

pgPool II Guide 1.0

EnterpriseDB

“Pgpool-II Guide”

Pgpool-IIは、クライアントアプリケーションとPostgreSQLデータベースサーバの間の仲介者として機能します。

Pgpool-IIを使用すると、アプリケーション接続インフラストラクチャに次の利点が追加されます。

- 透過的に再ルーティングし、読み取り専用トランザクションのスタンバイデータベースサーバーへの負荷分散を保証します。
- 再接続を防ぐために接続を再利用します
- 古い接続をキューに入れてPostgres接続を停止します
- フェールオーバーマネージャーと統合してプライマリをフォロー

EDBは、次のPgpool-II機能をサポートしています。

- 負荷分散
- 接続プーリング
- レプリケーション
- 高可用性
- 接続制限
- ウォッチドッグ
- 超過接続の制限
- メモリ内クエリキャッシュ

このガイドで使用されている規約

このガイドでは：

Postgresという用語は、PostgreSQLまたはEDB Postgres Advanced Serverを指します。

Note * このドキュメントのpdfバージョンを使用している場合、カット/貼り付けコマンドを使用すると、貼り付けたコマンドに余分なスペースまたは改行が含まれる場合があります。コマンドが失敗した場合は、追加の文字がないかコマンドを慎重に確認してください。

新機能

1. Pgpool-IIおよび2.2とのアップストリームマージ。4.2はメジャーリリースです。4.2への移行の詳細については、を参照してください。

connecting_a_client_to_pgpool-IIアップグレードuninstalling_pgpool-II結論を
configuring_connection_pooling installing_and_configuring_the_pgpool-IIの拡張機能

</ div>

Installing Pgpool-II

</ div>

Pgpool-IIは、Linuxシステム上でサービスとして実行されます。 WindowsシステムはPgpoolをサポートしていません。 EDB Postgres Advanced ServerとPostgreSQLのインストールに必要なPgpoolバージョンはバージョン固有ですが、文書化されサポートされている各バージョンの機能は同じです。 次の表に、Pgpoolのバージョンと、対応するEDB Postgres Advanced ServerおよびPostgreSQLのバージョンを示します。 このガイドの情報は、以下の表にリストされている各バージョンに適用されます。

Pgpool Version	Postgres Version	Supported Platforms
Pgpool 4.2	EDB Postgres Advanced Server and PostgreSQL 13	RHEL 7 - x86_64, RHEL 8 - x86_64 RHEL 7 - ppc64le Debian 9x Stretch and 10x Buster Ubuntu 18.04 LTS Bionic Beaver and Ubuntu 20.04 LTS Focal Fossa
Pgpool 4.2	EDB Postgres Advanced Server and PostgreSQL 12	RHEL 7 - x86_64, and RHEL 8 - x86_64 RHEL 7 - ppc64le Debian 9x Stretch and 10x Buster Ubuntu 18.04 LTS Bionic Beaver SLES 12
Pgpool 4.2	EDB Postgres Advanced Server and PostgreSQL 11	RHEL 7 - x86_64, RHEL 8 - x86_64 RHEL 7 - ppc64le Debian 9x Stretch Ubuntu 18.04 LTS Bionic Beaver SLES 12
Pgpool 4.2	EDB Postgres Advanced Server and PostgreSQL 9.6 and 10	RHEL 7 - x86_64 and RHEL 7 - ppc64le
Pgpool 4.1	EDB Postgres Advanced Server 13	RHEL 7 - x86_64, RHEL 8 - x86_64 RHEL 7 - ppc64le Debian 9x Stretch and 10x Buster Ubuntu 18.04 LTS Bionic Beaver and Ubuntu 20.04 LTS Focal Fossa

Pgpool Version	Postgres Version	Supported Platforms
Pgpool 4.1	EDB Postgres Advanced Server 12	RHEL 7 - x86_64, and RHEL 8 - x86_64 RHEL 7 - ppc64le Debian 9x Stretch and 10x Buster Ubuntu 18.04 LTS Bionic Beaver SLES 12
Pgpool 4.1	EDB Postgres Advanced Server 11	RHEL 7 - x86_64 RHEL 7 - ppc64le Debian 9x Stretch Ubuntu 18.04 LTS Bionic Beaver SLES 12
Pgpool 4.0	EDB Postgres Advanced Server 12	RHEL 7 - x86_64, and RHEL 8 - x86_64 RHEL 7 - ppc64le Debian 9x Stretch and 10x Buster Ubuntu 18.04 LTS Bionic Beaver SLES 12
Pgpool 4.0	EDB Postgres Advanced Server 11	RHEL 7 - x86_64 RHEL 7 - ppc64le Debian 9x Stretch Ubuntu 18.04 LTS Bionic Beaver SLES 12
Pgpool 3.7	EDB Postgres Advanced Server 10 and 11	RHEL 7 - x86_64 and RHEL 7 - ppc64le
Pgpool 3.6	EDB Postgres Advanced Server 9.6 and 10	RHEL 7 - x86_64 RHEL 7 - ppc64le Linux graphical installer

Note * -PgpoolはCentOS / RHEL / OEL 6.xプラットフォームをサポートしていません。EDBは、CentOS / RHEL / OEL 6.xプラットフォームで実行されているEDB製品をサポートされているプラットフォームに移行することをお勧めします。-Pgpoolは、EDB Postgres Advanced ServerおよびPostgreSQLに同梱されているPgpool-II拡張で認証されています。

このガイドでは、ユーザがインストールおよびシステム管理手順についてある程度の知識を持ち、ホストに対する管理権限を持っていることを前提としています。

リポジトリ構成をインストールする前に、EDBリポジトリへのアクセスを許可する資格情報が必要です。資格情報の要求については、[EDB website](#)をご覧ください。

CentOSホストへのPgpool-IIのインストール

CentOSホストにPgpool-IIをインストールするには、次の手順を実行します。

1.リポジトリ構成をインストールするには、スーパーユーザ特権を引き受けて、以下のプラットフォーム固有のコマンドのいずれかを呼び出します。

- CentOS 7の場合：

```
yum -y install https://yum.enterprisedb.com/edbrepos/edb-repo-latest.noarch.r
```

- CentOS 8の場合：

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

2.次のコマンドの USERNAME:PASSWORD 変数を、登録済みのEDBユーザ名とパスワードに置き換えます。

```
sed -i "s@<username>:<password>@USERNAME:PASSWORD@" /etc/yum.repos.d/edb.repo
```

3. Pgpool-IIをインストールする前に、次のコマンドを実行、Enterprise Linux（EPEL）の追加パッケージをインストールします。

- CentOS 7の場合：

```
yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.no
```

- CentOS 8の場合：

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

4. CentOS 8の場合、PowerToolsリポジトリを有効にして、追加のパッケージ依存関係を満たします。

```
dnf config-manager --set-enabled PowerTools
```

5. CentOS 8の場合、ビルトインのPostgreSQLモジュールを無効にします。

```
dnf -qy module disable postgresql
```

6.プラットフォーム固有のコマンドを使用して、Pgpool-IIをインストールします。

- CentOS 7の場合：

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

- CentOS 8の場合：

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

- ここで<xx>はPgpoolリリースバージョンです。

例、最新のpgpoolのバージョン4.2をインストールするには、次のコマンドを実行します。

- CentOS 7の場合：

```
yum install edb-pgpool42
```

- CentOS 8の場合：

```
dnf install edb-pgpool42
```

- システムで認識されないソースによって署名されたRPMパッケージをインストールすると、ローカルサーバにキーをインポートするパーミッションが求められる場合があります。プロンプトが表示され、パッケージソフトが信頼できるソースからのものであることに満足したら、`y` と入力し、`Return` を押して続行します。
- Pgpool-IIは `/usr/edb/pgpool<x.y>/` ディレクトリにインストールされます。ここで<xy>はインストールされたPgpool-IIバージョン番号です。設定ファイルは `/etc/sysconfig/edb/pgpool<x.y>` ディレクトリに作成されます。<xy>はPgpoolリリースバージョンです。デフォルトでは、`.sample` が構成ファイル名前に追加されます。カスタム設定を作成し、ファイルをコピーした後、構成ファイルから `.sample` を削除します。

</div>

RHELホストへのPgpool-IIのインストール

リポジトリ構成ファイルを作成する前に、EDBリポジトリへのアクセスを許可する資格情報が必要です。資格情報の要求については、[EDB website](#)をご覧ください。

Pgpool-IIをインストールするには、次の手順を実行します。

1.リポジトリ構成ファイルを作成するには、スーパーユーザ権限を引き受けて、次のプラットフォーム固有のコマンドのいずれかを起動します。

- RHEL 7の場合：

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

- RHEL 8の場合：

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

2.次のコマンドの `USERNAME:PASSWORD` 変数を、登録済みのEDBユーザ名とパスワードに置き換えます。

```
sed -i "s@<username>:<password>@USERNAME:PASSWORD@" /etc/yum.repos.d/edb.repo
```

3. Pgpoolをインストールする前に、次のコマンドを実行、Enterprise Linux (EPEL) の追加パッケージをインストールします。

- RHEL 7の場合：

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

- RHEL 8の場合：

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

4.リポジトリを有効にします。

- RHEL 7では、`optional`、`extras` および `HA` リポジトリを有効にして、追加のパッケージ依存関係を満たします。

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

- RHEL 8では、`codeready-builder-for-rhel-8-*-*rpm` リポジトリを有効にして、追加のパッケージ依存関係を満たします。

```
ARCH=$( /bin/arch )
```

```
subscription-manager repos --enable "codeready-builder-for-rhel-8-${ARCH}-rpm"
```

5. RHEL 8の場合、ビルトインのPostgreSQLモジュールを無効にします。

```
dnf -qy module disable postgresql
```

6. プラットフォーム固有のコマンドを使用して、Pgpool-IIをインストールします。

- RHEL 7の場合：

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

- RHEL 8の場合：

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

- ここで<xx>はPgpoolリリースバージョンです。

例、最新のpgpoolのバージョン4.2をインストールするには、次のコマンドを実行します。

- RHEL 7の場合：

```
yum install edb-pgpool42
```

- RHEL 8の場合：

```
dnf install edb-pgpool42
```

- システムで認識されないソースによって署名されたRPM/パッケージをインストールすると、ローカルサーバにキーをインポートするパーミッションが求められる場合があります。プロンプトが表示され、パッケージソフトが信頼できるソースからのものであることに満足したら、 `y` と入力し、 `Return` を押して続行します。
- Pgpool-IIは `/usr/edb/pgpool<x.y>/` ディレクトリにインストールされます。<xy>はインストールされたPgpool-IIのバージョン番号です。設定ファイルは `/etc/sysconfig/edb/pgpool<x.y>` ディレクトリに作成されます。<xy>はPgpoolリリースバージョンです。デフォルトでは、 `.sample` が構成ファイル名前に追加されます。カスタム設定を作成し、ファイルをコピーした後、構成ファイルから `.sample` を削除します。

