



# **Postgres Enterprise Manager Administraion Guide**

リリース 8.6.0

**EnterpriseDB Corporation**

2022 年 10 月 27 日



# 目次

|              |                                        |           |
|--------------|----------------------------------------|-----------|
| <b>第 1 章</b> | <b>Changing the default port</b>       | <b>3</b>  |
| 1.1          | RHEL および CentOS の場合                    | 3         |
| 1.2          | Debian および Ubuntu の場合                  | 4         |
| <b>第 2 章</b> | <b>Registering a Postgres server</b>   | <b>5</b>  |
| 2.1          | データベースサーバーの手動登録                        | 5         |
| 2.2          | 自動サーバー検出                               | 12        |
| 2.3          | pemworker ユーティリティを使用してサーバーを登録する        | 13        |
| 2.3.1        | pemworker ユーティリティを使用してサーバーの登録を解除する     | 14        |
| 2.4          | 接続とバインドの確認                             | 15        |
| 2.5          | AWS での Postgres クラスターの登録               | 15        |
| 2.5.1        | AWS EC2 で実行されている Postgres クラスターのモニタリング | 16        |
| 2.5.2        | AWS RDS で実行されている Postgres クラスターのモニタリング | 16        |
| <b>第 3 章</b> | <b>Registering a PEM agent</b>         | <b>17</b> |
| 3.1          | PEM エージェントの登録解除                        | 19        |
| 3.2          | PEM エージェント構成パラメーターの設定                  | 19        |
| 3.3          | 非 root ユーザーアカウントを使用して PEM エージェントを登録する  | 20        |
| <b>第 4 章</b> | <b>Managing a PEM server</b>           | <b>23</b> |
| 4.1          | PEM サーバーとエージェントの起動と停止                  | 23        |
| 4.2          | Linux での PEM サーバーまたは PEM エージェントの制御     | 24        |
| 4.3          | Windows での PEM サーバーまたは PEM エージェントの制御   | 24        |
| 4.4          | HTTPD サーバーの制御                          | 25        |
| 4.5          | PEM 認証の管理                              | 25        |
| 4.6          | PEM サーバー構成の編集                          | 26        |
| 4.7          | セキュリティの管理                              | 26        |
| 4.7.1        | ログインロール                                | 26        |
| 4.7.2        | グループロール                                | 27        |
| 4.7.3        | PEM の事前定義されたロールを使用して PEM 機能へのアクセスを管理する | 27        |
| 4.7.4        | チームロールの使用                              | 29        |
| 4.7.5        | オブジェクトの権限                              | 29        |
| 4.8          | サーバー構成パラメーター - リファレンス                  | 30        |
| <b>第 5 章</b> | <b>Managing Postgres servers</b>       | <b>39</b> |
| 5.1          | 監視対象データベースサーバーの起動と停止                   | 39        |
| 5.2          | pg_hba.conf ファイルの変更                    | 39        |
| 5.3          | データベースとオブジェクトの作成と保守                    | 41        |
| 5.4          | テンプレート Linux サービススクリプト                 | 42        |
| <b>第 6 章</b> | <b>Managing a PEM agent</b>            | <b>43</b> |

|              |                                        |           |
|--------------|----------------------------------------|-----------|
| 6.1          | ジョブ通知の管理 . . . . .                     | 43        |
| 6.1.1        | ジョブレベルでのジョブ通知の構成 . . . . .             | 43        |
| 6.1.2        | エージェントレベルでのジョブ通知の構成 . . . . .          | 43        |
| 6.1.3        | サーバーレベルでのジョブ通知の構成 . . . . .            | 44        |
| 6.2          | PEM スケジュールされたジョブの管理 . . . . .          | 44        |
| 6.3          | エージェント権限 . . . . .                     | 47        |
| 6.4          | エージェント構成 . . . . .                     | 49        |
| 6.5          | エージェントのプロパティ . . . . .                 | 53        |
| <b>第 7 章</b> | <b>Managing certificates</b>           | <b>55</b> |
| 7.1          | SSL 証明書の置き換え . . . . .                 | 55        |
| 7.2          | エージェント SSL 証明書の更新 . . . . .            | 58        |
| <b>第 8 章</b> | <b>Managing configuration settings</b> | <b>61</b> |

Postgres Enterprise Manager (PEM) 管理ガイドへようこそ。PEM は、EDB Postgres Advanced Server または PostgreSQL データベースの管理および分析機能を提供するコンポーネントで構成されています。PEM は、オープンソースの pgAdmin 4 プロジェクトに基づいています。

PEM は、包括的なデータベース設計および管理システムです。PEM は、初心者と経験豊富な Postgres ユーザーの両方のニーズを満たすように設計されており、データベースオブジェクトの作成、メンテナンス、使用を簡素化する強力なグラフィカルインターフェイスを提供します。



# 第1章 Changing the default port

デフォルトでは、PEM サーバーの構成時に 8443 ポートが Web サービスに割り当てられます。Apache 構成ファイルのいくつかのパラメーターを変更することにより、構成後にポートを変更できます。Apache 構成ファイルの名前と場所はプラットフォーム固有です。

## 1.1 RHEL および CentOS の場合

1. ファイル/etc/httpd/conf.d ファイルを編集し、次のパラメーターの 8443 をポート番号に置き換えます。

```
Listen 8443
<VirtualHost _default_:8443>
ServerName localhost:8443
RewriteRule ^(.*)$ https://%{HTTP_HOST}:8443%{REQUEST_URI} [L,R=301]
```

1. semanage がインストールされていない場合は、インストールします。

```
sudo yum -y install policycoreutils-python
```

1. ポートがリストされているかどうかを確認します。

```
semanage port -l | grep http
```

1. ポートがリストに表示されない場合は、次のコマンドを実行します。

```
sudo semanage port -a -t http_port_t -p tcp <your_port_number>
```

1. httpd サービスを再起動します。

```
sudo systemctl restart httpd
```

ポートを使用して PEM Web インターフェイスにアクセスできるようになりました。詳細については、

Accessing the web interface を参照してください。

## 1.2 Debian および Ubuntu の場合

1. ファイル/etc/apache2/sites-available/edb-ssl-pem.conf を編集し、次のパラメーターの 8443 を自分のポート番号に置き換えます。

```
Listen 8443
<VirtualHost _default_:8443>
ServerName HTTPD-EDBPEM-SERVER-v8:8443
```

1. httpd サービスを再起動します。

```
sudo systemctl restart apache2
```

ポートを使用して PEM Web インターフェイスにアクセスできるようになりました。詳細については、

[Accessing the web interface](#) を参照してください。

## 第2章 Registering a Postgres server

PEM でサーバーを管理またはモニタする前に、サーバーを PEM に登録し、エージェントをバインドする必要があります。サーバーをリモートエージェント（別のホストにあるエージェント）にバインドできます。ただし、エージェントが同じホストにない場合、インスタンスに関するすべての統計情報にアクセスできません。

### 2.1 データベースサーバーの手動登録

PEM でデータベースサーバーを管理または監視するには、次のことを行う必要があります。

- EDB Postgres Advanced Server または PostgreSQL サーバーを PEM サーバーに登録します。
- データベースサーバーを PEM エージェントにバインドします。

Create Server ダイアログボックスを使用して、サーバーの登録情報を提供し、PEM エージェントをバインドし、PEM クライアントツリーにサーバーを表示できます。[サーバーの作成] ダイアログボックスを開くには、**オブジェクト>作成>サーバー** を選択します。

---

**注釈:** 接続を試みる前に、登録している Postgres サーバーの `pg_hba.conf` ファイルで PEM クライアントのホストからの接続が許可されていることを確認する必要があります。

---

[ **General** ] タブを使用して、サーバーの一般的なプロパティを説明します。

- **Name** フィールドを使用して、サーバーの名前を指定します。名前は、PEM ブラウザーツリーでサーバーを識別します。
- グループを使用して、ツリー内のサーバーとエージェントを整理できます。グループを使用すると、多数のサーバーをより簡単に管理できます。たとえば、実稼働グループ、テストグループ、または LAN 固有のグループを持つことができます。**Group** リストを使用して、新しいサーバーを表示するサーバーグループを選択します。
- **Team** フィールドを使用して、PostgreSQL ロール名を指定します。このロールのメンバーであるか、サーバーを最初に作成したか、PEM サーバーでスーパーユーザー権限を持つ PEM ユーザーのみが、PEM にログオンしたときにこのサーバーを表示します。このフィールドを空白のままにすると、デフォルトですべての PEM ユーザーにサーバーが表示されます。[サーバーの構成] ダイアログボックスの `show_objects_with_no_team` パラメーターを使用して、動作を変更できます。`show_objects_with_no_team` が `false` に設定されている場合、チームのないサーバーは他のすべてのユーザーに表示されません。
- 背景カラーセレクターを使用して、サーバーに保存されているデータベースオブジェクトの背後にある PEM ツリーに表示する色を選択します。

- **前景色カラーセクター**を使用して、サーバーに保存されているオブジェクトの PEM ツリー内のラベルのフォントの色を選択します。
- **今すぐ接続しますか？** を選択して、**保存** を選択したときにサーバー接続を試みます。後の接続試行まで指定された接続パラメーターを PEM クライアントに検証しない場合は、**Connectnow?** をクリアします。
- **コメント** フィールドにサーバーに関するメモを入力します。

[接続] タブ を使用して、サーバーの接続の詳細を指定します。

- **ホスト名/アドレス** フィールドにサーバーホストの IP アドレスまたは完全修飾ドメイン名を指定します。Unix ベースのシステムでは、アドレスフィールドを空白のままにして、ローカルマシンでデフォルトの PostgreSQL Unix ドメインソケットを使用します。または、PostgreSQL ソケットを含む代替パスを設定できます。パスを入力する場合、パスはスラッシュ (/) で始める必要があります。
- **Port** フィールドにホストのポート番号を指定します。
- **メンテナンスデータベース** フィールドを使用して、PEM が接続する初期データベースの名前を指定します。これには、インストールされた pgAgent スキーマと adminpack オブジェクト（両方ともオプション）が含まれます。PostgreSQL バージョン 8.1 以降では、メンテナンス DB は通常 **postgres** と呼ばれます。以前のバージョンでは **template1** がよく使用されますが、テンプレートデータベースが乱雑になるのを避けるには **postgres** データベースを作成することをお勧めします。
- 監視対象サーバーで Kerberos 認証を使用するには、**Kerberos 認証** を **Yes** に設定します。デフォルトでは、監視対象サーバーは PEM サーバーと同じ認証方法を使用します。監視対象サーバーが Kerberos 認証を使用したくない場合は、**config\_local.py** ファイルで **ALLOW\_DATABASE\_CONNECTION\_WITHOUT\_KERBEROS** を **TRUE** に設定します。
- **ユーザー名** フィールドにサーバーで認証するときに使用する名前を指定します。
- 指定されたユーザーに関連付けられたパスワードを **Password** フィールドに入力します。
- **パスワードを保存しますか？** を選択して、パスワードを暗号化された形式で PEM バックエンドデータベースに保存します。各パスワードはユーザーごと、サーバーごとに保存され、他のチームメンバーと共有されることはありません。PEM は、保存されたパスワードを使用して、次回にデータベースサーバーに接続します。保存したパスワードを削除するには、最初にデータベースサーバーを切断してから、**オブジェクト>保存されたパスワードのクリア** を選択します。
- **ロール** フィールドを使用して、サーバーへの接続後に使用するクライアントの特権が割り当てられたロールの名前を指定します。この値により、1つのロールとして接続し、接続が確立されたときに別のロール（このフィールドで指定したロール）の権限を引き受けることができます。接続するロールは、指定されたロールのメンバーである必要があります。

[SSL] タブ を使用して SSL を構成します。

- **SSL モード** フィールドで、サーバーが使用する SSL 接続のタイプを選択します。SSL 暗号化の使用の詳細については、[the PostgreSQL documentation](#) を参照してください。

プラットフォーム固有のファイルマネージャーダイアログボックスを使用して、SSL 暗号化をサポートするファイルをサーバーにアップロードできます。ファイルマネージャーを開くには、次の各フィールドの右側にあるアイコンを選択します。

- **クライアント証明書**フィールドを使用して、クライアント SSL 証明書を含むファイルを指定します。このファイルは、PEM がデスクトップモードでインストールされている場合はデフォルトの `~/.postgresql/postgresql.crt` ファイルを置き換え、PEM が Web モードでインストールされている場合は `<STORAGE_DIR>/<USERNAME>/.postgresql/postgresql.crt` ファイルを置き換えます。SSL 接続が確立されていない場合、このパラメーターは無視されます。
- **クライアント証明書キー**フィールドを使用して、クライアント証明書に使用される秘密鍵を含むファイルを指定します。このファイルは、PEM がデスクトップモードでインストールされている場合はデフォルトの `~/.postgresql/postgresql.key` を置き換え、PEM が Web モードでインストールされている場合はデフォルトの `<STORAGE_DIR>/<USERNAME>/.postgresql/postgresql.key` を置き換えます。SSL 接続が確立されていない場合、このパラメーターは無視されます。
- **ルート証明書**フィールドを使用して、SSL 認証局を含むファイルを指定します。このファイルは、デフォルトの `~/.postgresql/root.crt` ファイルを置き換えます。SSL 接続が確立されていない場合、このパラメーターは無視されます。
- **証明書失効リスト**フィールドを使用して、SSL 証明書失効リストを含むファイルを指定します。このリストは、`~/.postgresql/root.crl` にあるデフォルトのリストを置き換えます。SSL 接続が確立されていない場合、このパラメーターは無視されます。
- **SSL 圧縮?** が **True** に設定されている場合、SSL 接続で送信されるデータは圧縮されます。デフォルト値は **False**（圧縮は無効）です。SSL 接続が確立されていない場合、このパラメーターは無視されます。

**警告:** 証明書、秘密鍵、および失効リストは、PEM サーバープロセスが実行されるユーザーアカウントが所有するサーバー上のユーザーごとのファイル格納領域に格納されます。これは、サーバーの管理者がこれらのファイルにアクセスできることを意味します。この機能を使用する前に注意してください。

[ **SSHTunnel** ] タブを使用して、SSH トンネリングを構成します。トンネルを使用して、中間プロキシホストを介してデータベースサーバーを、クライアントが直接接続できないネットワーク上にあるサーバーに接続できます。

- 指定したサーバーへの接続時に SSH トンネルを使用するには、**SSH トンネリングを使用する** を **はい** に設定します。
- **トンネルホスト**フィールドに（クライアント接続が転送される）SSH ホストの名前または IP アドレスを指定します。
- **トンネルポート**フィールドで、クライアント接続が転送される SSH ホストのポートを指定します。
- **Username** フィールドに SSH ホストへのログイン権限を持つユーザーの名前を指定します。
- **Authentication** フィールドで、SSH ホストに接続するときに使用する認証の種類を指定します。
- **Password** を選択して、SSH ホストへの認証にパスワードを使用します。これがデフォルトです。
- 接続時に秘密キーファイルを使用するには、[ **Identityfile** ] を選択します。
- SSH ホストが認証用のプライベートキーファイルを想定している場合は、[ **ID ファイル** ] フィールドを使用してキーファイルの場所を指定します。

- SSH ホストがパスワードを想定している場合は、[パスワード] フィールドを使用してパスワードを指定します。ID ファイルを使用している場合は、パスフレーズを指定します。

[ **Advanced** ] タブを使用して、サーバーの管理に使用される詳細を指定します。

- **HostAddress** フィールドにサーバーホストの IP アドレスを指定します。
- **DBRestriction** フィールドを使用して、  
`pg_database` テーブルに対して使用する SQL 制限を指定して、ツリーに表示されるデータベースを制限します。たとえば、  
`'live_db', 'test_db'` と入力して、`live_db` および `test_db` データベースのみを表示します。  
`pg_namespace` に対する制限を入力して、データベースの [プロパティ] ダイアログボックスからデータベースに表示されるスキーマを制限することもできます。
- **パスワードファイル** フィールドを使用して、パスワードファイル (`.pgpass`) の場所を指定します。`.pgpass` ファイルを使用すると、ユーザーは接続時にパスワードを入力せずにログインできます。ファイルは PEM サーバーに存在する必要があります。詳細については、

[Postgres documentation](#) を参照してください。

- **サービス ID** フィールドを使用して、データベースサービスプロセスにパラメーターを指定します。Enterprise Manager ディレクトリに保存されているサーバーの場合、サービス ID を入力します。Windows マシンでは、これは Windows サービスの識別子です。Linux マシンでは、サーバーの起動に使用される `init` スクリプトの名前は `/etc/init.d` であり、サーバーを起動する `systemd` スクリプトの名前は `systemctl` です。たとえば、EDB Postgres Advanced Server 11 サービスの名前は `edb-as-11` です。ローカルサーバーの場合、設定はオペレーティングシステムによって異なります。
- PEM クライアントが Windows マシンで実行されている場合、十分なアクセス権があれば `postmaster` サービスを制御できます。サービスの名前を入力します。リモートサーバーの場合は、マシン名 (`PSE1\pgsql-8.0` など) を追加します。PEM は、ローカルマシンで実行されているサービスを自動的に検出します。-PEM クライアントが Linux マシンで実行されている場合、十分なアクセス権があれば、ローカルマシンで実行されているプロセスを制御できます。`pg_ctl` プログラムにアクセスするためのフルパスと必要なオプションを指定します。サービス制御機能を実行するとき、PEM はこれに `status/start/stop` キーワードを追加します。例：

```
sudo /usr/pgsql-x/bin/pg_ctl -D /var/lib/pgsql/x/data
```

