pgpool-II Guide

Version 1.0

Built at 2021-01-19T09:29:55

1 Installing pgpool-II

pgpool-II runs as a service on Linux systems. It is not supported on Windows systems.

The pgpool version required by your Advanced Server installation is version-specific, but the documented and supported functionality of each version is the same. The following table lists the pgpool version and their corresponding Advanced Server versions. The information in this guide applies to each version listed in the table below.

Supported pgpool-II and Advanced Server Versions

pgpool Version	Advanced Server Version	Supported Platforms
pgpool 4.1.2	Advanced Server 11 and 12	RHEL 6, RHEL 7, and RHEL 8 - x86_64, RHEL 7 - ppc64le, Debian 9x Stretch, Ubuntu 18.04 Bionic Beaver, SLES 12
pgpool 4.0.9	Advanced Server 11 and 12	RHEL 6, RHEL 7, and RHEL 8 - x86_64, RHEL 7 - ppc64le, Debian 9x Stretch, Ubuntu 18.04 Bionic Beaver, SLES 12
pgpool 3.7.14	Advanced Server 10 and 11	RHEL 6 and RHEL 7 - x86_64, RHEL 7 - ppc64le, Debian 9x Stretch, Ubuntu 18.04 Bionic Beaver
pgpool 3.6.21	Advanced Server 9.6 and 10	RHEL 6 and RHEL 7 - x86_64, RHEL 7 - ppc64le, Linux graphical installer
pgpool 3.5.25	Advanced Server 9.6	RHEL 6 and RHEL 7 - x86_64, RHEL 7 - ppc64le, Linux graphical installer

This guide assumes that the user has some knowledge of installation and system administration procedures, and has administrative privileges on the host.

Installing pgpool-II on a RHEL/CentOS Host

You can install pgpool-II using the `yum` command on a RHEL/CentOS 6 and 7 host, or using the `dnf` command on a CentOS/RHEL 8 host (see supported platforms table for details about supported platforms).

If you have previously used an RPM package to install Advanced Server, you have probably already created the repository configuration file and have EDB credentials. If that is the case, you can install pgpool with the command:

On RHEL/CentOS 6 and 7:

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

On RHEL/CentOS 8:

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

Where <xx> is the pgpool version you wish to install.

If you encounter an installation-related issue, see Troubleshooting for a workaround.

Creating a Repository and Installing pgpool-II

If you need to create an EDB repository configuration file, assume superuser privileges and follow the steps described below.

Step 1: Before installing `pgpool-II`, invoke the following command to install the `epel-release` package:

On RHEL/CentOS 6 and 7:

```
yum -y install epel-release
```

On RHEL/CentOS 8:

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

Note

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

Ensure you have credentials that allow access to the EDB repository. For information about requesting credentials, visit the EDB website.

Step 2: After receiving your repository credentials, create the repository configuration file using the following command:

On RHEL/CentOS 6 and 7:

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

On RHEL/CentOS 8:

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

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

Use your choice of editor to ensure that the `enabled` parameter value is `1`, and replace the `username` and `password` placeholders in the `baseurl` specification with the credentials of a registered EDB 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
```

Step 3: After creating and modifying the `edb.repo` file, execute the following commands to update the metadata:

```
yum clean all

yum makecache
```

Step 4: After saving your changes to the configuration file, use the following yum command to install `pgpool-II`:

On RHEL/CentOS 6 and 7:

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

On RHEL/CentOS 8:

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

Where <xx> is the pgpool release version.

For example, to install `pgpool Version 4.1.2`, execute the following command:

On RHEL/CentOS 6 and 7:

```
yum install edb-pgpool41
```

On RHEL/CentOS 8:

```
dnf install edb-pgpool41
```

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

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

Installation Troubleshooting

This section provides a workaround for the following installation issue:

```
existing lock /var/run/yum.pid: another copy is running as pid 3104.  Another app is currently holding the yum lock; waiting for it to exit
```

To fix this issue, execute the following command:

```
rm -f /var/run/yum.pid
```

Installing pgpool-II on a Debian/Ubuntu Host

To install pgpool-II on a Debian or Ubuntu host, you must have credentials that allow access to the EDB repository. To request credentials for the repository, visit the EDB website.

The following steps walks you through using the EDB apt repository to install a DEB package. When using the commands, replace the

`username` and `password` with the credentials provided by EDB.

Please note that if you are using the pdf version of this document, using cut/paste to copy command may result in extra spaces or carriage returns in the pasted command. If a command fails, check the command carefully for additional characters.

1. Assume superuser privileges:

   ```
   sudo su –
   ```

2. Configure the EDB repository:

   ```
   sh -c 'echo "deb
   https://username:password@apt.enterprisedb.com/$(lsb_releas
   e -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<xx>
   ```

Where <xx> is the pgpool release version.

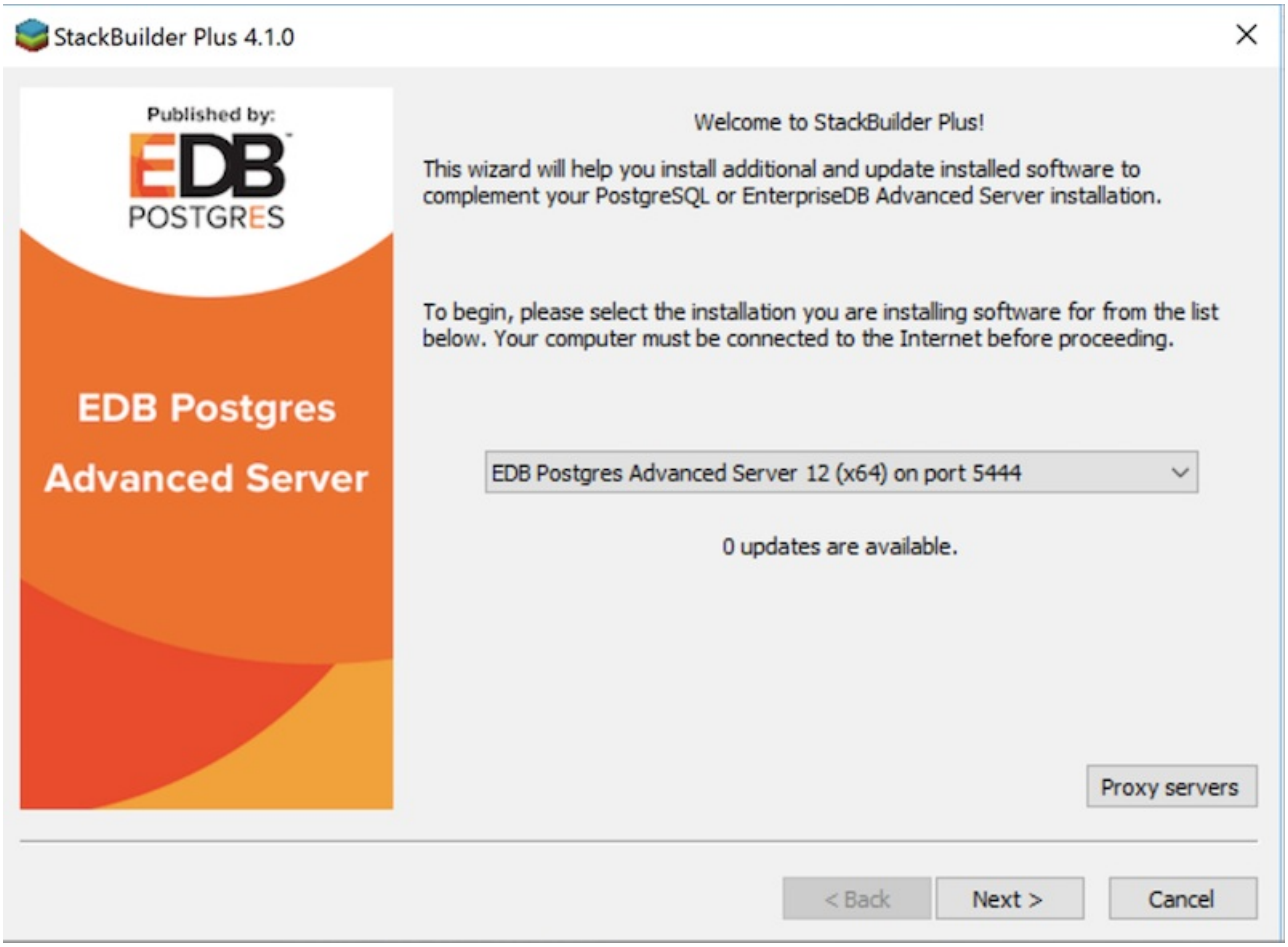
After installing pgpool-II on a Debian host, the configuration files are located in the `/etc/edb/edb-pgpool<x.x>` directory, where <x.x> is the pgpool release version. By default, `.sample` is appended to the configuration file name; `.sample` should be removed after copying the file to create your custom configuration.