</div>

Pgpool-IIをRHEL / CentOS 7 PPCLEホストにインストールする

リポジトリ構成ファイルを作成する前に、EDBリポジトリへのアクセスを許可する資格情報が必要です。資格情報の要求については、[EDB website](#)をご覧ください。

次の手順を実行して、RHEL / CentOS 7 PPC64LEホストにPgpool-IIをインストールします。

1. Advance Toolchainをインストールします。

```
rpm --import https://public.dhe.ibm.com/software/server/POWER/Linux/toolchain

cat > /etc/yum.repos.d/advance-toolchain.repo <<EOF

# Beginning of the configuration file
[advance-toolchain]
name=Advance Toolchain IBM FTP
baseurl=https://public.dhe.ibm.com/software/server/POWER/Linux/toolchain/at/r
failovermethod=priority
enabled=1
gpgcheck=1
gpgkey=ftp://public.dhe.ibm.com/software/server/POWER/Linux/toolchain/at/redh
# End of the configuration file
```

2.リポジトリ構成ファイルを作成するには、スーパーユーザ権限を引き受けて、次のコマンドを呼び出します。

```
yum -y install https://yum.enterprisedb.com/edbrepos/edb-repo-latest.noarch.r
```

3.次のコマンドの `USERNAME:PASSWORD` 変数を、登録済みのEDBユーザ名とパスワードに置き換えます。

```
sed -i "s@<username>:<password>@USERNAME:PASSWORD@" /etc/yum.repos.d/edb.repo
```

4. Pgpool-IIをインストールする前に、次のコマンドを実行、Enterprise Linux (EPEL) の追加パッケージをインストールします。

```
yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.no
```

5. RHEL 7では、`optional`、`extras` および `HA` リポジトリを有効にして、追加のパッケージ依存関係を満たします。

```
subscription-manager repos --enable "rhel-*-optional-rpms" --enable "rhel-*-e
```

6.次のコマンドを呼び出して、Pgpool-IIをインストールします。

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

- ここで `<xx>` は、インストールするPgpool-IIバージョンです。

</div>

Debian / UbuntuホストへのPgpool-IIのインストール

DebianまたはUbuntuホストにPgpool-IIをインストールするには、EDBリポジトリへのアクセスを許可する資格情報が必要です。リポジトリの資格情報を要求するには、[EDB website](#)にアクセスします。

次の手順では、EDB aptリポジトリを使用してDebianパッケージをインストールします。コマンドを使用する場合、`username` および `password` をEDBが提供する資格情報に置き換えます。

1.スーパーユーザ特権を想定します。

```
sudo su -
```

2.システムのEDBリポジトリへのアクセスを設定します。

- Debian 9、Ubuntu 18、およびUbuntu 20：

```
sh -c 'echo "deb https://username:password@apt.enterprisedb.com/$(lsb_release
```

- Debian 10の場合：

- a。 EDBリポジトリをセットアップします。

```
sh -c 'echo "deb [arch=amd64] https://apt.enterprisedb.com/$(lsb_release -cs)
```

- b。 次のコマンドで、`username` および `password` プレースホルダーをEDB資格情報に置き換えます。

```
sh -c 'echo "machine apt.enterprisedb.com login <username> password <password
```

3.安全なAPTリポジトリのサポートをシステムに追加します。

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

4. EDB署名キーを追加しキー。

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

5.リポジトリメタデータを更新します。

```
apt-get update
```

6. Debianパッケージをインストールします。

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

ここで<xx>はPgpoolリリースバージョンです。

DebianホストにPgpool-IIをインストールした後、構成ファイルは `/etc/edb/edb-pgpool<x.y>` ディレクトリにあります。ここで<xy>はPgpoolリリースバージョンです。デフォルトでは、`.sample` が構成ファイル名前に追加されます。ファイルをコピーしてカスタム構成を作成した後、`.sample` を削除する必要があります。

</div>

SLES 12ホストへのPgpool-IIのインストール

Zypperパッケージマネージャを使用して、SLES 12ホストにPgpool-IIをインストールできます。Zypperはパッケージの依存関係を満たそうとしますが、EDBがホストしない特定のリポジトリへのアクセスが必要です。

1.スーパーユーザ権限を引き受けます。

```
sudo su -
```

2.次のコマンドを使用して、EDBリポジトリをSLESホストに追加します。

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

3.次のコマンドを呼び出して、メタデータをリフレッシュします。

```
zypper refresh
```

4. `SUSEConnect` をインストールしてホストをSUSEに登録し、SUSEリポジトリへのアクセスを許可します。

```
zypper install SUSEConnect
```

5. SUSEリポジトリにアクセスできるようにホストをSUSEに登録し、`'REGISTRATION_CODE'` と `'EMAIL'` をSUSE登録情報に置き換えます。

```
SUSEConnect -r 'REGISTRATION_CODE' -e 'EMAIL'
SUSEConnect -p PackageHub/12.4/x86_64
SUSEConnect -p sle-sdk/12.4/x86_64
```

6. PEM依存関係の次のリポジトリをインストールします。

```
zypper addrepo https://download.opensuse.org/repositories/Apache:/Modules/SLE
```

7. メタデータを更新します。

```
zypper refresh
```

8. Javaベースのコンポーネント用のOpenJDK（バージョン1.8）をインストールします。

```
zypper -n install java-1_8_0-openjdk
```

9. 次に、Zypperユーティリティを使用してPgpool-IIをインストールします。

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

- ここで<xx>は、インストールするPgpoolバージョンです。

</div>

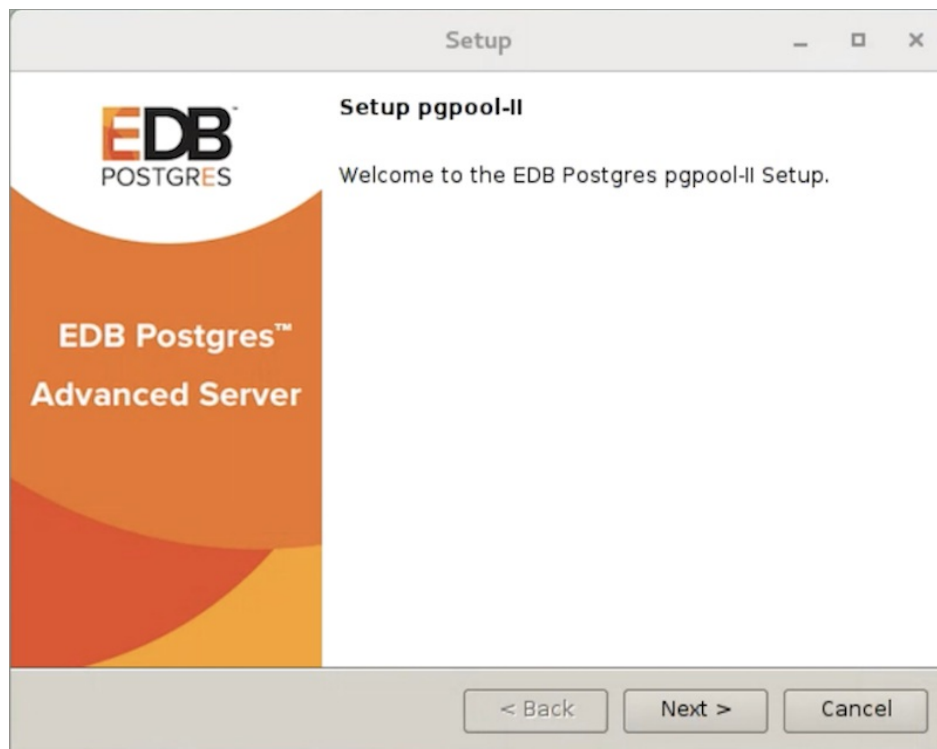
Linuxグラフィカルインストーラーを使用したPgpool-IIのインストール

Pgpool-IIのグラフィカルインストーラーは、StackBuilder Plus（EDB Postgres Advanced Serverホストの場合）またはStack Builder（PostgreSQLホストの場合）から利用できます。LinuxのスタートメニューからStackBuilder Plusにアクセスできます。PgpoolはWindowsシステムをサポートしていません。

Pgpool-IIをインストールするには、次の手順を実行します。

1. StackBuilder Plusを開き、Welcome ウィンドウのドロップダウンリストからEDB Postgres Advanced Serverインストールを選択します。Next をクリックして、アプリケーション選択ページに進みます。Add-ons, tools and utilities ノード、インストールするPgpool-IIバージョンの横のボックスをオンにして、Pgpool-IIインストーラーをダウンロード。Next をクリックして続行します。資格情報を入力して、Next .2をクリックします。選択したパッケージソフトとデフォルトのダウンロードディレクトリが表示されます。必要に応じて、ダウンロードディレクトリの場所を変更します。Next .3をクリックします。インストールファイルをダウンロードすると、確認メッセージが表示されます。Next をクリックして、Pgpool-IIのインストールをスタートし

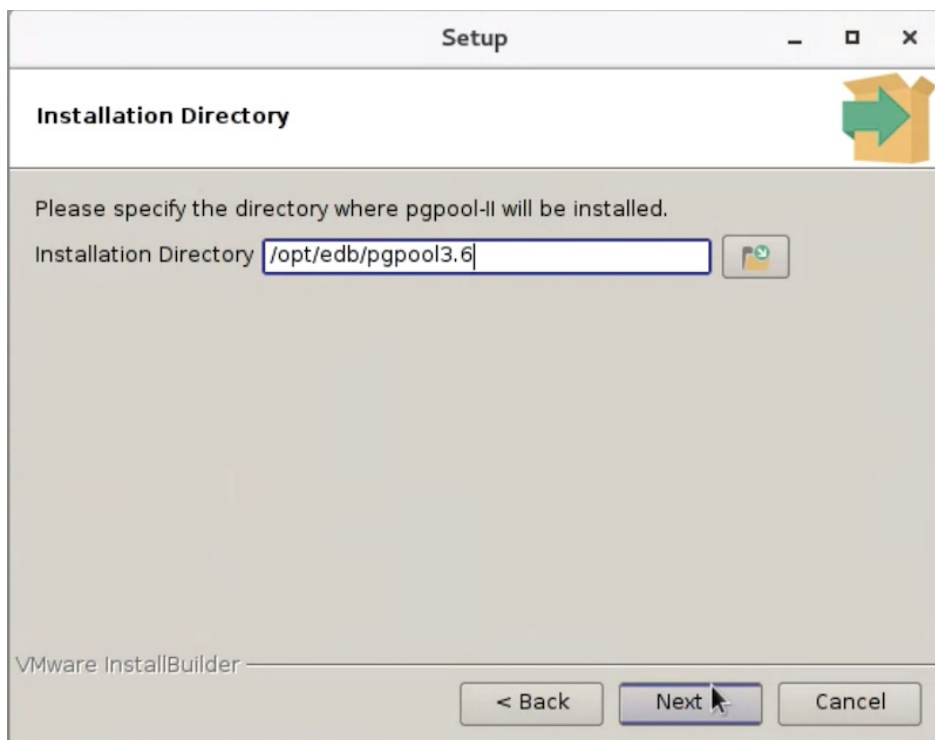
ます.4。インストール言語を選択して、**OK** をクリックします。Pgpoolインストーラがセットアップウィザードにようこそ。



