PGPOOL SHOW ALL commands where the last entry of command result was not shown.

Type	Description
Bug fix	Fixed an issue to allow the value of the parameter log_rotation_age to be set to 0 to disable it.
Bug fix	Fixed an incorrect display of the load balancing node status in raw clustering mode.
Bug fix	Fixed the pgpool logger process, which was utilizing 100% CPU.
Bug fix	Fixed the SI mode to acquire a snapshot with an internal transaction.
Bug fix	Fixed the backend_flag* parameter, which was shown twice while executing the PGPOOL SHOW ALL command.
Bug fix	Updated the other_pgpool parameter (removed in the 4.2 release) to watchdog in the SHOW ALL command.

2.10   EDB PgPool-II 4.1.12 release notes

Released: 05 Jul 2022

EDB Pgpool-II 4.1.12 includes the following upstream merge:

Type	Description
Upstream merge	Merged with community Pgpool-II 4.1.12. See the community Release Notes for details.

2.11   EDB PgPool-II 4.0.19 release notes

Released: 05 Jul 2022

EDB Pgpool-II 4.0.19 includes the following upstream merge:

Type	Description
Upstream merge	Merged with community Pgpool-II 4.0.19. See the community Release Notes for details.

2.12   EDB PgPool-II 3.7.24 release notes

Released: 05 Jul 2022

EDB Pgpool-II 3.7.24 includes the following upstream merge:

Type	Description
Upstream merge	Merged with community Pgpool-II 3.7.24. See the community Release Notes for details.

3   Supported platforms

EDB Pgpool-II is supported on the same Linux platforms as EDB Postgres Advanced Server. To determine the platform support for the EDB Pgpool-II, you can either refer to the platform support for EDB Postgres Advanced Server on the Platform Compatibility page on the EDB website or refer to Installing EDB Pgpool-II on Linux.

Supported database versions

This table lists the latest EDB Pgpool-II versions and their supported corresponding EDB Postgres Advanced Server (EPAS) versions.

EDB Pgpool-II	EPAS 16	EPAS 15	EPAS 14	EPAS 13	EPAS 12
4.4.x	Y	Y	Y	Y	Y
4.3.x	N	N	Y	Y	Y
4.2.x	N	N	N	Y	Y
4.1.x	N	N	N	Y	Y
4.0.x	N	N	N	N	Y
3.7.x	N	N	N	N	N

4   Installing EDB Pgpool-II on Linux

Select a link to access the applicable installation instructions:

Linux x86-64 (amd64)

Red Hat Enterprise Linux (RHEL) and derivatives

- RHEL 9, RHEL 8, RHEL 7

- Oracle Linux (OL) 9, Oracle Linux (OL) 8, Oracle Linux (OL) 7

- Rocky Linux 9, Rocky Linux 8

- AlmaLinux 9, AlmaLinux 8

- CentOS 7

SUSE Linux Enterprise (SLES)

- SLES 15, SLES 12

Debian and derivatives

- Ubuntu 22.04, Ubuntu 20.04

- Debian 11, Debian 10

Linux IBM Power (ppc64le)

Red Hat Enterprise Linux (RHEL) and derivatives

- RHEL 9, RHEL 8

SUSE Linux Enterprise (SLES)

- SLES 15, SLES 12

4.1   Installing EDB Pgpool-II on Linux x86 (amd64)

Operating system-specific install instructions are described in the corresponding documentation:

Red Hat Enterprise Linux (RHEL) and derivatives

- RHEL 9

- RHEL 8

- RHEL 7

- Oracle Linux (OL) 9

- Oracle Linux (OL) 8

- Oracle Linux (OL) 7

- Rocky Linux 9

- Rocky Linux 8

- AlmaLinux 9

- AlmaLinux 8

- CentOS 7

SUSE Linux Enterprise (SLES)

- SLES 15

- SLES 12

Debian and derivatives

- Ubuntu 22.04

- Ubuntu 20.04

- Debian 11

- Debian 10

4.1.1　Installing EDB Pgpool-II on RHEL 9 or OL 9 x86_64

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

    - Installing EDB Postgres Advanced Server

    - Installing PostgreSQL

- Set up the EDB repository.

    Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

    To determine if your repository exists, enter this command:

    ```
    dnf repolist | grep enterprisedb
    ```

    If no output is generated, the repository isn't installed.

    To set up the EDB repository:

    1. Go to EDB repositories.

    2. Select the button that provides access to the EDB repository.

    3. Select the platform and software that you want to download.

    4. Follow the instructions for setting up the EDB repository.

- Install the EPEL repository:

```
sudo dnf -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm
```

Install the package

```
sudo dnf -y install edb-pgpool<xx>
```

Where `<xx>` is the version of EDB PgPool-II you are installing. For example, if you are installing version 4.3, the package name would be `edb-pgpool43`.

4.1.2 Installing EDB Pgpool-II on RHEL 8 or OL 8 x86_64

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

  - [Installing EDB Postgres Advanced Server](#)

  - [Installing PostgreSQL](#)

- Set up the EDB repository.

  Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

  To determine if your repository exists, enter this command:

  ```
  dnf repolist | grep enterprisedb
  ```

  If no output is generated, the repository isn't installed.

  To set up the EDB repository:

  1. Go to [EDB repositories](#).

  2. Select the button that provides access to the EDB repository.

  3. Select the platform and software that you want to download.

  4. Follow the instructions for setting up the EDB repository.

- Install the EPEL repository:

  ```
  sudo dnf -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
  ```

Install the package

```
sudo dnf -y install edb-pgpool<xx>
```

Where `<xx>` is the version of EDB PgPool-II you are installing. For example, if you are installing version 4.3, the package name would be `edb-pgpool43`.

4.1.3   Installing EDB Pgpool-II on AlmaLinux 9 or Rocky Linux 9 x86_64

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

    - [Installing EDB Postgres Advanced Server](#)

    - [Installing PostgreSQL](#)

- Set up the EDB repository.

    Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

    To determine if your repository exists, enter this command:

    ```
    dnf repolist | grep enterprisedb
    ```

    If no output is generated, the repository isn't installed.

    To set up the EDB repository:

    1. Go to [EDB repositories](#).

    2. Select the button that provides access to the EDB repository.

    3. Select the platform and software that you want to download.

    4. Follow the instructions for setting up the EDB repository.

- Install the EPEL repository:

    ```
    sudo dnf -y install epel-release
    ```

- Enable additional repositories to resolve dependencies:

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

Install the package

```
sudo dnf -y install edb-pgpool<xx>
```

Where `<xx>` is the version of EDB PgPool-II you are installing. For example, if you are installing version 4.3, the package name would be `edb-pgpool43`.

4.1.4   Installing EDB Pgpool-II on AlmaLinux 8 or Rocky Linux 8 x86_64

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

    - [Installing EDB Postgres Advanced Server](#)

    - [Installing PostgreSQL](#)

- Set up the EDB repository.

    Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

    To determine if your repository exists, enter this command:

    ```
    dnf repolist | grep enterprisedb
    ```

    If no output is generated, the repository isn't installed.

    To set up the EDB repository:

    1. Go to [EDB repositories](#).

    2. Select the button that provides access to the EDB repository.

    3. Select the platform and software that you want to download.

    4. Follow the instructions for setting up the EDB repository.

- Install the EPEL repository:

    ```
    sudo dnf -y install epel-release
    ```

- Enable additional repositories to resolve dependencies:

    ```
    sudo dnf config-manager --set-enabled powertools
    ```

Install the package

```
sudo dnf -y install edb-pgpool<xx>
```

Where `<xx>` is the version of EDB PgPool-II you are installing. For example, if you are installing version 4.3, the package name would be `edb-pgpool43`.

4.1.5   Installing EDB Pgpool-II on RHEL 7 or OL 7 x86_64

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

    - [Installing EDB Postgres Advanced Server](#)

    - [Installing PostgreSQL](#)

- Set up the EDB repository.

    Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

    To determine if your repository exists, enter this command:

    ```
    dnf repolist | grep enterprisedb
    ```

    If no output is generated, the repository isn't installed.