Installing pgpool-II Using Linux Graphical Installer

Graphical installers for pgpool-II are available via StackBuilder Plus (for Advanced Server hosts) or Stack Builder (on PostgreSQL hosts). You can access StackBuilder Plus through your Linux start menu.

Perform the following steps to install pgpool-II:

1. Open StackBuilder Plus and select your Advanced Server installation from the drop-down list on the `Welcome` window. Click `Next` to continue to the application selection page.

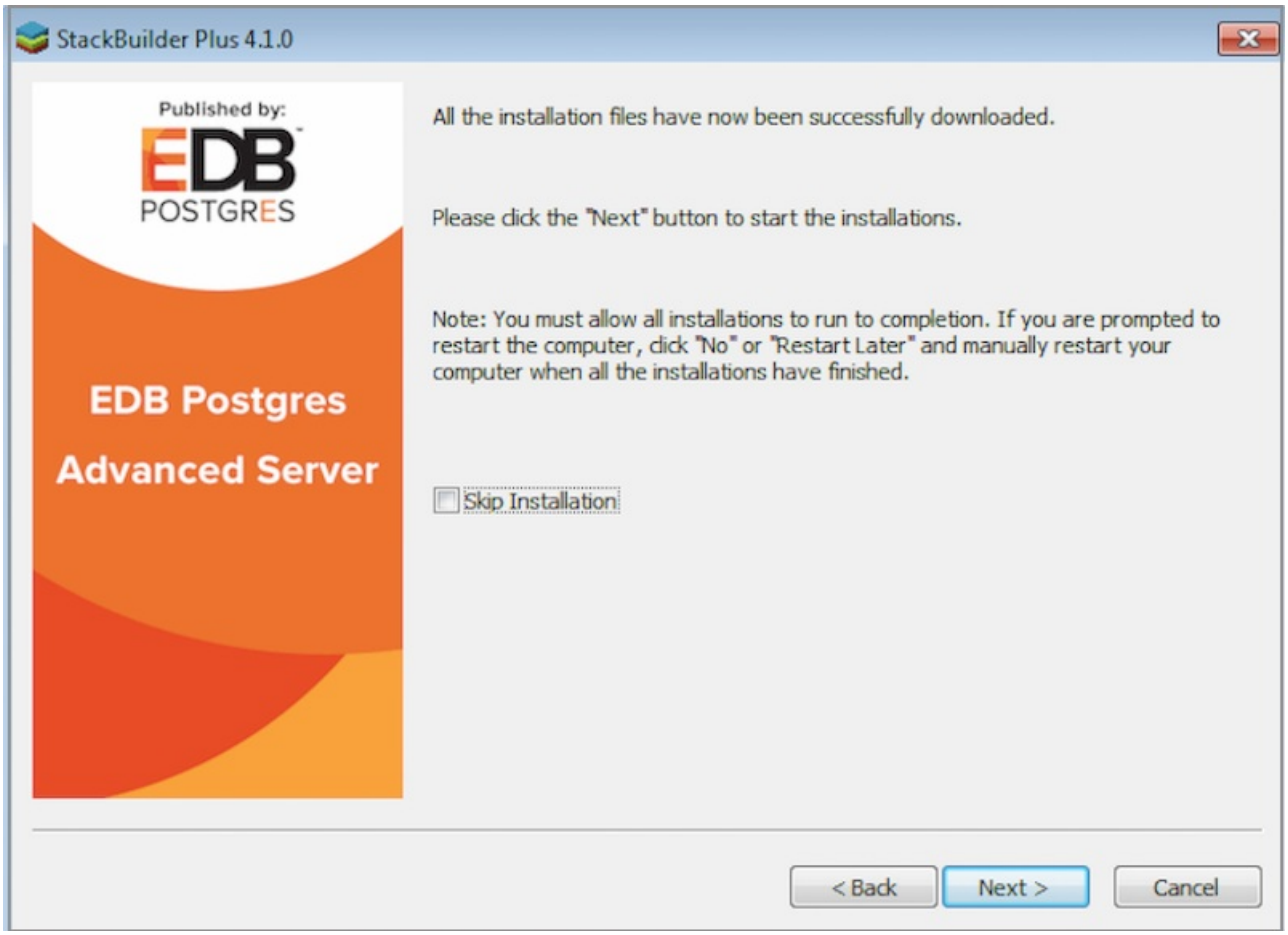


1. Expand the Add-ons, tools and utilities node, and check the box next to the pgpool-II version to select and download the pgpool-II installer. Click Next to continue.

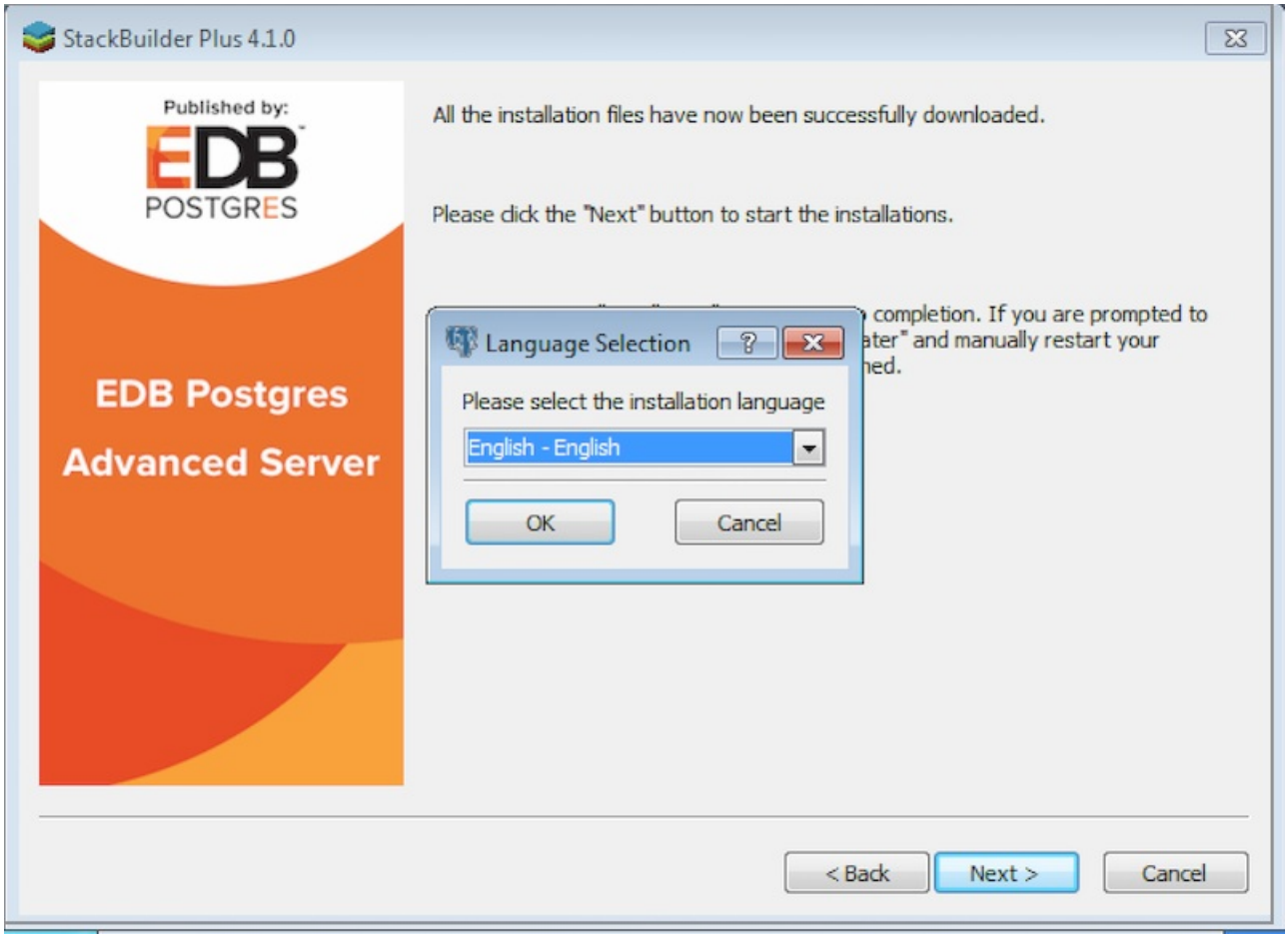
2. Provide the credentials and click Next.

3. The selected packages and the default download directory where the package will be installed are displayed; change the download directory location if required. Click Next.

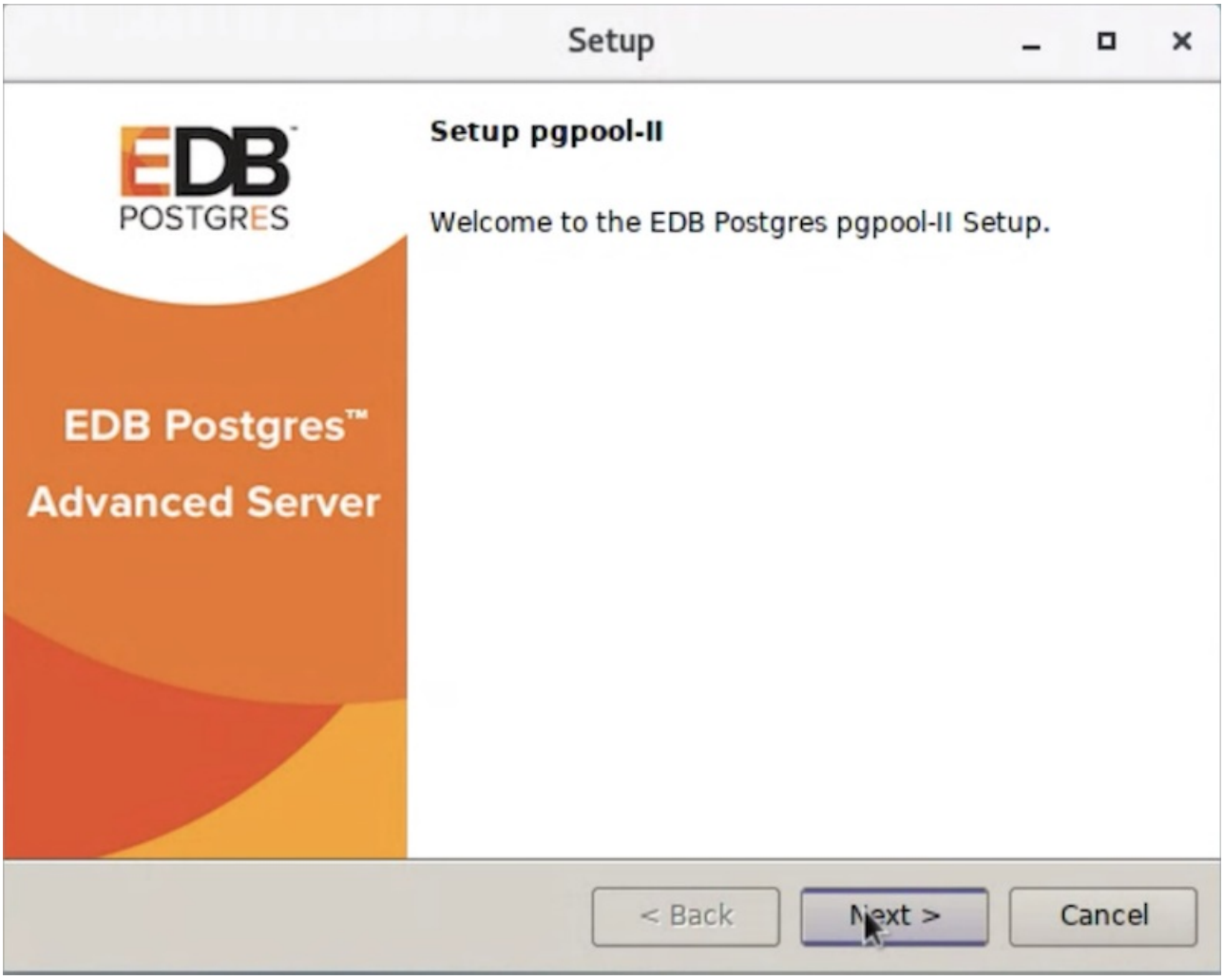
4. Once you have downloaded the installation files, a confirmation message is displayed. Click Next to start the pgpool-II installation.



