



EDB Postgres

Release 4.1.1

Pgpool-II Guide

Mar 06, 2020

Contents

1	Installing and Configuring Pgpool-II	2
1.1	Using an RPM Package to Install PgPool-II	2
1.2	Installing Pgpool on a Debian or Ubuntu Host	4
2	Pgpool-II Host Setup	5
3	Configuring Connection Pooling	6
4	Configuring Pgpool-II Load Balancing	7
5	Pgpool-II Client Authentication Configuration	11
6	Connecting a Client to Pgpool-II	12
7	PCP Configuration	13
8	Conclusion	15
	Index	16

The Pgpool-II utility package acts as a middleman between client applications and a Postgres database server. Pgpool-II functionality is transparent to client applications; client applications connect to Pgpool-II instead of directly to Advanced Server, and Pgpool-II manages the connection. EnterpriseDB supports the following Pgpool-II functionality:

- Load balancing
- Connection pooling
- High availability
- Connection limits

Installing and Configuring Pgpool-II

Pgpool-II runs as a service on Linux systems, and is not supported on Windows systems.

1.1 Using an RPM Package to Install PgPool-II

Before installing PgPool-II, you must:

Install the `epel-release` package:

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

Note: You may need to enable the `[extras]` repository definition in the `CentOS-Base.repo` file (located in `/etc/yum/repos.d`).

You must also have credentials that allow access to the EnterpriseDB repository. For information about requesting credentials, visit:

<https://info.enterprisedb.com/rs/069-ALB-339/images/Repository%20Access%2004-09-2019.pdf>

After receiving your repository credentials you can:

1. Create the repository configuration file.
2. Modify the file, providing your user name and password.
3. Invoke yum to install Pgpool-II.

Creating a Repository Configuration File

To create the repository configuration file, assume superuser privileges, and invoke the following command:

```
yum -y install https://yum.enterprisedb.com/edb-repo-rpms/edb-repo-latest.noarch.rpm
```

The repository configuration file is named `edb.repo`. The file resides in `/etc/yum.repos.d`.

Modifying the file, providing your user name and password

After creating the `edb.repo` file, use your choice of editor to ensure that the value of the `enabled` parameter is `1`, and replace the `username` and `password` placeholders in the `baseurl` specification with the name and password of a registered EnterpriseDB user.

```
[edb]
name=EnterpriseDB RPMs $releasever - $basearch
baseurl=https://<username>:<password>@yum.enterprisedb.com/edb/redhat/rhel-
↳$releasever-$basearch
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY
```

Installing Pgpool-II

After saving your changes to the configuration file, you can use the `yum install` command (`yum install edb-pgpool<xy>`, where `x.y.z` is the `pgpool` release version.) to install `Pgpool-II`. For example, the following command installs `Pgpool Version 4.1.1`:

```
yum install edb-pgpool41
```

When you install an RPM package that is signed by a source that is not recognized by your system, `yum` may ask for your permission to import the key to your local server. If prompted, and you are satisfied that the packages come from a trustworthy source, enter `y`, and press `Return` to continue.

During the installation, `yum` may encounter a dependency that it cannot resolve. If it does, it will provide a list of the required dependencies that you must manually resolve.

Configuring Pgpool-II

If you have used an RPM package to install `Pgpool-II`, the configuration files are located in `/etc/sysconfig/edb/pgpool<x.y>`, where `x.y.z` is the `pgpool` release version.). By default, in the configuration files names, `.sample` is added which should be removed after copying it on the same path, for example `cp pgpool.conf.sample to pgpool.conf`.

After modifying the parameter settings that implement `Pgpool-II` functionality for your installation, you must start the `Pgpool-II` service. For detailed information about controlling the `Pgpool-II` service, see [Managing an Advanced Server Installation in the EDB Postgres Advanced Server Installation Guide for Linux](#).

Please note that the configuration options for `Pgpool-II` are extensive; the options listed below should be considered a starting point only. For more information about configuring and using `Pgpool-II`, please consult the [project website](#).