    To set up the EDB repository:

    1. Go to [EDB repositories](#).

    2. Select the button that provides access to the EDB repository.

    3. Select the platform and software that you want to download.

    4. Follow the instructions for setting up the EDB repository.

- Install the EPEL repository:

    ```
    sudo yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
    ```

- Enable additional repositories to resolve dependencies:

    ```
    subscription-manager repos --enable "rhel-*-optional-rpms" --enable "rhel-*-extras-rpms"  --enable
    "rhel-ha-for-rhel-*-server-rpms"
    ```

Install the package

```
sudo yum -y install edb-pgpool<xx>
```

Where `<xx>` is the version of EDB PgPool-II you are installing. For example, if you are installing version 4.3, the package name would be `edb-pgpool43`.

4.1.6 Installing EDB Pgpool-II on CentOS 7 x86_64

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

  - [Installing EDB Postgres Advanced Server](#)

  - [Installing PostgreSQL](#)

- Set up the EDB repository.

  Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

  To determine if your repository exists, enter this command:

  ```
  dnf repolist | grep enterprisedb
  ```

  If no output is generated, the repository isn't installed.

  To set up the EDB repository:

  1. Go to [EDB repositories](#).

  2. Select the button that provides access to the EDB repository.

  3. Select the platform and software that you want to download.

  4. Follow the instructions for setting up the EDB repository.

- Install the EPEL repository:

  ```
  sudo yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
  ```

Install the package

```
sudo yum -y install edb-pgpool<xx>
```

Where `<xx>` is the version of EDB PgPool-II you are installing. For example, if you are installing version 4.3, the package name would be `edb-pgpool43`.

4.1.7 Installing EDB Pgpool-II on SLES 15 x86_64

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

    - [Installing EDB Postgres Advanced Server](#)

    - [Installing PostgreSQL](#)

- Set up the EDB repository.

    Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

    To determine if your repository exists, enter this command:

    ```
    zypper lr -E | grep enterprisedb
    ```

    If no output is generated, the repository isn't installed.

    To set up the EDB repository:

    1. Go to [EDB repositories](#).

    2. Select the button that provides access to the EDB repository.

    3. Select the platform and software that you want to download.

    4. Follow the instructions for setting up the EDB repository.

- Activate the required SUSE module:

    ```
    sudo SUSEConnect -p PackageHub/15.4/x86_64
    ```

- Refresh the metadata:

    ```
    sudo zypper refresh
    ```

Install the package

```
sudo zypper -n install edb-pgpool<xx>
```

Where `<xx>` is the version of EDB PgPool-II you are installing. For example, if you are installing version 4.3, the package name would be `edb-pgpool43`.

4.1.8   Installing EDB Pgpool-II on SLES 12 x86_64

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

    - Installing EDB Postgres Advanced Server

    - Installing PostgreSQL

- Set up the EDB repository.

    Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

    To determine if your repository exists, enter this command:

    ```
    zypper lr -E | grep enterprisedb
    ```

    If no output is generated, the repository isn't installed.

    To set up the EDB repository:

    1. Go to EDB repositories.

    2. Select the button that provides access to the EDB repository.

    3. Select the platform and software that you want to download.

    4. Follow the instructions for setting up the EDB repository.

- Activate the required SUSE module:

    ```
    sudo SUSEConnect -p PackageHub/12.5/x86_64
    sudo SUSEConnect -p sle-sdk/12.5/x86_64
    ```

- Refresh the metadata:

    ```
    sudo zypper refresh
    ```

Install the package

```
sudo zypper -n install edb-pgpool<xx>
```

Where `<xx>` is the version of EDB PgPool-II you are installing. For example, if you are installing version 4.3, the package name would be `edb-pgpool43`.

4.1.9   Installing EDB Pgpool-II on Ubuntu 22.04 x86_64

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

    - [Installing EDB Postgres Advanced Server](#)

    - [Installing PostgreSQL](#)

- Set up the EDB repository.

    Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

    To determine if your repository exists, enter this command:

    ```
    apt-cache search enterprisedb
    ```

    If no output is generated, the repository isn't installed.

    To set up the EDB repository:

    1. Go to [EDB repositories](#).

    2. Select the button that provides access to the EDB repository.

    3. Select the platform and software that you want to download.

    4. Follow the instructions for setting up the EDB repository.

Install the package

```
sudo apt-get -y install edb-pgpool<xx>
```

Where `<xx>` is the version of EDB PgPool-II you are installing. For example, if you are installing version 4.3, the package name would be `edb-pgpool43`.

4.1.10   Installing EDB Pgpool-II on Ubuntu 20.04 x86_64

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

  - Installing EDB Postgres Advanced Server

  - Installing PostgreSQL

- Set up the EDB repository.

  Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

  To determine if your repository exists, enter this command:

  ```
  apt-cache search enterprisedb
  ```

  If no output is generated, the repository isn't installed.

  To set up the EDB repository:

  1. Go to EDB repositories.

  2. Select the button that provides access to the EDB repository.

  3. Select the platform and software that you want to download.

  4. Follow the instructions for setting up the EDB repository.

Install the package

```
sudo apt-get -y install edb-pgpool<xx>
```

Where `<xx>` is the version of EDB PgPool-II you are installing. For example, if you are installing version 4.3, the package name would be `edb-pgpool43`.

4.1.11   Installing EDB Pgpool-II on Debian 11 x86_64

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

  - [Installing EDB Postgres Advanced Server](#)

  - [Installing PostgreSQL](#)

- Set up the EDB repository.

  Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

  To determine if your repository exists, enter this command:

  ```
  apt-cache search enterprisedb
  ```

  If no output is generated, the repository isn't installed.

  To set up the EDB repository:

  1. Go to [EDB repositories](#).

  2. Select the button that provides access to the EDB repository.

  3. Select the platform and software that you want to download.

  4. Follow the instructions for setting up the EDB repository.

Install the package

```
sudo apt-get -y install edb-pgpool<xx>
```

Where `<xx>` is the version of EDB PgPool-II you are installing. For example, if you are installing version 4.3, the package name would be `edb-pgpool43`.

4.1.12   Installing EDB Pgpool-II on Debian 10 x86_64

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

  - [Installing EDB Postgres Advanced Server](#)

  - [Installing PostgreSQL](#)

- Set up the EDB repository.

Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

To determine if your repository exists, enter this command:

```
apt-cache search enterprisedb
```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to EDB repositories.

2. Select the button that provides access to the EDB repository.

3. Select the platform and software that you want to download.

4. Follow the instructions for setting up the EDB repository.

Install the package

```
sudo apt-get -y install edb-pgpool<xx>
```

Where `<xx>` is the version of EDB PgPool-II you are installing. For example, if you are installing version 4.3, the package name would be `edb-pgpool43` .

4.2   Installing EDB Pgpool-II on Linux IBM Power (ppc64le)

Operating system-specific install instructions are described in the corresponding documentation:

Red Hat Enterprise Linux (RHEL)

- RHEL 9

- RHEL 8

SUSE Linux Enterprise (SLES)

- SLES 15

- SLES 12

4.2.1   Installing EDB Pgpool-II on RHEL 9 ppc64le

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

    - [Installing EDB Postgres Advanced Server](#)

    - [Installing PostgreSQL](#)

- Set up the EDB repository.

    Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

    To determine if your repository exists, enter this command:

    ```
    dnf repolist | grep enterprisedb
    ```

    If no output is generated, the repository isn't installed.

    To set up the EDB repository:

    1. Go to [EDB repositories](#).

    2. Select the button that provides access to the EDB repository.

    3. Select the platform and software that you want to download.

    4. Follow the instructions for setting up the EDB repository.

- Install the EPEL repository:

    ```
    sudo dnf -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm
    ```

- Refresh the cache:

    ```
    sudo dnf makecache
    ```

Install the package

```
sudo dnf -y install edb-pgpool<xx>
```

Where `<xx>` is the version of EDB PgPool-II you are installing. For example, if you are installing version 4.3, the package name would be `edb-pgpool43`.