ここで、`x` は PostgreSQL データベースサーバーのバージョンです。

- サーバーがフェールオーバーマネージャークラスターのメンバーである場合、PEM を使用してクラスターの状態を監視し、必要に応じてプライマリノードを置き換えることができます。PEM でフェールオーバーマネージャークラスターを監視できるようにするには、**EFMclustername** フィールドを使用してクラスター名を指定します。クラスター名は、フェールオーバーマネージャークラスタープロパティファイルの名前のプレフィックスです。たとえば、クラスタープロパティファイルの名前が `efm.properties` の場合、クラスター名は `efm`。
- PEM を使用してフェールオーバーマネージャークラスターのステータスを監視している場合は、**EFMインストールパス** フィールドを使用して、フェールオーバーマネージャークラスターバイナリファイルの場所を指定します。デフォルトでは、フェールオーバーマネージャークラスターバイナリファイルは `/usr/edb/efm-x.x/bin` にインストールされます。`x.x` はフェールオーバーマネージャークラスターのバージョンを指定します。

PEM エージェントタブを使用して、PEM エージェントの接続の詳細を指定します。

[ 接続パラメーター ] タブで：

- **リモートモニタリング?** を **Yes** に設定して、PEM エージェントが監視対象サーバーと同じホストにないことを示します。リモート監視が有効になっている場合、監視対象サーバーのエージェントレベルの統計はカスタムチャートとダッシュボードで利用できず、一部の PEM ユーティリティ（Audit Manager、Capacity Manager、Log Manager、Postgres Expert など）はリモートサーバーにアクセスできません。、およびチューニングウィザード）。
- **バインドされたエージェントリスト**を使用して、PEM エージェントを選択します。1つのエージェントで複数の Postgres サーバーを監視できます。
- **ホストフィールド**に、データベースサーバーへの接続時に使用するエージェントの IP アドレスまたはソケットパスを入力します。デフォルトでは、エージェントは [ 全般 ] タブに表示されるホストアドレスを使用します。Unix サーバーでは、/tmp などのソケットパスを指定できます。
- サーバーへの接続時に使用するエージェントのポート番号を入力します。デフォルトでは、エージェントは [ プロパティ ] タブで定義されたポートを使用します。
- **SSL フィールド**を使用して、SSL 操作モードを指定します。 **require**、 **prefer**、 **allow**、 **disable**、 **verify-ca**、または **verify-full** を指定します。

| Mode        | Description                                             |
|-------------|---------------------------------------------------------|
| require     | サーバーとエージェント間のトランザクションに SSL 暗号化を要求します。                   |
| prefer      | SSL 暗号化が使用可能な場合は、サーバーとエージェント間で SSL 暗号化を使用します。           |
| allow       | サーバーで必要な場合は、接続で SSL を使用できるようにします。                       |
| disable     | エージェントとサーバー間の SSL 暗号化を無効にします。                           |
| verify-ca   | SSL 暗号化を要求し、サーバーが認証局によって登録された証明書を使用して認証することを要求します。      |
| verify-full | SSL 暗号化を要求し、サーバーが信頼できる認証局によって登録された証明書を使用して認証することを要求します。 |

SSL 暗号化の使用の詳細については、 [PostgreSQL documentation](#) を参照してください。

- **データベースフィールド**を使用して、エージェントが最初に接続する Postgres データベースの名前を指定します。EDB Postgres 分散データベースノードを登録している場合は、このフィールドに EDB Postgres 分散対応データベース名を指定します。
- **ユーザー名フィールド**に、サーバーへの接続時に使用するエージェントのロールの名前を指定します。すべての機能が期待どおりに動作するには、指定されたロールがデータベースのスーパーユーザーである必要があります。指定されたロールがデータベースのスーパーユーザーではない場合に機能しない機能のリストについては、

[エージェント権限](#) を参照してください。

Kerberos 認証方法を使用している場合は、このフィールドに pgd\_monitor または pgd\_superuser ロールを持つユーザーを指定します。

Postgres バージョン 10 以降を使用している場合、pg\_monitor ロールを使用して、必要な特権を非スーパーユーザーに付与できます。pg\_monitor ロールの詳細については、[Default Roles](#) を参照してください。

- **ユーザー名** フィールドに、サーバーへの接続時に使用するエージェントのユーザーの名前を指定します。すべての機能が期待どおりに動作するには、指定されたロールがデータベースのスーパーユーザーである必要があります。Postgres バージョン 10 以降を使用している場合、pg\_monitor ロールを使用して、必要な特権を非スーパーユーザーに付与できます。pg\_monitor ロールの詳細については、[Default Roles](#) を参照してください。
- **Password** フィールドと **Confirmpassword** フィールドに、サーバーへの接続時に使用するエージェントのパスワードを指定します。パスワードを指定しない場合、エージェントの認証を手動で構成する必要があります。たとえば、.pgpass ファイルを使用できます。これは、エージェントがインストールされているシステムに存在し、アクセスできる必要があります。

[ 詳細設定 ] タブで：

- **Allowtakeover?** を **Yes** に設定して、別のエージェントがサーバーを引き継ぐことを指定します。この機能により、高可用性フェイルオーバープロセスの一部としてサーバーが別のホストに移動した場合などに、エージェントがデータベースサーバーの監視を行うことができます。
- プラス記号 (+) を使用して、PEM 監視から除外するデータベースを追加します。[ **PEM エージェント** ] タブの **接続パラメーター** タブに記載されているデータベースを除外することはできません。

---

**注釈：** データベースレベルのプロブは除外されたデータベースに対して実行されませんが、サーバーレベルのプロブはデータベース統計を収集できます。

---

**BART** タブの下 **General** タブを使用して、PEM サーバーにマッピングする BART サーバーの一般的なプロパティを説明します。

- **BART サーバー** フィールドを使用して、BART サーバー名を選択します。PEM コンソールで構成されたすべての BART サーバーがリストされます。
- **サーバー名** フィールドを使用して、BART サーバーを使用してバックアップするデータベースサーバーの名前を指定します。この名前は BART 構成ファイルに保存されます。
- **バックアップ名** フィールドを使用して、データベースサーバーのバックアップに割り当てるユーザー定義の名前のテンプレートを指定します。バックアップ名前テンプレートを指定しない場合、BART が割り当てた整数バックアップ識別子を使用してのみ BART サブコマンドでバックアップを参照できます。
- **ホストアドレス** フィールドを使用して、バックアップ用に構成するデータベースサーバーの IP アドレスを指定します。
- **ポート** フィールドを使用して、バックアップするデータベースに使用するポートを指定します。
- **User** フィールドを使用して、PEM コンソールから BART を使用してバックアップするデータベースのユーザーを指定します。このデータベースサーバーの増分バックアップを有効にする場合、ユーザーはスーパーユーザーである必要があります。
- **Password** フィールドを使用して、バックアップするデータベースのユーザーのパスワードを指定します。

- **クラスター所有者** フィールドを使用して、データベースクラスターを所有する Linux オペレーティングシステムのユーザーアカウントを指定します。これは通常、Oracle データベース互換モードでインストールされた EDB Postgres Advanced Server データベースクラスターの `enterprisedb` です。通常、PostgreSQL データベースクラスターおよび PostgreSQL データベース互換モードでインストールされた EDB Postgres Advanced Server データベースクラスターの場合は `postgres` です。
- **アーカイブコマンド** フィールドを使用して、`bart.cfg` ファイルで使用するアーカイブコマンド文字列の目的の形式を指定します。Archive コマンドに提供される入力、データベースサーバーの `Postgresql.conf` ファイルを上書きします。サーバーが追加されると、データベースサーバーが再起動するか、データベース構成がリロードされます。
- **増分バックアップを許可しますか?** を使用して、このデータベースサーバーの増分バックアップを有効にするかどうかを指定します。
- **SetpasswordlessSSH?** を使用して、データベースサーバーと BART サーバー間のパスワードなしのログインを許可する SSH 証明書を作成するかどうかを指定します。パスワードなしの SSH 認証を設定する前に、必ず PEM エージェントをバインドしてください。パスワードレス SSH は、PEM エージェントによってリモートで監視されているデータベースサーバーでは機能しません。

BART タブの下の **Misc** タブを使用して、BART Server のその他のプロパティを説明します。

- **デフォルト構成をオーバーライドしますか?** を使用して、BART サーバー構成を特定のデータベースサーバー構成でオーバーライドします。
- **Xlog** メソッドを使用して、`pg_basebackup` 実行中にトランザクションログを取得する方法を指定します。
- **アイテム保持ポリシー** フィールドを使用して、バックアップのアイテム保持ポリシーを指定します。これにより、アクティブなバックアップを古いもの、したがって削除の候補としてマークするタイミングが決まります。保持ポリシーは、バックアップの数または期間（日、週、または月）で指定できます。
- **WAL 圧縮** を使用して、アーカイブされた Xlog / WAL ファイルを `gzip` 形式で圧縮するかどうかを指定します。WAL 圧縮を有効にするには、BART ユーザーアカウントの PATH に `gzip` 圧縮プログラムが存在する必要があります。インクリメンタルバックアップを行う必要があるデータベースサーバーの **WAL 圧縮** 設定を有効にしないでください。
- **復元中に WAL をコピーする** フィールドを使用して、RESTORE 操作を呼び出すときにアーカイブされた WAL ファイルを収集する方法を指定します。**Enabled** に設定して、データベースサーバーのアーカイブリカバリの前にアーカイブされた WAL ファイルを BART バックアップカタログから `<restore_path>/archived_wals` ディレクトリにコピーします。**無効** に設定して、データベースサーバーのアーカイブリカバリ中にアーカイブされた WAL ファイルを BART バックアップカタログから取得します。
- **スレッド数** フィールドを使用して、ブロックをコピーするスレッドの数を指定します。`pg_basebackup` ユーティリティを使用してバックアップを取得する場合は、**スレッドカウント** を **1** に設定する必要があります。
- **バッチサイズ** フィールドを使用して、変更されたブロックのコピーに使用されるメモリブロック数を指定します。このオプションは、増分バックアップにのみ適用されます。
- **スキャン間隔** フィールドを使用して、WAL スキャナーが新しい WAL ファイルをスキャンするまで待機する秒数を指定します。

- **MBM スキャンのタイムアウト** フィールドを使用して、タイムアウトするまでに MBM ファイルを待機する秒数を指定します。このオプションは、増分バックアップにのみ適用されます。

サーバーのプロパティを表示するには、PEM クライアントツリーでサーバー名を右クリックし、コンテキストメニューから **Properties** を選択します。サーバーのプロパティを変更するには、サーバーから切断してから [プロパティ] ダイアログボックスを開きます。

## 2.2 自動サーバー検出

監視するサーバーが監視エージェントと同じホストにある場合は、[自動検出] ダイアログボックスを使用して登録とバインドのプロセスを簡素化できます。

特定のエージェントの自動検出を有効にするには、Server Auto Discovery プローブを有効にする必要があります。これを行うには、PEM クライアントツリーで PEM エージェントを選択し、**Management>ManageProbes** を選択します。**ManageProbes** タブが開いたら、**Enabled?** 列のスライダーが **Yes** に設定されていることを確認します。

Auto Discovery ダイアログボックスを開くには、PEM クライアントツリーで PEM エージェントを選択し、**Management>AutoDiscovery** を選択します。

[ Auto Discovery ] ダイアログボックスが開くと、[ **\*\*DiscoveredDatabaseServers\*\*** ] ボックスに現在 PEM エージェントによって監視されていないサーバーのリストが表示されます。サーバー名の横にあるボックスを選択して、サーバーに関する情報を **サーバー接続の詳細** ボックスに表示し、エージェントの接続プロパティを **エージェント接続の詳細** ボックスに表示します。

**CheckAll** を使用して、表示されているすべてのサーバーの横にあるボックスをオンにするか、**UncheckAll** を使用してサーバー名の左側にあるすべてのボックスをオフにします。

**ServerConnectionDetails** ボックスのフィールドは、PEM が監視するサーバーに関する情報を提供します。

- **Name** フィールドで監視対象サーバーの名前を受け入れるか、変更します。指定した名前が PEM クライアントのツリーに表示されます。
- **サーバーグループ** リストを使用して、サーバーが PEM クライアントツリーに表示されるサーバーグループを選択します。
- **Hostname/address** フィールドを使用して、監視対象サーバーの IP アドレスを指定します。
- **ポート** フィールドには、サーバーによって監視されているポートが表示されます。このフィールドは変更できません。
- **サービス ID** フィールドにサービスの名前を入力します。一部の PEM 機能を有効にするには、サービス名を指定する必要があります。
- デフォルトでは、**メンテナンスデータベース** フィールドは、選択したサーバーが Postgres メンテナンスデータベースを使用することを示します。インストールに合わせて **メンテナンスデータベース** フィールドの内容をカスタマイズします。

**AgentConnectionDetails** ボックスのフィールドは、サーバーへの接続時に使用する PEM エージェントのプロパティを指定します。

- **Host** フィールドには、PEM エージェントバインディングに使用される IP アドレスが表示されます。

- **ユーザー名** フィールドには、選択したサーバーに接続するときに PEM エージェントが使用する名前が表示されます。
- **Password** フィールドには、指定されたユーザー名に関連付けられたパスワードが表示されます。
- **SSL モード** フィールドを使用して、SSL 接続設定を指定します。

監視用にバインドしているサーバーの接続プロパティの指定が完了したら、**OK** を選択してサーバーを登録します。

**OK** を選択すると、新しく登録されたサーバーが PEM ツリーに表示され、PEM サーバーによって監視されます。

## 2.3 pemworker ユーティリティを使用してサーバーを登録する

pemworker ユーティリティを使用して、PEM サーバーによる監視のためにサーバーを登録したり、データベースサーバーの登録を解除したりできます。登録中に、pemworker ユーティリティは、登録コマンドを呼び出したシステムにあるエージェントに新しいサーバーをバインドします。サーバーを登録するには：

Linux ホストでは、次のコマンドを使用します。

```
pemworker --register-server
```

Windows ホストでは、次のコマンドを使用します。

```
pemworker.exe REGISTER-SERVICE
```

pemworker ユーティリティを呼び出すときに、コマンド文字列にコマンドラインオプションを追加します。各オプションの後に対応する値を続けます。

| Option                             | Description                                                                                                                                                                                   |
|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>--pem-user</code>            | PEM 管理ユーザーの名前を指定します。必須。                                                                                                                                                                       |
| <code>--server-addr</code>         | サーバーホストの IP アドレスまたは完全修飾ドメイン名を指定します。Unix ベースのシステムでは、アドレスフィールドを空白のままにして、ローカルマシンでデフォルトの PostgreSQL Unix ドメインソケットを使用できます。または、PostgreSQL ソケットを含む代替パスに設定できます。パスを入力する場合、パスはスラッシュ (/) で始める必要があります。必須。 |
| <code>--server-port</code>         | ホストのポート番号を指定します。必須。                                                                                                                                                                           |
| <code>--server-database</code>     | サーバーが接続するデータベースの名前を指定します。必須。                                                                                                                                                                  |
| <code>--server-user</code>         | サーバーの監視時にエージェントが使用するユーザーの名前を指定します。必須。                                                                                                                                                         |
| <code>--server-service-name</code> | 登録するサーバーでの操作 (STOP、START、RESTART など) を制御するデータベースサービスの名前を指定します。                                                                                                                                |

次のページに続く

表 2 – 前のページからの続き

| Option                                   | Description                                                                                                                                                                                                                                                                                           |
|------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>--remote-monitoring</code>         | <code>--remote-monitoring</code> 句と <code>'no'</code> (デフォルト) の値を含めて、サーバーが PEM エージェントと同じマシンにインストールされていることを示します。リモート監視が有効になっている場合 ( <code>yes</code> )、監視対象サーバーのエージェントレベルの統計はカスタムチャートとダッシュボードで利用できず、一部の PEM ユーティリティ (Audit Manager、Capacity Manager、Log Manager、Postgres Expert、および Tuning Wizard)。必須。 |
| <code>--efm-cluster-name</code>          | サーバーを監視するフェールオーバーマネージャークラスターの名前を指定します (該当する場合)。オプション。                                                                                                                                                                                                                                                 |
| <code>--efm-install-path</code>          | 必要に応じて、Failover Manager のインストールディレクトリへの完全なパスを指定します。オプション。                                                                                                                                                                                                                                             |
| <code>--asb-host-name</code>             | エージェントが接続するホストの名前を指定します。                                                                                                                                                                                                                                                                              |
| <code>--asb-host-port</code>             | データベースへの接続時に使用するエージェントのポート番号を指定します。                                                                                                                                                                                                                                                                   |
| <code>--asb-host-db</code>               | エージェントが接続するデータベースの名前を指定します。                                                                                                                                                                                                                                                                           |
| <code>--asb-host-user</code>             | データベースでの認証時にエージェントが提供するデータベースユーザー名を指定します。                                                                                                                                                                                                                                                             |
| <code>--asb-ssl-mode</code>              | 接続に使用する SSL 認証のタイプを指定します。サポートされている値は次のとおりです。 <b>prefer</b> 、 <b>require</b> 、 <b>disable</b> 、 <b>verify-CA</b> 、 <b>verify-full</b> 。                                                                                                                                                                |
| <code>--group</code>                     | サーバーが表示されるグループの名前を指定します。                                                                                                                                                                                                                                                                              |
| <code>--team</code>                      | サーバーへのアクセスが許可されるグループロールの名前を指定します。                                                                                                                                                                                                                                                                     |
| <code>--owner</code>                     | 監視対象サーバーを所有するロールの名前を指定します。                                                                                                                                                                                                                                                                            |
| <code>--display-name &lt;name&gt;</code> | 監視対象データベースサーバの表示名を指定します。デフォルトはシステムのホスト名です。                                                                                                                                                                                                                                                            |
| <code>--config-dir</code>                | エージェント構成ファイルのディレクトリパスを指定します。デフォルトは <code>"&lt;pemworker path&gt;/../etc"</code> です。                                                                                                                                                                                                                   |