1.2 Installing Pgpool on a Debian or Ubuntu Host

To install a DEB package on a Debian or Ubuntu host, you must have credentials that allow access to the EnterpriseDB repository. To request credentials for the repository, visit [the EDB website](#).

The following steps will walk you through on using the EnterpriseDB apt repository to install a DEB package. When using the commands, replace the `username` and `password` with the credentials provided by EnterpriseDB.

1. Assume superuser privileges:

```
sudo su -
```

2. Configure the EnterpriseDB repository:

```
sh -c 'echo "deb https://username:password@apt.enterprisedb.com/$(lsb_release -cs)-edb $(lsb_release -cs) main" > /etc/apt/sources.list.d/edb-$(lsb_release -cs).list'
```

3. Add support to your system for secure APT repositories:

```
apt-get install apt-transport-https
```

4. Add the EDB signing key:

```
wget -q -O - https://username:password@apt.enterprisedb.com/edb-deb.gpg.key | apt-key add -
```

5. Update the repository metadata:

```
apt-get update
```

6. Install DEB package:

```
apt-get install edb-pgpool<xy>
```

where `x.y.z` is the pgpool release version.

Note: By default, the Debian 9x and Ubuntu 18.04 platform installs Java version 10. Make sure you install Java version 8 on your system to run the EDB Java-based components.

Pgpool-II Host Setup

When Pgpool-II starts, it records its process ID in a 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
```

where `x.y.z` is the pgpool release version.

Please note that the contents of the `/var/run` directory (including the `pgpool` directory) may be removed by the operating system during a reboot. The `/var/run/edb/pgpool` directory should not be used as the location for the `pgpool.pid` file. Modify the `pid_file_name` parameter to specify a safer directory location.

Configuring Connection Pooling

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

A connection in the pool can be reused only if the target database and the connection user match a prior connection that is currently in the pool. Connection pooling configuration options (such as the number of child processes, and the maximum number of cached connections per child) are specified in the `pgpool.conf` file.

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`, and modify the file, setting the `connection_cache` parameter to `on`, and specifying connection properties for your database server.

For example:

```
connection_cache = on

backend_hostname0 = 'localhost' backend_port0 = 5444 backend_weight0 = 1
backend_data_directory0 = '/var/lib/edb/as12/data'
```

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

3. Optionally, configure Pgpool-II client authentication.
4. Optionally, configure the PCP administrative interface.
5. Start Pgpool-II and begin using your application.

Configuring Pgpool-II Load Balancing

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

When load balancing is enabled, 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 Pgpool-II load balancing, it is crucial that the initial database environments in all backend servers are identical:

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

If you use `password` authentication, the same password must be assigned to an associated user name on each database server. The same user name/password pair is used to connect Pgpool-II to each backend connection.

Within a replication scenario, each backend is uniquely identified by the host name (or IP address) and port number on which the database server instance is listening for connections. You must ensure that the `pool_hba.conf` and `pg_hba.conf` files allow a connection between that server and the host on which Pgpool-II will be running.

The following example demonstrates how to implement Pgpool-II load balancing with two servers (the master and replica nodes) in a Streaming Replication scenario. Configuring Pgpool-II load balancing for a Slony replication scenario is similar; please see the Slony documentation for information about configuring Slony replication.

Step 1 - Configuring the Master Node of the Replication Scenario

Open an SSH session with the master node of the replication scenario, and modify the `pg_hba.conf` file (located in the `/var/lib/edb/as12/data` directory), adding 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 should specify the address of the standby node of the replication scenario, and your preferred authentication method.

Modify the `postgresql.conf` file (located in `/var/lib/edb/as12/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:

```
systemctl restart edb-as-12
```

Note: The above command is applicable for RHEL 7 and Centos 7 platforms. To restart the server on RHEL 6 and Centos 6 platform, use the below command:

```
/etc/init.d/edb-as-12 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';
```

Step 2 - 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 are using a `.pgpass` file, you should ensure that only trusted users have access to the `.pgpass` file:

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

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

The server will enforce 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 Master node. For detailed information about controlling the Pgpool-II service, see [Managing an Advanced Server Installation in the EDB Postgres Advanced Server Installation Guide for Linux](#).

Then, delete the data directory on the Standby node:

```
rm -rf /var/lib/edb/as12/data
```

After deleting the existing data directory, use the `pg_basebackup` utility to copy the data directory of the Master 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 Master node and the name of the replication user created on the Master 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 at:

<https://www.postgresql.org/docs/12/static/app-pgbasebackup.html>

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/as12/data
```

Modify the `postgresql.conf` file (located in `/var/lib/edb/as12/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 Master node, and will contain the replication parameters specified previously.

