



PgBouncer Guide

Release 1.7.2.1

EDB Postgres™ PgBouncer Guide

Nov 08, 2019

Contents

1	Installing and Configuring PgBouncer	2
1.1	Performing an RPM Installation	2
1.2	Installing PgBouncer on a SLES 12 Host	4
1.3	Using the Graphical Installer	5
1.4	Configuring PgBouncer	12
1.4.1	Using the PgBouncer Admin Console	14
	Index	16

When a client application connects to a Postgres server, it negotiates a connection; that negotiation takes time. PgBouncer saves time by maintaining a pool of pre-established connections to the server. Instead of connecting directly to the server, the client connects to PgBouncer, minimizing the connection negotiation time by using a previously established connection made available to the client in the PgBouncer pool.

PgBouncer is a lightweight connection pooling utility for Postgres and Advanced Server installations that is based on the OpenSource PgBouncer project.

EnterpriseDB enhancements for the PgBouncer project are available via RPM Packages or a graphical PgBouncer installer that you can download with StackBuilder Plus. The enhancements allow pgBouncer to service clients that are using EDB Connectors that require compatible out parameter handling.

For more information about PgBouncer, including reference and usage information, please visit the [project site](#).

Installing and Configuring PgBouncer

PgBouncer is a lightweight connection pooler for Postgres servers. You can use an RPM installer or graphical installer to add PgBouncer to your system.

1.1 Performing an RPM Installation

For detailed information about creating and using EnterpriseDB repositories to install Advanced Server or its supporting components, please see the EDB Postgres Advanced Server Installation Guide, available at:

<https://www.enterprisedb.com/resources/product-documentation>

Before installing PgBouncer, 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. Install `edb-pgbouncer17`.

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 PgBouncer

After saving your changes to the configuration file, you can use the `yum install` command to install PgBouncer. For example, the following command installs PgBouncer:

```
yum install edb-pgbouncer17
```

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.

1.2 Installing PgBouncer on a SLES 12 Host

You can use the zypper package manager to install PgBouncer on an SLES 12 host. zypper will attempt to satisfy package dependencies as it installs a package, but requires access to specific repositories that are not hosted at EnterpriseDB.

Before installing PgBouncer, add the EnterpriseDB repository to your SLES host. Use the command:

```
zypper addrepo http://yum.enterprisedb.com/suse/epas96-sles.repo
```

The command creates the `edbas96suse.repo` file in the `/etc/zypp/repos.d` directory. The file contains:

```
[edbas96suse]
name=EDB Postgres Advanced Server 9.6 $releasever - $basearch
enabled=1
autorefresh=0
baseurl=http://username:password@yum.enterprisedb.com/9.6/suse/suse-$releasever-
↳$basearch
type=rpm-md
gpgcheck=1
gpgkey=http://yum.enterprisedb.com/9.6/suse/suse-$releasever-$basearch/repokey/
↳repomd.xml.key
```

Use your choice of editor to modify the file, replacing the `username` and `password` placeholders in the `baseurl` specification with your EnterpriseDB user name and repository password.

Then, use the following command to refresh the metadata on your SLES host to include the EnterpriseDB repository:

```
zypper refresh
```

```
Retrieving repository 'EDB Postgres Advanced Server 9.6 12 - x86_64'
metadata[]
New repository or package signing key received:
Repository: EDB Postgres Advanced Server 9.6 12 - x86_64
Key Name: EnterpriseDB Inc. (EnterpriseDB Yum Repositories)
&lt;packages@enterprisedb.com&gt;
Key Fingerprint: CA409F7C 635F2AE5 6C9E8B34 E5EDE919 7E30651C
Key Created: Wed Dec 31 11:37:58 2014
Key Expires: (does not expire)
Rpm Name: gpg-pubkey-7e30651c-54a3e016
Do you want to reject the key, trust temporarily, or trustalways?
[r/t/a/? shows all options] (r):
```

When prompted, specify `a` to always trust the provided key, and update the metadata to include the EnterpriseDB repository.

Before installing PgBouncer, you must add SUSEConnect and the SUSE Package Hub extension to the SLES host, and register the host with SUSE, allowing access to SUSE repositories. Use the commands:

```
zypper install SUSEConnect
SUSEConnect -p PackageHub/12/x86_64
```

For detailed information about registering a SUSE host, click [here](#).

Then, you can use the zypper utility to install PgBouncer:

```
zypper install edb-pgbouncer17
```

1.3 Using the Graphical Installer

Graphical installers for PgBouncer are available via StackBuilder Plus (for Advanced Server hosts) or Stack Builder (on PostgreSQL hosts). You can access StackBuilder Plus through your Windows or Linux start menu; after opening StackBuilder Plus and selecting the installation for which you wish to install PgBouncer, expand the component selection screen tree control to select and download the PgBouncer installer.

After downloading the PgBouncer installer, double-click the icon to start the PgBouncer installer. Select an installation language, and click OK to continue.