環境変数 `PEM_SERVER_PASSWORD` を設定して PEM サーバーのパスワードを提供し、`pemworker` が PEM 管理ユーザーとして接続できるようにします。

環境変数 `PEM_MONITORED_SERVER_PASSWORD` を設定して、`pemagent` によって登録および監視されるデータベースサーバのパスワードを提供します。

パスワードを入力しないと、パスワード認証エラーが発生します。PEM サーバーは、サーバーが適切に登録されたことを確認します。

### 2.3.1 pemworker ユーティリティを使用してサーバーの登録を解除する

`pemworker` ユーティリティを使用して、データベースサーバの登録を解除できます。サーバーの登録を解除するには、`pemworker` ユーティリティを呼び出します。

Linux ホストでは、次のコマンドを使用します。

```
pemworker --unregister-server
```

Windows ホストでは、次のコマンドを使用します。

```
pemworker.exe UNREGISTER-SERVICE
```

pemworker ユーティリティを呼び出すときに、コマンド文字列にコマンドラインオプションを追加します。各オプションの後に、対応する値を続けます。

| Option                     | Description                                                                                                                                                                                   |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>--pem-user</code>    | PEM 管理ユーザーの名前を指定します。必須。                                                                                                                                                                       |
| <code>--server-addr</code> | サーバーホストの IP アドレスまたは完全修飾ドメイン名を指定します。Unix ベースのシステムでは、アドレスフィールドを空白のままにして、ローカルマシンでデフォルトの PostgreSQL Unix ドメインソケットを使用できます。または、PostgreSQL ソケットを含む代替パスに設定できます。パスを入力する場合、パスはスラッシュ (/) で始める必要があります。必須。 |
| <code>--server-port</code> | ホストのポート番号を指定します。必須。                                                                                                                                                                           |
| <code>--config-dir</code>  | エージェント構成ファイルのディレクトリパスを指定します。デフォルトは <code>"&lt;pemworker path&gt;/../etc"</code> です。                                                                                                           |

PEM\_SERVER\_PASSWORD 環境変数を使用して PEM サーバーのパスワードを指定し、pemworker が PEM 管理ユーザーとして接続できるようにします。

パスワードを入力しないと、パスワード認証エラーが発生します。PEM サーバーは、サーバーが登録解除されたことを確認します。

## 2.4 接続とバインドの確認

登録されると、新しいサーバーが PEM ブラウザツリーに追加され、Global Overview ダッシュボードに表示されます。

新しくバインドされたサーバーに最初に接続すると、グローバル概要ダッシュボードのサーバーリストに「不明」のステータスで新しいサーバーが表示される場合があります。バインドされたエージェントは、サーバーを認識する前に、多くのプローブを実行する必要があります。ネットワークの可用性によっては、これらのプローブが完了するまでに数分かかる場合があります。

数分すると、グローバル概要ダッシュボードの棒グラフに、エージェントが正常に接続されたことが示されます。新しいサーバーが **PostgresServerStatus** リストに含まれています。

5 分後にもグローバル概要ダッシュボードに新しいサーバーがリストされない場合は、モニタリングエージェントのログファイルでエラーを確認します。ツリーでエージェントの名前を右クリックし、コンテキストメニューから **[ダッシュボード]>[プローブログ分析]** を選択します。

## 2.5 AWS での Postgres クラスターの登録

PEM を使用して AWS ホスト上の Postgres クラスターをモニタできる 2 つのシナリオがあります。次を監視できます。

- AWS EC2 で実行されている Postgres クラスター
- AWS RDS で実行されている Postgres クラスター

## 2.5.1 AWS EC2 で実行されている Postgres クラスターのモニタリング

AWS EC2 で Postgres クラスターを作成したら、PEM サーバーを使用してクラスターを登録およびモニタリングできます。現在、次のシナリオがサポートされています。

- 同じ AWS EC2 で実行されている Postgres クラスターと PEM エージェント、およびローカルマシンで実行されている PEM サーバー
- 同じローカルマシンで実行されている Postgres クラスターと PEM エージェント、および AWS EC2 で実行されている PEM サーバー
- 同じ AWS EC2 で実行されている Postgres クラスターと PEM エージェント、および別の AWS EC2 で実行されている PEM サーバー

---

**注釈:** 最初の 2 つのシナリオでは、AWS EC2 インスタンスが `pem` データベースにアクセスできるように AWS EC2 で VPN を構成する必要があります。必要に応じて、ネットワーク管理者に連絡して VPN を設定してください。

---

PEM エージェントは PEM サーバーとは別のホストにあるため、最初に PEM サーバーに PEM エージェントを登録します。また、PEM エージェントを PEM サーバーに登録するときは、AWS EC2 インスタンスのホスト名を使用してください。PEM エージェントの登録の詳細については、[Registering an agent](#) を参照してください。

次の操作を行った後、PEM を使用して Postgres クラスターを監視できます。

- PEM サーバーに PEM エージェントを登録する
- PEM サーバーに追加中に PEM エージェントを Postgres クラスターにバインドします

## 2.5.2 AWS RDS で実行されている Postgres クラスターのモニタリング

AWS RDS データベースの作成中に、エンジンオプションの入力を求められたら **PostgreSQL** を選択します。次の手順については、

[Remote monitoring](#) を参照してください。

## 第3章 Registering a PEM agent

別のホストにインストールされた各 PEM エージェントを PEM サーバーに登録する必要があります。(PEM エージェントは PEM サーバーホストとは異なります。)登録プロセスは、エージェントと通信するために必要な情報を PEM サーバーに提供します。Windows 用の PEM エージェントグラフィカルインストーラーは、エージェントの自己登録をサポートしています。Linux ホストでは、`pemworker` ユーティリティを使用してエージェントに登録する必要があります。

PEM エージェントパッケージインストーラーは、PEM エージェントを `/usr/edb/pem/agent/bin` ディレクトリに配置します。エージェントに登録するには、`pemworker` ユーティリティを呼び出すときに `--register-agent` キーワードに登録の詳細とともに含めます。

```
pemworker --register-agent
```

### 注釈:

- PEM サーバーホストにインストールされた PEM エージェントは、デフォルトで PEM サーバーに登録されます。- PEM エージェントをアップグレードした後、再起動する必要があります。登録は必要ありません。

`pemworker` ユーティリティを呼び出すときに、コマンド文字列にコマンドラインオプションを追加します。各オプションの後に対応する値を続けます。

| Option                        | Description                                                                                                      |
|-------------------------------|------------------------------------------------------------------------------------------------------------------|
| <code>--pem-server</code>     | PEM バックエンドデータベースサーバーの IP アドレス。このパラメーターは必須です。                                                                     |
| <code>--pem-port</code>       | PEM バックエンドデータベースサーバーのポート。デフォルト値は `5432` です。                                                                      |
| <code>--pem-user</code>       | PEM バックエンドデータベースサーバーのスーパーユーザー権限を持つデータベースユーザーの名前。このパラメーターは必須です。                                                   |
| <code>--pem-agent-user</code> | PEM サーバーバックエンドデータベースサーバーに接続するエージェントユーザー。                                                                         |
| <code>--cert-path</code>      | 証明書が作成されるディレクトリへの完全なパス。パスを指定しない場合、証明書は Linux では <code>~/pem</code> に、Windows <code>%APPDATA%/pem</code> に作成されます。 |
| <code>--config-dir</code>     | 構成ファイルのディレクトリパス。デフォルトは <code>&lt;pemworker path&gt;/../etc</code> です。                                            |
| <code>--display-name</code>   | PEM ブラウザーツリーに表示されるエージェントのわかりやすい名前。デフォルトはシステムのホスト名です。                                                             |

次のページに続く

表 1 – 前のページからの続き

| Option                                     | Description                                                                                                                                 |
|--------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <code>--force-registration</code>          | <code>`force_registration`</code> 句を含めて、指定された引数にエージェントを登録します。この句は、既存のエージェント構成をオーバーライドする場合に役立ちます。デフォルト値は <code>`Yes`</code> です。              |
| <code>--group</code>                       | エージェントが表示されるグループの名前。                                                                                                                        |
| <code>--team</code>                        | 監視対象データベースサーバーにアクセスする PEM バックエンドデータベースサーバー上のデータベースロールの名前。                                                                                   |
| <code>--owner</code>                       | エージェントを所有する PEM バックエンドデータベースサーバーのデータベースユーザーの名前。                                                                                             |
| <code>--allow_server_restart</code>        | <code>`allow_server_restart`</code> パラメーターを有効にして、PEM が監視対象サーバーを再起動できるようにします。デフォルト値は <code>`True`</code> です。                                 |
| <code>--allow_batch_probes</code>          | <code>`allow_batch_probes`</code> パラメーターを有効にして、PEM がこのエージェントでバッチプローブを実行できるようにします。デフォルト値は <code>`False`</code> です。                           |
| <code>--batch-script-user</code>           | バッチ/シェルスクリプトの実行に使用するオペレーティングシステムユーザー。デフォルト値はなしです。このパラメーターを空白のままにするか、指定したユーザーが存在しない場合、スクリプトは実行されません。                                         |
| <code>--enable-heartbeat-connection</code> | <code>`enable-heartbeat-connection`</code> パラメーターを有効にして、PEM エージェントとサーバー間に専用のハートビート接続を作成して、アクティブステータスを更新します。デフォルト値は <code>`False`</code> です。 |
| <code>--enable-smtp</code>                 | <code>`enable-smtp parameter`</code> を有効にして、PEM エージェントが PEM サーバーに代わって電子メールを送信できるようにします。デフォルト値は <code>`False`</code> です。                     |
| <code>--enable-snmp</code>                 | <code>`enable-snmp parameter`</code> を有効にして、PEM エージェントが PEM サーバーに代わって SNMP トラップを送信できるようにします。デフォルト値は <code>`False`</code> です。                |
| <code>-o</code>                            | 構成ファイルのオプションをオーバーライドするかどうかを指定します。                                                                                                           |

pemagent（監査マネージャー、ログマネージャー、チューニングウィザードなど）でデータベースサーバーの再起動が必要な PEM 機能を使用する場合は、`agent.cfg` ファイルで `allow_server_restart` の値を設定する必要があります。

**注釈:** PEM バージョン 7.11 以降がインストールされている PEM エージェントによって実行されるシェルスクリプトを構成する場合、`batch_script_user` パラメーターにユーザーを指定する必要があります。スクリプトの実行には root 以外のユーザーを使用することを強くお勧めします。root ユーザーを使用すると、データのセキュリティとオペレーティングシステムのセキュリティが危険にさらされる可能性があります。ただし、root ユーザーを使用してスクリプトを実行して pemagent を元の設定に復元する場合は、`batch_script_user` パラメーターを ``root`` に設定する必要があります。

PEM データベースで変更が行われる前に、接続エージェントは PEM データベースサーバーで認証されます。pemworker ユーティリティを呼び出すときは、PEM サーバー管理ユーザーロール（postgres）に関連付けられたパスワードを指定する必要があります。管理パスワードは次の 3 つの方法で指定できます。

- PEM\_MONITORED\_SERVER\_PASSWORD 環境変数を設定します。
- PGPASSWORD キーワードを使用してコマンドラインでパスワードを入力します。
- .pgpass ファイルにエントリーを作成します。

パスワードを入力しないと、パスワード認証エラーが発生します。その他の不足している必須情報の入力を求められます。登録が完了すると、サーバーはエージェントが正常に登録されたことを確認します。

### 3.1 PEM エージェントの登録解除

pemworker ユーティリティを使用して、PEM エージェントの登録を解除できます。エージェントの登録を解除するには、pemworker ユーティリティを呼び出すときに `--unregister-agent` キーワードを詳細とともに含めます。

```
pemworker --unregister-agent
```

pemworker ユーティリティを呼び出すときに、コマンド文字列にコマンドラインオプションを追加します。各オプションの後に対応する値を続けます。

| Option                                   | Description                                                                   |
|------------------------------------------|-------------------------------------------------------------------------------|
| <code>--pem-user &lt;username&gt;</code> | PEM バックエンドデータベースサーバーのデータベースユーザー (pem_admin ロールのメンバー) の名前を指定します。このパラメーターは必須です。 |
| <code>--config-dir</code>                | 構成ファイルのディレクトリパスを指定します。デフォルトは <code>"&lt;pemworker path&gt;/../etc"</code> です。 |

### 3.2 PEM エージェント構成パラメーターの設定

PEM エージェント RPM インストーラーは、`/usr/edb/pem/agent/etc` ディレクトリに `agent.cfg.sample` という名前のサンプル構成ファイルを作成します。PEM エージェントを登録すると、pemworker プログラムは `agent.cfg` という名前の実際のエージェント構成ファイルを作成します。 `agent.cfg` ファイルを変更して、次の構成パラメーターを追加する必要があります。

```
heartbeat_connection = true
```

`ca-bundle.crt` 認証局ファイルの場所も追加する必要があります。デフォルトでは、インストーラーは `agent.cfg.sample` ファイルで指定された場所に `ca-bundle.crt` ファイルを作成します。サンプルファイルからデフォルトのパラメーター値をコピーできます。別の場所に保存されている `ca-bundle.crt` ファイルを使用する場合は、その値を `ca_file` パラメーターに指定します。

```
ca_file=/usr/libexec/libcurl-pem7/share/certs/ca-bundle.crt
```

次に、プラットフォーム固有のコマンドを使用して PEM エージェントサービスを開始します。サービスの名前は `pemagent` です。

RHEL または CentOS 7.x または 8.x ホストでは、`systemctl` を使用してサービスを開始します。

```
systemctl start pemagent
```

サービスは、エージェントを起動していることを確認します。エージェントが登録されて起動されると、グローバル概要ダッシュボードと PEM Web インターフェースのオブジェクトブラウザツリーに表示されます。

pemworker ユーティリティを使用してサーバーを登録する方法については、  
データベースサーバーの手動登録 を参照してください。

### 3.3 非 root ユーザーアカウントを使用して PEM エージェントを登録する

非 root ユーザーアカウントを使用して PEM エージェントを登録するには、最初に root ユーザーとして PEM エージェントをインストールする必要があります。インストール後、edb などの非 root ユーザーの ID を引き継ぎます。次に：

1.edb でログインします。pem および logs ディレクトリを作成し、読み取り、書き込み、および実行のアクセス許可を割り当てます。

```
$ mkdir /home/edb/pem
$ mkdir /home/edb/pem/logs
$ chmod 700 /home/edb/pem
$ chmod 700 /home/edb/pem/logs
```

2. エージェントを PEM サーバーに登録します。

```
$ export PEM_SERVER_PASSWORD=edb

# Use the following command to create agent certificates and an agent
# configuration file (agent.cfg) in the `/home/edb/pem` directory.
$ /usr/edb/pem/agent/bin/pemworker --register-agent --pem-server <172.19.11.230> --
→pem-user postgres --pem-port 5432 --display-name non_root_pem_agent --cert-path /
→home/edb/pem --config-dir /home/edb/pem

# Use the following command to assign read and write permissions to
# these files:
$ chmod -R 600 /home/edb/pem/agent*
```

3. agent.cfg ファイルのパラメーターを変更します。

```
$ vi /home/edb/pem/agent.cfg
agent_ssl_key=/home/edb/pem/agent<id>.key
agent_ssl_cert=/home/edb/pem/agent<id>.crt
log_location=/home/edb/pem/worker.log
agent_log_location=/home/edb/pem/agent.log
```

<id> は、割り当てられた PEM エージェント ID です。

4. tmp ディレクトリを作成し、環境変数を設定して、エージェントを起動します。

```
$ mkdir /home/edb/pem/tmp

# Create a script file, add the environment variable, give permissions, and execute:
```

(次のページに続く)

(前のページからの続き)

```
$ vi /home/edb/pem/run_pemagent.sh
#!/bin/bash
export TEMP=/home/edb/agent/tmp
/usr/edb/pem/agent/bin/pemagent -c /home/edb/agent/agent.cfg
$ chmod a+x /home/edb/pem/run_pemagent.sh
$ cd /home/edb/pem
$ ./run_pemagent.sh
```