The Pgpool-II Welcome window

図1：Pgpool-IIのウェルカムウィンドウ

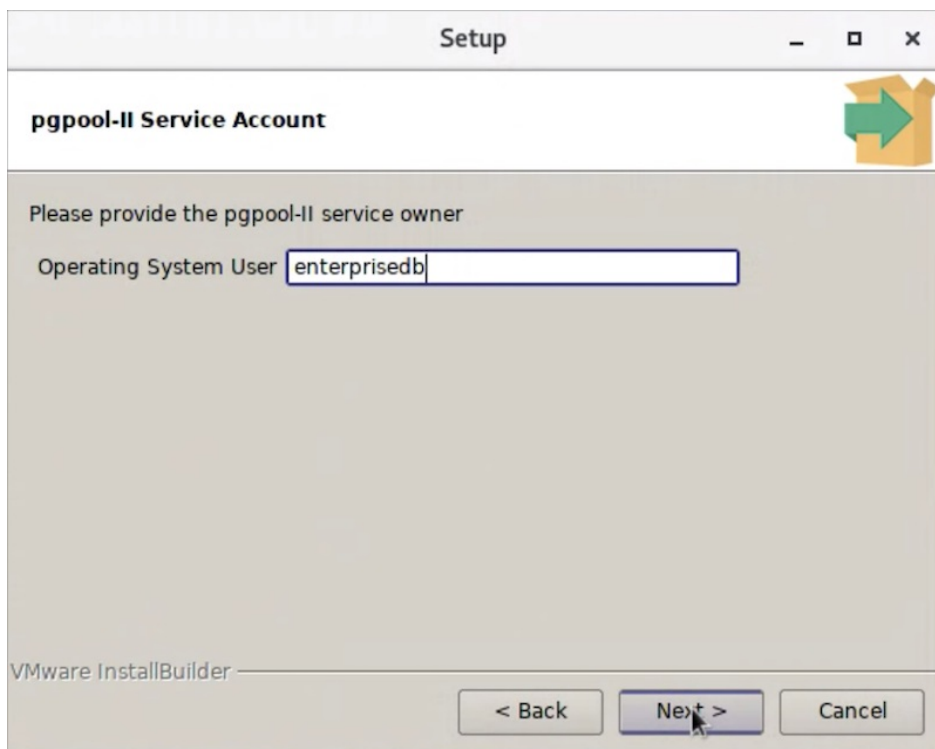
6. **Installation Directory** フィールドを使用して、Pgpool-IIソフトウェアをインストールするディレクトリを指定します（デフォルトのインストールディレクトリは `/opt/edb` です）。次に、**Next** をクリックして続行します。



The Pgpool-II Installation Details Window

図2：Pgpool-IIのインストール詳細ウィンドウ

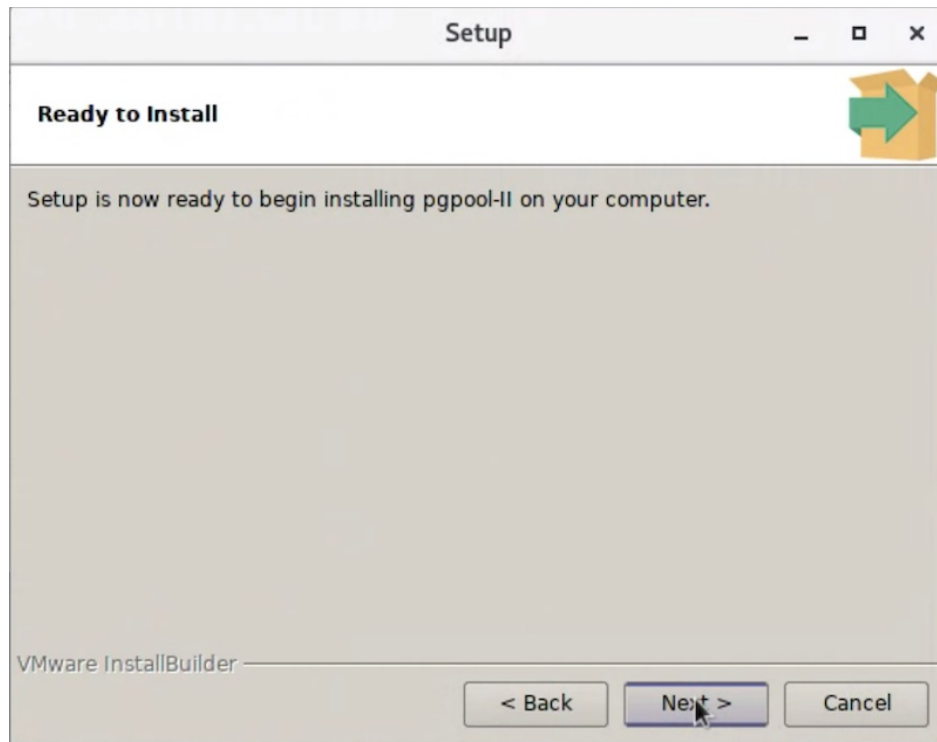
7. **Operating System User** フィールドを使用して、Pgpool-IIがスタートアップ後に変更するLinuxオペレーティングシステムユーザーの名前を指定します。次に、**Next** をクリックして続行します。



The Pgpool-II Operating User window

図3：Pgpool-II Operating Userウィンドウ

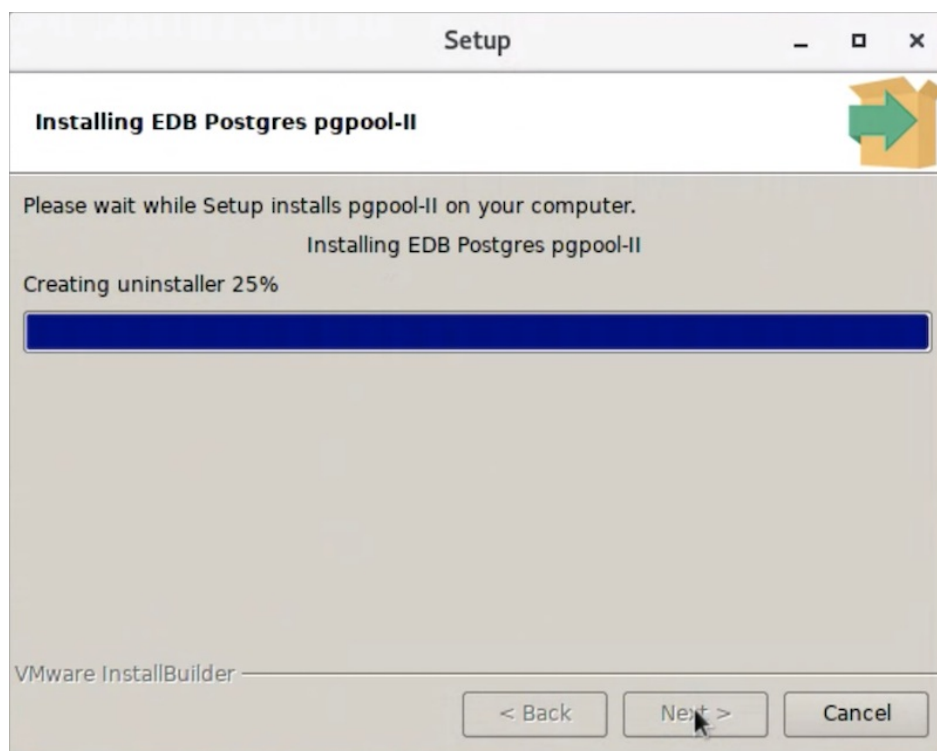
8. **Ready to Install** ウィンドウは、システムにPgpool-IIをインストールするために必要なすべての情報がインストーラを通知します。 **Next** をクリックして、Pgpool-IIをインストールします。



The Ready to Install window

図4：インストールの準備完了ウィンドウ

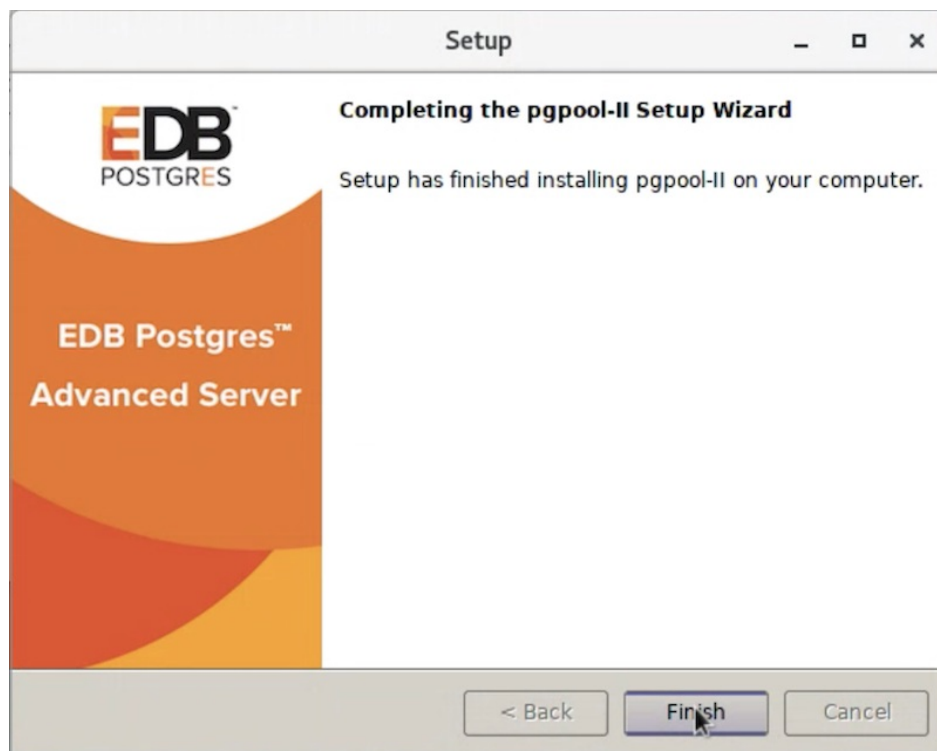
9. 進行状況バーは、インストールの進行状況を通知します。



The installation progresses

図5：インストールの進行

10. セットアップウィザードがPgpool-IIのインストールを完了すると、インストーラが通知します。インストーラを終了する **Finish** クリックしてください。



The installation is complete

図6：インストールが完了しました

```
title: "Installing and  
Managing Extensions"  
  
legacyRedirectsGenerated:  
* # This list is generated by  
a script. If you need add  
entries, use the  
legacyRedirects key. *  
- "/edb-docs/d/pgpool-  
ii/user-guides/pgpool-ii-  
guide/1.0/extensions.html"
```