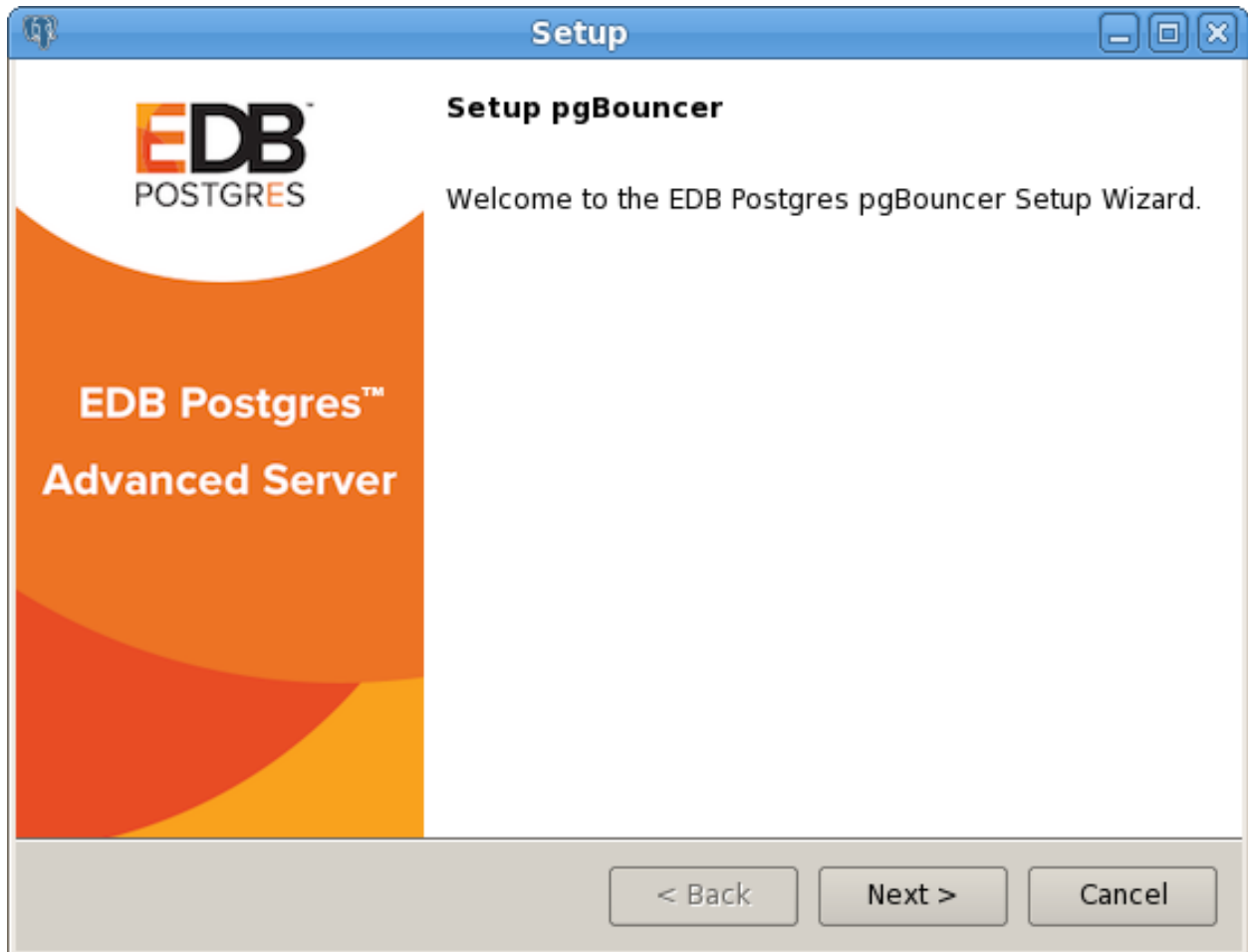


Fig. 1: *The PgBouncer Welcome window*

The PgBouncer installer welcomes you to the setup wizard, as shown in Figure above.