4.2.2   Installing EDB Pgpool-II on RHEL 8 ppc64le

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

    - [Installing EDB Postgres Advanced Server](#)

    - [Installing PostgreSQL](#)

- Set up the EDB repository.

    Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

    To determine if your repository exists, enter this command:

    ```
    dnf repolist | grep enterprisedb
    ```

    If no output is generated, the repository isn't installed.

    To set up the EDB repository:

    1. Go to [EDB repositories](#).

    2. Select the button that provides access to the EDB repository.

    3. Select the platform and software that you want to download.

    4. Follow the instructions for setting up the EDB repository.

- Install the EPEL repository:

    ```
    sudo dnf -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
    ```

- Refresh the cache:

    ```
    sudo dnf makecache
    ```

Install the package

```
sudo dnf -y install edb-pgpool<xx>
```

Where `<xx>` is the version of EDB PgPool-II you are installing. For example, if you are installing version 4.3, the package name would be `edb-pgpool43`.

4.2.3   Installing EDB Pgpool-II on SLES 15 ppc64le

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

    - [Installing EDB Postgres Advanced Server](#)

    - [Installing PostgreSQL](#)

- Set up the EDB repository.

    Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

    To determine if your repository exists, enter this command:

    ```
    zypper lr -E | grep enterprisedb
    ```

    If no output is generated, the repository isn't installed.

    To set up the EDB repository:

    1. Go to [EDB repositories](#).

    2. Select the button that provides access to the EDB repository.

    3. Select the platform and software that you want to download.

    4. Follow the instructions for setting up the EDB repository.

- Activate the required SUSE module:

    ```
    sudo SUSEConnect -p PackageHub/15.4/ppc64le
    ```

- Refresh the metadata:

    ```
    sudo zypper refresh
    ```

Install the package

```
sudo zypper -n install edb-pgpool<xx>
```

Where `<xx>` is the version of EDB PgPool-II you are installing. For example, if you are installing version 4.3, the package name would be `edb-pgpool43`.

4.2.4   Installing EDB Pgpool-II on SLES 12 ppc64le

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

    - [Installing EDB Postgres Advanced Server](#)

    - [Installing PostgreSQL](#)

- Set up the EDB repository.

    Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

    To determine if your repository exists, enter this command:

    ```
    zypper lr -E | grep enterprisedb
    ```

    If no output is generated, the repository isn't installed.

    To set up the EDB repository:

    1. Go to [EDB repositories](#).

    2. Select the button that provides access to the EDB repository.

    3. Select the platform and software that you want to download.

    4. Follow the instructions for setting up the EDB repository.

- Activate the required SUSE module:

    ```
    sudo SUSEConnect -p PackageHub/12.5/ppc64le
    sudo SUSEConnect -p sle-sdk/12.5/ppc64le
    ```

- Refresh the metadata:

    ```
    sudo zypper refresh
    ```

Install the package

```
sudo zypper -n install edb-pgpool<xx>
```

Where `<xx>` is the version of EDB PgPool-II you are installing. For example, if you are installing version 4.3, the package name would be `edb-pgpool43`.

4.3   Upgrading EDB Pgpool-II

You can perform an EDB Pgpool-II minor version upgrade, for example, to upgrade from 4.2.5 to 4.2.6.

For information about upgrading extensions, see Upgrading EDB Pgpool-II extensions.

> **Note**
>
> Version 4.3 is a major release. For more details on migrating from a version earlier than 4.3, see Migration.

To upgrade:

```
sudo <package-manager> upgrade edb-pgpool<xx>
```

Where:

- `<package-manager>` is the package manager used with your operating system.

Package manager	Operating system
dnf	RHEL 8/9 and derivatives
yum	RHEL 7 and derivatives, CentOS 7
zypper	SLES
apt-get	Debian and Ubuntu

- `<xx>` is the EDB Pgpool-II version you want to upgrade.

For example, to upgrade from EDB Pgpool-II 4.2.5 to 4.2.6 on RHEL 9:

```
sudo dnf upgrade edb-pgpool42
```

4.4   Uninstalling EDB Pgpool-II

For information about uninstallling extensions, see Uninstalling EDB Pgpool-II extensions.

Uninstalling EDB Pgpool-II on an RHEL//Rocky Linux/AlmaLinux host

To uninstall EDB Pgpool-II, assume the identity of the root user and invoke the appropriate command.

On RHEL/CentOS 7:

```
yum erase edb-pgpool<xx>
```

On RHEL/Rocky Linux/AlmaLinux 8:

```
dnf erase edb-pgpool<xx>
```

Where `<xx>` is the EDB Pgpool-II version.

Uninstalling EDB Pgpool-II on a Debian/Ubuntu host

To uninstall EDB Pgpool-II on a Debian/Ubuntu host:

```
apt-get remove -y edb-pgpool<xx>
```

Where `<xx>` is the EDB Pgpool-II version you want to uninstall.

Uninstalling EDB Pgpool-II on a SLES 12 host

To uninstall EDB Pgpool-II on a SLES host, assume the identity of the root user and invoke:

```
zypper remove edb-pgpool<xx>
```

Where `<xx>` is the EDB Pgpool-II version you want to uninstall.

Uninstalling EDB Pgpool-II Linux uninstaller

The EDB Pgpool-II graphical installer creates an uninstaller in the installation directory. If you used the default installation directory `/opt/edb` , then the uninstaller is in `/opt/edb/pgpool<x.y>` , where `<x.y>` is the EDB Pgpool-II version you installed.

1. Navigate into the directory that contains the uninstaller and assume superuser privileges. Open the uninstaller and select **Yes** to begin uninstalling EDB Pgpool-II.

   The uninstallation process begins.

2. Select **OK** when the uninstallation completes.

5   Installing EDB Pgpool-II Extensions on Linux

Select a link to access the applicable installation instructions:

Linux x86-64 (amd64)

Red Hat Enterprise Linux (RHEL) and derivatives

- RHEL 9, RHEL 8, RHEL 7

- Oracle Linux (OL) 9, Oracle Linux (OL) 8, Oracle Linux (OL) 7

- Rocky Linux 9, Rocky Linux 8

- AlmaLinux 9, AlmaLinux 8

- CentOS 7

SUSE Linux Enterprise (SLES)

- SLES 15, SLES 12

Debian and derivatives

- Ubuntu 22.04, Ubuntu 20.04

- Debian 11, Debian 10

Linux IBM Power (ppc64le)

Red Hat Enterprise Linux (RHEL) and derivatives

- RHEL 9, RHEL 8

SUSE Linux Enterprise (SLES)

- SLES 15, SLES 12

5.1 Installing EDB Pgpool-II Extensions on Linux x86 (amd64)

Operating system-specific install instructions are described in the corresponding documentation:

Red Hat Enterprise Linux (RHEL) and derivatives

- RHEL 9

- RHEL 8

- RHEL 7

- Oracle Linux (OL) 9

- Oracle Linux (OL) 8

- Oracle Linux (OL) 7

- Rocky Linux 9

- Rocky Linux 8

- AlmaLinux 9

- AlmaLinux 8

- CentOS 7

SUSE Linux Enterprise (SLES)

- SLES 15

- SLES 12

Debian and derivatives

- Ubuntu 22.04

- Ubuntu 20.04

- Debian 11

- Debian 10

5.1.1   Installing EDB Pgpool-II Extensions on RHEL 9 or OL 9 x86_64

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

  - Installing EDB Postgres Advanced Server

  - Installing PostgreSQL

- Set up the EDB repository.

  Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

To determine if your repository exists, enter this command:

```
dnf repolist | grep enterprisedb
```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to EDB repositories.

2. Select the button that provides access to the EDB repository.

3. Select the platform and software that you want to download.

4. Follow the instructions for setting up the EDB repository.

- Install the EPEL repository:

```
sudo dnf -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm
```

Install the package

```
sudo dnf -y install edb-as<xx>-pgpool<yy>-extensions
```

Where `<xx>` is the EDB Postgres Advanced Server version and `<yy>` is the EDB Pgpool-II version you are installing. For example, if you are installing EDB Pgpool-II version 4.4 and EDB Postgres Advanced Server version 15, the package name would be `edb-as15-pgpool44-extensions`.