Installing and Managing Extensions

</ div>

extensionsディレクトリのモジュールは、EDB Postgres Advanced Serverの追加機能であり、通常はコアデータベースには含まれていません。データベースにロードされると、ビルトインファンクションのように機能できます。シンプル `SELECT` コマンドを使用して、PCPをリモートで使用できます。

Note * Pgpool-II拡張機能は、EDB Postgres Advanced Serverのバージョンとオペレーティングシステムのサポートされている組み合わせに対してのみ提供されます。

Pgpool-II拡張機能をインストールする前に、ホストシステムにEDB Postgres Advanced Serverをインストールします。次のコマンドを使用して、次のPgpool固有の拡張機能をインストールできます。

- [Pgpool_adm](#)
- [Pgpool_recovery](#)

Pgpool-II拡張機能のインストール

次のセクションでは、Pgpool-II拡張機能をインストールする手順を説明します。リポジトリの資格情報を要求するには、[the EDB website](#)にアクセスしてください。

CentOSホストへのPgpool-II拡張機能のインストール

CentOSホストにPgpool-II拡張機能をインストールするには、スーパーユーザ特権を想定して次の手順を実行します。

1. リポジトリ構成をインストールするには、スーパーユーザ特権を引き受けて、以下のプラットフォーム固有のコマンドのいずれかを呼び出します。

- CentOS 7の場合：

```
yum -y install https://yum.enterprisedb.com/edbrepos/edb-repo-latest.noarch.r
```

- CentOS 8の場合：

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

2.次のコマンドの `USERNAME:PASSWORD` を、登録済みEDBユーザ名とパスワードに置き換えます。

```
sed -i "s@<username>:<password>@USERNAME:PASSWORD@" /etc/yum.repos.d/edb.repo
```

3. Pgpoolをインストールする前に、次のコマンドを実行、Enterprise Linux (EPEL) の追加パッケージをインストールします。

- CentOS 7の場合：

```
yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.no
```

- CentOS 8の場合：

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

4. CentOS 8の場合、PowerToolsリポジトリを有効にしてEPELパッケージの依存関係を満たします。

```
dnf config-manager --set-enabled PowerTools
```

- 5.プラットフォーム固有のコマンドを使用して、Pgpool-II拡張機能をインストールします。

- RHEL / CentOS 7の場合：

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

例、EDB Postgres Advanced Serverの13のためのPgpool42拡張をインストールするには、次のコマンドを実行します。

```
yum -y install edb-as13-pgpool42-extensions
```

- RHEL / CentOS 8の場合：

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

上記のコマンドで、`<xx>` はEDB Postgres Advanced Serverバージョンであり、`<yy>` はPgpool-II拡張バージョンです。拡張機能は `/usr/edb/as<xx>/share/extension` ディレクトリで利用できます。

</div>

RHELホストにPgpool-II拡張機能をインストールする

リポジトリ構成をインストールする前に、EDBリポジトリへのアクセスを許可する資格情報が必要です。資格情報の要求については、[EDB website](#)をご覧ください。

Pgpool-II拡張機能をRHELホストにインストールするには、次の手順を実行します。

1.リポジトリ構成をインストールするには、スーパーユーザ特権を引き受けて、以下のプラットフォーム固有のコマンドのいずれかを呼び出します。

- RHEL 7の場合：

```
yum -y install https://yum.enterprisedb.com/edbrepos/edb-repo-latest.noarch.r
```

- RHEL 8の場合：

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

2.次のコマンドの `USERNAME:PASSWORD` を、登録済みEDBユーザ名とパスワードに置き換えます。

```
sed -i "s@<username>:<password>@USERNAME:PASSWORD@" /etc/yum.repos.d/edb.repo
```

3. Pgpoolをインストールする前に、次のコマンドを実行、Enterprise Linux (EPEL) の追加パッケージをインストールします。

- RHEL 7の場合：

```
yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.no
```

- RHEL 8の場合：

```
dnf -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.no
```

4.リポジトリを有効にします。

- RHEL 7では、`optional`、`extras` および `HA` リポジトリを有効にして、EPELパッケージの依存関係を満たします。

```
subscription-manager repos --enable "rhel-*-optional-rpms" --enable "rhel-*-e
```

- RHEL 8では、EPELパッケージソフトの依存関係を満たすために `codeready-builder-for-rhel-8-*--rpms` リポジトリを有効にします。

```
ARCH=$( /bin/arch )
```

```
subscription-manager repos --enable "codeready-builder-for-rhel-8-${ARCH}-rpm
```

5.プラットフォーム固有のコマンドを使用して、Pgpool-II拡張機能をインストールします。

- RHEL / CentOS 7の場合：

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

例、EDB Postgres Advanced Serverの13のためのPgpool42拡張をインストールするには、次のコマンドを実行します。

```
yum -y install edb-as13-pgpool42-extensions
```

- RHEL / CentOS 8の場合：

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

上記のコマンドで、`<xx>` はEDB Postgres Advanced Serverバージョンであり、`<yy>` はPgpool-II拡張バージョンです。拡張機能は `/usr/edb/as<xx>/share/extension` ディレクトリで利用できます。

</ div>

RHEL / CentOS 7 PPCLEホストへのPgpool-II拡張機能のインストール

リポジトリ構成をインストールする前に、EDBリポジトリへのアクセスを許可する資格情報が必要です。資格情報の要求については、[EDB website](#)をご覧ください。

次の手順を実行して、Pgpool-II拡張機能をRHEL / CentOS 7 PPC64LEホストにインストール

します。

1. Advance Toolchainをインストールします。

```
rpm --import https://public.dhe.ibm.com/software/server/POWER/Linux/toolchain
cat > /etc/yum.repos.d/advance-toolchain.repo <<EOF

# Beginning of the configuration file
[advance-toolchain]
name=Advance Toolchain IBM FTP
baseurl=https://public.dhe.ibm.com/software/server/POWER/Linux/toolchain/at/r
failovermethod=priority
enabled=1
gpgcheck=1
gpgkey=ftp://public.dhe.ibm.com/software/server/POWER/Linux/toolchain/at/redh
# End of the configuration file
```

2. リポジトリ構成をインストールするには、スーパーユーザ特権を引き受けて、次のコマンドを呼び出します。

```
yum -y install https://yum.enterprisedb.com/edbrepos/edb-repo-latest.noarch.r
```

3. 次のコマンドの `USERNAME:PASSWORD` を、登録済みEDBユーザ名とパスワードに置き換えます。

```
sed -i "s@<username>:<password>@USERNAME:PASSWORD@" /etc/yum.repos.d/edb.repo
```

4. Pgpool-IIをインストールする前に、次のコマンドを実行、Enterprise Linux (EPEL) の追加パッケージをインストールします。

```
yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.no
```

5. RHEL 7で、`optional`、`extras` および `HA` リポジトリを有効にして、EPELパッケージの依存関係を満たします。

```
subscription-manager repos --enable "rhel-*-optional-rpms" --enable "rhel-*-e
```

6. 次のコマンドを呼び出して、Pgpool-II拡張機能をインストールします。

```
yum -y install edb-as13-pgpool<xx>-extensions
```

- ここで<xx>は、インストールするPgpool-IIバージョンです。

</div>

Debian / UbuntuホストへのPgpool-II拡張機能のインストール

DebianまたはUbuntuホストにPgpool-II拡張機能をインストールするには、EDBリポジトリへのアクセスを許可する資格情報が必要です。リポジトリの資格情報を要求するには、[EDB website](#)にアクセスします。

次の手順では、EDB aptリポジトリを使用してDEBパッケージをインストールします。コマンドを使用する場合、`username` および `password` をEDBが提供する資格情報に置き換えます。

Note * このドキュメントのpdfバージョンを使用している場合、カット/ペーストコマンドを使用すると、貼り付けられたコマンドに余分なスペースまたは改行が含まれる場合があります。コマンドが失敗した場合は、追加の文字がないかコマンドを慎重に確認してください。

1.スーパーユーザ特権を想定します。

```
sudo su -
```

2. EDBリポジトリを構成します。

- Debian 9およびUbuntu 18の場合：

```
sh -c 'echo "deb https://username:password@apt.enterprisedb.com/$(lsb_release
```

- Debian 10の場合：

- a. EDBリポジトリをセットアップします。

```
sh -c 'echo "deb [arch=amd64] https://apt.enterprisedb.com/$(lsb_release -cs)
```

- b. 次のコマンドで、`username` および `password` プレースホルダーをEDB資格情報に置き換えます。

```
sh -c 'echo "machine apt.enterprisedb.com login <username> password <password
```

3.安全なAPTリポジトリのサポートをシステムに追加します。

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

4. EDB署名キーを追加しキー。

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

5. リポジトリメタデータを更新します。

```
apt-get update
```

6. Debianパッケージをインストールします。

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

上記のコマンドで、`<xx>` はEDB Postgres Advanced Serverバージョンであり、`<yy>` はインストールするPgpool-II拡張バージョンです。

</div>

SLES 12ホストへのPgpool-II拡張機能のインストール

Zypperパッケージマネージャを使用して、SLES 12ホストにPgpool-II拡張機能をインストールできます。Zypperは、パッケージをインストールするときにパッケージの依存関係を満たそうとしますが、EDBでホストされていない特定のリポジトリへのアクセスが必要です。

1. スーパーユーザ権限を引き受けます。

```
sudo su -
```

2. 次のコマンドを使用して、EDBリポジトリをSLESホストに追加します。

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

3. 次のコマンドを呼び出して、メタデータをリフレッシュします。

```
zypper refresh
```

4. `SUSEConnect` をインストールしてホストをSUSEに登録し、SUSEリポジトリへのアクセスを許可します。

```
zypper install SUSEConnect
```

5. SUSEリポジトリにアクセスできるようにホストをSUSEに登録し、`'REGISTRATION_CODE'` と `'EMAIL'` をSUSE登録情報に置き換えます。

```
SUSEConnect -r 'REGISTRATION_CODE' -e 'EMAIL'  
SUSEConnect -p PackageHub/12.4/x86_64  
SUSEConnect -p sle-sdk/12.4/x86_64
```