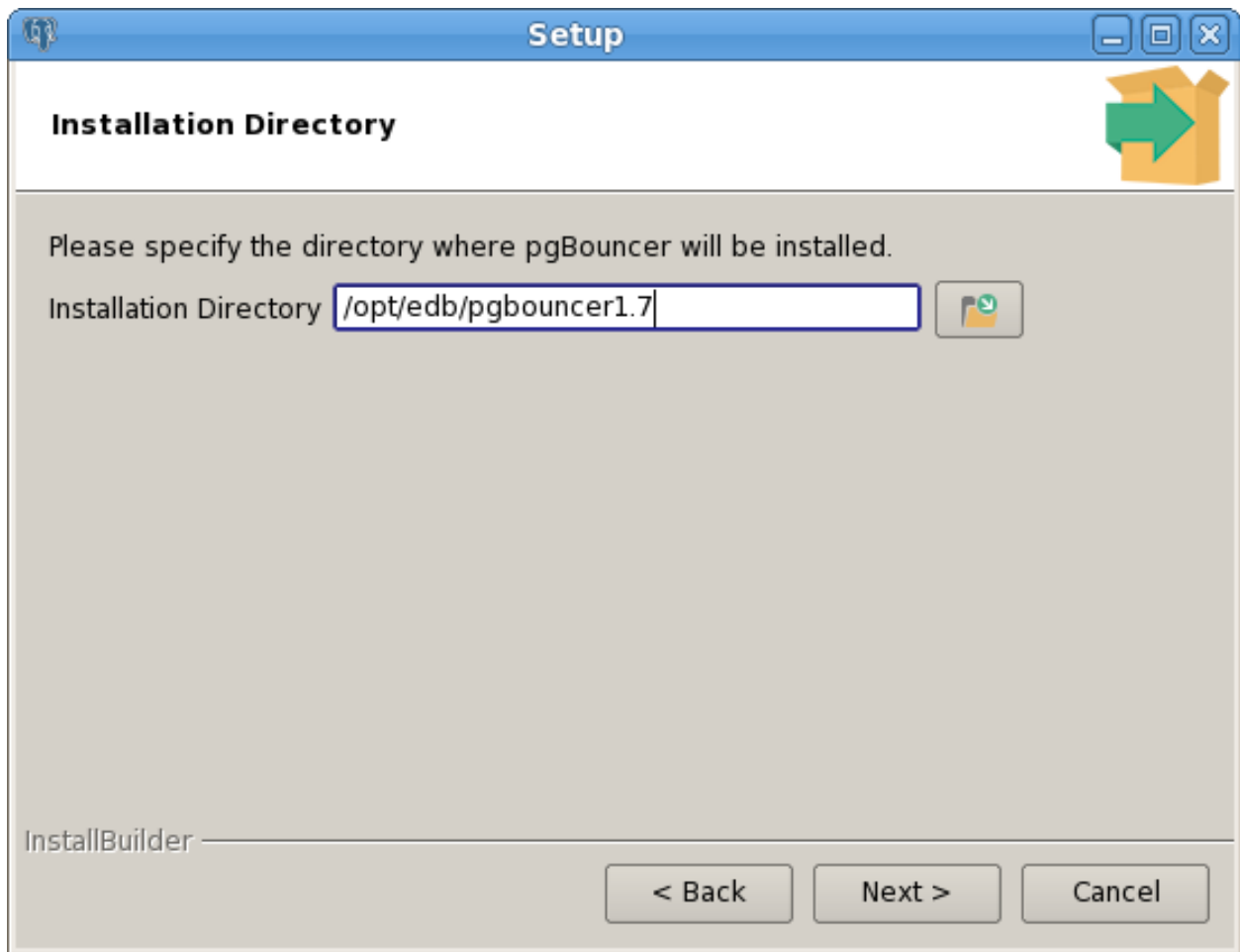
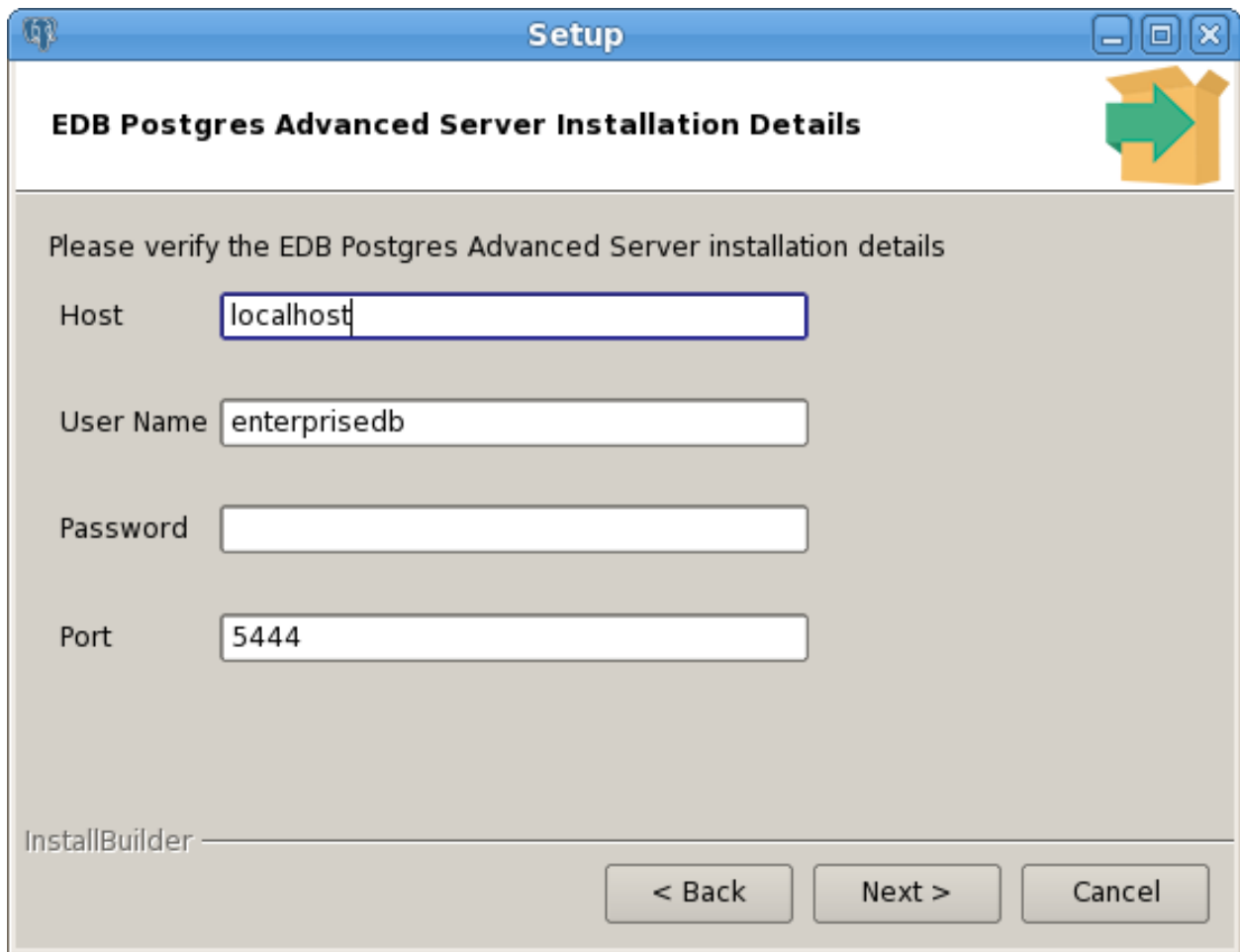