5.1.2   Installing EDB Pgpool-II Extensions on RHEL 8 or OL 8 x86_64

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

  - Installing EDB Postgres Advanced Server

  - Installing PostgreSQL

- Set up the EDB repository.

  Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

  To determine if your repository exists, enter this command:

```
dnf repolist | grep enterprisedb
```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to EDB repositories.

2. Select the button that provides access to the EDB repository.

3. Select the platform and software that you want to download.

4. Follow the instructions for setting up the EDB repository.

- Install the EPEL repository:

```
sudo dnf -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

Install the package

```
sudo dnf -y install edb-as<xx>-pgpool<yy>-extensions
```

Where `<xx>` is the EDB Postgres Advanced Server version and `<yy>` is the EDB Pgpool-II version you are installing. For example, if you are installing EDB Pgpool-II version 4.4 and EDB Postgres Advanced Server version 15, the package name would be `edb-as15-pgpool44-extensions`.

5.1.3   Installing EDB Pgpool-II Extensions on AlmaLinux 9 or Rocky Linux 9 x86_64

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

  - Installing EDB Postgres Advanced Server

  - Installing PostgreSQL

- Set up the EDB repository.

  Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

  To determine if your repository exists, enter this command:

  ```
  dnf repolist | grep enterprisedb
  ```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to EDB repositories.

2. Select the button that provides access to the EDB repository.

3. Select the platform and software that you want to download.

4. Follow the instructions for setting up the EDB repository.

- Install the EPEL repository:

```
sudo dnf -y install epel-release
```

- Enable additional repositories to resolve dependencies:

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

Install the package

```
sudo dnf -y install edb-as<xx>-pgpool<yy>-extensions
```

Where `<xx>` is the EDB Postgres Advanced Server version and `<yy>` is the EDB Pgpool-II version you are installing. For example, if you are installing EDB Pgpool-II version 4.4 and EDB Postgres Advanced Server version 15, the package name would be `edb-as15-pgpool44-extensions`.

5.1.4   Installing EDB Pgpool-II Extensions on AlmaLinux 8 or Rocky Linux 8 x86_64

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

  - Installing EDB Postgres Advanced Server

  - Installing PostgreSQL

- Set up the EDB repository.

  Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

  To determine if your repository exists, enter this command:

```
dnf repolist | grep enterprisedb
```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to EDB repositories.

2. Select the button that provides access to the EDB repository.

3. Select the platform and software that you want to download.

4. Follow the instructions for setting up the EDB repository.

- Install the EPEL repository:

```
sudo dnf -y install epel-release
```

- Enable additional repositories to resolve dependencies:

```
sudo dnf config-manager --set-enabled powertools
```

Install the package

```
sudo dnf -y install edb-as<xx>-pgpool<yy>-extensions
```

Where `<xx>` is the EDB Postgres Advanced Server version and `<yy>` is the EDB Pgpool-II version you are installing. For example, if you are installing EDB Pgpool-II version 4.4 and EDB Postgres Advanced Server version 15, the package name would be `edb-as15-pgpool44-extensions`.

5.1.5   Installing EDB Pgpool-II Extensions on RHEL 7 or OL 7 x86_64

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

  - Installing EDB Postgres Advanced Server

  - Installing PostgreSQL

- Set up the EDB repository.

  Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

To determine if your repository exists, enter this command:

```
dnf repolist | grep enterprisedb
```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to EDB repositories.

2. Select the button that provides access to the EDB repository.

3. Select the platform and software that you want to download.

4. Follow the instructions for setting up the EDB repository.

- Install the EPEL repository:

```
sudo yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

- Enable additional repositories to resolve dependencies:

```
subscription-manager repos --enable "rhel-*-optional-rpms" --enable "rhel-*-extras-rpms"  --enable
"rhel-ha-for-rhel-*-server-rpms"
```

Install the package

```
sudo yum -y install edb-as<xx>-pgpool<yy>-extensions
```

Where `<xx>` is the EDB Postgres Advanced Server version and `<yy>` is the EDB Pgpool-II version you are installing. For example, if you are installing EDB Pgpool-II version 4.4 and EDB Postgres Advanced Server version 15, the package name would be `edb-as15-pgpool44-extensions`.

5.1.6 Installing EDB Pgpool-II Extensions on CentOS 7 x86_64

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

  - Installing EDB Postgres Advanced Server

  - Installing PostgreSQL

- Set up the EDB repository.

Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

To determine if your repository exists, enter this command:

```
dnf repolist | grep enterprisedb
```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to EDB repositories.

2. Select the button that provides access to the EDB repository.

3. Select the platform and software that you want to download.

4. Follow the instructions for setting up the EDB repository.

- Install the EPEL repository:

```
sudo yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

Install the package

```
sudo yum -y install edb-as<xx>-pgpool<yy>-extensions
```

Where `<xx>` is the EDB Postgres Advanced Server version and `<yy>` is the EDB Pgpool-II version you are installing. For example, if you are installing EDB Pgpool-II version 4.4 and EDB Postgres Advanced Server version 15, the package name would be `edb-as15-pgpool44-extensions`.

5.1.7   Installing EDB Pgpool-II Extensions on SLES 15 x86_64

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

  - Installing EDB Postgres Advanced Server

  - Installing PostgreSQL

- Set up the EDB repository.

  Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

To determine if your repository exists, enter this command:

```
zypper lr -E | grep enterprisedb
```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to EDB repositories.

2. Select the button that provides access to the EDB repository.

3. Select the platform and software that you want to download.

4. Follow the instructions for setting up the EDB repository.

- Activate the required SUSE module:

```
sudo SUSEConnect -p PackageHub/15.4/x86_64
```

- Refresh the metadata:

```
sudo zypper refresh
```

Install the package

```
sudo zypper -n install edb-as<xx>-pgpool<yy>-extensions
```

Where `<xx>` is the EDB Postgres Advanced Server version and `<yy>` is the EDB Pgpool-II version you are installing. For example, if you are installing EDB Pgpool-II version 4.4 and EDB Postgres Advanced Server version 15, the package name would be `edb-as15-pgpool44-extensions`.

5.1.8   Installing EDB Pgpool-II Extensions on SLES 12 x86_64

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

  - Installing EDB Postgres Advanced Server

  - Installing PostgreSQL

- Set up the EDB repository.

Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

To determine if your repository exists, enter this command:

```
zypper lr -E | grep enterprisedb
```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to EDB repositories.

2. Select the button that provides access to the EDB repository.

3. Select the platform and software that you want to download.

4. Follow the instructions for setting up the EDB repository.

- Activate the required SUSE module:

```
sudo SUSEConnect -p PackageHub/12.5/x86_64
sudo SUSEConnect -p sle-sdk/12.5/x86_64
```

- Refresh the metadata:

```
sudo zypper refresh
```

Install the package

```
sudo zypper -n install edb-as<xx>-pgpool<yy>-extensions
```

Where `<xx>` is the EDB Postgres Advanced Server version and `<yy>` is the EDB Pgpool-II version you are installing. For example, if you are installing EDB Pgpool-II version 4.4 and EDB Postgres Advanced Server version 15, the package name would be `edb-as15-pgpool44-extensions`.

5.1.9   Installing EDB Pgpool-II Extensions on Ubuntu 22.04 x86_64

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

    - Installing EDB Postgres Advanced Server

    - Installing PostgreSQL

- Set up the EDB repository.

  Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

  To determine if your repository exists, enter this command:

  ```
  apt-cache search enterprisedb
  ```

  If no output is generated, the repository isn't installed.

  To set up the EDB repository:

  1. Go to EDB repositories.

  2. Select the button that provides access to the EDB repository.

  3. Select the platform and software that you want to download.

  4. Follow the instructions for setting up the EDB repository.

Install the package