PEM エージェントが登録され、edb ユーザーで起動されました。マシンが再起動しても、このエージェントは自動的に再起動しません。前のコマンドを使用して手動で起動する必要があります。

5. オプションで、次のように root ユーザーとしてこの PEM エージェントのサービスを作成して、マシンの再起動時にこのエージェントを自動的に起動できます。

a. スーパーユーザーとして `pemagent` サービスファイルの構成ファイルパスとユーザーの値を更新します。

```
$ sudo vi /usr/lib/systemd/system/pemagent.service
[Service]
Type=forking
WorkingDirectory=/home/edb/pem
Environment=LD_LIBRARY_PATH=/usr/edb/pem/agent/lib:/usr/libexec/edb-snmp++/lib
Environment=TEMP=/home/edb/pem/tmp
ExecStart=/usr/edb/pem/agent/bin/pemagent -c /home/edb/pem/agent.cfg
```

b. 実行中のエージェントプロセスを停止し、エージェントサービスを再起動します。

```
# Find the process id of the running pem agent and pem worker process and kill that
↳ process
$ ps -ax | grep pemagent
$ kill -9 <process_id_of_pemagent>
$ ps -ax | grep pemworker
$ kill -9 <process_id_of_pemworker>
# Enable and start pemagent service
$ sudo systemctl enable pemagent
$ sudo systemctl start pemagent
$ sudo systemctl status pemagent
```

6. PEM ダッシュボードでエージェントのステータスを確認します。

**注釈:** root 権限または別のユーザー（たとえば、enterprisedb）が所有するファイルへのアクセスを必要とするプローブとジョブは失敗します。



## 第4章 Managing a PEM server

PEM サーバーの管理に関連するタスクには次のものがあります。

- PEM サーバーとエージェントの再起動
- PEM サーバーまたは PEM エージェントの制御
- Linux および Windows での HTTPD サービスの制御
- HTTPD サーバーの制御
- PEM 認証とセキュリティの管理
- `pg_hba.conf` ファイルの変更
- プロキシサーバーを使用するように PEM を変更

### 4.1 PEM サーバーとエージェントの起動と停止

PEM サーバーは、それが存在する場所の Postgres サーバーインスタンスが起動、停止、または再起動します。Postgres サーバーの制御に使用するのと同じコマンドを使用して PEM サーバーを制御します。Linux プラットフォームでは、サービススクリプトを停止および開始するコマンドは、プラットフォームと OS バージョンによって異なります。

PEM エージェントは、`pemagent` という名前のサービスによって制御されます。

Windows オペレーティングシステムには、サーバーの状態を表示し、対話型のサーバー制御を提供するグラフィカルサービスコントローラーが含まれています。サービスユーティリティには、Windows のコントロールパネルからアクセスできます。ユーティリティが開いたら、リストされたサービスをナビゲートし、サービス名を選択します。

**Stop**、**Pause**、**Start**、または **Restart** を選択して、サービスの状態を制御します。

サービスを停止すると、Postgres サーバーに接続されているユーザーまたはクライアントアプリケーションが突然切断されます。サービスの制御の詳細については、  
EDB Postgres Advanced Server を参照してください。

## 4.2 Linux での PEM サーバーまたは PEM エージェントの制御

Linux プラットフォームでは、以下を制御するサービススクリプトの名前。

- EDB Postgres Advanced Server 上の PEM サーバーは `edb-as-<x>` または `ppas-<x>` です。
- PostgreSQL 上の PEM サーバーは `postgresql-<x>` です。
- PEM エージェントは `pemagent` です。

`x` は、サーバーのバージョン番号を示します。

サービススクリプトを使用して、サービスを制御できます。

- RHEL または CentOS バージョン 7.x または 8.x でサービスを制御するには、コマンドプロンプトでスーパーユーザ権限を引き受けて、次のコマンドを発行します。

```
systemctl <service_name> <action>
```

そこで：

`service_name` はサービスの名前です。

`action` は、サービスが実行するアクションを指定します。次を指定します。

- `start` でサービスを開始します。
- `stop` でサービスを停止します。
- `restart` を停止して、サービスを開始します。
- `status` でサービスのステータスを確認します。

## 4.3 Windows での PEM サーバーまたは PEM エージェントの制御

Windows オペレーティングシステムには、サーバーの状態を表示し、対話型のサーバー制御を提供するグラフィカルサービスコントローラーが含まれています。以下を制御するサービスの登録名。

- PostgreSQL 上の PEM サーバーホストは `postgresql-<x>` です。
- Advanced Server の PEM サーバーホストは `edb-as-<x>` または `ppas-<x>` です。
- PEM エージェントは Postgres Enterprise Manager - `pemAgent` です。

`x` は、サーバーのバージョン番号を示します。

Windows のコントロールパネルで、サービスユーティリティを開き、サービスのリストを表示します。

[ **サービスを停止する** ] を選択して、サービスを停止します。サービスを停止すると、サーバーに接続されているユーザーまたはクライアントアプリケーションが突然切断されます。

[ **サービスを一時停止** ] を選択して、サービスの構成パラメーターをリロードします。**サービスを一時停止** は、多くの構成パラメーターのユーザーセッションを中断せずにパラメーターをリセットする効果的な方法です。

[ サービスを開始する ] を選択して、サービスを開始します。

## 4.4 HTTPD サーバーの制御

Linux では、コマンドプロンプトで次のコマンドを入力して、PEM-HTTPD サービスのステータスを確認できます。

```
ps -ef | grep httpd
```

Linux が次のような回答で応答した場合、httpd は実行されていません。

```
user 13321 13267 0 07:37 pts/1 00:00:00 grep httpd
```

CentOS または RHEL 7.x または 8.x システムでサービスを開始するには、次のコマンドを使用します。

```
systemctl start httpd
```

Windows、サービスアプレットを使用して PEM HTTPD サービスのステータスを確認できます。サービスアプレットを開いたら、リストから PEM HTTPD サービスを選択します。

Status 列には、サーバーの現在の状態が表示されます。サービスが実行されていない場合は、[ **Start** ] を選択して PEM HTTPD を起動します。

## 4.5 PEM 認証の管理

Postgres は多くの認証方法をサポートしています：

- 安全なパスワード (md5)
- GSSAPI
- SSPI
- ケルベロス
- アイデント
- LDAP
- RADIUS
- 証明書 (SSL)
- PAM

Postgres と PEM の認証は、`pg_hba.conf` 構成ファイルによって制御されます。構成ファイルのエントリは以下を指定します。

- 特定のデータベースに接続できる人
- そのユーザーが接続を許可される前に必要な認証の種類

postgres という名前のユーザーが安全なパスワード (MD5) 認証接続を使用してローカルホスト (127.0.0.1/32) からすべてのデータベースに接続できるようにする `pg_hba.conf` ファイルの一般的なエントリーは次のとおりです。

```
host all postgres 127.0.0.1/32 md5
```

システム構成によっては、PEM エージェントがサーバーへの接続に使用するユーザーアカウントのパスワードファイルを作成する必要がある場合があります。このファイルにより、エージェントはサーバーの認証要求に適切に応答できます。パスワードが `1safepwd` である postgres という名前のユーザーのパスワードファイルのエントリーは次の形式です。

```
localhost:5432:*:postgres:1safepwd
```

パスワードファイルの名前は通常、Linux システムでは `~root/.pgpass`、Windows%APPDATA%\postgresql\pgpass.conf です。パスワードファイルの構成の詳細については、[PostgreSQL website](#) を参照してください。

Postgres でサポートされている認証方法の詳細については、[PostgreSQL core documentation](#) を参照してください。

## 4.6 PEM サーバー構成の編集

PEM クライアントを使用して PEM サーバーの構成パラメーターを管理し、機能を有効にしたり、デフォルト設定を変更したりできます。[ Server Configuration ] ダイアログボックスを開くには、**Management>ServerConfiguration** を選択します。次に、必要に応じてパラメーター値を編集します。

## 4.7 セキュリティの管理

PEM は、Postgres のロールとサーバーを管理するためのインターフェイスを提供します。

### 4.7.1 ログインロール

PEM サーバーに接続するときは、PEM サーバーがデータを保存するデータベースへのアクセスを許可するロールの資格情報を提供する必要があります。デフォルトでは、postgres スーパーユーザーアカウントがサーバーへの最初の接続に使用されますが、接続するユーザーごとに個々のロールを作成することを強くお勧めします (セキュリティと監査の両方の目的)。PEM クエリツール、PEM Web インタフェースの **Create - Login/Group Role** ダイアログ、またはコマンドラインクライアント (psql など) を使用してロールを作成できます。

Create - Login/Group Role ダイアログボックスを使用してロールを作成するには：

1. PEM ツリーで、ロールを作成するサーバーのノードを展開します。
2. **Login/GroupRoles** ノードを右クリックします。
1. コンテキストメニューから、**作成>ログイン/グループロール**を選択します。
  1. [Create - Login/Group Role] ダイアログボックスのタブを使用して、ロールを定義します。

1. 新しいロールの定義が完了したら、**保存**を選択してロールを作成します。

既存のログインロールのプロパティを変更するには、ツリーでログインロールの名前を右クリックし、コンテキストメニューから**プロパティ**を選択します。ログインロールを削除するには、ロールの名前を右クリックし、コンテキストメニューから **Delete/Drop** を選択します。

ロールの作成と管理の詳細については、 [PostgreSQL online documentation](#) を参照してください。

## 4.7.2 グループロール

グループロールはコンテナとして機能し、データベースの作成などのシステム権限や、テーブルへのデータの挿入などのオブジェクト権限を分配するために使用されます。グループロールの主な目的は、DBA がシステムとオブジェクトの権限を簡単に管理できるようにすることです。さまざまなログインアカウントに特権を個別に割り当てまたは変更する代わりに、単一のロールの特権を割り当てまたは変更し、そのロールを一度に多くのログインロールに付与できます。

(PEM ツリーの各登録済みサーバーの名前の下にある) **GroupRoles** ノードを使用して、グループロールを作成および管理します。コンテキストメニューのオプションから、新しいロールを作成したり、既存のロールのプロパティを変更したりできるダイアログボックスにアクセスできます。ロールの作成の詳細については、

[PostgreSQL documentation](#) を参照してください。

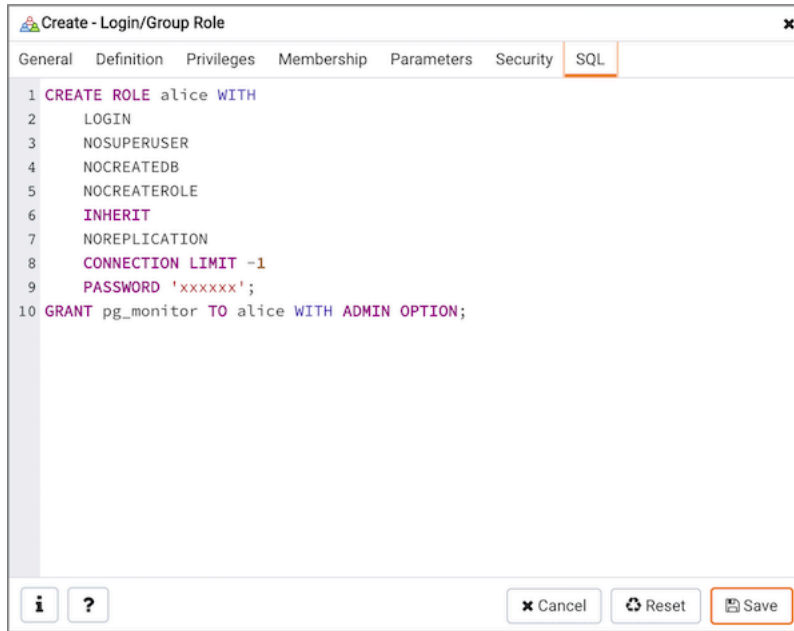
## 4.7.3 PEM の事前定義されたロールを使用して PEM 機能へのアクセスを管理する

Login/Group Role ダイアログボックスを使用して、権限が制限されたロールに、監査マネージャー、Capacity Manager、SQL プロファイラーなどの PEM 機能へのアクセスを許可できます。PEM の事前定義されたロールにより、PEM 機能へのアクセスが許可されます。これらのロールのメンバーシップが割り当てられているロールは、関連する機能にアクセスできます。

ユーザーを定義するときに、[ **メンバーシップ** ] タブを使用して、新しいユーザーがメンバーであるロールを指定します。新しいユーザーは、自分がメンバーである各ロールに関連付けられた特権を共有します。ユーザーが PEM 拡張機能にアクセスするには、ロールが pem\_user ロールと、機能へのアクセスを許可する事前定義されたロールのメンバーである必要があります。**Roles** フィールドを使用して、リストから事前定義されたロール名を選択します。

[ **SQL** ] タブには、**保存**を選択したときにサーバーが実行する SQL コマンドが表示されます。

この例では、監査マネージャーにアクセスする acctg\_clerk という名前のログインロールを作成します。このロールは、任意の時点でサーバーに無制限に接続できます。



PEM の事前定義されたロールを使用して、表にリストされている機能へのアクセスを許可できます。

| Value                            | Parent role                     | Description                                                 |
|----------------------------------|---------------------------------|-------------------------------------------------------------|
| pem_super_admin                  |                                 | PEM 上のすべてを管理/構成するロール                                        |
| pem_admin                        | pem_super_admin                 | すべての可視エージェント/サーバーおよび監視対象オブジェクトの管理/管理/構成のロール                 |
| pem_config                       | pem_admin                       | PEM の構成管理のロール                                               |
| pem_component                    | pem_admin                       | ウィザード/ダイアログボックススペースのすべてのコンポーネントを実行/実行するロール                  |
| pem_rest_api                     | pem_admin                       | REST API にアクセスするロール                                         |
| pem_server_service_management    | pem_admin                       | 監視対象データベースサーバの再起動/リロードを許可するロール（サーバー ID が提供されている場合）          |
| pem_manage_schedule_task         | pem_admin                       | スケジュールタスクを設定するロール                                           |
| pem_manage_alert                 | pem_admin                       | アラートとそのテンプレートを管理・設定するロール                                    |
| pem_config_alert                 | pem_config,<br>pem_manage_alert | 監視対象オブジェクトのアラートを構成するためのロール                                  |
| pem_manage_probe                 | pem_admin                       | カスタムプローブを作成、更新、削除し、カスタムプローブ構成を変更するロール                       |
| pem_config_probe                 | pem_config,<br>pem_manage_probe | 表示されているすべての監視対象オブジェクトに対するプローブ構成（履歴保持、実行頻度、プローブの有効化/無効化）のロール |
| pem_database_server_registration | pem_admin                       | データベースサーバを登録するロール                                           |
| pem_comp_postgres_expert         | pem_component                   | Postgres Expert を実行するロール                                    |
| pem_comp_auto_discovery          | pem_component                   | データベースサーバの自動検出ダイアログを実行するロール                                 |
| pem_comp_log_analysis_expert     | pem_component                   | Log Analysis Expert を実行するロール                                |

次のページに続く

表 1 – 前のページからの続き

| Value                     | Parent role   | Description               |
|---------------------------|---------------|---------------------------|
| pem_comp_sqlprofiler      | pem_component | SQL プロファイラーを実行するロール       |
| pem_manage_efm            | pem_admin     | Failover Manager を管理するロール |
| pem_comp_capacity_manager | pem_component | Capacity Manager を実行するロール |
| pem_comp_log_manager      | pem_component | ログマネージャーを実行するロール          |
| pem_comp_audit_manager    | pem_component | Audit Manager を実行するロール    |
| pem_comp_tuning_wizard    | pem_component | Tuning Wizard を実行するロール    |

#### 4.7.4 チームロールの使用

PEM による監視用にサーバーを登録するときに、サーバーに関連付けるチームを指定できます。チームは、1 つ以上の監視対象サーバーへのアクセスをロールメンバーの制限されたグループに許可または制限するために使用できるグループロールです。PEM クライアントは、次のユーザーにのみ指定されたチームのサーバーを表示します。

- チームロールのメンバー
- サーバーを作成したロール
- PEM サーバーでスーパーユーザー権限を持つロール

チームロールを作成するには：

1. PEM ツリーで、ロールを作成するサーバーのノードを展開します。
  2. **Login/GroupRoles** ノードを右クリックします。
1. コンテキストメニューから、**作成>ログイン/グループロール**を選択します。
  1. [作成 - ログイン/グループロール] ダイアログボックスで、フィールドを使用してチームロールのプロパティを指定します。

#### 4.7.5 オブジェクトの権限

データベースオブジェクトにアクセス、実行、または作成する前に、ロールに十分な権限を付与する必要があります。PEM では、PEM クライアントインターフェイスを使用して、グループロールまたはログインアカウントにオブジェクト権限を割り当て（GRANT）および削除（REVOKE）できます。

オブジェクトの権限は、特定のオブジェクトごとにグラフィカルオブジェクトエディターで管理されます。たとえば、データベーステーブルにアクセスする特権を割り当てるには、ツリーでテーブル名を右クリックし、コンテキストメニューから**プロパティ**を選択します。[ **Privileges** ] タブを使用して、テーブルに特権を割り当てます。