Fig. 2: *The Installation Directory window*

Use the `Installation Directory` field to specify the directory in which you wish to install the PgBouncer software. Then, click `Next` to continue.



Setup

EDB Postgres Advanced Server Installation Details

Please verify the EDB Postgres Advanced Server installation details

Host

User Name

Password

Port

InstallBuilder

< Back Next > Cancel

Fig. 3: The Advanced Server Installation Details window

Use fields on the EDB Postgres Advanced Server Installation Details window to provide connection information for the Advanced Server host:

- Use the `Host` field to identify the system on which Advanced Server resides.
- Provide the name of the role that PgBouncer will use for connections to the server in the `User Name` field.
- Provide the password associated with the role in the `Password` field.
- Use the `Port` field to identify the listener port that Advanced Server monitors for client connections.

Then, click `Next` to continue.

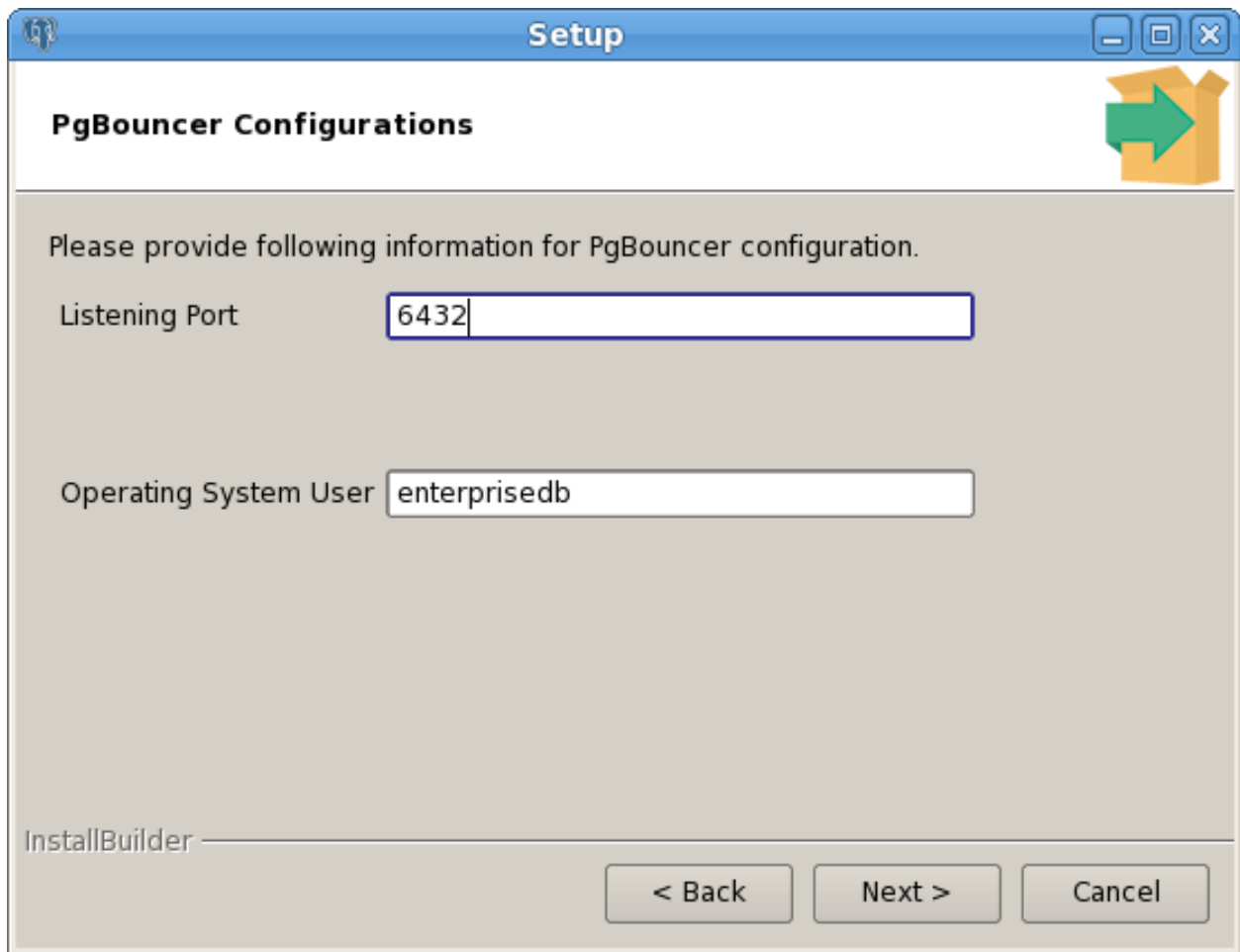


Fig. 4: The PgBouncer Configuration window

Use fields on the PgBouncer Configuration window to provide your preferences for the PgBouncer installation:

- Use the `Listening Port` field to specify the port that PgBouncer monitors for connections.
- Use the `Operating System User` field to specify the name of the Linux operating system user that PgBouncer will change to after startup. This option is not supported on Windows hosts.

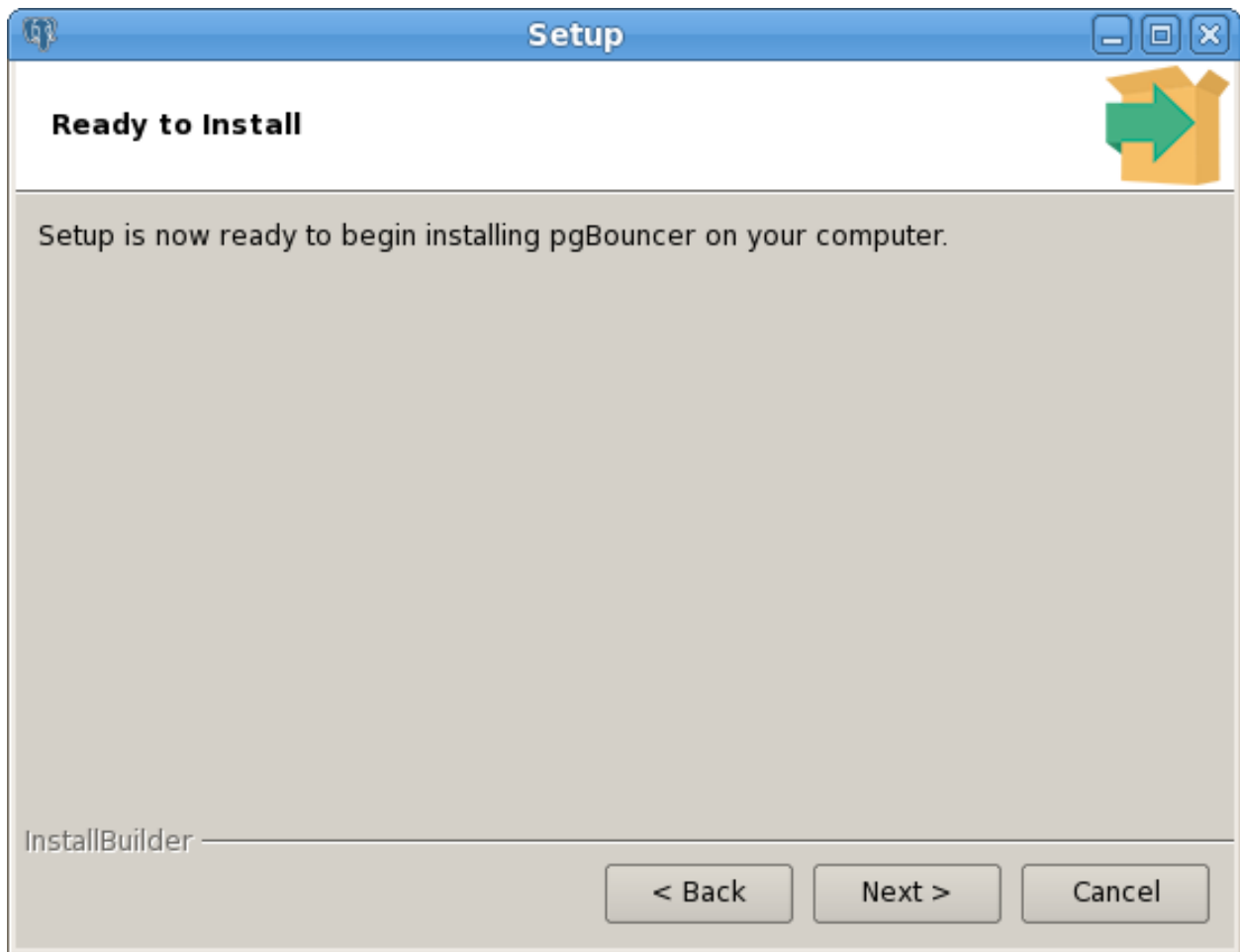


Fig. 5: *The Ready to Install window*

The Ready to Install window notifies you when the installer has all of the information needed to install PgBouncer on your system. Click Next to install PgBouncer.