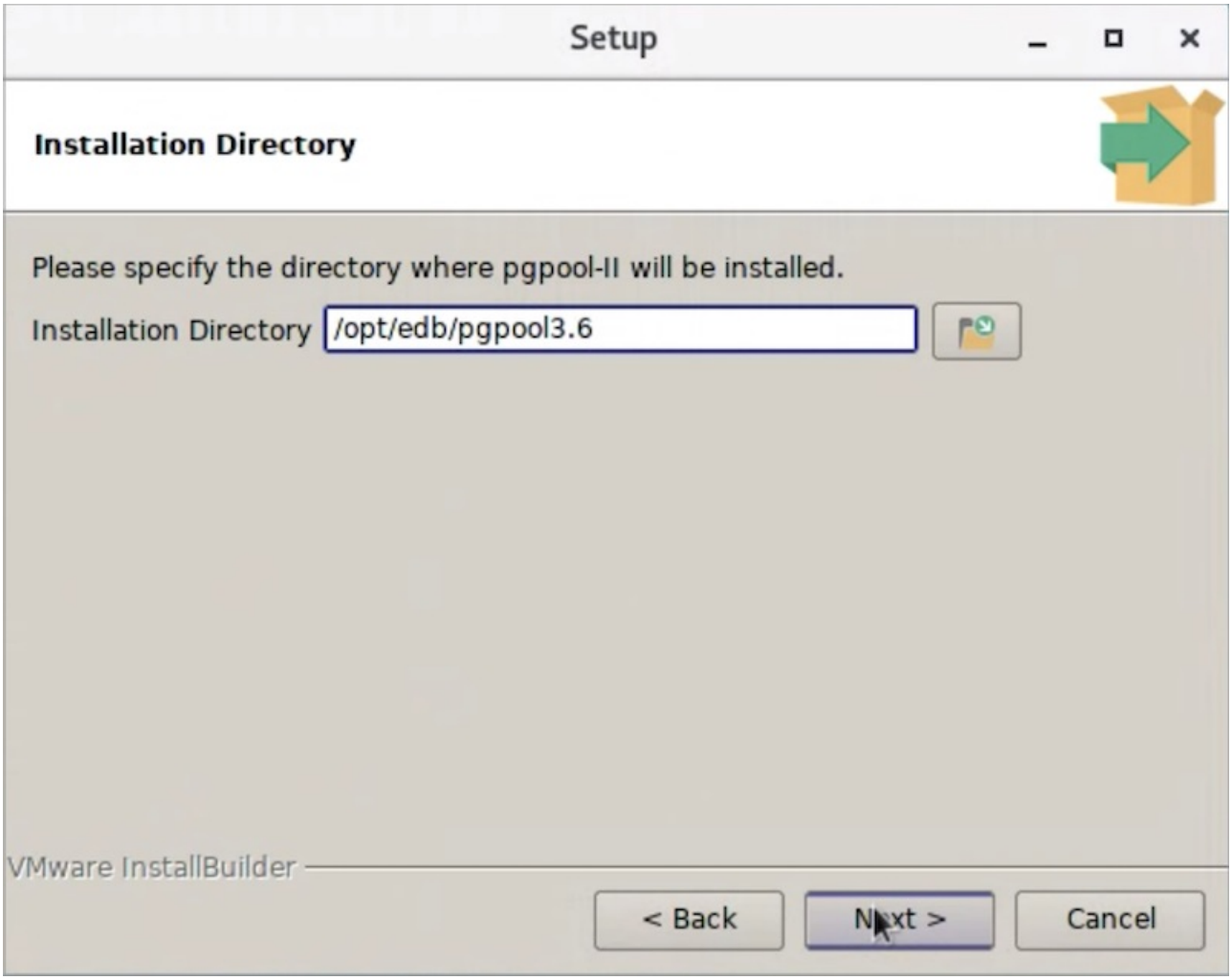
1. Select an installation language and click OK.



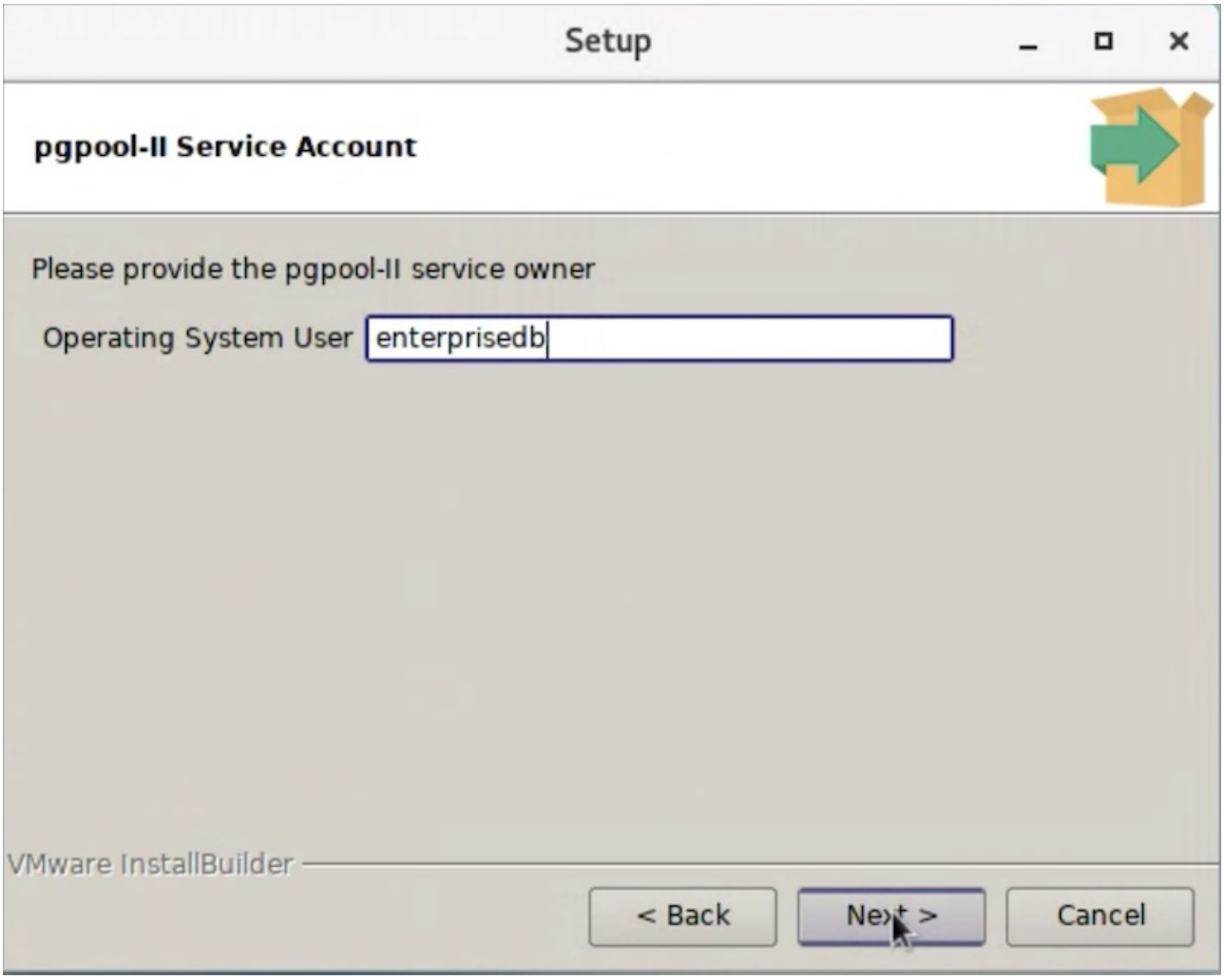
1. The pgpool installer welcomes you to the setup wizard.



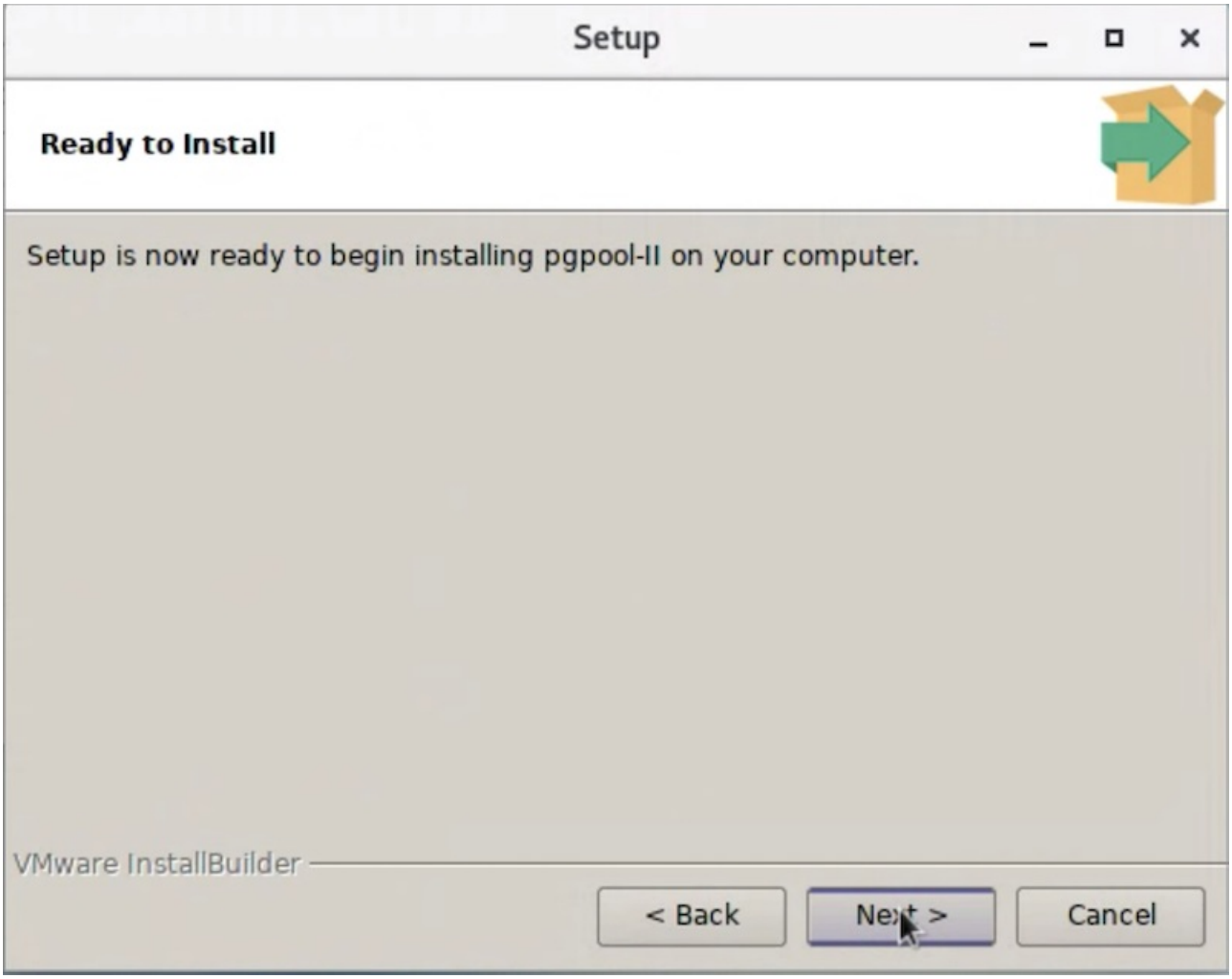
1. Use the Installation Directory field to specify the directory in which you wish to install the pgpool-II software (the default installation directory is /opt/edb) Then, click Next to continue.



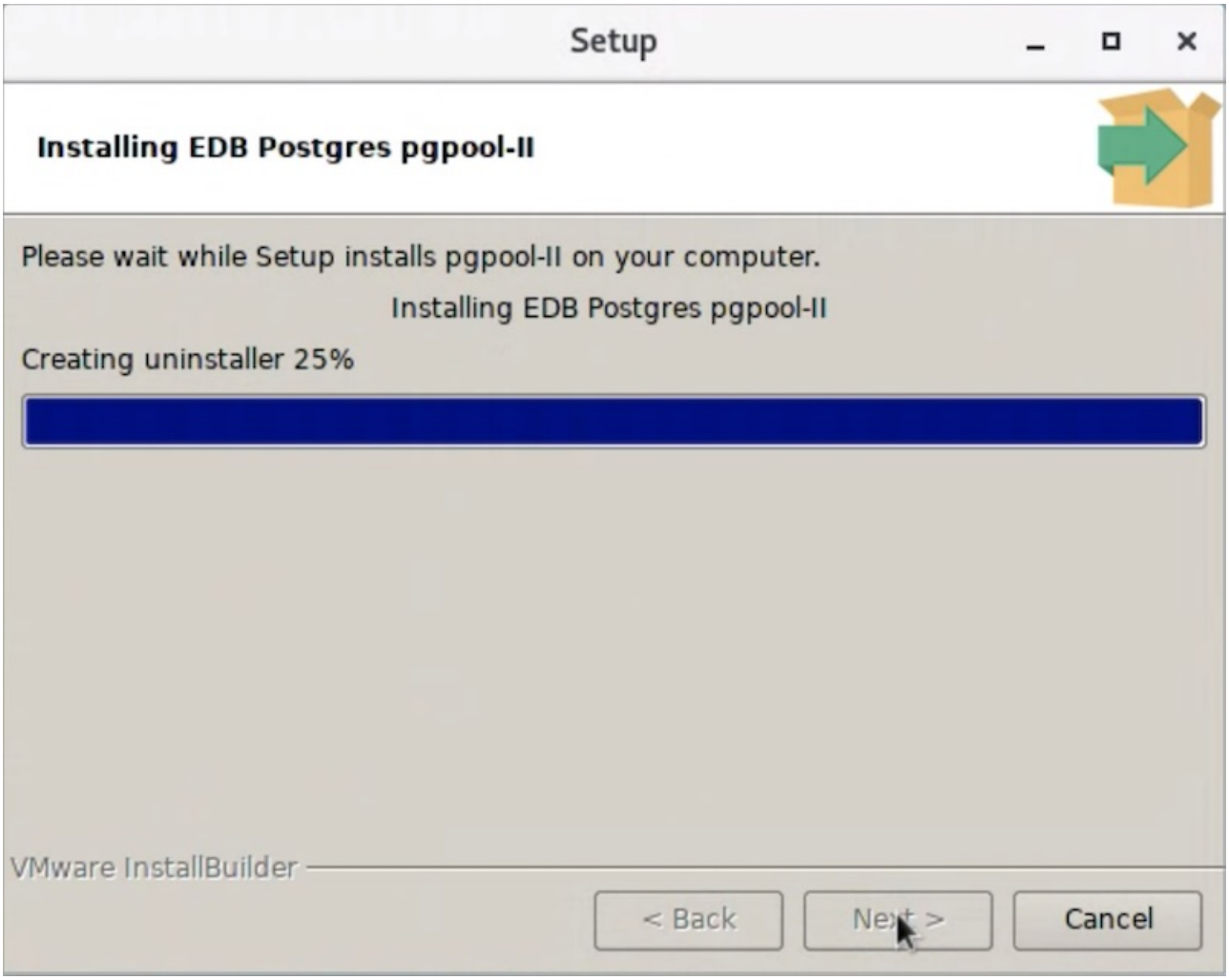
1. Use the Operating System User field to specify the name of the Linux operating system user that pgpool-II will change to after startup. Then, click Next to continue.



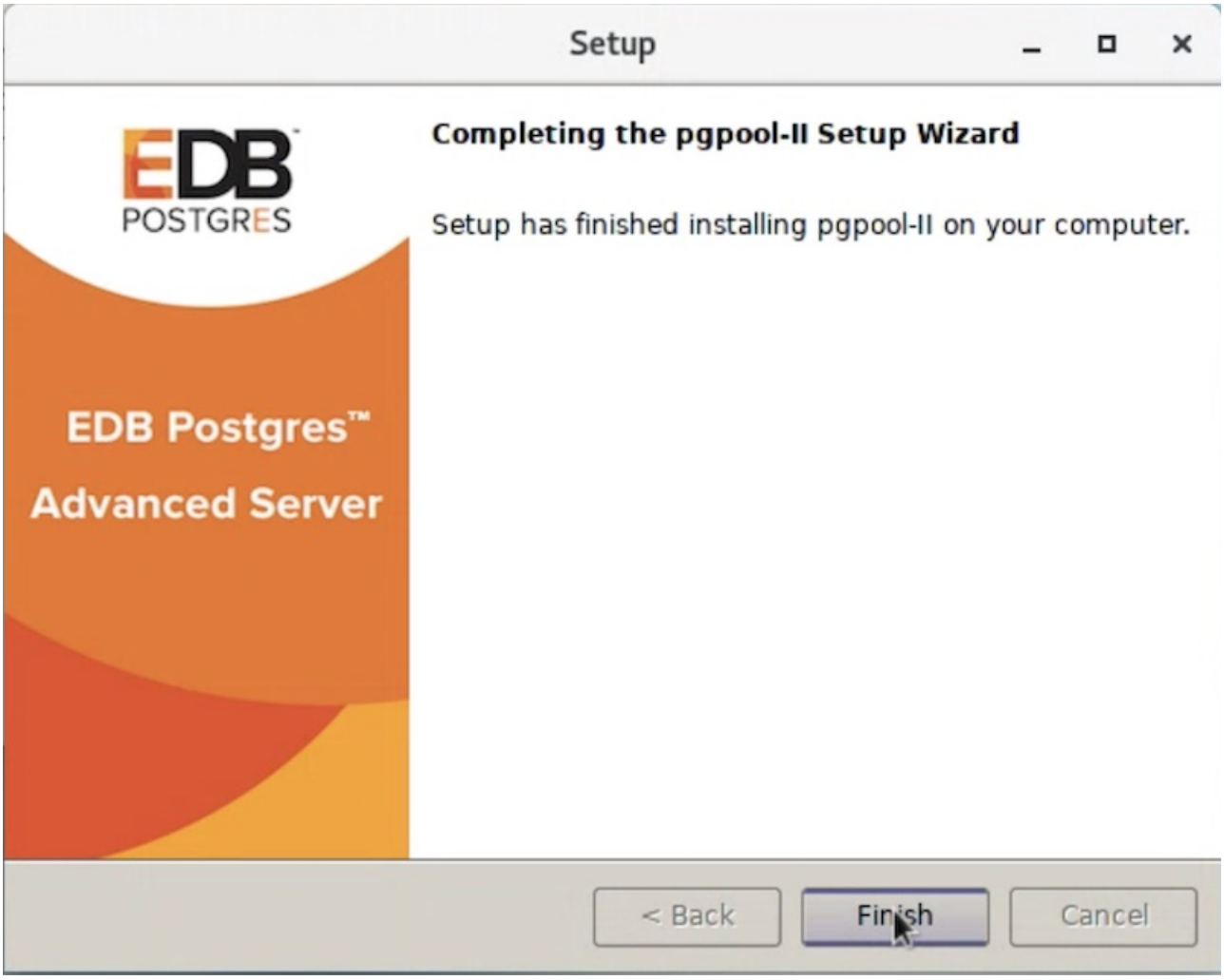
1. The Ready to Install window notifies you when the installer has all of the information needed to install pgpool-II on your system. Click Next to install pgpool-II.



1. Progress bars inform you as the installation progresses.



1. The installer notifies you when the setup wizard has completed the pgpool-II installation. Click Finish to exit the installer.



Installing pgpool-II Using an SLES 12 Host

You can use the zypper package manager to install pgpool-II 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 EDB.

1. Assume superuser privileges.