Then, restart the server:

```
systemctl restart edb-as-12
```

Note: The above command is applicable for RHEL 7 and Centos 7 platforms. To restart the server on RHEL 6 and Centos 6 platform, use the below command:

```
/etc/init.d/edb-as-12 restart
```

At this point, the Master node will be replicating data to the Standby node.

Step 3 - Configuring Pgpool-II Load Balancing

Use your choice of editor to modify the `pgpool.conf` file. Within 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 master 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/as12/data'
```

Then, specify the connections settings for each node to which queries will be 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/as12/data'
```

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

Step 4 - Restart Pgpool-II and begin using your application

For detailed information about controlling the Pgpool-II service, see [Managing an Advanced Server Installation in the EDB Postgres Advanced Server Installation Guide for Linux](#).

Commonly used Pgpool-II Parameters

The following table lists `pgpool.conf` parameters that are used when implementing connection pooling:

Parameter Name	Description
<code>pool_conn_dbname</code>	Database name to which Pgpool-II will connect. By default, Pgpool-II will connect with Postgres.
<code>listen_addresses</code>	Host name or IP address used by Pgpool-II to listen for connections. Default is localhost. Change to '*' for all addresses.
<code>port</code>	Port for Pgpool-II connections. Default is 9999.
<code>pcp_port</code>	Port for PCP connections. Default is 9898.
<code>backend_hostname0</code>	Host name or IP address for backend 0. You can specify '' if the backend and Pgpool-II are running on the same host.
<code>backend_port0</code>	Port number for backend 0.
<code>backend_weight0</code>	Weight for backend 0 (only in load balancing mode). Specify 1 for each backend if you want the load equally balanced, or decimal values (.9, .1, etc.) to weight the load towards certain backends.
<code>backend_data_directory0</code>	Data directory for backend 0.
<code>enable_pool_hba</code>	Set to on to use <code>pool_hba.conf</code> for client authentication.
<code>num_init_children</code>	Number of pools. Default is 32.
<code>max_pool</code>	Number of connections per pool. Default is 4.
<code>connection_cache</code>	Set to on to enable connection pooling.

The following table lists `pgpool.conf` parameters that are used when implementing replication and load balancing:

Parameter Name	Description
<code>allow_sql_comments</code>	If on, ignore SQL comments; modifications to this parameter require a reload of the <code>pgpool.conf</code> file.
<code>load_balance_mode</code>	Set to on to activate load balancing mode. If <code>load_balance_mode</code> is on and <code>replicate_select</code> is off, SELECT statements are sent to one backend. The proportion of SELECT statements each backend receives is determined by parameter <code>backend_weight<N></code> .
<code>ignore_leading_white_space</code>	Ignore leading white spaces of each query. Certain APIs such as DBI/DBD::Pg for Perl add white space that the user cannot control. Default is on.

Pgpool-II Client Authentication Configuration

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

Pgpool-II authentication properties are determined by parameter settings in the `pool_hba.conf` configuration file. The `pool_hba.conf` file is similar in format and function to the Postgres `pg_hba.conf` configuration file. Please refer the Pgpool-II documentation for detailed information about `pool_hba.conf` entries.

To enable 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 that 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 Pgpool-II to reload the pgpool-II configuration files.

Note: User names and passwords specified in the `pool_hba.conf` file will be used for authentication with the database server; you must also specify those user names and passwords in the database server's `pg_hba.conf` file.