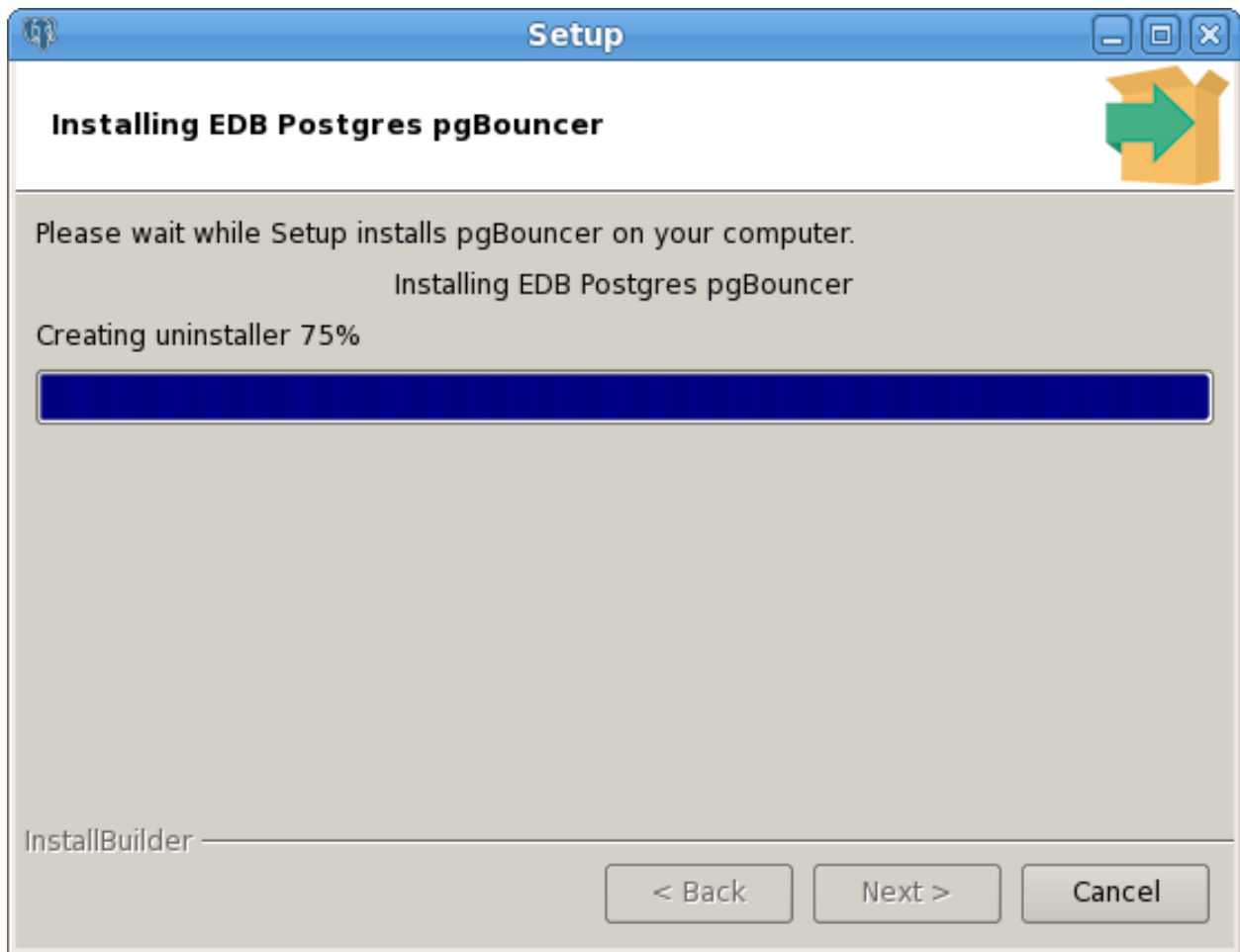


Fig. 6: *Installing EDB Postgres PgBouncer*

Progress bars inform you as the installation progresses.