```
sudo apt-get -y install edb-as<xx>-pgpool<yy>-extensions
```

Where `<xx>` is the EDB Postgres Advanced Server version and `<yy>` is the EDB Pgpool-II version you are installing. For example, if you are installing EDB Pgpool-II version 4.4 and EDB Postgres Advanced Server version 15, the package name would be `edb-as15-pgpool44-extensions`.

5.1.10   Installing EDB Pgpool-II Extensions on Ubuntu 20.04 x86_64

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

  - Installing EDB Postgres Advanced Server

  - Installing PostgreSQL

- Set up the EDB repository.

  Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

  To determine if your repository exists, enter this command:

```
apt-cache search enterprisedb
```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to EDB repositories.

2. Select the button that provides access to the EDB repository.

3. Select the platform and software that you want to download.

4. Follow the instructions for setting up the EDB repository.

Install the package

```
sudo apt-get -y install edb-as<xx>-pgpool<yy>-extensions
```

Where `<xx>` is the EDB Postgres Advanced Server version and `<yy>` is the EDB Pgpool-II version you are installing. For example, if you are installing EDB Pgpool-II version 4.4 and EDB Postgres Advanced Server version 15, the package name would be `edb-as15-pgpool44-extensions`.

5.1.11 Installing EDB Pgpool-II Extensions on Debian 11 x86_64

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

  - Installing EDB Postgres Advanced Server

  - Installing PostgreSQL

- Set up the EDB repository.

  Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

  To determine if your repository exists, enter this command:

  ```
  apt-cache search enterprisedb
  ```

  If no output is generated, the repository isn't installed.

  To set up the EDB repository:

1. Go to EDB repositories.

2. Select the button that provides access to the EDB repository.

3. Select the platform and software that you want to download.

4. Follow the instructions for setting up the EDB repository.

Install the package

```
sudo apt-get -y install edb-as<xx>-pgpool<yy>-extensions
```

Where `<xx>` is the EDB Postgres Advanced Server version and `<yy>` is the EDB Pgpool-II version you are installing. For example, if you are installing EDB Pgpool-II version 4.4 and EDB Postgres Advanced Server version 15, the package name would be `edb-as15-pgpool44-extensions`.

5.1.12 Installing EDB Pgpool-II Extensions on Debian 10 x86_64

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

    - Installing EDB Postgres Advanced Server

    - Installing PostgreSQL

- Set up the EDB repository.

  Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

  To determine if your repository exists, enter this command:

  ```
  apt-cache search enterprisedb
  ```

  If no output is generated, the repository isn't installed.

  To set up the EDB repository:

    1. Go to EDB repositories.

    2. Select the button that provides access to the EDB repository.

    3. Select the platform and software that you want to download.

    4. Follow the instructions for setting up the EDB repository.

Install the package

```
sudo apt-get -y install edb-as<xx>-pgpool<yy>-extensions
```

Where `<xx>` is the EDB Postgres Advanced Server version and `<yy>` is the EDB Pgpool-II version you are installing. For example, if you are installing EDB Pgpool-II version 4.4 and EDB Postgres Advanced Server version 15, the package name would be `edb-as15-pgpool44-extensions`.

5.2 Installing EDB Pgpool-II Extensions on Linux IBM Power (ppc64le)

Operating system-specific install instructions are described in the corresponding documentation:

Red Hat Enterprise Linux (RHEL)

- RHEL 9

- RHEL 8

SUSE Linux Enterprise (SLES)

- SLES 15

- SLES 12

5.2.1 Installing EDB Pgpool-II Extensions on RHEL 9 ppc64le

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

    - Installing EDB Postgres Advanced Server

    - Installing PostgreSQL

- Set up the EDB repository.

    Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

    To determine if your repository exists, enter this command:

    ```
    dnf repolist | grep enterprisedb
    ```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to EDB repositories.

2. Select the button that provides access to the EDB repository.

3. Select the platform and software that you want to download.

4. Follow the instructions for setting up the EDB repository.

- Install the EPEL repository:

```
sudo dnf -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm
```

- Refresh the cache:

```
sudo dnf makecache
```

Install the package

```
sudo dnf -y install edb-as<xx>-pgpool<yy>-extensions
```

Where `<xx>` is the EDB Postgres Advanced Server version and `<yy>` is the EDB Pgpool-II version you are installing. For example, if you are installing EDB Pgpool-II version 4.4 and EDB Postgres Advanced Server version 15, the package name would be `edb-as15-pgpool44-extensions`.

5.2.2   Installing EDB Pgpool-II Extensions on RHEL 8 ppc64le

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

  - Installing EDB Postgres Advanced Server

  - Installing PostgreSQL

- Set up the EDB repository.

  Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

  To determine if your repository exists, enter this command:

```
dnf repolist | grep enterprisedb
```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to EDB repositories.

2. Select the button that provides access to the EDB repository.

3. Select the platform and software that you want to download.

4. Follow the instructions for setting up the EDB repository.

- Install the EPEL repository:

```
sudo dnf -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

- Refresh the cache:

```
sudo dnf makecache
```

Install the package

```
sudo dnf -y install edb-as<xx>-pgpool<yy>-extensions
```

Where `<xx>` is the EDB Postgres Advanced Server version and `<yy>` is the EDB Pgpool-II version you are installing. For example, if you are installing EDB Pgpool-II version 4.4 and EDB Postgres Advanced Server version 15, the package name would be `edb-as15-pgpool44-extensions`.

5.2.3   Installing EDB Pgpool-II Extensions on SLES 15 ppc64le

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

  - Installing EDB Postgres Advanced Server

  - Installing PostgreSQL

- Set up the EDB repository.

  Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

To determine if your repository exists, enter this command:

```
zypper lr -E | grep enterprisedb
```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to EDB repositories.

2. Select the button that provides access to the EDB repository.

3. Select the platform and software that you want to download.

4. Follow the instructions for setting up the EDB repository.

- Activate the required SUSE module:

```
sudo SUSEConnect -p PackageHub/15.4/ppc64le
```

- Refresh the metadata:

```
sudo zypper refresh
```

Install the package

```
sudo zypper -n install edb-as<xx>-pgpool<yy>-extensions
```

Where `<xx>` is the EDB Postgres Advanced Server version and `<yy>` is the EDB Pgpool-II version you are installing. For example, if you are installing EDB Pgpool-II version 4.4 and EDB Postgres Advanced Server version 15, the package name would be `edb-as15-pgpool44-extensions`.

5.2.4   Installing EDB Pgpool-II Extensions on SLES 12 ppc64le

Prerequisites

Before you begin the installation process:

- Install Postgres on the same host. See:

  - Installing EDB Postgres Advanced Server

  - Installing PostgreSQL

- Set up the EDB repository.

Setting up the repository is a one-time task. If you have already set up your repository, you don't need to perform this step.

To determine if your repository exists, enter this command:

```
zypper lr -E | grep enterprisedb
```

If no output is generated, the repository isn't installed.

To set up the EDB repository:

1. Go to EDB repositories.

2. Select the button that provides access to the EDB repository.

3. Select the platform and software that you want to download.

4. Follow the instructions for setting up the EDB repository.

- Activate the required SUSE module:

```
sudo SUSEConnect -p PackageHub/12.5/ppc64le
sudo SUSEConnect -p sle-sdk/12.5/ppc64le
```

- Refresh the metadata:

```
sudo zypper refresh
```

Install the package

```
sudo zypper -n install edb-as<xx>-pgpool<yy>-extensions
```

Where `<xx>` is the EDB Postgres Advanced Server version and `<yy>` is the EDB Pgpool-II version you are installing. For example, if you are installing EDB Pgpool-II version 4.4 and EDB Postgres Advanced Server version 15, the package name would be `edb-as15-pgpool44-extensions`.

5.3   Creating Pgpool-II extensions

You must install and create the extensions in each database where you plan to use EDB Pgpool-II. To ensure all extensions are available for future databases, you can add the extension to the `template1` database. Any extensions installed in the `template1` database are created in each of the databases that uses `template1` as a template during creation.