6. PEM依存関係の次のリポジトリをインストールします。

```
zypper addrepo https://download.opensuse.org/repositories/Apache:/Modules/SLE
```

7. メタデータを更新します。

```
zypper refresh
```

8. Javaベースのコンポーネント用のOpenJDK（バージョン1.8）をインストールします。

```
zypper -n install java-1_8_0-openjdk
```

9. 次に、Zypperユーティリティを使用してPgpool-II拡張機能をインストールします。

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

- ここで、< xx>はEDB Postgres Advanced Serverバージョンであり、< yy>はインストールするPgpoolバージョンです。

</div>

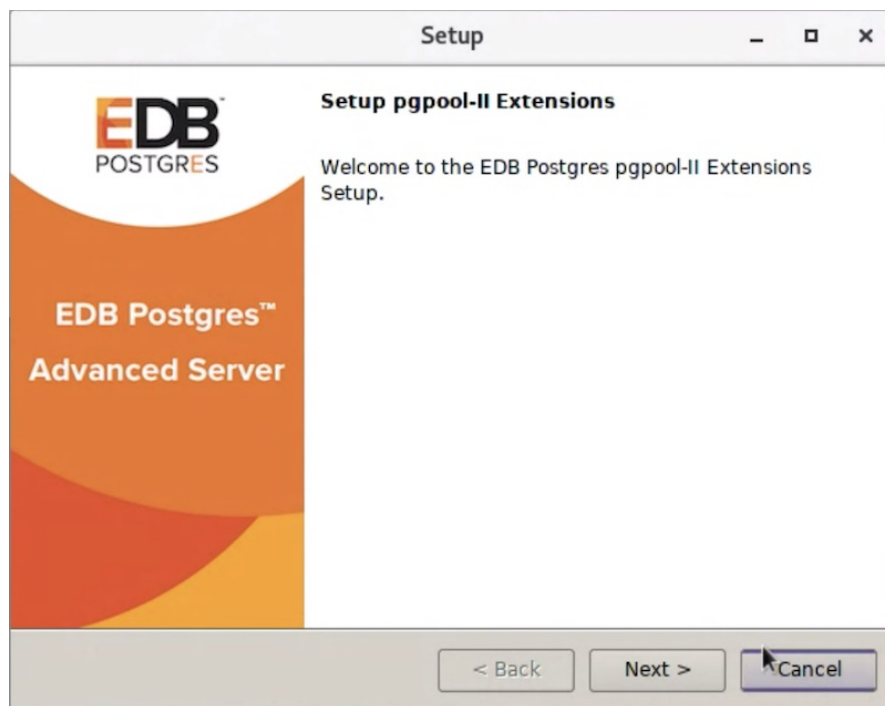
Linuxインストーラーを使用してPgpool-II拡張機能をインストールする

Pgpool-II拡張のグラフィカルインストーラーは、StackBuilder Plus（EDB Postgres Advanced Serverホストの場合）またはStack Builder（PostgreSQLホストの場合）を介して利用できます。LinuxのスタートメニューからStackBuilder Plusにアクセスできます。Windowsシステムではサポートされていません。

次の手順を実行して、Pgpool-II拡張機能をインストールします。

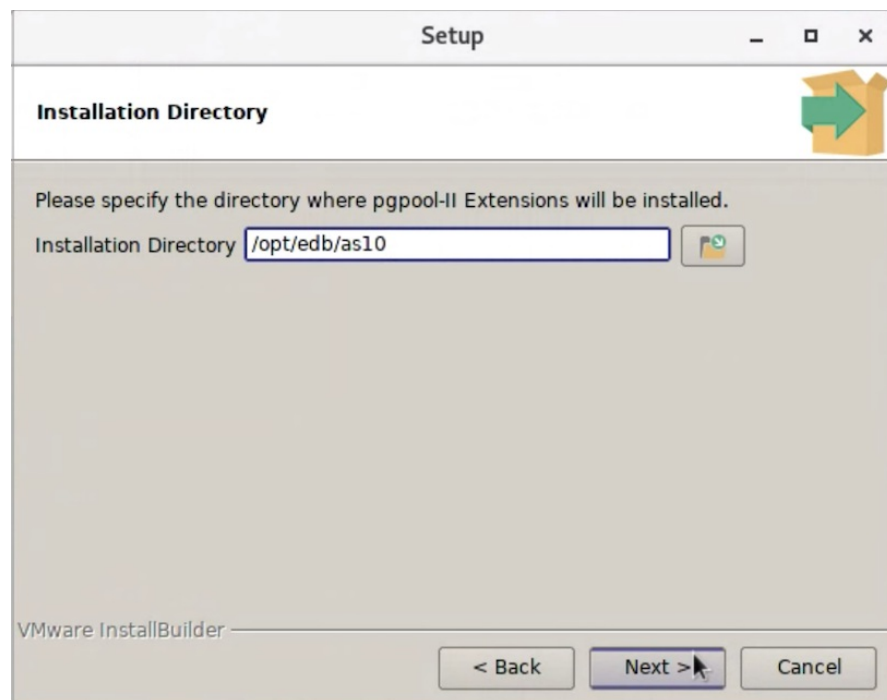
1. StackBuilder Plusを開き、Welcome ウィンドウのドロップダウンリストからEDB Postgres Advanced Serverインストールを選択します。Next をクリックして、アプリケーション選択ページに進みます。Add-ons, tools and utilities ノードを展開し、選択し、インストーラをダウンロードするには、次のpgpool-IIの拡張機能のチェックボックスをオンにします。Next をクリックして続行します。資格情報を入力して、Next をクリックします。
2. 選択したパッケージソフトと、パッケージがインストールされるデフォルトのダウンロードディレクトリが表示されます。必要に応じて、ダウンロードディレクトリの場所を変更します。Next をクリックします。

3. インストールファイルをダウンロードすると、確認メッセージが表示されます。 **Next** をクリックして、インストールをスタートします。
4. インストール言語を選択して、 **OK** をクリックします。
5. Pgpool-II拡張機能インストーラがセットアップウィザードにようこそ。



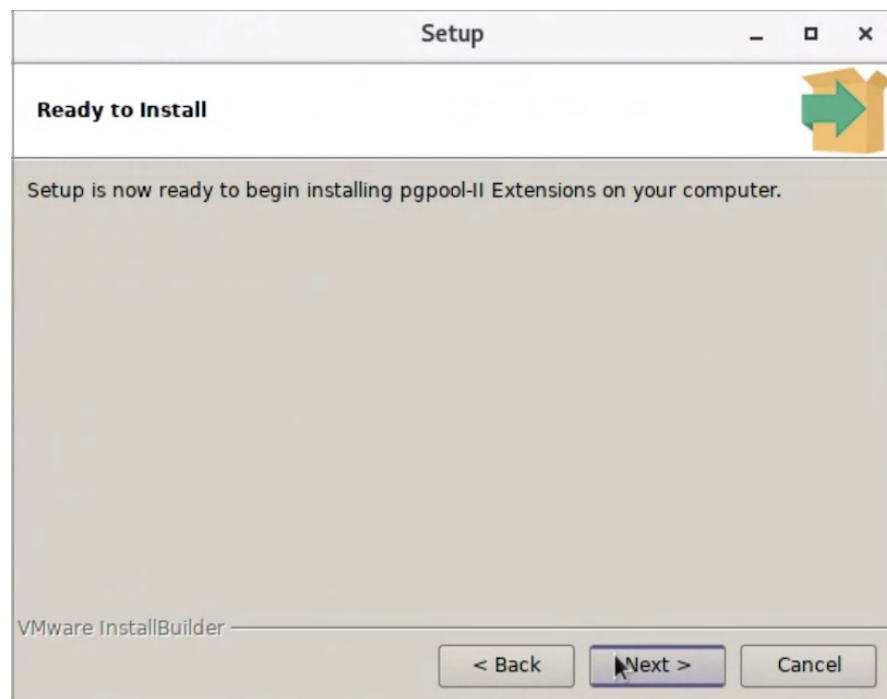
The Pgpool-II Extensions Welcome window

- 。 図1：Pgpool-II拡張機能のウェルカムウィンドウ
6. **Installation Directory** フィールドを使用して、Pgpool-II拡張ソフトウェアをインストールするディレクトリを指定します（デフォルトのインストールディレクトリは `/opt/edb/as<xx>` です）。次に、 **Next** をクリックして続行します。



The Pgpool-II Extensions Installation Details Window

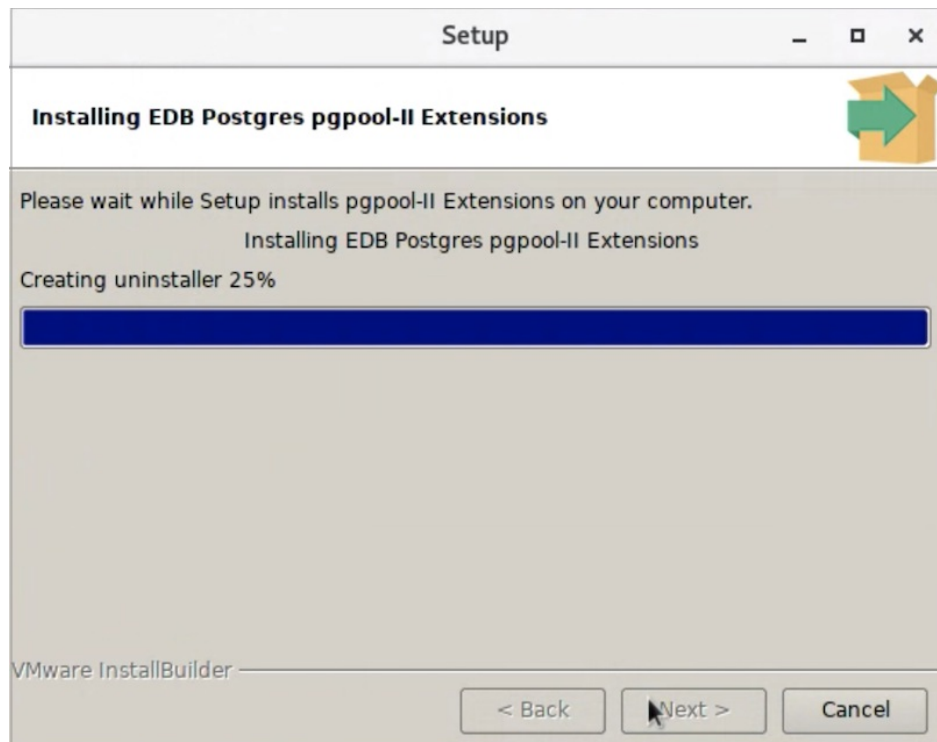
- 図2：Pgpool-II拡張機能のインストール詳細ウィンドウ
7. **Ready to Install** ウィンドウは、システムにPgpool-II拡張機能をインストールするために必要な情報がすべてインストーラ場合に通知します。 **Next** をクリックして、Pgpool-II拡張機能をインストールします。