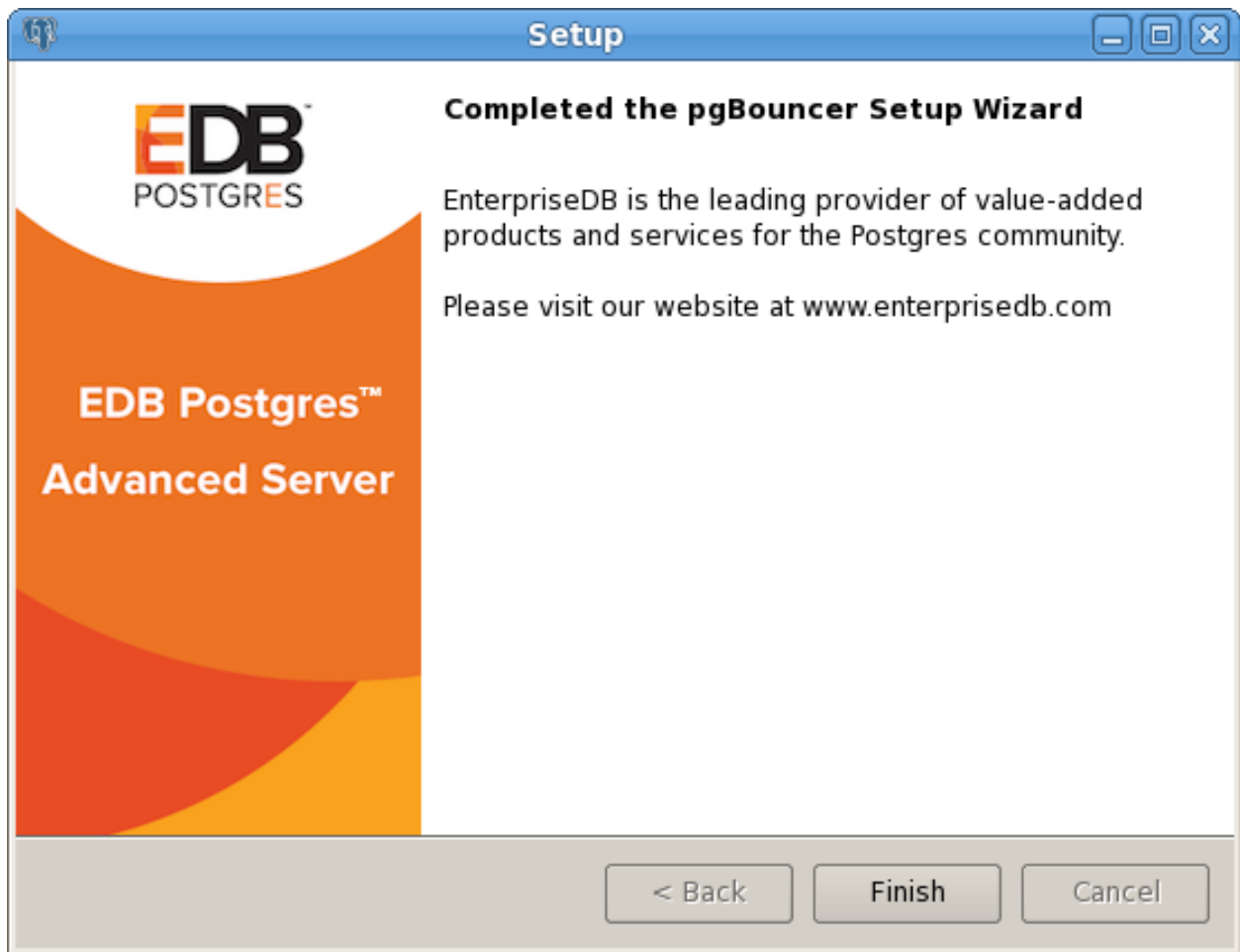


Fig. 7: The installation is complete

The installer notifies you when the setup wizard has completed the PgBouncer installation. Click **Finish** to exit the installer.

1.4 Configuring PgBouncer

When the PgBouncer service is running, any Postgres client connecting to the PgBouncer listener port specified in the configuration file will use connection pooling. PgBouncer connection and configuration information is stored in the `pgbouncer.ini` file, located in the following directory:

On Linux:

```
/opt/edb/pgbouncer1.7/share/
```

On Windows:

```
C:\Program Files\edb\pgbouncer1.7\share
```

The PgBouncer configuration file is divided into two sections: `[databases]` and `[pgbouncer]`.

The `[databases]` section of the configuration file contains a list of databases and the associated connection information. In an Advanced Server installation, the configuration file contains an entry for the installation of Advanced Server that installed PgBouncer:

```
edb = host=127.0.0.1 port=5444
```

You can specify additional database connection information in the configuration file in the form of `keyword=value` pairs. You can include the following parameters:

Parameter	Description
<code>name</code>	The name of the database to which the client application will connect.
<code>host</code>	The IP address of the host.
<code>port</code>	The port on which the host is listening.
<code>dbname</code>	The (optional) database name.
<code>user</code>	A username (if different than the information specified by the connecting client).
<code>password</code>	A password (if different than the information specified by the connecting client).

The following example demonstrates the syntax allowed in the `[databases]` section of the configuration file:

```
[databases]
edb = host=127.0.0.1 port=5444
acctg = host=192.168.10.101 port=5432 user=bob password=1safe_pwd!
```

Include the `dbname` parameter to map the connection name to an alternate database name. For example:

```
hr = host=127.0.0.1 port=5444 dbname=humanresources
```

When the client provides authentication information, that information is used to connect to PgBouncer, which in turn uses the information specified in the PgBouncer configuration file to connect to the database server. The user information provided in the configuration file must match a role defined in the Postgres database cluster.

Note: If you do not specify user details in `pgbouncer.ini`, the username and password will be authenticated by the database server and PgBouncer. As such, the username and password should be included in the `userlist.txt` file and the database cluster.