Pgpool_adm extension

`Pgpool_adm` is a set of extensions that allows SQL access to PCP commands. For information, see PCP commands.

After installing the `Pgpool_adm` extension, use the psql client application to connect to the database, and execute the following SQL command:

```
CREATE EXTENSION
pgpool_adm;
```

For more information about `Pgpool_adm` , see Pgpool_adm docs.

Pgpool_recovery extension

The `Pgpool_recovery` extension is required for online recovery and future fail-back mechanisms.

After installing the `Pgpool_recovery` extension, use psql to connect to the database. Create a `Pgpool_recovery` extension:

```
CREATE EXTENSION
pgpool_recovery;
```

For more information about using the `CREATE EXTENSION` command, see the PostgreSQL core documentation.

5.4 Upgrading EDB Pgpool-II extensions

You can upgrade the version of EDB PgPool-II extensions, for example, from 4.2.5 to 4.2.6.

For information about upgrading PgPool, see Upgrading EDB Pgpool-II.

> Note
>
> 4.3 is a major release. For more details on migrating from earlier versions to version 4.3, see Migration.

To upgrade PgPool-II extensions:

```
sudo <package-manager> upgrade edb-as<xx>-pgpool<yy>-extensions
```

Where:

- `<package-manager>` is the package manager used with your operating system.

Package manager	Operating system
dnf	RHEL 8/9 and derivatives
yum	RHEL 7 and derivatives, CentOS 7
zypper	SLES
apt-get	Debian and Ubuntu

- `<xx>` is the EDB Postgres Advanced Server version, and `<yy>` is the EDB Pgpool-II extension version.

> Note

Only minor version upgrade is supported. For example, you can upgrade from 3.6.20 to 3.6.21 extension but not to 3.7.14.

5.5   Uninstalling EDB Pgpool-II extensions

Uninstalling EDB Pgpool-II extensions on an RHEL/CentOS/Rocky Linux/AlmaLinux host

To remove extensions from the server, execute the appropriate command.

On RHEL/CentOS 7:

```
yum erase edb-as<xx>-pgpool<yy>-extensions
```

On RHEL/Rocky Linux/AlmaLinux 8:

```
dnf erase edb-as<xx>-pgpool<yy>-extensions
```

Where `<xx>` is the EDB Postgres Advanced Server version, and `<yy>` is the EDB Pgpool-II extension version.

Uninstalling EDB Pgpool-II extensions on a SLES 12 host

To uninstall EDB Pgpool-II extensions on a SLES host, assume the identity of the root user and invoke:

```
zypper remove edb-as<xx>-pgpool<yy>-extensions
```

Where `<xx>` is the EDB Postgres Advanced Server version, and `<yy>` is the EDB Pgpool-II extension version you want to uninstall.

Uninstalling EDB Pgpool-II extensions on a Debian/Ubuntu host

To uninstall EDB Pgpool-II extensions on a Debian/Ubuntu host:

```
apt-get remove -y edb-as<xx>-pgpool<yy>-extensions
```

Where `<xx>` is the EDB Postgres Advanced Server version, and `<yy>` is the EDB Pgpool-II extension version you want to uninstall.

Uninstalling EDB Pgpool-II extensions Linux graphical uninstaller

The EDB Pgpool-II extensions graphical installer creates an uninstaller in the installation directory. If you used the default installation directory `/opt/edb/as<xx>`, then the uninstaller is in the `/opt/edb/as<xx>`, where `<xx>` is the EDB Postgres Advanced Server version you installed.

1. Navigate into the directory that contains the uninstaller and assume superuser privileges. Open the uninstaller and select **Yes** to begin

uninstalling EDB Pgpool-II extensions. The uninstallation process begins.

2. Select **OK** when the uninstallation completes.

6   Configuring EDB Pgpool-II

The configuration files are created in the `/etc/sysconfig/edb/pgpool<x.y>` directory, where *x.y* is the EDB Pgpool-II release version. By default, `.sample` is appended to the configuration file name. remove `.sample` from the configuration file after copying the file to create your custom configuration.

> **Note**
>
> The configuration options for Pgpool-II are extensive. Consider the options listed here as a starting point. For more information about configuring and using Pgpool-II, consult the [project website](#).

Commonly used parameters

The table lists `pgpool.conf` parameters to use when implementing connection pooling:

Parameter name	Description
`listen_addresses`	Hostname or IP address used by EDB Pgpool-II to listen for connections. The default is `localhost`. Change to '*' for all addresses.
`port`	Port for EDB Pgpool-II connections. The default is `9999`.
`pcp_port`	Port for PCP connections. The default is `9898`.
`backend_hostname0`	Hostname or IP address for backend 0. You can specify '' if the backend and EDB Pgpool-II are running on the same host.
`backend_port0`	Port number for backend 0.
`backend_weight0`	Weight for backend 0 (only in load balancing mode). Specify `1` for each backend if you want to balance the load equally or decimal values (`.9` , `.1` , etc.) to weigh the load toward specific backends.
`backend_data_directory0`	Data directory for backend 0.
`enable_pool_hba`	Set to `on` to use `pool_hba.conf` for client authentication.
`num_init_children`	Number of pools. Default is `32`.
`max_pool`	Number of connections per pool. Default is `4`.
`connection_cache`	Set to `on` to enable connection pooling.
`pool_conn_dbname`	Database name to which EDB Pgpool-II connects. By default, EDB Pgpool-II connects with Postgres. As of v4.3, the deprecated `pool_conn_dbname` parameter is removed.
`sr_check_user`	User name to perform streaming replication check. Required as of EDB Pgpool-II v4.3.
`sr_check_password`	Password of the `sr_check_user` user to perform the streaming replication checks. Required as of EDB Pgpool-II v4.3.

The following table lists `pgpool.conf` parameters to use when implementing replication and load balancing:

Parameter name	Description
`Allow_sql_comments`	If on, ignore SQL comments. Changes to this parameter require reloading the `pgpool.conf` file.
`load_balance_mode`	Set to `on` to activate load balancing mode. If `load_balance_mode` is on and `replicate_select` is off, `SELECT` statements are sent to one backend. The parameter backend_weight<N>.z determines the proportion of `SELECT` statements each backend receives.
`ignore_leading_white_space`	Ignore leading white spaces of each query. Certain APIs such as DBI/DBD::Pg for Perl add white space that you can't control. Default is `on`.

Configuring connection pooling

EDB Pgpool-II provides a set of child processes that maintain cached connections to one or more database servers. When a client connects, EDB Pgpool-II attempts to reuse a connection from its pool, thus avoiding the overhead of opening and closing client connections.

You can reuse a connection in the pool only if the target database and the connection user match a prior connection that is currently in the pool. The `pgpool.conf` file specifies the connection pooling configuration options (such as the number of child processes and the maximum number of cached connections per child).

To configure connection pooling with one database server:

1. Configure the `pg_hba.conf` file on the `Pgpool-II` host to permit connections between the clients and the server.

2. Copy the `pgpool.conf.sample` file to `pgpool.conf`, modify the file, set the `connection_cache` parameter to `on`, and specify connection properties for your database server.

   The following example shows how to connect with the EDB Postgres Advanced Server:

   ```
   connection_cache = on

   backend_hostname0 =
   'localhost'

   backend_port0 = 5444

   backend_weight0 = 1

   backend_data_directory0 =
   '/var/lib/edb/as14/data'

   sr_check_user =
   'enterprisedb'

   sr_check_password = 'enterprisedb_password'
   ```

   The following example shows how to connect with the PostgreSQL Server:

   ```
   connection_cache = on

   backend_hostname0 =
   'localhost'

   backend_port0 = 5432
   ```

```
backend_weight0 = 1

backend_data_directory0 =
'/var/lib/pgsql/14/data'

sr_check_user =
'enterprisedb'

sr_check_password = 'enterprisedb_password'
```

> **Note**
>
> In the `pgpool.conf` file, connection parameters have an appended digit that specifies a cluster node identifier. Database node `0` specifies values for the primary node.