The Ready to Install window

。図3：インストールの準備完了ウィンドウ

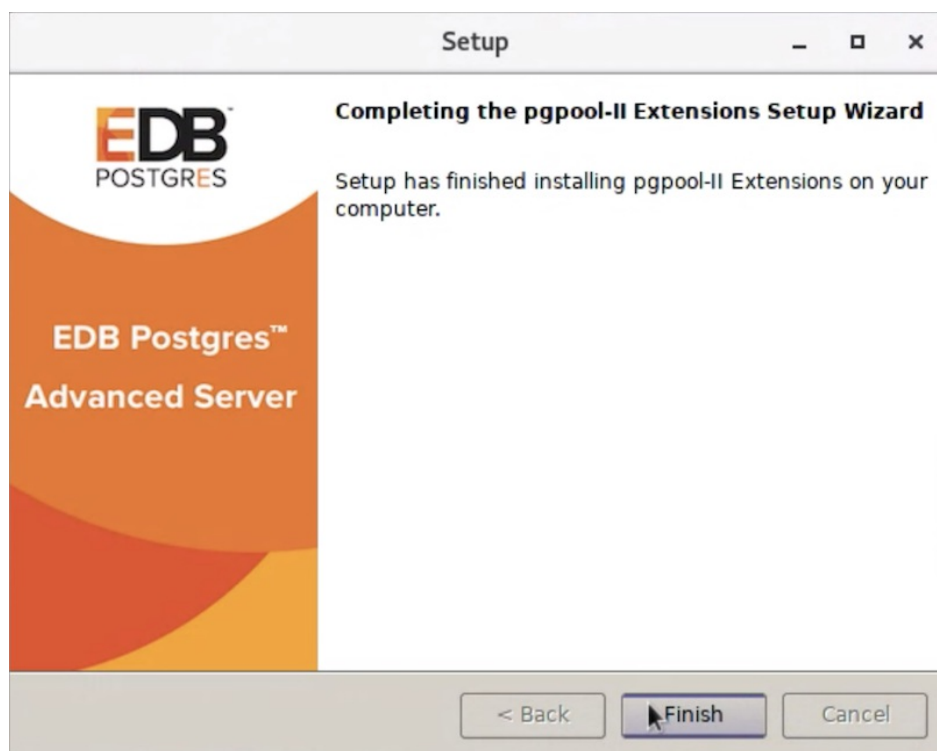
8.インストールの進行状況が進行状況バーに表示されます。



The installation progresses

図4：インストールの進行

9.インストーラは、セットアップウィザードがPgpool-IIのインストールを完了すると通知します。インストーラを終了する **Finish** クリックしてください。



The installation is complete

図5：インストールが完了しました

10.拡張機能は `/opt/edb/as<xx>/share/extension/pgpool*` ディレクトリで利用可能になります。

Pgpool-II拡張機能の作成

Pgpool-II機能を使用する各データベースに拡張機能をインストールして作成する必要があります。すべての拡張機能は、将来のデータベースのために利用可能である保証するために、あなたは `template1` データベースに拡張子を追加することができます。 `template1` データベースにインストールされた拡張機能は、作成時に `template1` をテンプレートとして使用する各データベースに作成されます。

**** Pgpool_adm拡張機能****

</ div>

`Pgpool_adm` は、PCPコマンドへのSQLアクセスを可能にする拡張機能のセットです。PCPコマンドに関する情報をビューするには、<
<https://www.pgpool.net/docs/latest/en/html/pcp-commands.html> >を参照してください。

`Pgpool_adm` 拡張機能をインストールした後、`psql`クライアントアプリケーションを使用してデータベースに接続し、次のSQLコマンドを実行します。

```
CREATE EXTENSION pgpool_adm;
```

Pgpool_adm に関する詳細情報をビューするには、<参照<https://www.pgpool.net/docs/latest/en/html/pgpool-adm.html>>。

**** Pgpool_recovery拡張機能****

</ div>

Pgpool_recovery 拡張は、オンラインリカバリおよび将来のフェールバックメカニズムに必要です。

Pgpool_recovery 拡張機能をインストールした後、psqlを使用してデータベースに接続し、次のSQLコマンドを実行して Pgpool_recovery 拡張機能を作成します。

```
CREATE EXTENSION pgpool_recovery;
```

CREATE EXTENSION コマンドの使用の詳細については、[PostgreSQL core documentation](#)を参照してください。

title: “Configuring Pgpool-II”

legacyRedirectsGenerated: * # This list is generated by a script. If you need add entries, use the legacyRedirects key. * - “/edb-docs/d/pgpool-ii/user-guides/pgpool-ii-guide/1.0/configuring_connection_pooling.html”

Configuring Pgpool-II

Note * Pgpool-IIの設定オプションは豊富です。以下にリストされているオプションを出発点としてのみ検討してください。 Pgpool-IIの設定と使用の詳細については、[project website](#)を参照してください。

</ div>

一般的に使用されるパラメーター

次の表に、コネクションプーリングを実装するときに使用する pgpool.conf パラメータを示します。

Parameter Name | Description |

-----	-----
-----|

listen_addresses | Pgpool-IIが接続を監視するために使用するホスト名またはIPアドレス。デフォルトはローカルホストです。すべてのアドレスを「*」に変更します。 |

`port` | Pgpool-II接続のポート。デフォルトは9999です。|

`pcp_port` | PCP接続用のポート。デフォルトは9898です。|

`backend_hostname0` | バックエンド0のホスト名またはIPアドレス。バックエンドとPgpool-IIが同じホストで実行されている場合、”を指定できます。|

`backend_port0` | バックエンド0のポート番号。|

`backend_weight0` | バックエンド0の重み（負荷分散モードのみ）。ロードを均等にまたは10進数進値（.9、.1など）のバランスをとり、特定のバックエンドに向かってロードを重み付けする場合は、各バックエンドに1を指定します。|

||NA|| `backend_data_directory0` | バックエンド0のデータディレクトリ。|| `enable_pool_hba` | クライアント認証にpool_hba.confを使用するには、on に設定します。||

`num_init_children` | プールの数。デフォルトは32です。|| `max_pool` | プールごとの接続数。デフォルトは4です。|| `connection_cache` | コネクションプーリングを有効にするには、onに設定します。|| `pool_conn_dbname` | Pgpool-IIが接続するデータベース名前。デフォルトでは、Pgpool-IIはPostgresに接続します。 `pool_conn_dbname` パラメータは廃止されることにノートしてください。|

次の表に、レプリケーションと負荷分散を実装するときに使用する `pgpool.conf` パラメータを示します。

Parameter Name| Description|

-----|-----

-----|

`Allow_sql_comments` | オンの場合、SQLコメントを無視します。このパラメータを変更するには、pgpool.confファイルのリロードが必要です。|

`load_balance_mode` | オンに設定すると、負荷分散モードが有効になります。
`load_balance_mode`がオンで`replicate_select`がオフの場合、SELECTステートメントは1つのバックエンドに送信されます。パラメータ`backend_weight`は、各バックエンドが受け取るSELECTステートメントの割合を決定します。|

`ignore_leading_white_space` | 各クエリーの先頭の空白を無視します。Perl用のDBI / DBD :: Pgなどの特定のAPIは、ユーザが制御できない空白文字を追加します。デフォルトはオンです。|

</ div>

接続プーリングの構成

Pgpool-IIは、1つ以上のデータベースサーバーへのキャッシュ接続を維持する子プロセスの

セットを提供します。クライアントが接続すると、Pgpool-IIはそのプールから接続を再利用しようとするため、クライアント接続を開いたり閉じたりするオーバーヘッドを回避できます。

ターゲットデータベースと接続ユーザが、現在プール内にある以前の接続とマッチする場合にのみ、プール内の接続を再利用できます。 `pgpool.conf` ファイルは、コネクションプーリング構成オプション（子プロセスの数や、子ごとのキャッシュ接続の最大数など）を指定します。

1つのデータベースサーバでコネクションプーリングを構成するには：

1. Pgpool-II ホストで `pg_hba.conf` ファイルを構成して、クライアントとサーバー間の接続を許可します。

2. コピーし `pgpool.conf` に `pgpool.conf.sample` ファイルは、ファイルを修正し、`on` に `connection_cache` パラメータを設定し、データベースサーバの接続プロパティを指定します。

- 次の例は、EDB Postgres Advanced Serverに接続する方法を示しています。

- `connection_cache = on`
- `backend_hostname0 = 'localhost'`
- `backend_port0 = 5444`
- `backend_weight0 = 1`
- `backend_data_directory0 = '/var/lib/edb/as13/data'`

- 次の例は、PostgreSQLサーバーに接続する方法を示しています。

- `connection_cache = on`
- `backend_hostname0 = 'localhost'`
- `backend_port0 = 5432`
- `backend_weight0 = 1`
- `backend_data_directory0 = '/var/lib/pgsql/13/data'`

- !!! Note

- `pgpool.conf` ファイルでは、接続パラメーターに、クラスターノード識別子を指定する数字が追加されています。データベースノード `0` は、プライマリノードの値を指定します。

3. オプションで、[Pgpool-II client authentication](#)を構成します。

4. オプションで、[PCP administrative interface](#)を構成します。

5. Pgpool-IIを起動し、次のコマンドを使用してアプリケーションの使用を開始します。

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

- ここで<xy>はPgpoolリリースバージョンです。

6.次のプラットフォーム固有のコマンドを実行して、Pgpool42に接続します。

- CentOS 7用EDB Postgres Advanced Serverの場合：

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

- Debian用EDB Postgres Advanced Serverの場合：

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

- CentOS 7用のPostgreSQLサーバー：

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

- Debian用PostgreSQLサーバーの場合：

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

</div>

ロードバランシングの設定

EDBは、PostgreSQLストリーミングレプリケーションまたはSlonyレプリケーションでPgpool-II負荷分散を使用するレプリケーションシナリオをサポートしています。サポートされるレプリケーション方法は、クライアントアプリケーションによって行われたそのデータベースの更新は、マルチプル>バックエンドサーバーに適用されます保証。各レプリケーションメソッドの利点と詳細な設定手順の詳細については、各ユーティリティのプロジェクト文書をご覧ください。

負荷分散が有効な場合、Pgpool-IIはいくつかのタイプの `SELECT` ステートメントをバックエンドサーバーに配布し、マルチプルのデータベースサーバーとホストがクライアントアプリケーションによって発行された `SELECT` ステートメントの処理ロードを共有できるようにします。

Pgpool-II負荷分散を設定する場合、すべてのバックエンドサーバーの初期データベース環境は同一である必要があります。

- テーブルの名前、定義、および行のコンテンツは同じでなければなりません。
- スキーマは各バックエンドアプリケーションデータベースに存在する必要があります。
- SQL文の結果セットを確保する保証に、各バックエンドサーバ上の設定の役割と権限

は、すべてのサーバーで同一です。

`password` 認証を使用する場合、各データベースサーバ上の関連するユーザ名前に同じパスワードを割り当てます。同じユーザ名前/パスワードのペアを使用して、Pgpool-IIを各バックエンド接続に接続します。

レプリケーションシナリオでは、各バックエンドはホスト名（またはIPアドレス）とデータベースサーバインスタンスが接続をリッスンしているポート番号によって一意に識別されます。`pool_hba.conf` および `pg_hba.conf` ファイルが、そのサーバーとPgpool-IIが実行されるホストとの間の接続を許可することを保証する必要があります。