Connecting a Client to Pgpool-II

Client applications should connect directly to the Pgpool-II listener port on the Pgpool-II host. For example, to connect to the `edb` database (while using pgpool-II functionality), enter:

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

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

Command	Information Provided
<code>SHOW pool_status</code>	Displays Pgpool-II configuration parameters and their name, value, and description.
<code>SHOW pool_nodes</code>	Displays a list of all configured nodes.
<code>SHOW pool_processes</code>	Displays a list of all Pgpool-II processes waiting for connections or dealing with a connection.
<code>SHOW pool_pools</code>	Displays a list of pools.
<code>SHOW pool_version</code>	Displays the Pgpool-II release number.

PCP Configuration

PCP is an administrative interface for Pgpool-II that allows you to retrieve information about database nodes, Pgpool-II child processes, etc. You should issue PCP commands from the Linux command line.

Before using PCP commands, you must modify the `pcp.conf` file, providing user names and passwords that you supply whenever you issue a PCP command. The user names in the `pcp.conf` file are completely independent of the database server user names and passwords.

Use the following steps to enable 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
```

where:

`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 as shown below:

```
$ 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 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 5432 1 1.000000
```

After configuring PCP, you can use PCP commands to control Pgpool-II and retrieve information. Specify the following arguments when calling PCP commands:

Argument	Description
timeout	Timeout value in seconds. PCP will disconnect if Pgpool-II does not respond within the specified number of seconds.
host	The name of the Pgpool-II host.
port	The PCP port number; the default value is 9898.
username	The PCP user name (as specified in pcp.conf.)
password	The password associated with the user name (as specified in pcp.conf).

PCP recognizes the following commands:

PCP Command	Description
pcp_node_count timeout host port username password	Total number of nodes defined in pgpool.conf
pcp_node_info timeout host port username password nodeid	Displays information on the node given by <nodeid>
pcp_proc_count timeout host port username password	Displays the pgpool-II child process IDs
pcp_proc_info timeout host port username password processid	Displays information on the pgpool-II child process given by <processid>
pcp_detach_node [-g] timeout host port username password nodeid	Detaches the node specified by <nodeid> from pgpool-II. If -g is given, wait until all clients are disconnected (unless <code>client_idle_limit_in_recovery</code> is -1 or <code>recovery_timeout</code> is expired).
pcp_attach_node timeout host port username password nodeid	Attaches the node specified by <nodeid> to pgpool-II.

EDB Postgres™ Pgpool-II Guide

Copyright © 2020 EnterpriseDB Corporation. All rights reserved.

EnterpriseDB® Corporation 34 Crosby Drive, Suite 201, Bedford, MA 01730, USA

T +1 781 357 3390 F +1 978 467 1307 E info@enterprisedb.com www.enterprisedb.com

- EnterpriseDB and Postgres Enterprise Manager are registered trademarks of EnterpriseDB Corporation. EDB and EDB Postgres are trademarks of EnterpriseDB Corporation. Oracle is a registered trademark of Oracle, Inc. Other trademarks may be trademarks of their respective owners.
- EDB designs, establishes coding best practices, reviews, and verifies input validation for the logon UI for EDB Postgres products where present. EDB follows the same approach for additional input components, however the nature of the product may require that it accepts freeform SQL, WMI or other strings to be entered and submitted by trusted users for which limited validation is possible. In such cases it is not possible to prevent users from entering incorrect or otherwise dangerous inputs.
- EDB reserves the right to add features to products that accept freeform SQL, WMI or other potentially dangerous inputs from authenticated, trusted users in the future, but will ensure all such features are designed and tested to ensure they provide the minimum possible risk, and where possible, require superuser or equivalent privileges.
- EDB does not warrant that we can or will anticipate all potential threats and therefore our process cannot fully guarantee that all potential vulnerabilities have been addressed or considered.

C

Conclusion, 15

Configuring Connection Pooling, 6

Configuring Pgpool-II Load Balancing, 7

Connecting a Client to Pgpool-II, 12

I

Installing and Configuring Pgpool-II, 2

P

PCP Configuration, 13

Pgpool-II Client Authentication Configuration, 11

Pgpool-II Host Setup, 5