3. Optionally, configure EDB Pgpool-II client authentication.

4. Optionally, configure the PCP administrative interface.

5. Start EDB Pgpool-II:

```
systemctl start edb-pgpool-<x.y>.service
```

<x.y> is the EDB Pgpool release version.

6. Run the following platform-specific command to connect to Pgpool43:

On EDB Postgres Advanced Server for CentOS 7:

```
./psql -d edb -p 9999 -U enterprisedb -h /tmp
```

On EDB Postgres Advanced Server for Debian:

```
./psql -d edb -p 9999 -U enterprisedb
```

On PostgreSQL Server for CentOS 7:

```
./psql -d postgres -p 9999 -U postgres -h /tmp
```

On PostgreSQL Server for Debian:

```
./psql -d postgres -p 9999 -U postgres
```

Configuring load balancing

EDB supports replication scenarios that use EDB Pgpool-II load balancing with PostgreSQL streaming replication or Slony replication. The supported replication methods ensure that database updates made by client applications apply to multiple backend servers. For detailed information about the benefits of each replication method and configuration instructions, see the project documentation for each utility.

When load balancing is enabled, EDB Pgpool-II distributes some types of `SELECT` statements to backend servers, allowing multiple database

servers and hosts to share the processing load of `SELECT` statements issued by client applications.

When configuring EDB Pgpool-II load balancing, the initial database environments in all backend servers must be identical:

- Tables must have the same name, definition, and row content.
- Schemas must exist in each backend application database.
- Roles and privileges on each backend server must be configured to ensure the result set of SQL statements are identical on all servers.

If you use `password` authentication, assign the same password to an associated user name on each database server. Use the same user name/password pair to connect EDB Pgpool-II to each backend connection.

In a replication scenario, each backend is uniquely identified by the hostname (or IP address) and the port number on which the database server instance is listening for connections. Make sure that the `pool_hba.conf` and `pg_hba.conf` files allow a connection between that server and the host on which EDB Pgpool-II is running.

The following example shows how to implement EDB Pgpool-II load balancing with two servers (the primary and replica nodes) in a streaming replication scenario. Configuring EDB Pgpool-II load balancing for a Slony replication scenario is similar. See the Slony documentation for information about configuring Slony replication.

Configuring the primary node of the replication scenario

Open an SSH session with the primary node of the replication scenario, and modify the `pg_hba.conf` file (located in the `/var/lib/edb/as14/data` directory). Add connection information for the replication user. (In the example that follows, edbrepuser resides on a standby node with an IP address of `107.178.217.178`):

```
host replication edbrepuser 107.178.217.178/32 md5
```

The connection information must specify the address of the replication scenario's standby node and your preferred authentication method.

Modify the `postgresql.conf` file (located in `/var/lib/edb/as14/data`), adding the following replication parameter and values to the end of the file:

```
 wal_level = replica
 max_wal_senders = 10
 checkpoint_segments =
8
 wal_keep_segments = 0
```

Save the configuration file, and restart the server:

To restart on RHEL/CentOS 7 and RHEL/Rocky Linux/AlmaLinux 8 platforms:

```
systemctl restart edb-as-14
```

To restart on Debian 10.x or Ubuntu 18.04 | 20.04 platforms:

```
/usr/edb/as14/bin/epas_ctlcluster 14 main restart
```

Use the `sudo su -` command to assume the identity of the enterprisedb database superuser:

```
  sudo su - enterprisedb
```

Then, start a psql session, connecting to the edb database:

```
psql -d edb
```

At the psql command line, create a user with the `replication` attribute:

```
 CREATE ROLE edbrepuser WITH REPLICATION LOGIN PASSWORD
'password';
```

Configuring the standby node of the replication scenario

Open an SSH session with the standby server and assume the identity of the database superuser (enterprisedb):

```
sudo su - enterprisedb
```

With your choice of editor, create a `.pgpass` file in the home directory of the enterprisedb user. The `.pgpass` file holds the password of the replication user in plain-text form. If you're using a `.pgpass` file, make sure that only trusted users have access to it:

Add an entry that specifies connection information for the replication user:

```
*:5444:*:edbrepuser:password
```

The server enforces restrictive permissions on the `.pgpass` file. Use the following command to set the file permissions:

```
chmod 600 .pgpass
```

Relinquish the identity of the database superuser:

```
exit
```

Then, assume superuser privileges:

```
sudo su -
```

Use your platform-specific command to stop the database server before replacing the data directory on the standby node with the `data` directory of the primary node.

Then, delete the `data` directory on the standby node:

```
rm -rf /var/lib/edb/as14/data
```

After deleting the existing `data` directory, use the `pg_basebackup` utility to copy the `data` directory of the primary node to the standby:

```
pg_basebackup --format=p --label=standby --host=146.148.46.44 --username=edbrepuser --password --wal-
method=stream -R
```

The call to `pg_basebackup` specifies the IP address of the primary node and the name of the replication user created on the primary node.

Including the `-R` option creates the `standby.signal` file and appends connection settings to `postgresql.auto.conf` in the output directory (or into the base archive file when using tar format) to ease setting up a standby server.

For more information about the options available with the `pg_basebackup` utility, see the PostgreSQL core documentation.

When prompted by `pg_basebackup`, provide the password associated with the replication user.

After copying the `data` directory, change ownership of the directory to the database superuser (enterprisedb):

```
chown -R enterprisedb /var/lib/edb/as14/data
```

Modify the `postgresql.conf` file (located in `/var/lib/edb/as14/data`), specifying the following values at the end of the file:

```
wal_level = replica
hot_standby = on
```

The `data` file has been copied from the primary node and contains the replication parameters specified previously.

Then, restart the server. At this point, the primary node is replicating data to the standby node.

Configuring EDB Pgpool-II load balancing

In the `pgpool.conf` file, modify the parameter settings to specify that load balancing is enabled:

```
load_balance_mode = on
```

Then, specify the connections settings for the primary database node in the parameter set that ends with a `0`. For example:

```
backend_hostname0 = '146.148.46.44'
backend_port0 = 5444
backend_weight0 = 1
backend_data_directory0 =
'/var/lib/edb/as14/data'
```

Then, specify the connections settings for each node to which queries are distributed. Increment the number that follows the parameter name for each node, and provide connection details:

```
backend_hostname1 = '107.178.217.178'
backend_port1 = 5444
backend_weight1 = 1
backend_data_directory1 =
'/var/lib/edb/as14/data'
```

Use the `backend_weight` parameter to specify how to distribute queries distributed among the nodes. Specify a value of `1` to indicate that you want (qualified) queries to be equally distributed across the nodes of the replication scenario.

Restart EDB Pgpool-II

```
systemctl restart edb-pgpool-<x.y>.service
```

<x.y> is the EDB Pgpool release version.

Configuring client authentication

When EDB Pgpool-II is enabled, client applications connect to EDB Pgpool-II, which acts as a middleman for a Postgres server. A connecting client application is first authenticated with the EDB Pgpool-II server and then with the Postgres server.

Parameter settings in the `pool_hba.conf` configuration file determine the EDB Pgpool-II authentication properties. The `pool_hba.conf` file is similar in format and function to the Postgres `pg_hba.conf` configuration file. See the Pgpool-II documentation for detailed information about `pool_hba.conf` entries.

To enable EDB Pgpool-II authentication:

1. Copy the `pool_hba.conf.sample` file to `pool_hba.conf`.
2. Modify the `pool_hba.conf` file, specifying authentication information for servers or users you want to connect. Entries must follow the same format used in the `pg_hba.conf` file.
3. Modify the `pgpool.conf` file, setting the `enable_pool_hba` parameter to `on`.
4. Restart EDB Pgpool-II to reload the EDB Pgpool-II configuration files.

> **Note**
>
> When authenticating with the database server, use the user names and passwords specified in the `pool_hba.conf` file. You must also specify these user names and passwords in the database server's `pg_hba.conf` file.

Configuring PCP

PCP is an administrative interface for EDB Pgpool-II that allows you to retrieve information about database nodes, EDB Pgpool-II child processes, and other information. Issue PCP commands from the Linux command line.