PEM クライアントには、一度に多くのオブジェクト権限を管理できる付与ウィザード（[ツール] メニューからアクセス）も含まれています。

## 4.8 サーバー構成パラメーター - リファレンス

グローバル構成オプションを使用して、PEM サーバーの動作の側面を変更できます。構成パラメーターのリストは変更される場合があります。

| Parameter name             | Value and unit | Description                                                                                |
|----------------------------|----------------|--------------------------------------------------------------------------------------------|
| audit_log_retention_time   | 30 days        | PEM サーバーに監査ログを保持する日数を指定します。                                                                |
| auto_create_agent_alerts   | true           | エージェントの登録時にデフォルトのエージェントレベルのアラートを自動的に作成するかどうかを指定します。                                        |
| auto_create_server_alerts  | true           | サーバーがエージェントにバインドされたときに、デフォルトのサーバーレベルのアラートを自動的に作成するかどうかを指定します。                              |
| chart_disable_bullets      | false          | ダッシュボードと Capacity Manager レポートの折れ線グラフの簡条書きを有効/無効にします。                                      |
| cm_data_points_per_report  | 60             | Capacity Manager レポートのグラフにプロットするデータポイントの数を指定します。                                           |
| cm_max_end_date_in_years   | 5 years        | Capacity Manager がデータを推定する最大時間を指定します。レポートのしきい値ベースの終了日が無期限に推定されないようにします。                    |
| dash_alerts_timeout        | 60 seconds     | アラート ダッシュボードのコンポーネントが更新されるまでの秒数を指定します。                                                     |
| dash_db_comrol_span        | 7 days         | Database Analysis および Server Analysis ダッシュボードのコミット/ロールバック分析グラフにプロットするデータの日数を指定します。         |
| dash_db_comrol_timeout     | 1800 seconds   | Database Analysis および Server Analysis ダッシュボードでコミット/ロールバック折れ線グラフが更新されるまでの秒数を指定します。          |
| dash_db_connovervw_timeout | 300 seconds    | Database Analysis ダッシュボードで接続の概要の円グラフが更新されるまでの秒数を指定します。                                     |
| dash_db_eventlag_span      | 7 days         | データベース分析ダッシュボードのスロニーレプリケーションのイベント数ラグチャートにプロットするデータの日数を指定します。                               |
| dash_db_eventlag_timeout   | 1800 seconds   | Database Analysis ダッシュボードでスロニーレプリケーションの Number of Events Lag 折れ線グラフが更新されるまでの秒数を指定します。      |
| dash_db_hottable_rows      | 25 rows        | データベース分析ダッシュボードの HOT Table Analysis テーブルに表示する行数を指定します。                                     |
| dash_db_hottable_timeout   | 300 seconds    | Database Analysis ダッシュボードで Hot Tables テーブルが更新されるまでの秒数を指定します。                               |
| dash_db_io_span            | 7 days         | Database Analysis および I/O Analysis ダッシュボードの Database I/O Analysis チャートにプロットするデータの日数を指定します。 |

次のページに続く

表 2 – 前のページからの続き

| Parameter name               | Value and unit | Description                                                                                  |
|------------------------------|----------------|----------------------------------------------------------------------------------------------|
| dash_db_io_timeout           | 1800 seconds   | Database Analysis および I/O Analysis ダッシュボードで Database I/O 折れ線グラフが更新されるまでの秒数を指定します。            |
| dash_db_rowact_span          | 7 days         | Database Analysis、I/O Analysis、および Server Analysis ダッシュボードの行アクティビティ分析グラフにプロットするデータの日数を指定します。 |
| dash_db_rowact_timeout       | 1800 seconds   | Database Analysis、I/O Analysis、および Server Analysis ダッシュボードで行アクティビティ折れ線グラフが更新されるまでの秒数を指定します。  |
| dash_db_storage_timeout      | 300 seconds    | Database Analysis ダッシュボードでストレージ棒グラフが更新されるまでの秒数を指定します。                                        |
| dash_db_timelag_span         | 7 days         | Database Analysis ダッシュボードの Slony レプリケーションのタイムラグチャートにプロットするデータの日数を指定します。                      |
| dash_db_timelag_timeout      | 1800 seconds   | Slony レプリケーションのタイムラグ折れ線グラフが Database Analysis ダッシュボードで更新されるまでの秒数を指定します。                      |
| dash_db_useract_span         | 7 days         | データベース分析ダッシュボードのユーザーアクティビティ分析グラフにプロットするデータの日数を指定します。                                         |
| dash_db_useract_timeout      | 1800 seconds   | データベース分析ダッシュボードでユーザーアクティビティ折れ線グラフが更新されるまでの秒数を指定します。                                          |
| dash_efm_timeout             | 300 seconds    | ストリーミングレプリケーションダッシュボードでフェールオーバーマネージャノードのステータスとフェールオーバーマネージャクラスター情報の折れ線グラフが更新されるまでの秒数を指定します。  |
| dash_global_overview_timeout | 30 seconds     | グローバル概要ダッシュボードのコンポーネントが更新されるまでの秒数を指定します。                                                     |
| dash_header_timeout          | 60 seconds     | すべてのダッシュボードのヘッダーの情報が更新されるまでの秒数を指定します。                                                        |
| dash_io_chkpt_span           | 7 days         | I/O 分析ダッシュボードのチェックポイントチャートにプロットするデータの日数を指定します。                                               |
| dash_io_chkpt_timeout        | 1800 seconds   | I/O Analysis ダッシュボードでチェックポイント折れ線グラフが更新されるまでの秒数を指定します。                                        |
| dash_io_hotindx_timeout      | 300 seconds    | I/O Analysis ダッシュボードで Hot Indexes 棒グラフが更新されるまでの秒数を指定します。                                     |

次のページに続く

表 2 – 前のページからの続き

| Parameter name                     | Value and unit | Description                                                                            |
|------------------------------------|----------------|----------------------------------------------------------------------------------------|
| dash_io_hottbl_timeout             | 300 seconds    | I/O Analysis ダッシュボードで Hot Tables 棒グラフが更新されるまでの秒数を指定します。                                |
| dash_io_index_objectio_rows        | 25 rows        | I/O Analysis および Object Activity Analysis ダッシュボードのインデックスアクティビティテーブルに表示される行数を指定します。     |
| dash_io_index_objectio_timeout     | 300 seconds    | I/O Analysis および Object Activity Analysis ダッシュボードでインデックスアクティビティテーブルが更新されるまでの秒数を指定します。  |
| dash_io_objectio_rows              | 25 rows        | I/O Analysis および Object Activity Analysis ダッシュボードのオブジェクト I/O の詳細表に表示される行数を指定します。       |
| dash_io_objectio_timeout           | 300 seconds    | I/O Analysis および Object Activity Analysis ダッシュボードでオブジェクト I/O の詳細テーブルが更新されるまでの秒数を指定します。 |
| dash_memory_hostmemact_days        | 7 days         | メモリ分析ダッシュボードのホストメモリアクティビティ分析グラフにプロットするデータの日数を指定します。                                    |
| dash_memory_hostmemact_timeout     | 800 seconds    | メモリ分析ダッシュボードでホストメモリアクティビティ線グラフが更新されるまでの秒数を指定します。                                       |
| dash_memory_hostmemcomp_timeout    | 300 seconds    | メモリ分析およびサーバー分析ダッシュボードでホストメモリ構成の円グラフが更新されるまでの秒数を指定します。                                  |
| dash_memory_servmemact_days        | 7 days         | メモリ分析ダッシュボードのサーバーのメモリアクティビティ分析グラフにプロットするデータの日数を指定します。                                  |
| dash_memory_servmemact_timeout     | 800 seconds    | メモリ分析ダッシュボードでサーバーメモリアクティビティの折れ線グラフが更新されるまでの秒数を指定します。                                   |
| dash_memory_servmemcomp_timeout    | 300 seconds    | メモリ分析ダッシュボードでサーバーメモリ構成の円グラフが更新されるまでの秒数を指定します。                                          |
| dash_objectact_objstorage_rows     | 5 rows         | Object Activity Analysis ダッシュボードの Object Storage テーブルに表示する行数を指定します。                    |
| dash_objectact_objstorage_timeout  | 300 seconds    | Object Activity Analysis ダッシュボードで Object Storage テーブルが更新されるまでの秒数を指定します。                |
| dash_objectact_objtopindex_timeout | 300 seconds    | オブジェクト アクティビティ分析ダッシュボードで上位 5 つのインデックスの棒グラフが更新されるまでの秒数を指定します。                           |
| dash_objectact_objtoptable_timeout | 300 seconds    | オブジェクト アクティビティ分析ダッシュボードで上位 5 つのテーブルの棒グラフが更新されるまでの秒数を指定します。                             |

次のページに続く

表 2 – 前のページからの続き

| Parameter name          | Value and unit | Description                                                |
|-------------------------|----------------|------------------------------------------------------------|
| dash_os_cpu_span        | 7 days         | オペレーティング システム分析ダッシュボードの CPU チャートにプロットするデータの日数を指定します。       |
| dash_os_cpu_timeout     | 1800 seconds   | オペレーティングシステム分析ダッシュボードで CPU 折れ線グラフが更新されるまでの秒数を指定します。        |
| dash_os_data_span       | 7 days         | オペレーティング システム分析ダッシュボードの I/O 折れ線グラフにプロットするデータの日数を指定します。     |
| dash_os_disk_span       | 7 days         | オペレーティング システム分析ダッシュボードの使用率グラフにプロットするデータの日数を指定します。          |
| dash_os_hostfs_timeout  | 1800 seconds   | オペレーティングシステム分析ダッシュボードでホストファイルシステムの詳細テーブルが更新されるまでの秒数を指定します。 |
| dash_os_io_timeout      | 1800 seconds   | オペレーティングシステム分析ダッシュボードで I/O 折れ線グラフが更新されるまでの秒数を指定します。        |
| dash_os_memory_span     | 7 days         | オペレーティング システム分析ダッシュボードのメモリ チャートにプロットするデータの日数を指定します。        |
| dash_os_memory_timeout  | 1800 seconds   | オペレーティング システム分析ダッシュボードでメモリの折れ線グラフが更新されるまでの秒数を指定します。        |
| dash_os_packet_span     | 7 days         | オペレーティング システム分析ダッシュボードの パケット チャートにプロットするデータの日数を指定します。      |
| dash_os_packet_timeout  | 1800 seconds   | オペレーティングシステム分析ダッシュボードでネットワークパケット折れ線グラフが更新されるまでの秒数を指定します。   |
| dash_os_process_span    | 7 days         | オペレーティング システム分析ダッシュボードのプロセス チャートにプロットするデータの日数を指定します。       |
| dash_os_process_timeout | 1800 seconds   | オペレーティングシステム分析ダッシュボードでプロセス折れ線グラフが更新されるまでの秒数を指定します。         |
| dash_os_storage_timeout | 1800 seconds   | オペレーティングシステム分析ダッシュボードでストレージの円グラフが更新されるまでの秒数を指定します。         |
| dash_os_traffic_span    | 7 days         | オペレーティング システム分析ダッシュボードのトラフィック チャートにプロットするデータの日数を指定します。     |

次のページに続く

表 2 – 前のページからの続き

| Parameter name                  | Value and unit | Description                                                      |
|---------------------------------|----------------|------------------------------------------------------------------|
| dash_os_traffic_timeout         | 1800 seconds   | オペレーティングシステム分析ダッシュボードでトラフィック ライン チャートが更新されるまでの秒数を指定します。          |
| dash_os_util_timeout            | 1800 seconds   | オペレーティングシステム分析ダッシュボードで使用率折れ線グラフが更新されるまでの秒数を指定します。                |
| dash_probe_log_timeout          | 300 seconds    | プローブログテーブルが更新されるまでの秒数を指定します。                                     |
| dash_replication_archiveset_lag | 7 days         | ストリーミングレプリケーション分析ダッシュボードの WAL アーカイブステータスチャートにプロットするデータの日数を指定します。 |
| dash_replication_archiveset_lag | 1800 seconds   | ストリーミングレプリケーションダッシュボードで WAL アーカイブステータス線グラフが更新されるまでの秒数を指定します。     |
| dash_replication_pagelag        | 7 days         | ストリーミングレプリケーションダッシュボードの WAL ラグページチャートにプロットするデータの日数を指定します。        |
| dash_replication_pagelag        | 1800 seconds   | ストリーミングレプリケーションダッシュボードで WAL ラグページ折れ線グラフが更新されるまでの秒数を指定します。        |
| dash_replication_segment_lag    | 7 days         | ストリーミングレプリケーションダッシュボードの WAL ラグセグメントチャートにプロットするデータの日数を指定します。      |
| dash_replication_segment_lag    | 1800 seconds   | ストリーミングレプリケーションダッシュボードで WAL ラグセグメント折れ線グラフが更新されるまでの秒数を指定します。      |
| dash_replication_timelag        | 7 days         | ストリーミングレプリケーションダッシュボードのレプリケーションラグタイムチャートにプロットするデータの日数を指定します。     |
| dash_replication_timelag        | 1800 seconds   | ストリーミングレプリケーションダッシュボードでレプリケーションラグタイム線グラフが更新されるまでの秒数を指定します。       |
| dash_server_buffers_writes      | 68 hours       | サーバー分析ダッシュボードのバックグラウンドライター統計グラフにプロットするデータの日数を指定します。              |
| dash_server_buffers_writes      | 300 seconds    | サーバー分析ダッシュボードでバックグラウンドライター統計の折れ線グラフが更新されるまでの秒数を指定します。            |
| dash_server_connoverview        | 300 seconds    | Server Analysis ダッシュボードで接続の概要の円グラフが更新されるまでの秒数を指定します。             |
| dash_server_database_time       | 300 seconds    | Server Analysis ダッシュボードでデータベーステーブルが更新されるまでの秒数を指定します。             |

次のページに続く

表 2 – 前のページからの続き

| Parameter name                   | Value and unit | Description                                                                         |
|----------------------------------|----------------|-------------------------------------------------------------------------------------|
| dash_server_dbsize_span          | 7 days         | Server Analysis ダッシュボードのデータベースサイズ分析にプロットするデータの日数を指定します。                             |
| dash_server_dbsize_timeout       | 1800 seconds   | Server Analysis ダッシュボードでデータベースサイズの折れ線グラフが更新されるまでの秒数を指定します。                          |
| dash_server_disk_timeout         | 1800 seconds   | Server Analysis ダッシュボードでディスク折れ線グラフが更新されるまでの秒数を指定します。                                |
| dash_server_global_span          | 7 days         | Server Analysis ダッシュボードのディスク折れ線グラフにプロットするデータの日数を指定します。                              |
| dash_server_sharedbuff_span      | 7 days         | Server Analysis ダッシュボードの共有バッファグラフにプロットするデータの日数を指定します。                               |
| dash_server_sharedbuff_timeout   | 1800 seconds   | Server Analysis ダッシュボードで共有バッファ折れ線グラフが更新されるまでの秒数を指定します。                              |
| dash_server_tabspacesize_span    | 7 days         | Server Analysis ダッシュボードのテーブルスペースサイズチャートにプロットするデータの日数を指定します。                         |
| dash_server_tabspacesize_timeout | 1800 seconds   | Server Analysis ダッシュボードでテーブルスペースサイズの折れ線グラフが更新されるまでの秒数を指定します。                        |
| dash_server_useract_span         | 7 days         | Server Analysis ダッシュボードのユーザー アクティビティ グラフにプロットするデータの日数を指定します。                        |
| dash_server_useract_timeout      | 1800 seconds   | Server Analysis ダッシュボードでユーザーアクティビティ折れ線グラフが更新されるまでの秒数を指定します。                         |
| dash_sessact_lockact_timeout     | 300 seconds    | セッションアクティビティ分析ダッシュボードでセッションロックアクティビティテーブルが更新されるまでの秒数を指定します。                         |
| dash_sessact_workload_timeout    | 300 seconds    | セッションアクティビティ分析ダッシュボードでセッションワークロードテーブルが更新されるまでの秒数を指定します。                             |
| dash_sess_waits_timewait_timeout | 300 seconds    | Session Waits By Time Waited 円グラフが Session Waits Analysis ダッシュボードで更新されるまでの秒数を指定します。 |
| dash_sess_waits_waitdtl_timeout  | 300 seconds    | セッション待機分析ダッシュボードでセッション待機の詳細テーブルが更新されるまでの秒数を指定します。                                   |