```
Sudo su -
```

1. Use the following command to add the EDB repository to your SLES host:

```
zypper addrepo https://zypp.enterprisedb.com/suse/edb-sles.repo
```

Ensure you have credentials that allow access to the EDB repository. For information about requesting credentials, click this link.

1. Use the following command to refresh the metadata on your SLES host to include the EDB repository:

```
zypper refresh
```

1. Then, use the zypper utility to install pgpool-II:

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

Where <xx> is the pgpool version you wish to install. Currently, 4.0.9 and 4.1.2 versions are supported on an SLES 12 host.

pgpool-II Host Setup

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 the *Managing an Advanced Server Installation* section of the EDB Postgres Advanced Server Installation Guide for Linux available at the EDB website.

When 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.x>/edb-pgpool-<x.x>.pid
```

Where <x.x> 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.

2 Installing and Managing Extensions

Modules in the extensions directory are additional features to Advanced Server, which are generally not included in the core database. Once loaded in a database, they can function just like built-in features. They allow you to use simple `SELECT` commands to use PCP remotely.

Before installing pgpool extensions, ensure that Advanced Server is installed on your host system. You can install the following pgpool-specific extensions using the following command:

- pgpool_adm <pgpool_adm>
- pgpool_recovery <pgpool_recovery>

Installing pgpool-II Extensions

The following section walks you through the steps of installing pgpool-II extensions.

Installing pgpool-II Extension on a RHEL/CentOS Host

Assume superuser privileges and execute the following command to install pgpool-II extensions:

On RHEL/CentOS 6 and 7:

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

For example, to install pgpool41 extensions for Advanced Server 12, execute the following command:

```
yum install edb-as12-pgpool41-extensions
```

On RHEL/CentOS 8:

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

In the above command, `<xx>` is the Advanced Server version, and `<yy>` is the pgpool-II extension version. The extensions will be available in the `/usr/edb/as<xx>/share/extension` directory.

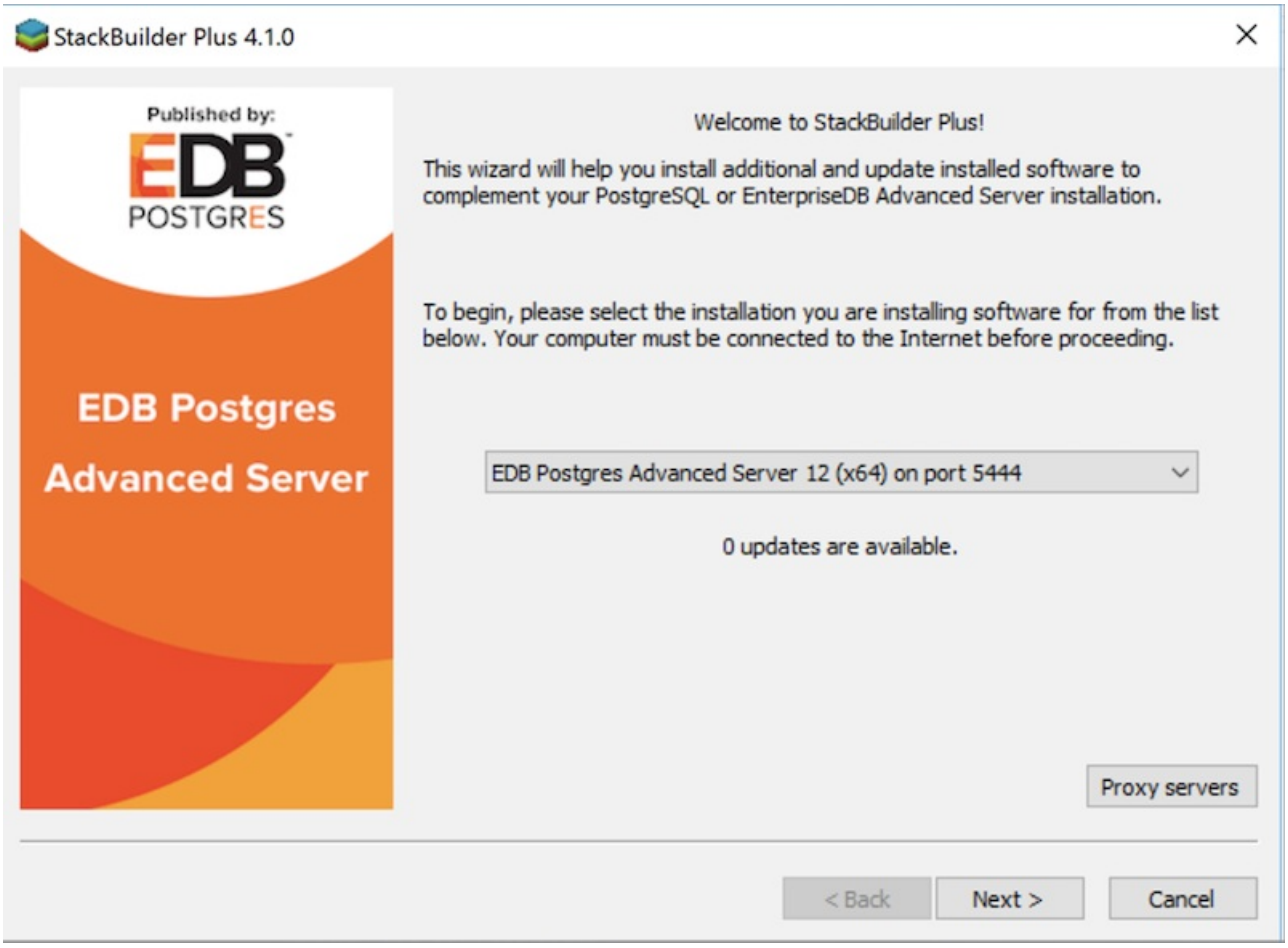
Currently, pgpool-II 4.0.<x> and 4.1.<x> extensions can be installed on RHEL8 platforms.

Installing pgpool-II Extension Using Linux Graphical Installer

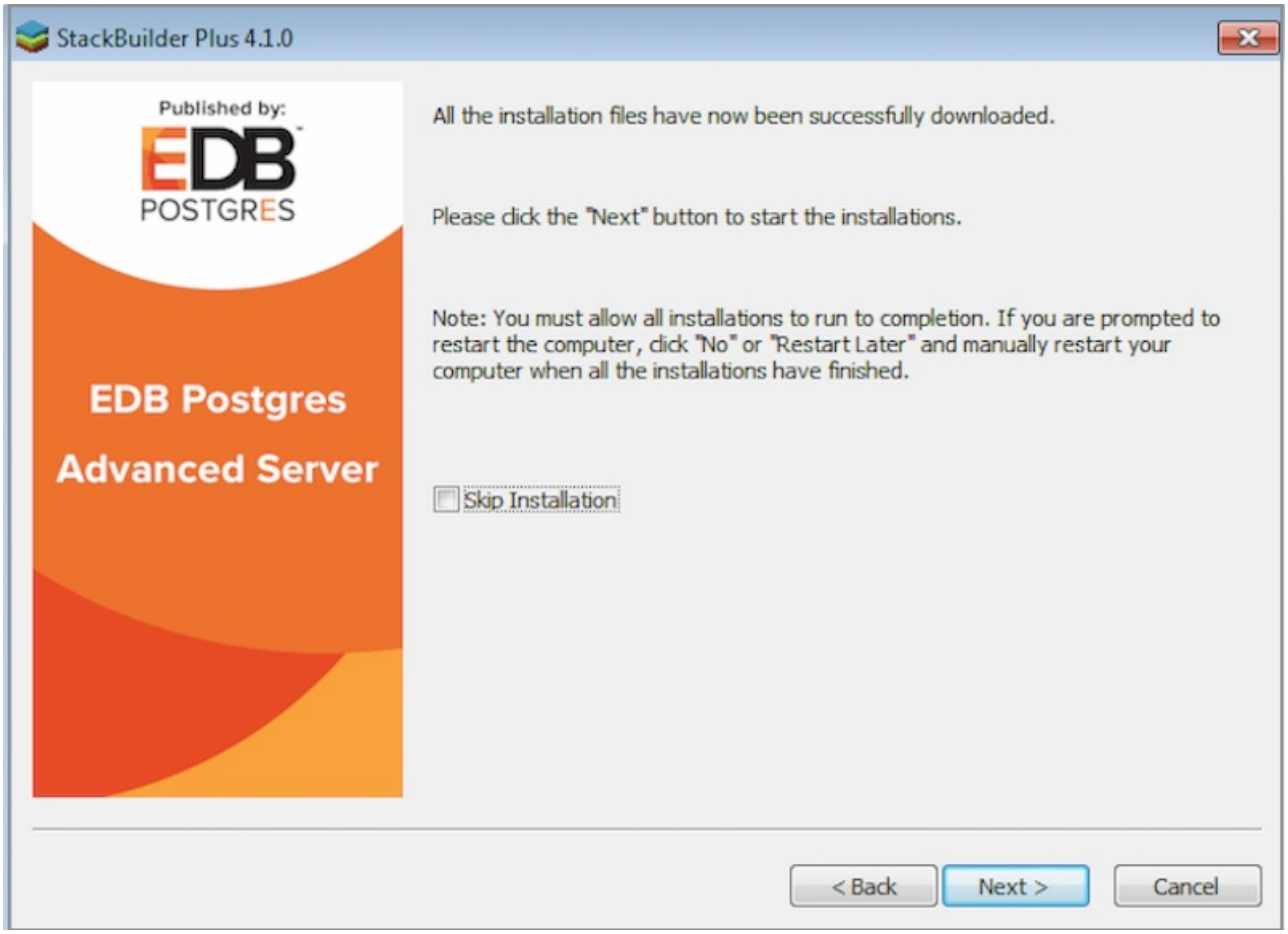
Graphical installers for pgpool-II extensions are available via StackBuilder Plus (for Advanced Server hosts) or Stack Builder (on PostgreSQL hosts). You can access StackBuilder Plus through your Linux start menu.