The `[pgbouncer]` section of the configuration file contains configuration details specific to PgBouncer:

Parameter	Description
admin_users	A comma-delimited list of users that are allowed to access the Admin Console (for management and monitoring purposes). By default, PgBouncer is installed with an admin_users = enterprisedb.
auth_file	The path to the authentication file that contains username and passwords of clients that may connect to PgBouncer. The authentication file (userlist.txt) is located in /opt/edb/pgbouncer-1.7/etc, and contains <i>username/password</i> pairs that specify the identities that clients may use to access PgBouncer. Within the authentication file, the username and password must be specified within double-quotes, as shown below: <i>“user_name” “password”</i> To make changes to the identities that can access PgBouncer, you can edit the existing authentication file, or specify an alternate authentication file with the auth_file parameter. auth_type
auth_type	The authentication method used by PgBouncer. May be: md5, crypt, plain, trust or any. The default value is md5.
default_pool_size	The amount of user connections that are allowed to access the server. The default is 20 active connections.
group_connections	Clients providing the same application_name will be grouped to use the same connection. The default is 0.
ignore_startup_parameters	A comma-delimited list of application startup packets that PgBouncer should ignore. The default is application_name .
listen_addr	The IP address on which PgBouncer listens for client connections. If omitted, only Unix socket connections are allowed; the client must also reside on the same host as PgBouncer and may not specify a host IP address when connecting to PgBouncer.
listen_port	The port that PgBouncer monitors for client connections. By default, PgBouncer listens on port 6432.
logfile	The path to the PgBouncer log file.
max_client_conn	The maximum number of connections allowed. The default is 100.
pidfile	The path to the process ID file.
pool_mode	The value of pool_mode specifies when the server connection can be made available to the connection pool. May be: session, transaction or statement. The default value is session.
server_reset_query	The default is DISCARD ALL which instructs PgBouncer to clean any changes made to a database session.
stats_users	A comma delimited list of users who are allowed to connect and run read-only queries. The default is stats_users = enterprisedb.

The following example demonstrates the syntax allowed in the [pgbouncer] section of the configuration file:

```
[pgbouncer]
logfile = /var/log/edb/pgbouncer1.7/edb-pgbouncer-1.7.log
pidfile = /var/run/edb/pgbouncer1.7/edb-pgbouncer-1.7.pid
listen_addr = *
listen_port = 6432
auth_type = md5
auth_file = /opt/edb/pgbouncer-1.7/etc/userlist.txt
admin_users = enterprisedb
stats_users = enterprisedb
pool_mode = session
```

(continues on next page)

(continued from previous page)

```
server_reset_query = DISCARD ALL
ignore_startup_parameters = application_name
max_client_conn = 100
default_pool_size = 20
group_connections = 0
```

For more information about the settings used in the `pgbouncer.ini` file, click [here](#).

After editing the PgBouncer configuration file to reflect your environment, you must restart the PgBouncer service for the changes to take effect. The name of the PgBouncer service is `edb-pgbouncer-1.7`; use platform specific commands to stop, start, or restart the service as needed.

1.4.1 Using the PgBouncer Admin Console

The Admin Console allows you to retrieve statistical information about PgBouncer activity, and to control the PgBouncer process. You can use the `edb-psql` client to access the PgBouncer Admin Console by connecting to the `pgbouncer` database. The following example connects to the `pgbouncer` database with the `edb-psql` client on a Linux system. PgBouncer is listening on port 6432, with a user name of `enterprisedb`:

Enter following command after navigating to the `bin` directory under your Advanced Server installation:

```
# ./psql -p 6432 -U enterprisedb pgbouncer
```

Please note that the required connection information will vary according to the connecting client, platform and authentication information required by the server.

After connecting to the `pgbouncer` database, you can use the `SHOW CLIENTS` command to retrieve client-related information:

```
# SHOW CLIENTS;
```

The `SHOW CLIENTS` command returns:

```
--[ RECORD 1 ]--+-----
type           | C
user            | postgres
database       | pgbouncer
state          | active
addr           | unix
port           | 6432
local_addr     | unix
local_port     | 6432
connect_time   | 2010-05-25 05:26:20
request_time   | 2010-05-25 05:39:46
ptr            | 0x8655d20
link           |
```

You can use other variations of the `SHOW` command to retrieve information about PgBouncer:

`SHOW STATS`

`SHOW SERVERS`

`SHOW POOLS`

`SHOW LISTS`

`SHOW USERS`

`SHOW DATABASES`

SHOW FDS
SHOW CONFIG

You can use the following commands to control the PgBouncer process:

PAUSE

Use the PAUSE command to disconnect all servers after waiting for current queries to complete.

SUSPEND

Use the SUSPEND command to flush the socket buffers and suspend the PgBouncer process.

RESUME

Use the RESUME command to resume work after a PAUSE or SUSPEND command.

SHUTDOWN

Use the SHUTDOWN command to stop the PgBouncer process and exit.

RELOAD

Use the RELOAD command to reload the PgBouncer configuration files.

For more information about using PgBouncer, click [here](#).

C

Configuring PgBouncer, [12](#)

I

Installing and Configuring PgBouncer, [2](#)

Installing PgBouncer on a SLES 12
Host, [4](#)

P

Performing an RPM Installation, [2](#)

U

using pgbouncer admin console, [14](#)

Using the Graphical Installer, [5](#)