次の例は、ストリーミングレプリケーションシナリオで2つのサーバー（プライマリノードとレプリカノード）でPgpool-II負荷分散を実装する方法を示しています。SlonyレプリケーションシナリオのPgpool-II負荷分散の構成も同様です。Slonyレプリケーションの設定については、Slonyの文書を参照してください。

レプリケーションシナリオのプライマリノードの構成

レプリケーションシナリオのプライマリノードでSSHセッションを開き、`pg_hba.conf` ファイル（`/var/lib/edb/as13/data` ディレクトリ）を変更し、レプリケーションユーザの接続情報を追加します（次の例では、`edbrepuser` はIPアドレスを持つスタンバイノードにあります） `107.178.217.178` の）：

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

接続情報には、レプリケーションシナリオのスタンバイノードのアドレスと優先認証メソッドを指定する必要があります。

`postgresql.conf` ファイル（`/var/lib/edb/as13/data` にあります）を変更し、次のレプリケーションパラメータと値をファイルの最後に追加します。

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

構成ファイルを保存し、サーバーをリスタートします。

```
systemctl restart edb-as-13
```

Note * 上記のコマンドは、RHEL / CentOS 7および8プラットフォームに適用できます。

`sudo su -` コマンドを使用して、`enterprisedb` データベーススーパーユーザのIDを想定します。

```
sudo su - enterprisedb
```

次に、`psql` セッションをスタートし、`edb` データベースに接続します。

```
psql -d edb
```

`psql` コマンドラインで、`replication` 属性を持つユーザを作成します。

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

レプリケーションシナリオのスタンバイノードの構成

スタンバイサーバとのSSHセッションを開き、データベーススーパーユーザ（`enterprisedb`）のIDを引き継ぎます。

```
sudo su - enterprisedb
```

エディタを選択して、`enterprisedb` ユーザのホームディレクトリに `.pgpass` ファイルを作成します。`.pgpass` ファイルには、レプリケーションユーザのパスワードがプレーンテキスト形式です。あなたは `.pgpass` ファイルを使用している場合は、信頼できるユーザのみが `.pgpass` ファイルへのアクセス権を持っていることを保証必要があります。

レプリケーションユーザの接続情報を指定するエントリーを追加します。

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

サーバは、`.pgpass` ファイルに制限付きアクセス許可を適用します。次のコマンドを使用して、ファイルのアクセス許可を設定します。

```
chmod 600 .pgpass
```

データベーススーパーユーザのIDを放棄します。

```
exit
```

次に、スーパーユーザ権限を引き継ぎます。

```
sudo su -
```

プラットフォーム固有のコマンドを使用してデータベースサーバを停止してから、スタンバイノードのデータディレクトリをプライマリノードの `data` ディレクトリに置き換えます。

次に、スタンバイノードの `data` ディレクトリを削除します。

```
rm -rf /var/lib/edb/as13/data
```

既存の `data` ディレクトリを削除した後、`pg_basebackup` ユーティリティを使用して、プライマリノードの `data` ディレクトリをスタンバイにコピーします。

```
pg_basebackup --format=p --label=standby --host=146.148.46.44 --username=edbr
```

`pg_basebackup` の呼び出しは、プライマリノードのIPアドレスと、プライマリノードで作成されたレプリケーションユーザの名前を指定しプライマリノード。

`-R` オプションを含めると、`standby.signal` ファイルが作成され、出力ディレクトリの `postgresql.auto.conf`（またはtarフォーマットを使用する場合はベースアーカイブファイル）に接続設定が追加され、スタンバイサーバのセットアップが容易になります。

`pg_basebackup` ユーティリティで使用可能なオプションの詳細については、次の

PostgreSQLコア文書を参照してください。

< <https://www.postgresql.org/docs/current/ 静的/app-pgbasebackup.html> >

`pg_basebackup` によってプロンプトが表示されたら、レプリケーションユーザに関連付けられたパスワードです。

`data` ディレクトリをコピーした後、ディレクトリの所有権をデータベーススーパーユーザ（`enterprisedb`）に変更します。

```
chown -R enterprisedb /var/lib/edb/as13/data
```

`postgresql.conf` ファイル（`/var/lib/edb/as13/data` にあります）を変更し、ファイルの最後に次の値を指定しファイル。

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

`data` ファイルはプライマリノードからコピーされ、以前に指定されたレプリケーションパラメーターが含まれています。

次に、サーバーをリスタートします。この時点で、プライマリノードはスタンバイノードにデータを複製します。

**** Pgpool-II負荷分散の設定****

選択したエディタを使用して、`pgpool.conf` ファイルを変更します。`pgpool.conf` ファイル内で、パラメータ設定を変更して、負荷分散が有効になるように指定します。

```
load_balance_mode = on
```

次に、`0` で終わるパラメータセットでプライマリデータベースノードの接続設定を指定します。例：

```
backend_hostname0 = '146.148.46.44' backend_port0 = 5444  
backend_weight0 = 1 backend_data_directory0 = '/var/lib/edb/as13/data'
```

次に、クエリを配布する各ノードの接続設定を指定します。各ノードのパラメータ名前に続く番号を増やし、接続の詳細を提供します。

```
backend_hostname1 = '107.178.217.178' backend_port1 = 5444  
backend_weight1 = 1 backend_data_directory1 = '/var/lib/edb/as13/data'
```

`backend_weight` パラメータを使用して、ノード間でクエリを分散する方法を指定します。`1` の値を指定して、レプリケーションシナリオのノード間で（修飾された）クエリを均等に分散することを示します。

**** Pgpool-IIを再起動します****

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

ここで`<xy>`はPgpoolリリースバージョンです。

</div>

クライアント認証の構成

Pgpool-IIが有効な場合、クライアントアプリケーションはPgpool-IIに接続します。Pgpool-IIはPostgresサーバーの仲介者として機能します。接続しているクライアントアプリケーションは、最初にPgpool-IIサーバーで認証され、次にPostgresサーバーで認証されます。

`pool_hba.conf` 構成ファイルのパラメーター設定は、Pgpool-II認証プロパティを決定します。`pool_hba.conf` ファイルは、Postgresの `pg_hba.conf` 構成ファイルとフォーマットとファンクションが似ています。`pool_hba.conf` エントリの詳細については、Pgpool-IIの文書を参照してください。

Pgpool-II認証を有効にするには：

1. `pool_hba.conf.sample` ファイルを `pool_hba.conf` にコピーします。
`pool_hba.conf` ファイルを変更し、接続するサーバーまたはユーザーの認証情報を指定します。エントリは、`pg_hba.conf` ファイルで使用されているのと同じフォーマットに従う必要があります。`pgpool.conf` ファイルを変更し、`enable_pool_hba` パラメータを `on` に設定します。Pgpool-IIを再起動して、Pgpool-II構成ファイルをリロードします。

Note * データベースサーバで認証するときは、`pool_hba.conf` ファイルで指定されているユーザ名とパスワードを使用します。また、データベースサーバーの `pg_hba.conf` ファイルでそれらのユーザ名とパスワードを指定する必要があります。

</div>

PCPの構成

PCPは、データベースノード、Pgpool-II子プロセス、およびその他の情報に関する情報を取得できるようにする、Pgpool-IIの管理インタフェースです。LinuxコマンドラインからPCPコマンドを発行する必要があります。

`pcp.conf` は、PCPクライアントのパスワード構成ファイルです。PCPコマンドを使用する前に、`pcp.conf` ファイルを変更して、PCPコマンドを呼び出すときに指定したユーザ名とパスワードを指定する必要があります。`pcp.conf` ファイル内のユーザ名は、データベースサーバのユーザ名とパスワードから完全に独立しています。

PCPを有効にするには、次の手順を使用します。

1. `pcp.conf.sample` ファイルを `pcp.conf` にコピーします。
2. 次のフォームで `pcp.conf` ファイルにエントリーを追加します。
 - `username:md5_password`
 - ここで：

- `username` はPCPユーザ名です。
- `md5_password` は、`md5` 形式のPCPパスワードフォーマット。
- 以下に示すように、`pg_md5` プログラムを使用して、クリアテキストフォームから暗号化されたパスワードを生成できます。
- `$ pg_md5 mypassword`
- `34819d7beeabb9260a5c854bc85b3e44`

例、`mypassword` のパスワードで `pcpuser` 名前付けのPCPユーザの `pcp.conf` ファイルのエントリは次のとおりです。

- `# USERID:MD5PASSWD pcpuser:34819d7beeabb9260a5c854bc85b3e44`

3. Pgpoolサービスを再起動します。

4. PCPコマンドを発行するとき、PCPユーザ名前とパスワードの暗号化されていないフォームを指定します。

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

PCPを構成したら、次のPCPコマンドを使用してPgpool-IIを制御し、情報を取得できます。

PCP Commandl Descriptionl

```
-----|-----
-----|
```

`pcp_common_options` | PCPコマンドで使用される一般的なオプションl

`pcp_node_count` | データベースノードの総数を表示しますl

`pcp_node_info` | 指定されたノードIDに関する情報を表示しますl

`pcp_health_check_stats` | 特定のノードIDのヘルスチェック統計データを表示しますl

`pcp_watchdog_info` | Pgpool-IIのウォッチドッグステータスを表示しますl

`pcp_proc_count` | Pgpool-IIの子プロセスIDのリストを表示しますl

`pcp_proc_info` | 指定されたPgpool-IIの子プロセスIDに関する情報を表示しますl

`pcp_pool_status` | `pgpool.conf`で定義されているパラメータ値を表示しますl

`pcp_detach_node` | 指定されたノードをPgpool-IIから切り離します。Pgpool-IIへの既存の接続は強制的に切断されます。l

`pcp_attach_node` | 指定されたノードをPgpool-IIに接続します。|

`pcp_promote_node` | 指定されたノードを新しいメインとしてPgpool-IIに昇格させます|

`pcp_stop_pgpool` | Pgpool-IIプロセスを終了します|

`pcp_reload_config` | pgpool-II設定ファイル|

`pcp_recovery_node` | 指定されたバックエンドノードをリカバリ|

Note * `pcp_health_check_stats` および `pcp_reload_config` コマンドは、Pgpoolバージョン4.2以降で使用できます。

PCPコマンドオプションの詳細をビューするには、[Pgpool project site](#)にアクセスしてください。

接続数とプーリングの構成

Pgpoolには、プーリングと接続処理を調整するための構成がいくつかあります。この構成に応じて、`max_connections` のPostgres構成も設定して、必要に応じてすべての接続が受け入れられるようにする必要があります。また、ノート全てPgpoolinstancesにわたって拡散 `num_init_children` にニーズクラウドアーキテクチャworkswithアクティブ/アクティブの場合、（アクティブなインスタンスの数で通常使用される値を分割する）こと。以下のテキストでは、構成を変更した場合の影響について説明し、オンプレミスとクラウドアーキテクチャの両方の値を推奨しています。