Perform the following steps to install pgpool-II extensions:

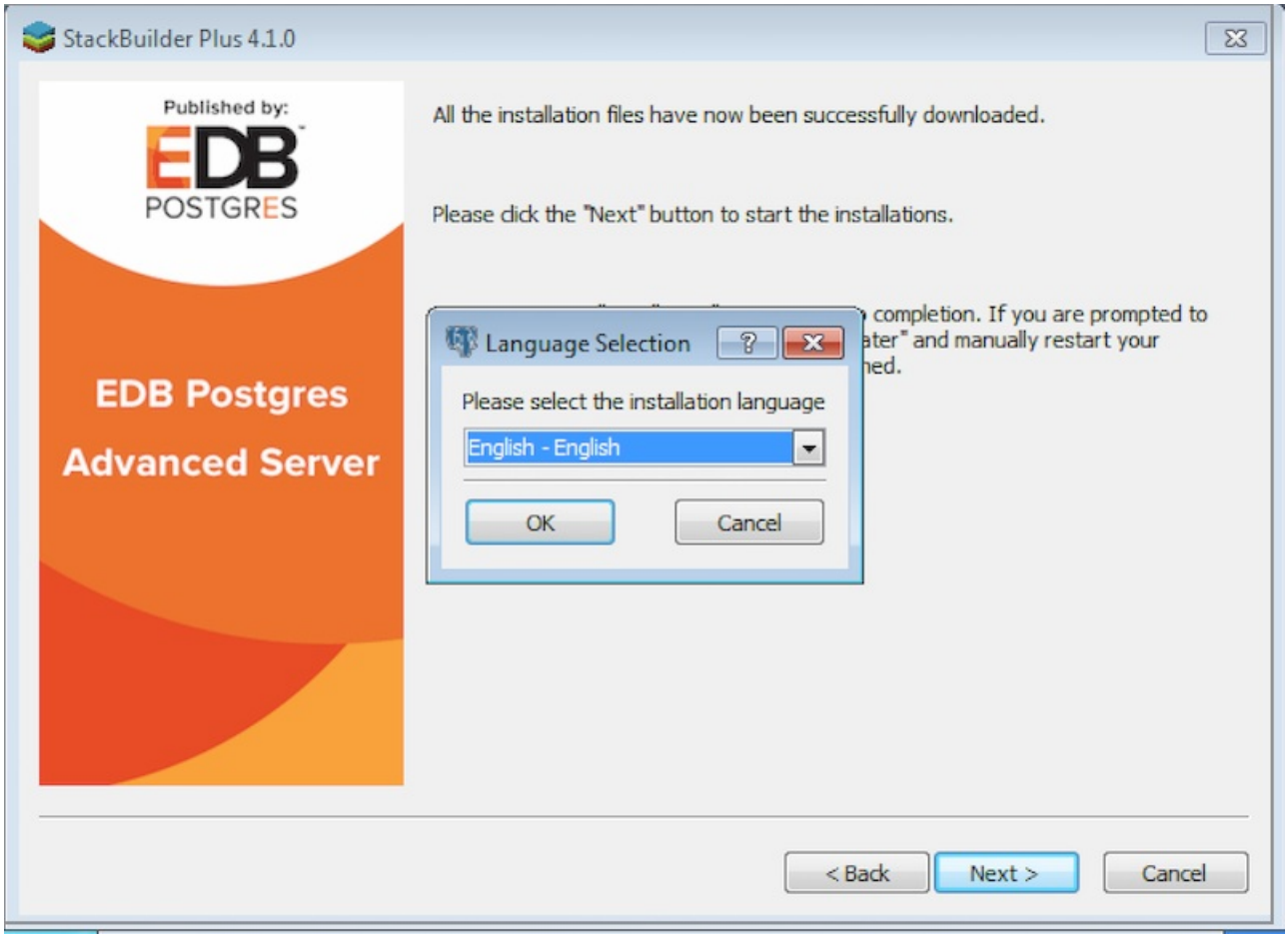
1. Open StackBuilder Plus and select your Advanced Server installation from the drop-down list on the `Welcome` window. Click `Next` to continue to the application selection page.



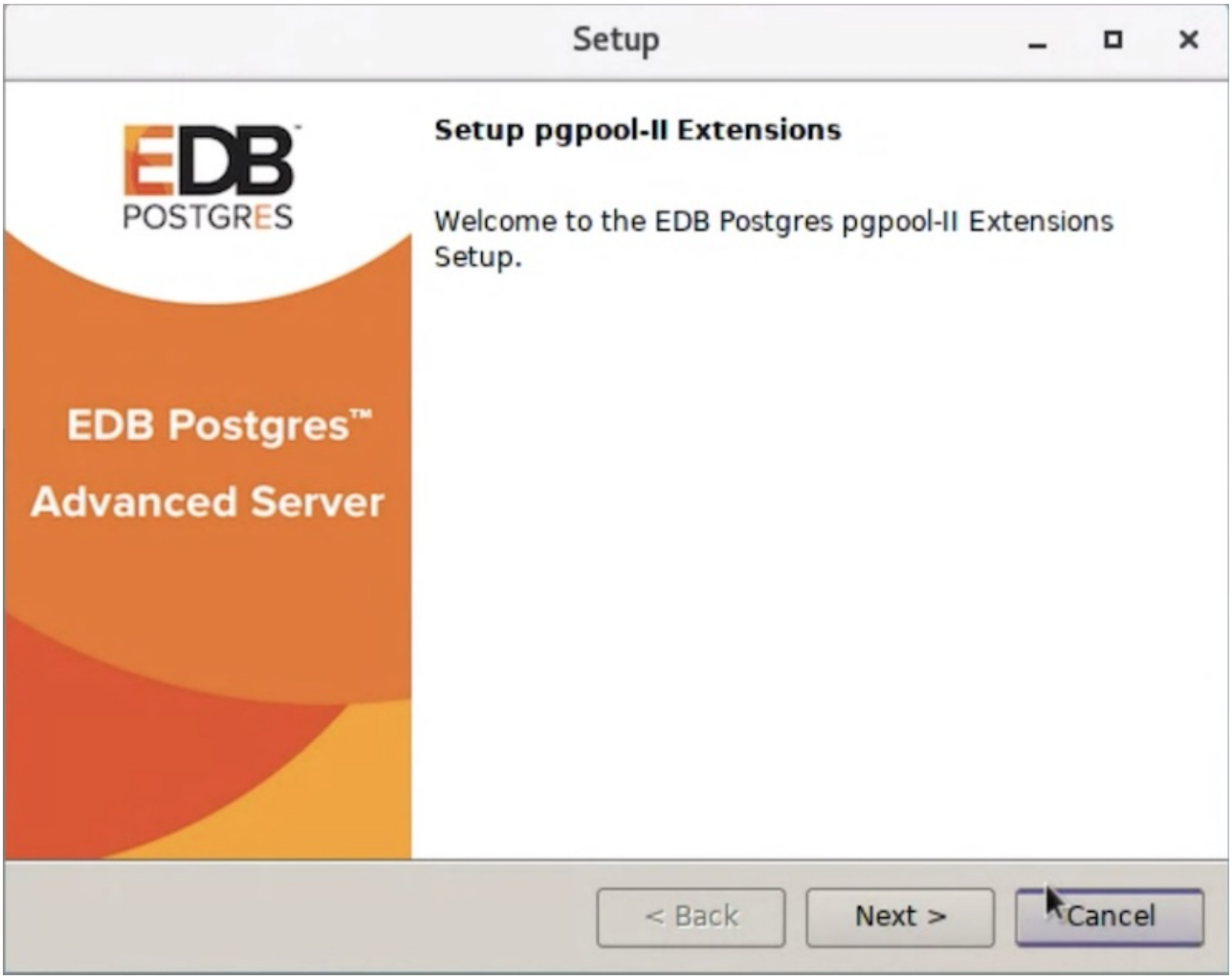
1. Expand the Add-ons, tools and utilities node, and check the box next to the pgpool-II extension to select and download the installer. Click Next to continue.
2. Provide the credentials and click Next.
3. The selected packages and the default download directory where the package will be installed are displayed; change the download directory location if required. Click Next.
4. Once you have downloaded the installation files, a confirmation message is displayed. Click Next to start the installation.



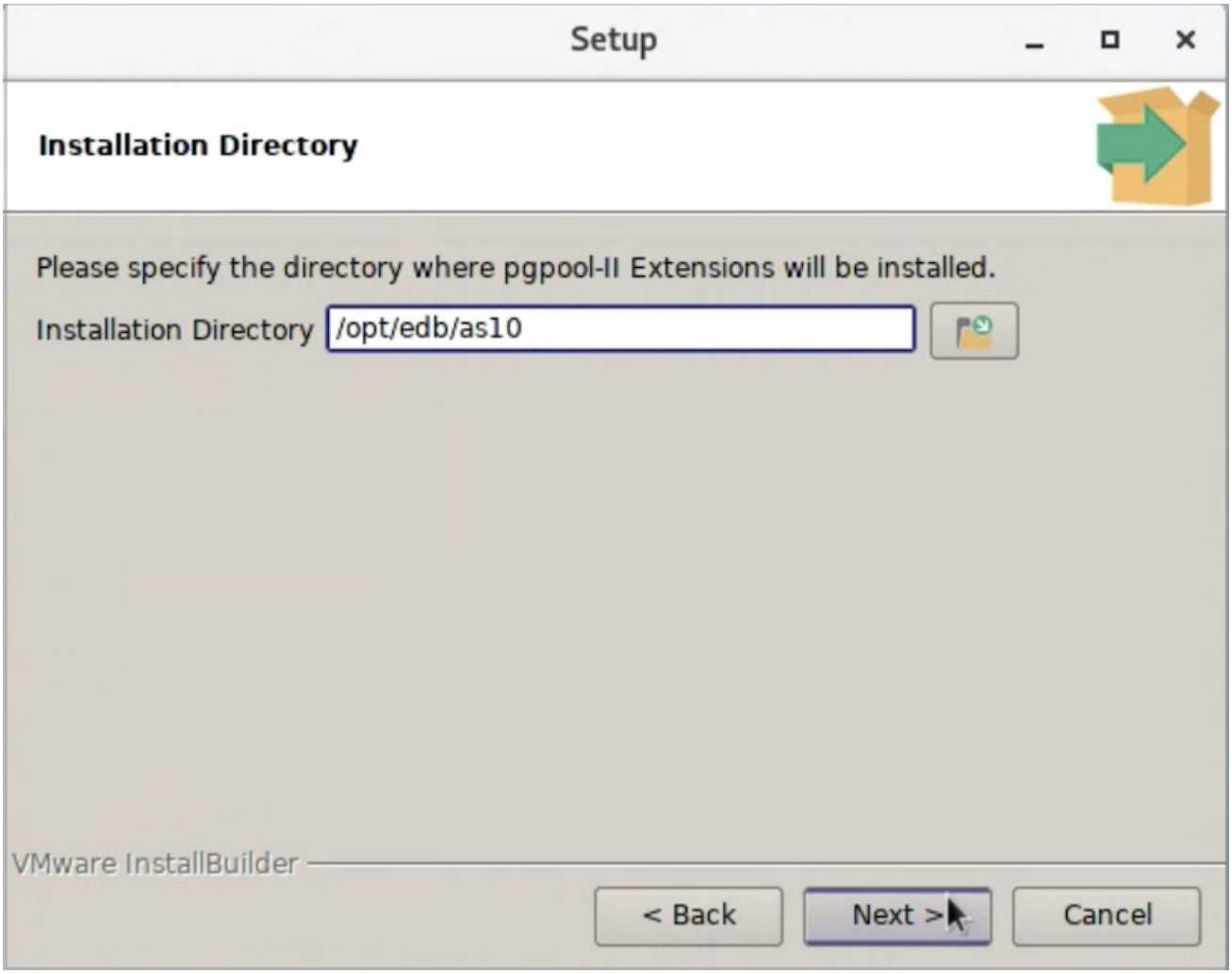
1. Select an installation language and click OK.