`pcp.conf` is the password configuration file for the PCP client. Before using PCP commands, modify the `pcp.conf` file, providing the user names and passwords you provide when invoking a PCP command. The user names in the `pcp.conf` file are independent of the database server user names and passwords.

Use the following steps to configure PCP:

1. Copy the `pcp.conf.sample` file to `pcp.conf`.

2. Add an entry to the `pcp.conf` file in the following form:

   ```
   username:md5_password
   ```

   `username` is a PCP user name.

   `md5_password` is the PCP password in `md5` format.

   You can use the `pg_md5` program to generate the encrypted password from the clear-text form:

   ```
   $ pg_md5 mypassword
   ```

   ```
   34819d7beeabb9260a5c854bc85b3e44
   ```

   For example, the entry in the `pcp.conf` file for a PCP user named pcpuser with the password of `mypassword` is:

```
# USERID:MD5PASSWD
pcpuser:34819d7beeabb9260a5c854bc85b3e44
```

3. Restart the EDB Pgpool service.

4. When issuing a PCP command, specify the PCP user name and the unencrypted form of the password:

```
$ pcp_node_info 5 localhost 9898 pcpuser mypassword 0
localhost 5444 1 1.000000
```

After configuring PCP, you can use the following PCP commands to control EDB Pgpool-II and retrieve information.

PCP command	Description
`pcp_common_options`	Common options used in PCP commands
`pcp_node_count`	Displays the total number of database nodes
`pcp_node_info`	Displays the information on the given node ID
`pcp_health_check_stats`	Displays health check statistics data on given node ID
`pcp_watchdog_info`	Displays the watchdog status of the EDB Pgpool-II
`pcp_proc_count`	Displays the list of EDB Pgpool-II children process IDs
`pcp_proc_info`	Displays the information on the given EDB Pgpool-II child process ID
`pcp_pool_status`	Displays the parameter values as defined in `pgpool.conf`
`pcp_detach_node`	Detaches the given node from EDB Pgpool-II, forcing existing connections to EDB Pgpool-II to be disconnected
`pcp_attach_node`	Attaches the given node to EDB Pgpool-II
`pcp_promote_node`	Promotes the given node as new main to EDB Pgpool-II
`pcp_stop_pgpool`	Terminates the EDB Pgpool-II process
`pcp_reload_config`	Reloads EDB Pgpool-II config file
`pcp_recovery_node`	Attaches the given backend node with recovery

> **Note**
>
> `pcp_health_check_stats` and `pcp_reload_config` commands are available from EDB Pgpool version 4.2 onwards.

To view more information about PCP command options, see the Pgpool project site.

Configuring number of connections and pooling

EDB Pgpool-II has some configuration to tune the pooling and connection processing. Depending on this configuration, you must also set the Postgres configuration for `max_connections` to ensure all connections can be accepted as required. Furthermore, the cloud architecture works with active/active instances, which needs to spread `num_init_children` over all EDB Pgpool instances (divide the normally used value by the number of active instances).

`max_pool`: Generally, advised to set `max_pool` to `1`. Alternatively, for applications with many reconnects, you can set `max_pool` to the number of distinct combinations of users, databases, and connection options for the application connections. All but one connection in the pool are stale connections, which consume a connection slot from Postgres without adding to the performance. We therefore recommend that you don't configure `max_pool` beyond 4 to preserve a healthy ratio between active and stale connections. As an example, for an application that constantly reconnects

and uses two distinct users, both connecting to their own database, set it to `2` . If both users can connect to both databases, set it to `4` . Increasing `max_pool` requires that you tune down `num_init_children` in EDB Pgpool or tune up `max_connections` in Postgres.

`num_init_children`: We recommend setting `num_init_children` to the number of connections that could be running active in parallel, but the divide value by the number of active EDB Pgpool-II instances (one with the on-premise architecture and all instances for the cloud architecture). As an example: in an architecture with three EDB Pgpool instances, to allow the application to have 100 active connections in parallel, set `num_init_children` to 100 for the on-premises architecture, and set `num_init_children` to 33 for the cloud architecture. Increasing `num_init_children` generally requires that you tune up `max_connections` in Postgres.

`listen_backlog_multiplier`: Can be set to multiply the number of open connections (as perceived by the application) with the number of active connections (`num_init_children`). As an example, when the application might open 500 connections, of which 100 should be active in parallel, with the on-premises architecture, set `num_init_children` to `100` and `listen_backlog_multiplier` to `4` . This setup can process 100 connections active in parallel, and another 400 (`listen_backlog_multiplier` x `num_init_children`) connections are queued before connections are blocked. The application perceives a total of 500 open connections, and Postgres processes the load of 100 connections maximum at all times. Increasing `listen_backlog_multiplier` causes the application to perceive more connections but doesn't increase the number of parallel active connections (which is determined by `num_init_children`).

`max_connections`: We recommend setting `max_connections` in Postgres higher than [number of active pgpool instances] x [`max_pool`] x [`num_init_children`] + [`superuser_reserved_connections`] (Postgres). As an example: in the on-premises setup with three instances active/passive, `max_pool` set to `2` , `num_init_children` set to `100` , and `superuser_reserved_connections` (Postgres) set to `5` , set Postgres `max_connections` equal to or higher than [1 x 2 x 100+5], which is 205 connections, or higher. A similar setup in the cloud setup runs with three active instances, `max_pool` set to `2` , `num_init_children` set to `33` , and `superuser_reserved_connections` (Postgres) set to `5` . In this case set Postgres `max_connections` equal or higher than [3x 2 x 33+5] which is 203 or higher. Configuring below the advised setting can cause issues opening new connections and, in combination with `max_pool` , can cause unexpected behavior (low or no active connections but still connection issues due to stale pooled connections using connection slots from Postgres.

EDB Pgpool-II host setup

After modifying the parameter settings that implement EDB Pgpool-II functionality for your installation, start the EDB Pgpool-II service.

When EDB Pgpool-II starts, it records its process ID in a `pgpool.conf` file whose name is determined by the `pid_file_name` configuration parameter. The initial value of the `pid_file_name` parameter in the sample file is:

```
pid_file_name = /var/run/edb/pgpool<x.y>/edb-pgpool-
<x.y>.pid
```

<x.y> is the EDB Pgpool release version.

> **Note**
>
> The operating system might remove the contents of the `/var/run` directory (including the `pgpool` directory) during a reboot. Don't use the `/var/run/edb/pgpool` directory as the location for the `pgpool.pid` file. Modify the `pid_file_name` parameter to specify a safer directory location.

7   Connecting a client to EDB Pgpool-II

Connect client applications directly to the EDB Pgpool-II listener port on the EDB Pgpool-II host. For example, to connect to the edb database (while using EDB Pgpool-II), enter:

```
psql -d edb -U enterprisedb -h localhost -p 9999
```

When invoked at the psql prompt, the following `SHOW` command keywords display EDB Pgpool-II information:

Command	Information provided
`SHOW pool_status`	Displays EDB Pgpool-II configuration parameters and their name, value, and description
`SHOW pool_nodes`	Displays a list of all configured nodes
`SHOW pool_processes`	Displays a list of all EDB Pgpool-II processes waiting for connections or dealing with a connection
`SHOW pool_pools`	Displays a list of pools
`SHOW pool_version`	Displays the EDB Pgpool-II release number
`PGPOOL SHOW`	Displays the configuration parameter value
`PGPOOL SET`	Changes a configuration parameter
`PGPOOL RESET`	Restores the configuration parameter to the default value
`SHOW POOL_CACHE`	Displays cache storage statistics
`SHOW POOL_HEALTH_CHECK_STATS`	Displays health check statistics
`SHOW POOL_BACKEND_STATS`	Displays backend SQL command statistics

To view more information about `SHOW` command options, see the Pgpool project site.

> **Note**
>
> `SHOW POOL_HEALTH_CHECK_STATS` and `SHOW POOL_BACKEND_STATS` commands are available from EDB Pgpool version 4.2 onwards.