次のページに続く

表 2 – 前のページからの続き

| Parameter name                  | Value and unit | Description                                                                                                                                                                                                                                                                                                                |
|---------------------------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| dash_storage_dbdttl_time        | 300 seconds    | Storage Analysis ダッシュボードでデータベースの詳細テーブルが更新されるまでの秒数を指定します。                                                                                                                                                                                                                                                                   |
| dash_storage_dbovervw_time      | 300 seconds    | Storage Analysis ダッシュボードでデータベースの概要円グラフが更新されるまでの秒数を指定します。                                                                                                                                                                                                                                                                   |
| dash_storage_hostdttl_time      | 300 seconds    | ホストの詳細テーブルが更新されるまでの秒数を指定します。                                                                                                                                                                                                                                                                                               |
| dash_storage_hostovervw_time    | 300 seconds    | Storage Analysis ダッシュボードでホストの概要円グラフが更新されるまでの秒数を指定します。                                                                                                                                                                                                                                                                      |
| dash_storage_tblspcdttl_time    | 300 seconds    | Storage Analysis ダッシュボードで Tablespace Details テーブルが更新されるまでの秒数を指定します。                                                                                                                                                                                                                                                        |
| dash_storage_tblspcovervw_time  | 300 seconds    | Storage Analysis ダッシュボードでテーブルスペースの概要円グラフが更新されるまでの秒数を指定します。                                                                                                                                                                                                                                                                 |
| dash_sys_waits_nowaits_time     | 300 seconds    | System Waits By Number Of Waits 円グラフが System Waits Analysis ダッシュボードで更新されるまでの秒数を指定します。                                                                                                                                                                                                                                      |
| dash_sys_waits_timewait_time    | 300 seconds    | System Waits By Time Waited 円グラフが System Waits Analysis ダッシュボードで更新されるまでの秒数を指定します。                                                                                                                                                                                                                                          |
| dash_sys_waits_waitdttl_time    | 300 seconds    | システム待機分析ダッシュボードでシステム待機の詳細テーブルが更新されるまでの秒数を指定します。                                                                                                                                                                                                                                                                            |
| deleted_charts_retention        | 7 days         | カスタムチャート（ユーザー定義のダッシュボードに表示される）が保存される日数を指定します。                                                                                                                                                                                                                                                                              |
| deleted_probes_retention        | 7 days         | カスタムプローブ（ユーザー定義のダッシュボードに表示）が保存される日数を指定します。                                                                                                                                                                                                                                                                                 |
| download_chart_format           | jpeg           | ダウンロードしたチャートを保存する形式を指定します。 jpeg または png になります。                                                                                                                                                                                                                                                                             |
| flapping_detection_state_change | 3              | 指定した間隔内に検出された状態の変化の数を指定して、特定のアラートをフラッピングとして定義します。フラッピングは、 $[N + 1 * (\min(\text{probe\_interval}) * 2)]$ 分にわたって $N$ を超える状態変化が発生し、細かい状態が None でない場合に開始されます。 $N$ のデフォルト値は 2 または 3 であり、 $\min(\text{probe\_interval})$ はアラートで使用されるすべてのプローブの最小間隔です。フラッピングは、ZERO の状態変化が $[2 * N * \min(\text{probe\_interval})]$ 分以上発生すると終了します。 |

次のページに続く

表 2 – 前のページからの続き

| Parameter name                   | Value and unit | Description                                                          |
|----------------------------------|----------------|----------------------------------------------------------------------|
| job_retention_time               | 30 days        | 非定期的なスケジュールされたタスクとそれに関連するジョブを保持する日数を指定します。                           |
| long_running_transaction_timeout | 5 minutes      | クエリが長時間実行されるまでに実行される分数を指定します。                                        |
| na-gios_cmd_file_name            | <file_name>    | パッシブサービスチェックの結果を送信する na-gios コマンドファイルを指定します。                         |
| nagios_enabled                   | t              | アラート通知を nagios に送信するかどうかを指定します。                                      |
| na-gios_medium_alert_as_critical | f              | nagios で中レベルの PEM アラートがクリティカルと見なされるかどうかを指定します。                       |
| na-gios_spool_retention_time     | 7 days         | nagios メッセージが破棄されるまでにスプールテーブルに保持する日数を指定します。                          |
| probe_log_retention_time         | 30 days        | プローブログレコードを保持する日数を指定します。                                             |
| re-minder_notification_interval  | 24 hours       | アラートがクリアされなかった場合にリマインダー メールが送信されるまでの時間数を指定します。                       |
| server_log_retention_time        | 30 days        | サーバーログが PEM サーバーに保持される日数を指定します。                                      |
| show_data_tab_on_graph           | false          | `true` の場合、各グラフに**データ**タブが追加されます。[データ] タブを選択して、グラフにプロットされたデータを確認します。 |
| smtp_authentication              | false          | SMTP 経由の認証を有効/無効にするかどうかを指定します。                                       |
| smtp_enabled                     | true           | メールの送信を有効/無効にするかどうかを指定します。                                           |
| smtp_encryption                  | false          | 暗号化された接続を使用して SMTP 電子メールを送信するかどうかを指定します。                             |
| smtp_password                    |                | SMTP サーバーへの接続に使用するパスワードを指定します。                                       |
| smtp_port                        | 25             | 電子メールの送信に使用する SMTP サーバーのポートを指定します。                                   |
| smtp_server                      | 127.0.0.1      | 電子メールの送信に使用する SMTP サーバーのホストアドレスを指定します。                               |
| smtp_spool_retention_time        | 7 days         | 送信された電子メールメッセージが破棄されるまでにスプールテーブルに保持する日数を指定します。                       |
| smtp_username                    |                | SMTP サーバーへの接続に使用するユーザー名を指定します。                                       |
| snmp_community                   | public         | トラップ送信時に使用する SNMP コミュニティを指定します。SNMPv1 および SNMPv2 でのみ使用されます。          |

次のページに続く

表 2 – 前のページからの続き

| Parameter name               | Value and unit           | Description                                                                                                                          |
|------------------------------|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| snmp_enabled                 | true                     | SNMP トラップの送信を有効/無効にするかどうかを指定します。                                                                                                     |
| snmp_port                    | 162                      | SNMP トラップの送信に使用する SNMP サーバーポートを指定します。                                                                                                |
| snmp_server                  | 127.0.0.1                | SNMP トラップの送信に使用する SNMP サーバーのホストアドレスを指定します。                                                                                           |
| snmp_spool_retention_time    | 7days                    | 送信されたトラップが破棄されるまでにスプールテーブルに保持する日数を指定します。                                                                                             |
| snmp_security_name           |                          | SNMP トラップを送信するためのユーザー名またはセキュリティ名を指定します。SNMPv3 でのみ使用されます。                                                                             |
| snmp_security_engine_id      |                          | SNMP サーバー上の SNMP エージェントのエンジン ID を指定します。SNMPv3 でのみ使用されます。                                                                             |
| snmp_security_level          | NOAUTH_NOPRIV            | セキュリティレベルを指定します。その可能な値は次のとおりです。AUTH_NOPRIV - 認証、プライバシーなしまたは AUTH_PRIV - 認証、プライバシーまたは NOAUTH_NOPRIV - 認証なし、プライバシーなし。SNMPv3 でのみ使用されます。 |
| snmp_context_name            |                          | コンテキスト名、SNMP トラップを送信するときの MIB オブジェクトの識別子を指定します。SNMPv3 でのみ使用されます。                                                                     |
| snmp_context_engine_id       |                          | SNMP トラップを送信するときの MIB オブジェクトの識別子であるコンテキストエンジン ID を指定します。指定しない場合、snmp_security_engine_id が使用されます。SNMPv3 でのみ使用されます。                    |
| snmp_authentication_protocol | NONE                     | SNMP トラップの認証タイプを指定します。可能な値は、NONE、HMACMD5、および HMACSHA です。SNMPv3 でのみ使用されます。                                                            |
| snmp_privacy_protocol        | NONE                     | SNMP トラップのプライバシー プロトコルを指定します。その可能な値は、NONE、DES、AES128、IDEA、AES192、または AES256 です。SNMPv3 でのみ使用されます。                                     |
| snmp_authentication_password |                          | snmp_security_name に記載されているセキュリティ名に関連付けられた認証パスワードを指定します。SNMPv3 でのみ使用されます。                                                            |
| snmp_privacy_password        |                          | snmp_security_name に記載されているセキュリティ名に関連付けられたプライバシーパスワードを指定します。SNMPv3 でのみ使用されます。                                                        |
| webclient_help_pg            | EDB hosted documentation | オンラインの PostgreSQL コードキュメントの場所を指定します。                                                                                                 |

## 第5章 Managing Postgres servers

### 5.1 監視対象データベースサーバーの起動と停止

PEM を使用すると、PEM クライアントで管理対象サーバーインスタンスを起動およびシャットダウンできます。PEM がサービスを管理できるようにサーバーを構成するには、Server Registration ダイアログボックスに入力します。ダイアログボックスを使用して、データベースサーバーと同じホストにインストールされたローカルエージェントである PEM エージェントにデータベースサーバーを登録します。次のこともできます。

- [プロパティ] ダイアログボックスで [ **PEM サーバーに保存する** ] オプションを指定します。
- [ 詳細 ] タブの **サービス ID** フィールドにサービススクリプトの名前を指定します。
- EDB Postgres Advanced Server の場合、サービス名は `edb-as-<x>` または `ppas-<x>` です。• PostgreSQL の場合、サービス名は `postgresql-<x>` です。

<x> は、サーバーのバージョン番号を示します。

サーバーに接続した後、ツリーでサーバー名を選択し、**ツール>QueueServerStartup** または **ツール>QueueServerShutdown** を選択して、サーバーを起動または停止できます。

### 5.2 pg\_hba.conf ファイルの変更

`pg_hba.conf` ファイルのエントリは、ネットワークの認証と承認を制御します。PEM サーバーホスト上の `pg_hba.conf` ファイルは、PEM サーバーと PEM-HTTPD、PEM エージェント、および監視対象サーバー間の接続を許可する必要があります。

PEM サーバーのインストールプロセスでは、監視する PEM のホストの IP アドレスと接続情報の入力を求められます。この情報は、PEM バッキングデータベースの `pg_hba.conf` ファイルの先頭に追加されます。

PEM サーバーと他のコンポーネント間の接続を許可するには、`pg_hba.conf` ファイルを手動で変更する必要があります場合があります。たとえば、PEM-HTTPD インストールが PEM サーバーと同じホストにない場合は、PEM サーバーホスト上の `pg_hba.conf` ファイルを変更して、PEM-HTTPD がサーバーに接続できるようにする必要があります。

デフォルトでは、`pg_hba.conf` ファイルは Postgres インストールの下に `data` ディレクトリにあります。たとえば、EDB Postgres Advanced Server 10 ホストでは、`pg_hba.conf` のデフォルトの場所は次のとおりです。

```
/var/lib/edb/as10/data/pg_hba.conf
```

選択したエディターで `pg_hba.conf` ファイルを変更できます。ファイルを変更した後、サーバーを再起動して変更を有効にします。

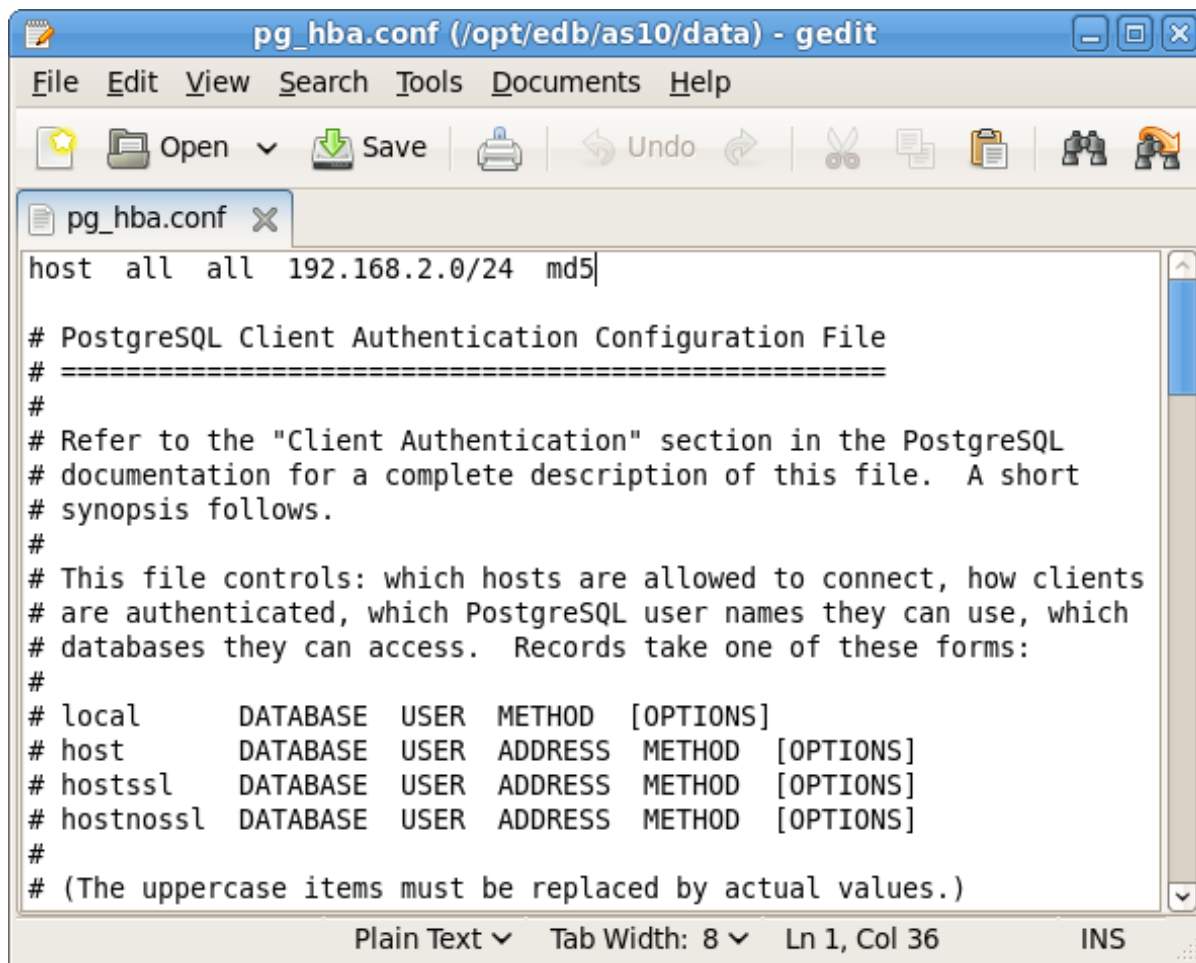


図 1: PEM entries in the pg\_hba.conf file

次の例は、postgres という名前のユーザーから pg\_hba.conf が存在するホスト上の postgres データベースへの md5 パスワード認証接続を許可する pg\_hba.conf エントリーを示しています。接続は 192.168.10.102 の IP アドレスから来ています。

| #                         | TYPE | DATABASE | USER     | CIDR-ADDRESS      | METHOD |
|---------------------------|------|----------|----------|-------------------|--------|
| # IPv4 local connections: |      |          |          |                   |        |
|                           | host | postgres | postgres | 192.168.10.102/32 | md5    |

ネットワークホストのアドレスまたはネットワークアドレス範囲を指定できます。たとえば、アドレス 192.168.10.23、192.168.10.76、および 192.168.10.184 のサーバーからの接続を許可する場合、192.168.10.0/24 の CIDR-ADDRESS を入力して、そのネットワーク内のすべてのホストからの接続を許可します。

| #                         | TYPE | DATABASE | USER | CIDR-ADDRESS    | METHOD |
|---------------------------|------|----------|------|-----------------|--------|
| # IPv4 local connections: |      |          |      |                 |        |
|                           | host | postgres | all  | 192.168.10.0/24 | md5    |

pg\_hba.conf ファイルエントリーのフォーマットの詳細については、

[PostgreSQL core documentation](#) を参照してください。

PEM を使用して Postgres サーバーに接続する前に、両方のサーバーの pg\_hba.conf ファイルで接続が許可されていることを確認する必要があります。データベースサーバへの接続時にエラーを受け取った場合は、pg\_hba.conf ファイルを変更して、接続を許可するエントリーを追加します。

## 5.3 データベースとオブジェクトの作成と保守

Postgres サーバーの各インスタンスは、1 つ以上のデータベースを管理します。各ユーザーは、データベースに含まれる情報にアクセスする前に、データベースに接続するための認証情報を提供する必要があります。PEM クライアントを使用すると、データベースと、テーブル、インデックス、ストアドプロシージャなどのデータベースを構成するオブジェクトを作成および管理できます。

PEM でデータベースを作成するには、管理対象サーバーの **Databases** ノードを右クリックし、**Create>Database** を選択します。データベースを定義したら、新しいデータベースにオブジェクトを作成できます。

たとえば、テーブルを作成するには、**Tables** ノードを右クリックし、**Create>Table** を選択します。[ New Table ] ダイアログボックスでテーブルの属性を指定します。

PEM は、次のような他のデータベースオブジェクトを作成および管理するための同様のダイアログボックスを提供します。

- テーブル
- インデックス
- ストアドプロシージャ

- 機能
- トリガー
- ビュー
- 制約

各オブジェクトタイプがツリーにリストされます。オブジェクトタイプに対応するノードを右クリックして、[作成] メニューにアクセスし、オブジェクトを作成します。選択したオブジェクトの管理タスクを実行するには、名前付けノードのコンテキストメニューから**プロパティ** を選択します。

## 5.4 テンプレート Linux サービススクリプト

データベースサーバーのサービススクリプトを使用すると、PEM サーバーでデータベースサーバーを起動、停止、または再起動できます。これは、構成管理、証明書管理、およびその他の管理タスクを実行するときに必要になる場合があります。サービススクリプトはプラットフォーム固有です。

PEM サーバーが存在する Postgres サーバーには、サービススクリプトが含まれている必要があります。EDB によって生成された Windows の Postgres インストーラーは、サービススクリプトを作成します。ネイティブパッケージなどの別のソースから Postgres サーバーを使用している場合は、サービススクリプトを提供する必要があります。