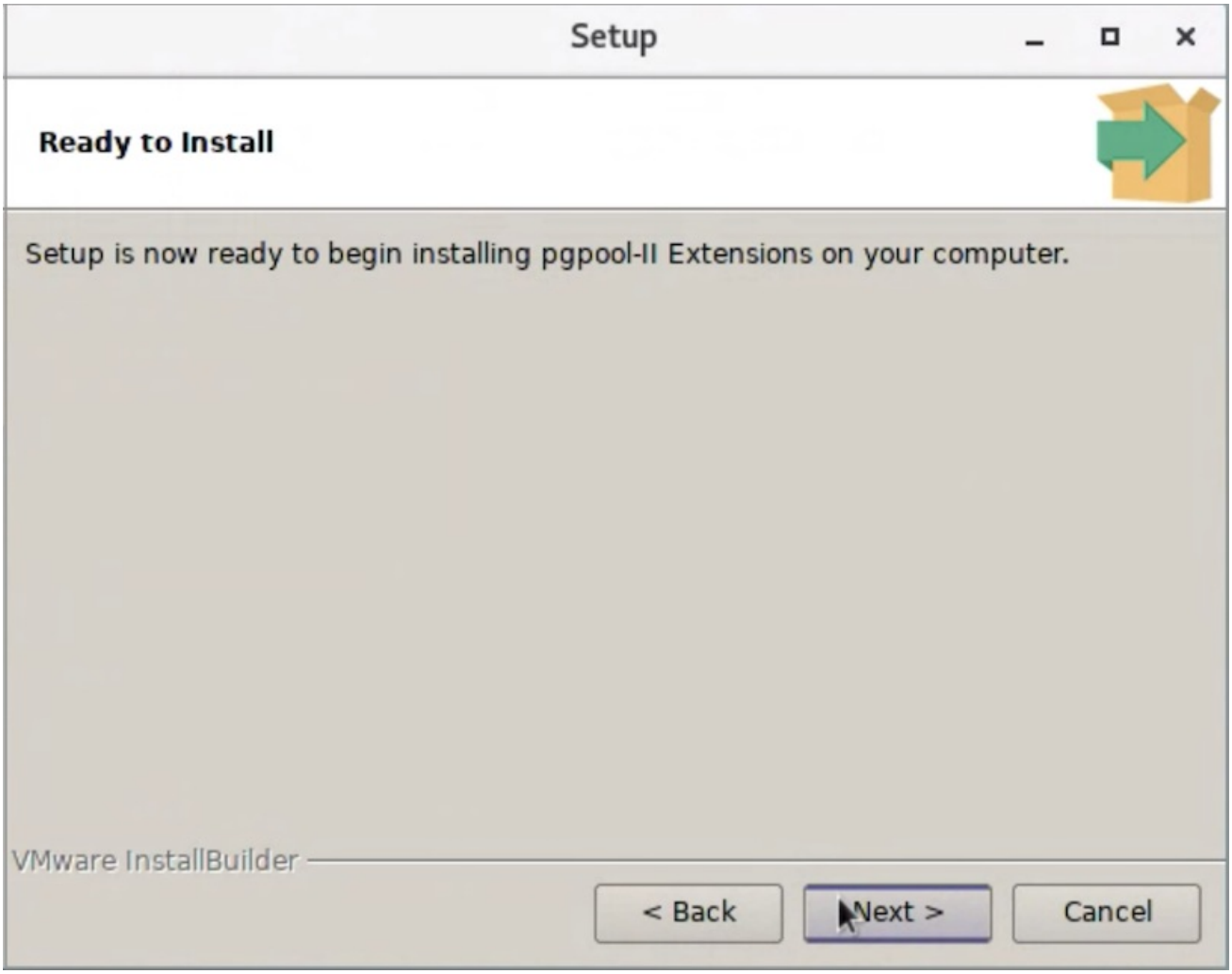
The pgpool extensions installer welcomes you to the setup wizard.



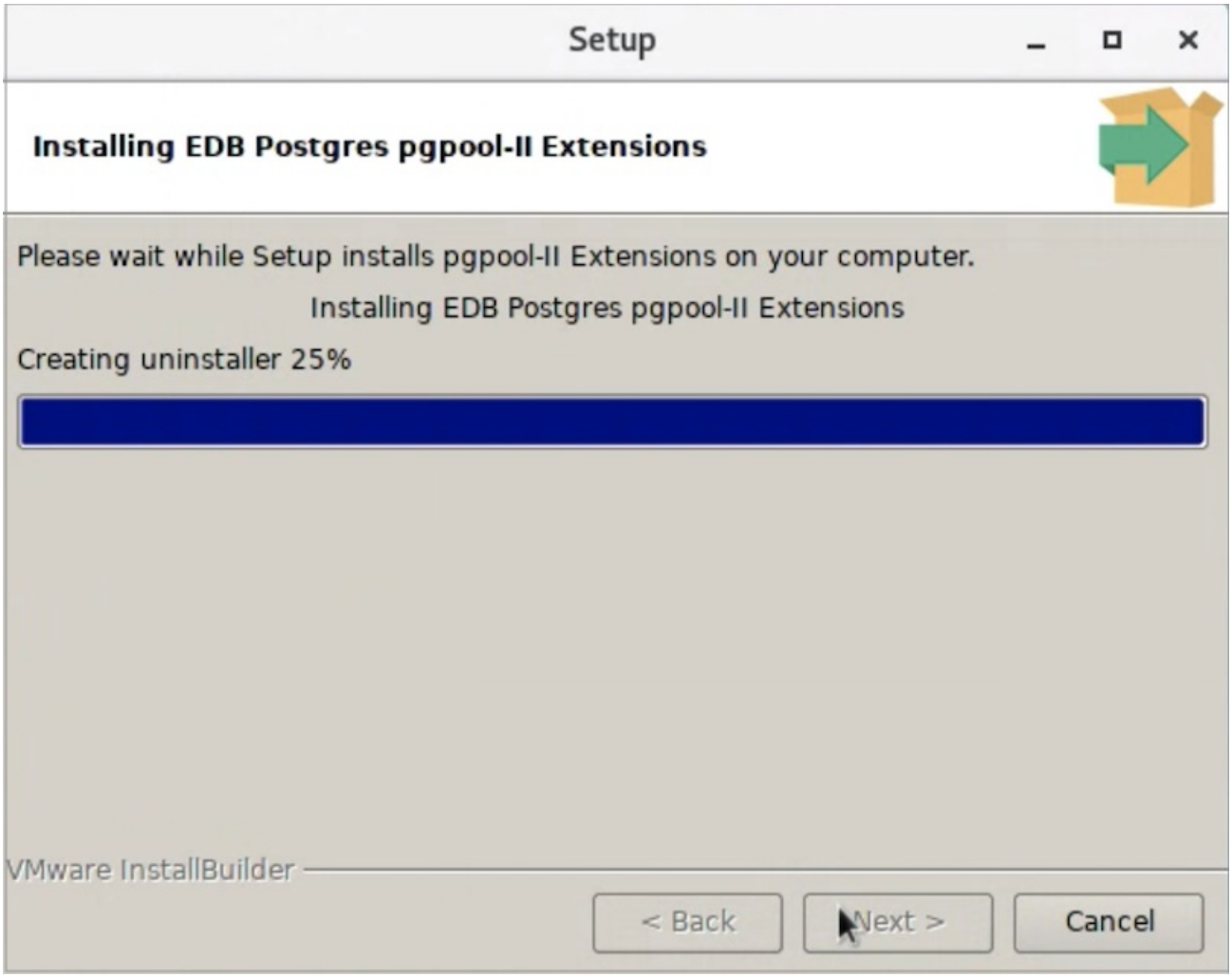
Use the Installation Directory field to specify the directory in which you wish to install the pgpool-II extensions software (the default installation directory is /opt/edb/as<xx>) Then, click Next to continue.