**** max_pool **** : 通常、`max_pool` を1に設定することをお勧めします。また、多くの再接続を行うアプリケーションの場合、`max_pool` をアプリケーション接続のユーザー、データベース、および接続オプションの異なる組み合わせの数に設定できます。プール内の1つを除くすべての接続は古い接続であり、パフォーマンスに追加せずにPostgresからのコネクションスロットを消費します。したがって、アクティブな接続と古い接続の健全な比率を維持するために、4を超える `max_pool` を構成しないことをお勧めします。例として、常に再接続し、2人の異なるユーザーを使用するアプリケーションで、両方とも自分のデータベースに接続する場合は、2に設定します。両方のユーザーが両方のデータベースに接続できる場合は、4に設定します。Pgpoolの `num_init_children` 、またはPostgresの `max_connections` を調整します。

**** num_init_children **** : `num_init_children` をアクティブに並行して実行できる接続の数に設定することをお勧めしますが、値はactivePgpool-IIインスタンスの数（オンプレミスアーキテクチャとクラウドアーキテクチャ）。例：3つのPgpoolインスタンスを持つアーキテクチャで、アプリケーションが100個のアクティブな接続を並行して実行できるようにするには、オンプレミスアーキテクチャの `num_init_children` を100に設定し、クラウドアーキテクチャの `num_init_children` を33に設定します。通常、`num_init_children` を増やすには、Postgresで `max_connections` を調整する必要があります。

**** listen_backlog_multiplier **** : 100 beactiveべきでアプリケーション500個の接続を開くかも

しれないアクティブな接続（`num_init_children`）の.as例の数、と（アプリケーションによって知覚されるように）開いている接続の数を乗算するように設定することができオンプレミスアーキテクチャでは、`num_init_children` を100に設定し、`listen_backlog_multiplier` を4に設定する必要があります。このセットアップでは、100個のアクティブな接続を並列で処理できます。アプリケーションは合計500のオープン接続を認識し、Postgresは常に最大100接続の負荷を処理しロード。`listen_backlog_multiplier` を増やすと、アプリケーションはより多くの接続を認識しますが、並列アクティブ接続の数は増加しません（`num_init_children` によって決定されます）。

**** max_connections ****：Postgresの[アクティブなpgpoolインスタンスの数] x [`max_pool`] x [`num_init_children`] + [`superuser_reserved_connections`]（Postgres）よりも大きい `max_connections` を設定することをお勧めします。例：3つのインスタンスがアクティブ/パッシブであるオンプレミスセットアップで、`max_pool` が2に設定され、`num_init_children` が100に設定され、`superuser_reserved_connections`（Postgres）が5に設定され、Postgres `max_connections` は[1 x 2 x 100 + 5]、これは205接続以上です。cloudsetupの同様のセットアップは、3つのアクティブなインスタンス、`max_pool` を2、`num_init_children` setを33、`superuser_reserved_connections`（Postgres）を5に設定して実行します。この場合、Postgres `max_connections` は[3x 2 x 33 + 5] 203以上です。お勧めの設定の下に設定すると、新しい接続を開くの問題を引き起こす可能性があり、かつ `max_pool` との組み合わせで、予期しない動作（低または無アクティブな接続が、接続スロットfromPostgresを使用してプールされた接続をSTALEに起因する、まだ接続の問題を引き起こす可能性があることに注意してください。リレーションの詳細については、`num_init_children`、`max_pool`、および `max_connections` の間で、この背景情報を参照してください。

</div>

Pgpool-IIホストのセットアップ

インストールにPgpool-II機能を実装するパラメータ設定を変更した後、Pgpool-IIサービスをスタートする必要があります。

Pgpool-IIが起動すると、`pid_file_name` 設定パラメータによって名前が決定される `pgpool.conf` ファイルにプロセスIDを記録します。サンプルファイルの `pid_file_name` パラメータの初期値は次のとおりです。

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

ここで<lt; xy>はPgpoolリリースバージョンです。

Note * オペレーティングシステムは再起動時（`pgpool` ディレクトリを含む）`/var/run` ディレクトリの内容を削除してもよいです。`pgpool.pid` ファイルの場所として `/var/run/edb/pgpool` ディレクトリを使用しないでください。`pid_file_name` パラメータを変更して、より安全なディレクトリの場所を指定します。

```
title: "Connecting a Client to Pgpool-II"

legacyRedirectsGenerated: * # This list is
generated by a script. If you need add
entries, use the legacyRedirects key.
* - "/edb-docs/d/pgpool-ii/user-
guides/pgpool-ii-
guide/1.0/connecting_a_client_to_pgpool-
II.html"
```

Connecting a Client to Pgpool-II

</ div>

クライアントアプリケーションは、Pgpool-IIホスト上のPgpool-IIリスナポートに直接接続する必要があります。例、（pgpool-IIの機能を使用しながら）、入力 edb データベースに接続します。

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

psql プロンプトで呼び出されると、次の SHOW コマンドキーワードはPgpool-II情報を表示します。

Command	Information Provided
SHOW pool_status	Displays 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 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.
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.

SHOW コマンドのオプションの詳細情報をビューするには、[Pgpool project site](#)をご覧ください

い。

Note * SHOW POOL_HEALTH_CHECK_STATSおよびSHOW POOL_BACKEND_STATSコマンドは、Pgpoolバージョン4.2以降で使用できます。

title: “Upgrading Pgpool-II and Extensions”

legacyRedirectsGenerated:
* # This list is generated by
a script. If you need add
entries, use the
legacyRedirects key. *
- “/edb-docs/d/pgpool-
ii/user-guides/pgpool-ii-
guide/1.0/upgrade.html”

Upgrading Pgpool-II and Extensions

</div>

次のセクションでは、PgpoolおよびPgpool拡張機能のアップグレードプロセスの概要を説明します。

Pgpool-IIのアップグレード

次のセクションでは、（例、3.7.13から3.7.14にアップグレードする）pgpool-IIのマイナーバージョンアップグレード処理の概要を示します。

ルートユーザのIDを想定して、次のコマンドを呼び出します。

RHEL / CentOS 7の場合：

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

ここで、<xx>はアップグレードするPgpoolバージョンです。例、pgpoolの3.7.13から3.7.14にアップグレードするには、次のコマンドを実行します。

```
yum upgrade edb-pgpool37
```

RHEL / CentOS 8の場合：

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

Debian / Ubuntuの場合：

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

SLES 12の場合：

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

Pgpool-II拡張機能のアップグレード

次のセクションでは、Pgpool-II拡張機能のアップグレードプロセスの概要を説明します。

pgpoolの拡張の古いバージョンから最新バージョンにアップグレードするには、スーパーユーザを想定し、次のコマンドを実行します。

RHEL / CentOS 7の場合：

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

ここで、<xx> はEDB Postgres Advanced Serverバージョンであり、<yy> はPgpool拡張バージョンです。

Note * マイナーバージョンアップグレードのみがサポートされています（例、3.6.20から3.6.21拡張にアップグレードできますが、3.7.14にはアップグレードできません）。

RHEL / CentOS 8の場合：

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

Debian / Ubuntuの場合：

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

SLES 12の場合：

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

title: “Uninstalling Pgpool-II
and Extensions”

```
legacyRedirectsGenerated: *  
# This list is generated by a  
script. If you need add  
entries, use the  
legacyRedirects key. * -  
"/edb-docs/d/pgpool-ii/user-  
guides/pgpool-ii-  
guide/1.0/uninstalling_pgpool-  
II.html"
```

Uninstalling Pgpool-II and Extensions

次のセクションでは、Pgpool-IIとその拡張機能をアンインストールするプロセスの概要を説明します。

Pgpool-IIのアンインストール

RHEL / CentOSホスト上のPgpool-IIのアンインストール

Pgpool-IIをアンインストール、 ルートユーザのIDを想定して、次のコマンドを呼び出します。

RHEL / CentOS 7の場合：

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

RHEL / CentOS 8の場合：

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

<xx> はPgpoolのバージョンです。

Debian / Ubuntuホスト上のPgpool-IIのアンインストール

Debian / UbuntuホストでPgpool-IIをアンインストール、次のコマンドを呼び出します。

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

<xx> は、アンインストールするPgpoolバージョンです

SLES 12ホストでのPgpool-IIのアンインストール

SLESホストでPgpool-IIをアンインストール、 ルートユーザのIDを想定して、次のコマンドを呼び出します。

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

ここで、<xx>はアンインストールするPgpoolバージョンです。

Pgpool-II Linuxアンインストーラーのアンインストール

Pgpool-IIグラフィカルインストーラは、インストールインストレーションディレクトリにアンインストーラーを作成します。デフォルトのインストレーションディレクトリ、つまり /opt/edb を使用している場合、アンインストーラーは /opt/edb/pgpool<x.y> にあります（<xy>はインストールしたPgpoolバージョンです）。

- 1.アンインストーラーが含まれるディレクトリに、スーパーユーザ権限を引き受けます。アンインストーラーを開き、 **Yes** をクリックしてPgpool-IIのアンインストールを開始します。
- 2.アンインストールプロセスが開始されます。アンインストールが完了したら、 **OK** をクリックします。

Pgpool-II拡張機能のアンインストール

次のセクションでは、Pgpool-IIとその拡張機能をアンインストールするプロセスの概要を説明します。

RHEL / CentOSホスト上のPgpool-II拡張機能のアンインストール

サーバーから拡張機能を削除実行には、次のコマンドを実行します。

RHEL / CentOS 7の場合：

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

RHEL / CentOS 8の場合：

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

ここで、<xx> はEDB Postgres Advanced Serverのバージョン、<yy> はPgpool-II拡張バージョンです。

SLES 12ホストでのPgpool-II拡張機能のアンインストール

SLESホスト上のPgpool-II拡張機能をアンインストール、 ルートユーザのIDを想定して、次のコマンドを呼び出します。

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

ここで、`<xx>` はEDB Postgres Advanced Serverのバージョン、`<yy>` はアンインストールするPgpool-II拡張バージョンです。

Debian / Ubuntuホスト上のPgpool-IIエクステンションのアンインストール

Debian / UbuntuホストでPgpool-II拡張機能をアンインストール、次のコマンドを呼び出します。

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

ここで、`<xx>` はEDB Postgres Advanced Serverのバージョン、`<yy>` はアンインストールするPgpool-II拡張バージョンです。

Pgpool-II Extensions Linux Graphical Uninstallerのアンインストール

Pgpool-II拡張機能のグラフィカルインストーラは、インストールインストレーションディレクトリにアンインストーラーを作成します。デフォルトのインストレーションディレクトリ（`/opt/edb/as<xx>`）を使用している場合、アンインストーラーは `/opt/edb/as<xx>` にあります（< xx>はインストールしたEDB Postgres Advanced Serverバージョンです）。

1.アンインストーラーが含まれるディレクトリし、スーパーユーザ権限を引き受けます。アンインストーラーを開き、`Yes` をクリックして、Pgpool-II拡張機能のアンインストールを開始します。アンインストールプロセスが開始されます。アンインストールが完了したら、`OK` をクリックします。