---

**注釈:** CentOS または RHEL 7.x の場合 | Rocky Linux または RHEL 8.x、サービススクリプトは `/usr/lib/systemd/system` にあります。

---

Postgres サービスのカスタマイズについては、[PostgreSQL documentation](#) を参照してください。

## 第6章 Managing a PEM agent

### 6.1 ジョブ通知の管理

PEM コンソールでは、スケジュールされたタスクまたはカスタムエージェントジョブの下にリストされたシステム生成のジョブが成功または失敗したときに SMTP トラップを送信するための設定を構成できます。これらのメール通知設定を次の3つのレベルで構成して、指定したユーザーグループにメール通知を送信できます。これらのレベルは優先順に示されています。

- ジョブレベル
- エージェントレベル
- PEM サーバーレベル（デフォルト）

#### 6.1.1 ジョブレベルでのジョブ通知の構成

次のいずれかの方法で、カスタムエージェントジョブのジョブレベルでのみメール通知設定を構成できます。

- 新しいエージェントジョブの場合、ジョブの作成中にエージェントジョブの作成ウィザードの [通知] タブでメール通知設定を構成できます。
- 既存のカスタムジョブの場合、ジョブのプロパティを編集し、通知設定を構成できます。

[通知] タブを使用して、ジョブレベルでメール通知設定を構成します。

- **通知を送信するフィールド**を使用して、メール通知を送信するタイミングを指定します。
- **メールグループフィールド**を使用して、メール通知を送信するメールグループを指定します。

#### 6.1.2 エージェントレベルでのジョブ通知の構成

ツリービューでエージェントを選択して右クリックし、[プロパティ]を選択します。[プロパティ]ダイアログボックスで、[ジョブ通知]タブを選択します。

[ジョブ通知] タブを使用して、エージェントレベルでメール通知設定を構成します。

- **Overridedefaultconfiguration?** スイッチを使用して、エージェントレベルのジョブ通知設定でデフォルトのジョブ通知設定をオーバーライドするかどうかを指定します。**はい**を選択して、このダイアログボックスの残りの設定を有効にして、ジョブ通知をいつ誰に送信するかを定義します。
- **ジョブ完了時にメールを送信しますか?** スイッチを使用して、ジョブが正常に完了したときにジョブ通知を送信するかどうかを指定します。

- ジョブが失敗したときにメールを送信しますか? スイッチを使用して、ジョブが失敗したときにジョブ通知を送信するかどうかを指定します。
- メールグループフィールドを使用して、ジョブ通知を送信するメールグループを指定します。

### 6.1.3 サーバーレベルでのジョブ通知の構成

サーバー構成ダイアログボックスを使用して、PEM サーバーレベルでの電子メール通知構成に関する情報を提供できます。[ Server Configuration ] ダイアログボックスを開くには、**Management>ServerConfiguration** を選択します。

4つのサーバー構成パラメーターは、PEM サーバーレベルでのジョブ通知設定に関する情報を指定します。

- **job\_failure\_notification** スイッチを使用して、ジョブが失敗するたびにメール通知を送信するかどうかを指定します。
- **job\_notification\_email\_group** パラメーターを使用して、メール通知の送信先のメールグループを指定します。
- **job\_retention\_time** パラメーターを使用して、非定期的なスケジュールされたタスクをシステムに保持する日数を指定します。
- **job\_status\_change\_notification** スイッチを使用して、各ジョブのステータスが変更された後、つまり失敗、成功、または中断後にメール通知を送信するかどうかを指定します。

## 6.2 PEM スケジュールされたジョブの管理

定義した一連のステップを指定された順序で実行する PEM スケジュールされたジョブを作成できます。これらの手順には、エージェントにバインドされたサーバーで実行できる SQL コードまたはバッチ/シェルスクリプトを含めることができます。これらのジョブは、ビジネス要件に合わせてスケジュールできます。たとえば、特定のデータベースサーバーのバックアップを作成するジョブを作成し、毎月の特定の日時に実行するようにスケジュールできます。

PEM スケジュールされたジョブを作成または管理するには、PEM ツリーを使用して、ジョブを作成する PEM エージェントを参照します。ツリーには**ジョブ**ノードが表示され、その下に現在定義されているジョブがリストされます。ジョブを追加するには、**Jobs** ノードを右クリックし、コンテキストメニューから **CreateJob** を選択します。

[ Create - Agent Job ] ダイアログボックスのタブを使用して、PEM スケジュールされたジョブを構成するステップとスケジュールを定義します。

[ **General** ] タブを使用して、ジョブに関する一般的な情報を提供します。

- **名前** フィールドにジョブの名前を入力します。
- **Enabled** スイッチを **Yes** に設定して、ジョブを有効にします。ジョブを無効にするには **いいえ** に設定します。
- **コメント** フィールドを使用して、ジョブに関するメモを保存します。

[ **ステップ** ] タブを使用して、ジョブが実行するステップを定義および管理します。追加 (+) を選択して、ステップを追加します。次に、ヘッダーの左側にある作成アイコンを選択して、[ **ステップ定義** ] ダイアログボックスを開きます。

ステップ定義ダイアログボックスを使用して、ステップを定義します。

- **名前** フィールドにステップの名前を入力します。ステップは名前の英数字順に実行されます。
- **Enabled** スイッチを使用して、ジョブの実行時にステップを含める (**True**) か、ステップを無効にします (**False**)。
- **Kind** スイッチを使用して、ジョブステップが **SQL** コード (**SQL**) を呼び出すか、バッチスクリプト (**Batch**) を呼び出すかを示します。
- **SQL** を選択した場合、[ **コード** ] タブを使用してステップの **SQL** コードを指定します。- **Batch** を選択した場合、**Code** タブを使用して、ステップ中に実行するバッチスクリプトを指定します。
- **エラー** リストを使用して、ステップの実行中にエラーが発生した場合の pgAgent の動作を指定します。次から選択します。
- **失敗** - このステップの処理中にエラーが発生した場合は、ジョブを停止します。- **成功** - ステップを正常に完了したものとしてマークし、続行します。- **無視** エラーを無視して続行します。
- **Kind** スイッチの入力として **SQL** を選択した場合：
- **Server** フィールドを使用して、PEM スケジュールされたジョブを作成するエージェントにバインドされたサーバーを指定します。- **データベース** フィールドを使用して、選択したサーバーに関連付けられているデータベースを指定します。
- **Comment** フィールドを使用して、ステップに関するコメントを入力します。
- [ **ステップ定義** ] ダイアログボックスの [ **コード** ] タブの状況依存フィールドを使用して、ステップ中に実行する **SQL** コードまたはバッチスクリプトを提供します。
- ステップが **SQL** コードを呼び出す場合、**SQLquery** フィールドに 1 つ以上の **SQL** ステートメントを入力します。
- ステップがバッチスクリプトを呼び出す場合は、スクリプトを **Code** フィールドに入力します。Windows サーバーで実行している場合は、標準のバッチファイル構文を使用します。Linux サーバーでは、最初の行で適切なインタープリター (`#!/bin/sh` など) を指定すれば、任意のシェルスクリプトを使用できます。定義されたインラインコードとともに、ファイルシステム上のバッチスクリプト、シェルスクリプト、または **SQL** ファイルのパスを指定することもできます。
- Linux システムでスクリプトを呼び出すには、`agent.cfg` ファイルの `batch_script_user` パラメーターのエントリを変更し、スクリプトを実行するユーザーを指定する必要があります。このパラメーターには、非 root ユーザーまたは root を指定できます。ユーザーを指定しない場合、または指定したユーザーが存在しない場合、スクリプトは実行されません。ファイルを変更した後、エージェントを再起動します。

Windows システムでスクリプトを呼び出すには、`AllowBatchJobSteps` のレジストリエントリを `true` に設定し、PEM エージェントを再起動します。PEM レジストリエントリは `HKEY_LOCAL_MACHINE\Software\Wow6432Node\EnterpriseDB\PEM\agent` にあります。

ステップで必要な情報をすべて入力したら、[ **保存** ] を選択します。

追加 (+) を選択して各ステップを追加するか、[ **スケジュール** ] タブを選択してジョブ スケジュールを定義します。

[ **スケジュール** ] タブで、**追加 (+)** を選択して、ジョブのスケジュールを追加します。次に、ヘッダーの左側にある作成アイコンを選択して、[ **スケジュール定義** ] ダイアログボックスを開きます。

[ **スケジュール定義** ] タブを使用して、ジョブを実行する日時を指定します。

- **名前** フィールドにスケジュールの名前を入力します。
- **Enabled** スイッチを使用して、pgAgent がスケジュールを使用する (**Yes**) か、スケジュールを無効にする (**No**) かを示します。
- **開始** フィールドのカレンダーセレクターを使用して、スケジュールの開始日時を指定します。
- **終了** フィールドのカレンダーセレクターを使用して、スケジュールの終了日時を指定します。
- **コメント** フィールドを使用して、スケジュールに関するコメントを入力します。

繰り返しタブを選択して、スケジュールを cron スタイルの形式で実行する日を定義します。ジョブは、[ **繰り返し** ] タブで選択された各日付または時刻要素に実行されます。

各フィールドで、値を選択して、フィールドで選択した値のリストに追加します。フィールドから値をクリアするには、フィールドの右側にある X を選択します。

- **Days** ボックスのフィールドを使用して、ジョブを実行する日を指定します。 - **WeekDays** フィールドを使用して、ジョブを実行する日を選択します。 - **MonthDays** フィールドを使用して、ジョブを実行する日数を選択します。日付に関係なく、月の最終日にジョブを実行するには、**最終日** を選択します。 - **Months** フィールドを使用して、ジョブを実行する月を選択します。
- **Times** ボックスのフィールドを使用して、ジョブを実行する時刻を時間と分で指定します。

[ **Exceptions** ] タブを選択して、スケジュールを実行しない日を指定します。たとえば、休日にジョブを実行したくない場合があります。

**追加 (+)** を選択して、例外テーブルに行を追加します。次に：

- **日付列**で、カレンダーセレクターを開き、ジョブを実行しない日付を選択します。\*\*\*\* <Any> \*\*\*\* を指定して、選択した時刻にジョブを実行しないことを示します。
- **時間列**で、時間セレクターを開き、ジョブを実行しない時間を指定します。\*\*\*\* <Any> \*\*\*\* を指定して、選択した日にジョブを実行しないことを示します。

[ **Notifications** ] タブを選択して、ジョブレベルでメール通知設定を構成します。

- **通知を送信する** フィールドを使用して、メール通知を送信するタイミングを指定します。
- **メールグループ** フィールドを使用して、メール通知を送信するメールグループを指定します。

スケジュールの定義が完了したら、[ **SQL** ] タブを使用して、ジョブを作成または変更するコードを確認できます。

**Save** を選択して、ジョブ定義を保存します。

ジョブを保存すると、ジョブは定義されたサーバーの PEM ツリーの **Jobs** ノードの下にリストされます。PEM コンソールの **プロパティ** タブには、選択したジョブの高レベルの概要が表示され、**統計** タブにはジョ

ブの各実行の詳細が表示されます。既存のジョブを変更したり、ジョブに関する詳細情報を確認したりするには、ジョブ名を右クリックし、コンテキストメニューから**プロパティ**を選択します。

## 6.3 エージェント権限

デフォルトでは、PEM エージェントはオペレーティングシステムホストの root 権限とデータベースサーバーのスーパーユーザー権限でインストールされます。これらの権限により、PEM エージェントは、システムの使用状況について監視対象ホストおよびデータベースサーバーで無制限のプロブを呼び出し、情報を取得して PEM サーバーに返すことができます。

PEM エージェントの権限が低下すると、PEM 機能が低下します。完全な機能を得るには、root として PEM エージェントを実行します。PEM エージェントがデータベースサーバーのサービスアカウントで実行されている場合、PEM プロブはレポートの生成に使用される統計情報に完全にアクセスできず、機能はそのアカウントの機能に制限されます。PEM エージェントが別のより低い特権のアカウントで実行されている場合、機能はさらに制限されます。

PEM エージェントのオペレーティングシステム権限を制限すると、一部の PEM プロブが情報を返さず、次の機能が影響を受ける可能性があります。

**注釈:** このリストは包括的なものではありませんが、制限されている機能の種類の概要を提供します。

| Probe or action                                 | Operating system | PEM functionality affected                                                                                                 |
|-------------------------------------------------|------------------|----------------------------------------------------------------------------------------------------------------------------|
| Data And Logfile Analysis                       | Linux/ Windows   | The Postgres Expert can't access complete information.                                                                     |
| Session Information                             | Linux            | The per-process statistics are incomplete.                                                                                 |
| PG HBA                                          | Linux/ Windows   | The Postgres Expert can't access complete information.                                                                     |
| Service restart functionality                   | Linux/ Windows   | The Audit Log Manager, Server Log Manager Log Analysis Expert, and PEM might not be able to apply requested modifications. |
| Package Deployment                              | Linux/ Windows   | PEM can't run downloaded installation modules.                                                                             |
| Batch Task                                      | Windows          | PEM can't run scheduled batch jobs in Windows.                                                                             |
| Collect data from server (root access required) | Linux/ Windows   | Columns such as swap usage, CPU usage, IO read, IO write appear as 0 in the session activity dashboard.                    |

PEM エージェントのデータベース特権を制限すると、次の PEM 機能が影響を受ける可能性があります。

| Probe                 | Operating system | PEM functionality affected                     |
|-----------------------|------------------|------------------------------------------------|
| Audit Log Collection  | Linux/Windows    | PEM receives empty data from the PEM database. |
| Server Log Collection | Linux/Windows    | PEM can't collect server log information.      |

次のページに続く

表 2 – 前のページからの続き

| Probe                      | Operating system | PEM functionality affected                                                         |
|----------------------------|------------------|------------------------------------------------------------------------------------|
| Database Statistics        | Linux/Windows    | The Database/Server Analysis dashboards contain incomplete information.            |
| Session Waits/System Waits | Linux/Windows    | The Session/System Waits dashboards contain incomplete information.                |
| Locks Information          | Linux/Windows    | The Database/Server Analysis dashboards contain incomplete information.            |
| Streaming Replication      | Linux/Windows    | The Streaming Replication dashboard doesn't display information.                   |
| Slony Replication          | Linux/Windows    | Slony-related charts on the Database Analysis dashboard don't display information. |
| Tablespace Size            | Linux/Windows    | The Server Analysis dashboard doesn't display complete information.                |
| xDB Replication            | Linux/Windows    | PEM can't send xDB alerts and traps.                                               |

プローブが十分な特権なしでオペレーティングシステムを照会している場合、プローブは **permission denied** エラーを返す場合があります。プローブが十分な権限なしでデータベースを照会している場合、プローブは **permission denied** エラーを返すか、返されたデータを空の値として PEM チャートまたはグラフに表示する場合があります。

プローブが失敗すると、プローブの名前、プローブが失敗した理由、および問題の解決に役立つヒントを含むエントリがログファイルに書き込まれます。

サーバーで発生したプローブ関連のエラーをプローブログダッシュボードで表示したり、PEM ワーカーログファイルのエラーメッセージを確認したりできます。Linux では、ログファイルのデフォルトの場所は次のとおりです。

```
/var/log/pem/worker.log
```

Windows、イベントビューアでログ情報を利用できます。

## 6.4 エージェント構成

多数の構成可能なパラメーターとレジストリエントリが PEM エージェントの動作を制御します。一部の PEM 機能を有効にするには、PEM エージェントのパラメーター設定を変更する必要がある場合があります。PEM エージェント構成ファイルの値を変更した後、PEM エージェントを再起動して変更を適用します。

PEM\_MAXCONN パラメーターを除き、PEM 機能を有効にするために変更が必要な場合を除き、EDB サポートの専門家に相談することなく、これらの構成パラメーターまたはレジストリエントリを変更しないことを強くお勧めします。