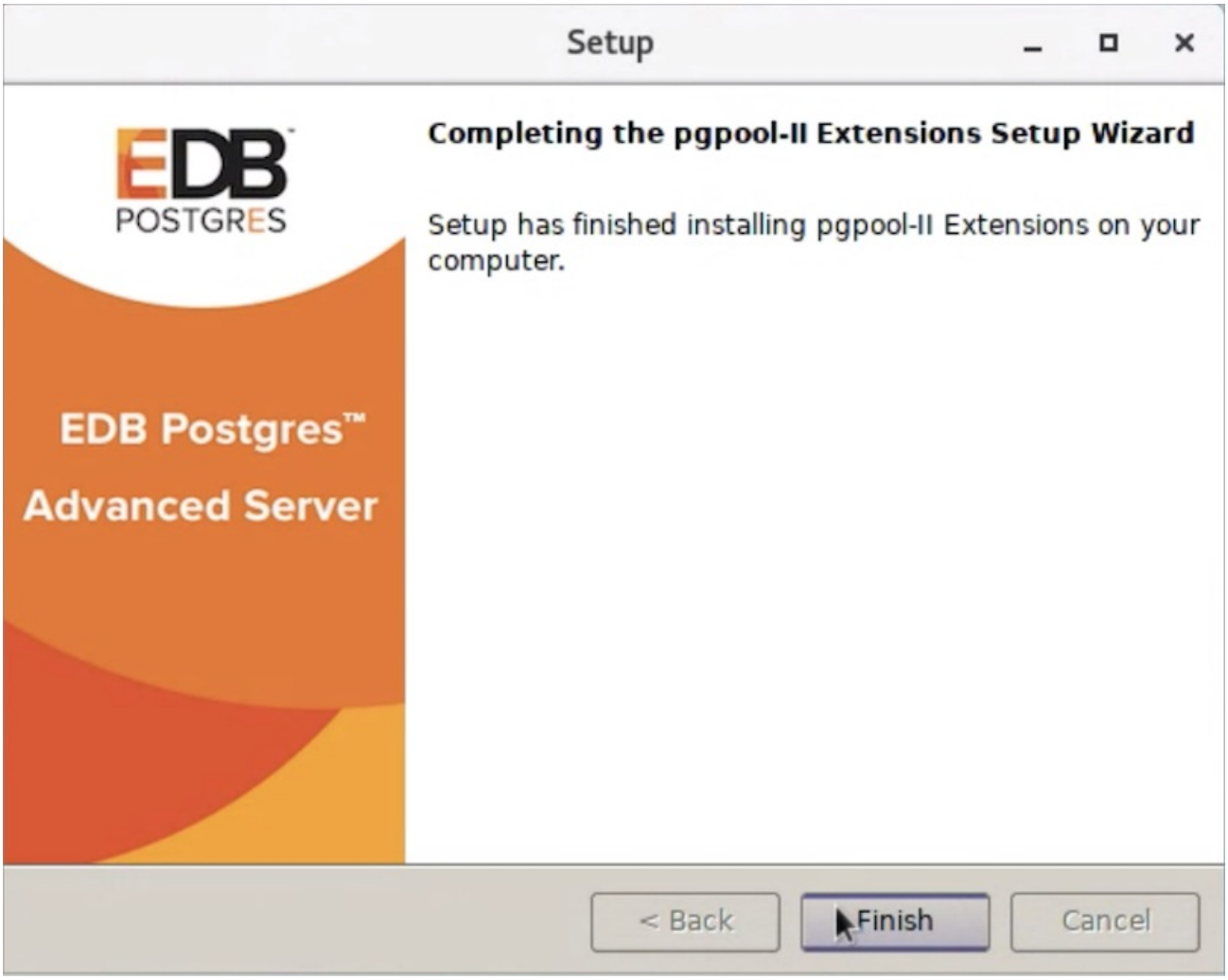
The Ready to Install window notifies you when the installer has all of the information needed to install pgpool-II extensions on your system. Click Next to install pgpool-II extensions.



Progress bars inform you as the installation progresses.



The installer notifies you when the setup wizard has completed the pgpool-II installation. Click Finish to exit the installer.



The extensions will be available in the `/opt/edb/as<xx>/share/extension/pgpool*` directory.

Installing pgpool-II Extension Using an SLES 12 Host

You can use the zypper package manager to install pgpool-II extension 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.

1. Assume superuser privileges.

```
Sudo su -
```

1. Use the following command to add the EnterpriseDB repository to

your SLES host.

```
zypper addrepo https://zypp.enterprisedb.com/suse/edb-sles.repo
```

Ensure you have credentials that allow access to the EnterpriseDB repository. For information about requesting credentials, click this link.

1. Use the following command to refresh the metadata on your SLES host to include the EnterpriseDB repository:

```
zypper refresh
```

1. Then, use the zypper utility to install pgpool-II extension:

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

Where <xx> is the advanced server version and <yy> is the pgpool version you wish to install. Currently, 4.0.9 and 4.1.2 versions are supported on Advanced server 11 and 12 on an SLES 12 host.

Creating pgpool-II Extensions

You must install and create the extensions in each database in which you will be using pgpool-II functionality. 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 will be created in each of the database that uses `template1` as a template during creation.

pgpool_adm Extension

The `pgpool_adm` extension is available for the following pgpool-II versions.

3.5.24, 3.5.25, 3.6.20, 3.6.21, 3.7.13, 3.7.14, 4.0.9, and 4.1.2

`pgpool_adm` is a set of extensions that allows SQL access to PCP commands. To view information about PCP commands, see https://www.pgpool.net/docs/41/en/html/pcp-commands.html.

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;
```

To view more information about `pgpool_adm`, see https://www.pgpool.net/docs/41/en/html/pgpool-adm.html.

pgpool_recovery Extension

The `pgpool_recovery` extension is required for online recovery and future fail-back mechanism.

After installing the `pgpool_recovery` extension, use psql to connect to the database, and execute the following SQL command to create a `pgpool_recovery` extension:

```
CREATE EXTENSION pgpool_recovery;
```

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

3 Configuring pgpool-II

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.

Commonly Used pgpool-II Parameters

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

Parameter Name	Description
pool_conn_dbname	Database name to which pgpool-II will connect. By default, pgpool-II will connect with Postgres.
listen_addresses	Host name or IP address used by pgpool-II to listen for connections. Default is localhost. Change to '*' for all addresses.
port	Port for pgpool-II connections. Default is 9999.
pcp_port	Port for PCP connections. Default is 9898.
backend_hostname0	Host name or IP address for backend 0. You can specify '' if the backend and 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 the load equally balanced, or decimal values (.9, .1, etc.) to weight the load towards certain backends.
backend_data_directory0	Data directory for backend 0.
enable_pool_hba	Set to on to use pool_hba.conf for client authentication.

Parameter Name	Description
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.

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

Parameter Name	Description
Allow_sql_comments	If on, ignore SQL comments; modifications to this parameter require a reload of 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 proportion of SELECT statements each backend receives is determined by parameter backend_weight<N>.
ignore_leading_white_space	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.

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 primary node.

3. Optionally, configure pgpool-II client authentication.

4. Optionally, configure the PCP administrative interface.

5. Start pgpool-II and begin using your application using the following command:

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

> where <x.y> is the pgpool release version.

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, 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 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 hostname (or IP address) and the 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 primary 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.

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/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/CentOS 7 platforms. To restart the server on RHEL/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';
```

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 primary 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 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 at:

```
https://www.postgresql.org/docs/13/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 primary node, and contains the replication parameters specified previously.

Then, restart the server:

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

Note

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

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

At this point, the primary node will be replicating data to the standby node.

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 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/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.

Restart pgpool-II and begin using your application

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

where <x.y> is the pgpool release version.

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.

pgpool-II Client Authentication

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.

Parameter settings in the `pool_hba.conf` configuration file determine the pgpool-II authentication properties. The `pool_hba.conf` file is similar in format and function to the Postgres `pg_hba.conf` configuration file. Please refer to 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.

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

where <x.y> is the pgpool release version.

Note

When authenticating with the database server, use the user names and passwords specified in the `pool_hba.conf` file; you must also specify those user names and passwords in the database server's `pg_hba.conf` file.

Configuring PCP

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

`pcp.conf` is the password configuration file for the PCP client. Before using PCP commands, you must modify the `pcp.conf` file, providing

the user names and passwords that you supply when invoking 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.

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

   where <x.y> is the pgpool release version.

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 Command	Description
`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 client_idle_limit_in_recovery is -1 or recovery_timeout is expired).
`pcp_attach_node timeout host port username password nodeid`	Attaches the node specified by <nodeid> to pgpool-II.

4 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 display pgpool-II information:

Command	Information Provided
`SHOW pool_status`	Displays pgpool-II configuration parameters and their name, value, and description.

Command	Information Provided
SHOW pool_nodes	Displays a list of all configured nodes.
SHOW pool_processes	Displays a list of all pgpool-II processes waiting for connections or dealing with a connection.
SHOW pool_pools	Displays a list of pools.
SHOW pool_version	Displays the pgpool-II release number.

5 Upgrading pgpool-II and extensions

The following section outlines the process of upgrading pgpool and pgpool extensions.

Upgrading pgpool-II

The following section outlines the process of upgrading pgpool-II minor version (for example, to upgrade from 3.7.13 to 3.7.14):

Assume the identity of the root user and invoke the following command:

On RHEL/CentOS 6 and 7:

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

where <xx> is the pgpool version you want to upgrade to. For example, to upgrade from pgpool 3.7.13 to 3.7.14, execute the following command:

```
yum upgrade edb-pgpool37
```

On RHEL/CentOS 8:

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

Upgrading pgpool-II Extensions

The following section outlines the process of pgpool-II extensions.

To upgrade from older versions of pgpool extensions to the latest version, assume superuser privileges and execute the following command:

On RHEL/CentOS 6 and 7:

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

Where `<xx>` is the Advanced Server version, and `<yy>` is the pgpool 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).

On RHEL/CentOS 8:

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

6 Uninstalling pgpool-II and Extensions

The following section outlines the process of uninstalling pgpool-II and its extensions.

Uninstalling pgpool-II on a RHEL/CentOS Host

To uninstall pgpool-II, assume the identity of the root user and invoke the following command:

On RHEL/CentOS 6 and 7:

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

On RHEL/CentOS 8:

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

Where `<xx>` is the pgpool version.

Uninstalling pgpool-II on a Debian/Ubuntu Host

To uninstall pgpool-II on a Debian/Ubuntu host, invoke the following command:

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

Where `<xx>` is the pgpool version you want to uninstall

Uninstalling pgpool-II Using a Linux Uninstaller

The pgpool-II graphical installer creates an uninstaller that you can use to remove pgpool-II. The uninstaller is created in the installation directory that you have specified while installing pgpool. If you have used the default installation directory i.e. `/opt/edb`, then uninstaller will be in the `/opt/edb/pgpool<x.x>` (where <x.x> is the pgpool version you have installed).

1. Navigate into the directory that contains the uninstaller and assume superuser privileges. Open the uninstaller and click `Yes` to begin uninstalling pgpool-II.
2. The uninstallation process begins. Click `OK` when the uninstallation completes.

Uninstalling pgpool-II Extensions

The following section outlines the process of uninstalling pgpool-II and its extensions.

Uninstalling pgpool-II Extensions on a RHEL/CentOS Host

To remove extensions from the server, execute the following command:

On RHEL/CentOS 6 and 7:

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

On RHEL/CentOS 8:

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

Where <xx> is the Advanced Server version, and <yy> is the pgpool-II extension version.

Uninstalling pgpool-II Extesnions Using Linux Graphical Uninstaller

The pgpool-II extensions graphical installer creates an uninstaller that you can use to remove pgpool-II extensions. The uninstaller is created in the installation directory that you have specified. If you have used the default installation directory i.e. /opt/edb/as<xx>, then uninstaller will be in the /opt/edb/as<xx> (where <xx> is the advanced server version you have installed).

1. Navigate into the directory that contains the uninstaller and assume superuser privileges. Open the uninstaller and click Yes to begin uninstalling pgpool-II extensions.
2. The uninstallation process begins. Click OK when the uninstallation completes.