Linux システムでは、PEM 構成オプションは `/usr/edb/pem/agent/etc` にある `agent.cfg` ファイルに保存されます。 `agent.cfg` ファイルには、次のエントリが含まれています。

| Parameter name                  | Description                                                                                                    | Default value                                                                              |
|---------------------------------|----------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| <code>pem_host</code>           | PEM サーバーの IP アドレスまたはホスト名。                                                                                      | 127.0.0.1.                                                                                 |
| <code>pem_port</code>           | エージェントが PEM サーバーと通信するために接続するデータベースサーバーポート。                                                                     | Port 5432.                                                                                 |
| <code>pem_agent</code>          | PEM エージェントに割り当てられた一意の識別子。                                                                                      | The first agent is '1', the second agent is '2', and so on.                                |
| <code>agent_ssl_key</code>      | PEM エージェントのキーファイルへの完全なパス。                                                                                      | <code>/root/.pem/agent.key</code>                                                          |
| <code>agent_ssl_cert</code>     | PEM エージェントの証明書ファイルへの完全なパス。                                                                                     | <code>/root/.pem/agent.crt</code>                                                          |
| <code>agent_flag_dir</code>     | HA サポートに使用されます。別のサーバーの監視を引き継ぐための要求をチェックするディレクトリパスを指定します。要求は、指定されたフラグディレクトリにファイルの形式で作成されます。                     | Not set by default.                                                                        |
| <code>log_level</code>          | ログレベルは、PEM ログファイルに書き込むイベントの種類を指定します。                                                                           | warning                                                                                    |
| <code>log_location</code>       | PEM ワーカーログファイルの場所を指定します。                                                                                       | 127.0.0.1.                                                                                 |
| <code>agent_log_location</code> | PEM エージェントログファイルの場所を指定します。                                                                                     | <code>/var/log/pem/agent.log</code>                                                        |
| <code>long_wait</code>          | 最初の接続に失敗した場合に PEM サーバーに接続するまでに待機する最大時間（秒）。                                                                     | 30 seconds                                                                                 |
| <code>short_wait</code>         | 実行を待機しているキューに次のプローブがあるかどうかを確認するまでに PEM エージェントが待機する最小時間（秒）。                                                     | 10 seconds                                                                                 |
| <code>alert_threads</code>      | エージェントによって生成されるアラートスレッドの数。                                                                                     | Set to 1 for the agent that resides on the host of the PEM server, 0 for all other agents. |
| <code>enable_smtp</code>        | 複数の PEM エージェント（7.13 以前）を true に設定すると、より多くの重複したメールが送信される場合があります。一方、PEM エージェント（7.14 以降）の場合、送信するメールが少なくなる場合があります。 | true for PEM server host, false for all others.                                            |

次のページに続く

表 3 – 前のページからの続き

| Parameter name          | Description                                                                                                                                                      | Default value                                                                                                 |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| enable_snmp             | 複数の PEM エージェント (7.13 以前) を true に設定すると、より多くの重複トラップを送信する場合があります。一方、PEM エージェント (7.14 以降) の場合、送信される重複トラップが少なくなる場合があります。                                             | true for PEM server host, false for all others.                                                               |
| enable_nagios           | true に設定すると、Nagios アラートが有効になります。                                                                                                                                 | true for PEM server host, false for all others.                                                               |
| enable_webhook          | true に設定すると、Webhook アラートが有効になります。                                                                                                                                | true for PEM server host, false for all others.                                                               |
| max_webhook_retries     | pemAgent が失敗したときに Webhook の呼び出しを再試行する最大回数を設定します。                                                                                                                 | Default 3.                                                                                                    |
| connect_timeout         | エージェントが接続を待機する最大時間 (秒単位)。                                                                                                                                        | Not set by default. Set to 0 to indicate for the agent to wait indefinitely.                                  |
| allow_server_restart    | TRUE に設定すると、エージェントは監視するデータベースサーバーを再起動できます。このパラメーターの値によっては、一部の PEM 機能が有効/無効になる場合があります。                                                                            | False                                                                                                         |
| max_connections         | 接続スロットルによって使用されるプローブ接続の最大数。                                                                                                                                      | 0 (an unlimited number)                                                                                       |
| connection_lifetime     | ConnectionLifetime (または connection_lifetime) を使用して、開いているがアイドル状態の接続を保持する最小秒数を指定します。MaxConnections で指定された値に到達し、待機中の要求を満たすために別のデータベースへの新しい接続が必要な場合、このパラメーターは無視されます。 | By default, set to 0 (a connection is dropped when the connection is idle after the agent's processing loop). |
| allow_batch_probes      | TRUE に設定した場合、ユーザーはカスタムプローブ機能を使用してバッチプローブを作成できません。                                                                                                                | false                                                                                                         |
| heartbeat_connection    | TRUE に設定すると、ハートビートの送信に専用の接続が使用されます。                                                                                                                              | false                                                                                                         |
| batch_script_dir        | スクリプトファイル (アラート用) が保存されているパスを指定します。                                                                                                                              | /tmp                                                                                                          |
| connection_custom_setup | 監視対象サーバーとの新しい接続が確立されたときに呼び出す SQL コードを提供するために使用します。                                                                                                               | Not set by default.                                                                                           |
| ca_file                 | CA 証明書が存在するパスを指定します。                                                                                                                                             | Not set by default.                                                                                           |
| batch_script_user       | バッチ/シェルスクリプトの実行に使用するユーザーの名前を指定します。                                                                                                                               | None                                                                                                          |
| webhook_ssl_key         | Webhook の SSL クライアントキーファイルへの完全なパス。                                                                                                                               |                                                                                                               |

次のページに続く

表 3 – 前のページからの続き

| Parameter name          | Description                                 | Default value |
|-------------------------|---------------------------------------------|---------------|
| webhook_ssl_crt         | Webhook の SSL クライアント証明書ファイルへの完全なパス。         |               |
| webhook_ssl_crl         | Webhook サーバー証明書を検証するための CRL ファイルの完全なパス。     |               |
| webhook_ssl_ca_crt      | Webhook の SSL CA 証明書ファイルへの完全なパス。            |               |
| allow_insecure_webhooks | true に設定すると、Webhook が安全でないフラグで呼び出すことを許可します。 | false         |

64 ビット Windows システムでは、PEM レジストリエントリは次の場所にあります。

HKEY\_LOCAL\_MACHINE\Software\Wow6432Node\EnterpriseDB\PEM\agent

レジストリには、表に示すエントリが含まれています。

| Parameter name | Description                                                                                | Default value                                                                              |
|----------------|--------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| PEM_HOST       | PEM サーバーの IP アドレスまたはホスト名。                                                                  | 127.0.0.1.                                                                                 |
| PEM_PORT       | エージェントが PEM サーバーと通信するために接続するデータベースサーバーポート。                                                 | Port 5432.                                                                                 |
| AgentID        | PEM エージェントに割り当てられた一意の識別子。                                                                  | The first agent is '1', the second agent is '2', and so on.                                |
| AgentKeyPath   | PEM エージェントのキーファイルへの完全なパス。                                                                  | %APP-DATA%\Roaming\pem\agent.key.                                                          |
| AgentCrtPath   | PEM エージェントの証明書ファイルへの完全なパス。                                                                 | %APP-DATA%\Roaming\pem\agent.crt                                                           |
| AgentFlagDir   | HA サポートに使用されます。別のサーバーの監視を引き継ぐための要求をチェックするディレクトリパスを指定します。要求は、指定されたフラグディレクトリにファイルの形式で作成されます。 | Not set by default.                                                                        |
| LogLevel       | PEM ログファイルに書き込むイベントの種類を指定します。                                                              | warning                                                                                    |
| LongWait       | 最初の接続に失敗した場合に PEM サーバーに接続するまでに待機する最大時間（秒）。                                                 | 30 seconds                                                                                 |
| shortWait      | PEM エージェントがキューに次のプローブがあるかどうかを確認する（実行を待機している）までに待機する秒単位の最小時間。                               | 10 seconds                                                                                 |
| AlertThreads   | エージェントが生成するアラートスレッドの数。                                                                     | Set to 1 for the agent that resides on the host of the PEM server, 0 for all other agents. |
| EnableSMTP     | true に設定すると、SMTP 電子メール機能が有効になります。                                                          | true for PEM server host, false for all others.                                            |

次のページに続く

表 4 – 前のページからの続き

| Parameter name         | Description                                                                                                                                                    | Default value                                                                                                 |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| EnableSNMP             | true に設定すると、SNMP トラップ機能が有効になります。                                                                                                                               | true for PEM server host, false for all others.                                                               |
| EnableWebhook          | true に設定すると、Webhook アラートが有効になります。                                                                                                                              | true for PEM server host, false for all others.                                                               |
| MaxWebhookRetries      | pemAgent が失敗したときに Webhook の呼び出しを再試行する最大回数を設定します。                                                                                                               | Default 3.                                                                                                    |
| ConnectTimeout         | エージェントが接続を待機する秒単位の最大時間（10 進の整数文字列）。                                                                                                                            | Not set by default. If set to 0, the agent waits indefinitely.                                                |
| AllowServerRestart     | TRUE に設定すると、エージェントは監視するデータベースサーバーを再起動できます。このパラメーターの値によっては、一部の PEM 機能が有効/無効になる場合があります。                                                                          | true                                                                                                          |
| MaxConnections         | 接続スロットルによって使用されるプローブ接続の最大数。                                                                                                                                    | 0 (an unlimited number)                                                                                       |
| ConnectionLifetime     | ConnectionLifetime（または connection_lifetime）を使用して、開いているがアイドル状態の接続を保持する最小秒数を指定します。MaxConnections で指定された値に到達し、待機中の要求を満たすために別のデータベースへの新しい接続が必要な場合、このパラメーターは無視されます。 | By default, set to 0 (a connection is dropped when the connection is idle after the agent's processing loop). |
| AllowBatchProbes       | TRUE に設定した場合、ユーザーはカスタムプローブ機能を使用してバッチプローブを作成できません。                                                                                                              | false                                                                                                         |
| HeartbeatConnection    | TRUE に設定すると、ハートビートの送信に専用の接続が使用されます。                                                                                                                            | false                                                                                                         |
| BatchScriptDir         | アラート用のスクリプトファイルを保存するパスを指定します。                                                                                                                                  | /tmp                                                                                                          |
| ConnectionCustom-Setup | 監視対象サーバーとの新しい接続が確立されたときに呼び出す SQL コードを提供するために使用します。                                                                                                             | Not set by default.                                                                                           |
| ca_file                | CA 証明書が存在するパスを指定します。                                                                                                                                           | Not set by default.                                                                                           |
| AllowBatchJobSteps     | true に設定すると、管理者ユーザーアカウントを使用してバッチ/シェルスクリプトが実行されます。                                                                                                              | None                                                                                                          |
| WebhookSSLKey          | Webhook の SSL クライアントキーファイルへの完全なパス。                                                                                                                             |                                                                                                               |
| WebhookSSLCrt          | Webhook の SSL クライアント証明書ファイルへの完全なパス。                                                                                                                            |                                                                                                               |
| WebhookSSLCrI          | Webhook サーバー証明書を検証するための CRL ファイルの完全なパス。                                                                                                                        |                                                                                                               |
| WebhookSSLCaCrt        | Webhook の SSL CA 証明書ファイルへの完全なパス。                                                                                                                               |                                                                                                               |

次のページに続く

表 4 – 前のページからの続き

| Parameter name        | Description                                 | Default value |
|-----------------------|---------------------------------------------|---------------|
| AllowInsecureWebhooks | true に設定すると、Webhook が安全でないフラグで呼び出すことを許可します。 | false         |

## 6.5 エージェントのプロパティ

PEM エージェントのプロパティダイアログボックスには、ダイアログボックスが開かれた PEM エージェントに関する情報が表示されます。ダイアログボックスを開くには、PEM クライアントツリーでエージェント名を右クリックし、コンテキストメニューから**プロパティ**を選択します。

PEM エージェントのプロパティダイアログボックスを使用して、PEM エージェントに関する情報を確認または変更します。

- **Description** フィールドには、PEM エージェントの変更可能な説明が表示されます。この説明は、PEM クライアントのツリーに表示されます。
- グループを使用して、PEM クライアントツリーでサーバーとエージェントを整理できます。**Group** リストを使用して、エージェントが表示されるグループを選択します。
- **チーム** フィールドを使用して、エージェントによって監視されているサーバーにアクセスできるグループロールの名前を指定します。このエージェントによって監視されているサーバーは、接続されたチームメンバーに PEM クライアントツリーで表示されます。これは便利な機能です。**Team** フィールドは、真の分離を提供しません。セキュリティ目的で使わないでください。
- **Heartbeatinterval** フィールドには、PEM エージェントから PEM サーバーまでの時間が表示されます。**Minutes** または **Seconds** フィールドの横にあるセクターを使用して、間隔を変更します。

[ **ジョブ通知** ] タブを使用して、エージェントレベルでメール通知設定を構成します。

- **Overridedefaultconfiguration?** スイッチを使用して、エージェントレベルのジョブ通知設定でデフォルトのジョブ通知設定をオーバーライドするかどうかを指定します。**はい**を選択して、このダイアログボックスの残りの設定を有効にして、ジョブ通知をいつ誰に送信するかを定義します。- **ジョブ完了時にメールを送信しますか?** スイッチを使用して、ジョブが正常に完了したときにジョブ通知を送信するかどうかを指定します。- **ジョブが失敗したときにメールを送信しますか?** スイッチを使用して、ジョブが失敗したときにジョブ通知を送信するかどうかを指定します。- **メールグループ** フィールドを使用して、ジョブ通知を送信するメールグループを指定します。

[ **エージェント構成** ] タブには、エージェントの現在の構成と機能がすべて表示されます。

- **パラメーター** 列には、パラメーターのリストが表示されます。- **Value** 列には、対応するパラメーターの現在の値が表示されます。- **Category** 列には、対応するパラメーターのカテゴリが表示されます。**構成** または **機能** のいずれかです。



## 第7章 Managing certificates

PEM サーバーバックアップデータベースのデータディレクトリに保存されているファイルには、PEM サーバーが安全な接続を使用するのに役立つ情報が含まれています。

- `ca_certificate.crt`
- `ca_key.key`
- `server.crt`
- `server.key`
- `root.crl`
- `root.crt`

PEM サーバーと共にインストールされる PEM エージェントは、`ca_certificate.crt` ファイルの有効期限を監視します。証明書の有効期限が近づくと、PEM は次のことを行います。

- 既存の証明書ファイルのバックアップを作成します。
- 新しい証明書ファイルを作成し、新しい CA 証明書ファイルを PEM サーバー上の `root.crt` ファイルに追加します。
- アクティブなエージェントの証明書ファイルを更新するジョブを作成します。
- PEM サーバーを再起動します。

エージェントをアンインストールすると、そのエージェントに関連付けられた証明書が `root.crl` ファイルで維持されている証明書失効リストに追加され、証明書を使用して PEM サーバーに接続できないようにします。

証明書ファイルは手動で置き換えることができます。

### 7.1 SSL 証明書の置き換え

既存の PEM インストールの SSL 証明書を置き換えることができます。サーバーを新しいバージョンに同時にアップグレードする予定がある場合は、SSL 証明書を置き換える前に、すべての PEM インストーラー（最初にサーバーインストーラー、次にエージェントインストーラー）を呼び出します。次に：

1. 実行中のすべての PEM エージェントを、最初にサーバーホストで、次に監視対象ノードで停止します。

Linux ホストで PEM エージェントを停止するには、ターミナルウィンドウを開き、スーパーユーザ権限を引き受けて、次のコマンドを入力します。

Centos 7 または 8 など、systemd を使用する Linux の場合

```
systemctl stop pemagent
```

Windows ホストでは、サービスアプレットを使用して PEM エージェントを停止できます。PEM エージェントサービスの名前は Postgres Enterprise Manager Agent です。[ Services ] ダイアログボックスでサービス名を選択し、**Stoptheservice** を選択します。

2. 既存の SSL キーと証明書のバックアップを取ります。SSL キーと証明書は、PEM インストールの下の data ディレクトリに保存されます。たとえば、Linux システムのデフォルトの場所は次のとおりです。

/var/lib/pgsql/x/data、ここで x は PostgreSQL データベースのバージョンです

次のファイルをコピーし、各ファイルに拡張子を追加して名前を一意にします。

- ca\_certificate.crt - ca\_key.key - root.crt - root.crl - server.key - server.crt

たとえば、エントリに old という単語を追加して ca\_certificate ファイルのバックアップを作成するには、次のコマンドを使用します。

```
# cp ca_certificate.crt ca_certificate_old.crt
```

3. openssl\_rsa\_generate\_key() ファンクションを使用して、ca\_key.key ファイルを生成します。

```
/usr/pgsql-x.x/bin/psql -U postgres -d pem --no-psqlrc -t -A -c "SELECT public.  
→openssl_rsa_generate_key(1024)" > /var/lib/pgsql/x/data/ca_key.key
```

ca\_key.key ファイルを作成したら、ca\_certificate.crt ファイルを生成するとき使用する変数 CA\_KEY に内容を cat します。ca\_key.key ファイルの特権を変更します。

```
CA_KEY=$(cat /var/lib/pgsql/x/data/ca_key.key)
```

```
chmod 600 /var/lib/pgsql/x/data/ca_key.key
```

4. キーを使用して、ca\_certificate.crt ファイルを生成します。簡単にするために、SQL クエリを一意的な名前で一時ファイルに配置します。

```
echo "SELECT openssl_csr_to_cert(openssl_rsa_key_to_csr(${CA_KEY},  
PEM,US, MA, Bedford, Postgres Enterprise Manager,  
support@enterprisedb.com), NULL,  
/var/lib/pgsql/x/data/ca_key.key)" > /tmp/_random.$$
```

次に、変数を使用してクエリを実行し、コンテンツを ca\_certificate.crt ファイルに配置します。

```
/usr/pgsql-x.x/bin/psql -U postgres -d pem --no-psqlrc -t -A -f /tmp/_random.$$ > /var/  
→lib/pgsql/x/data/ca_certificate.crt
```

ca\_certificate.crt ファイルのアクセス許可を変更し、SQL コマンドを含む一時ファイルを削除します。

```
chmod 600 /var/lib/pgsql/x/data/ca_certificate.crt

rm -f /tmp/_random.$$
```

5. ca\_certificate.crt ファイルを root.crt ファイルとして再利用します。

```
cp /var/lib/pgsql/x/data/ca_certificate.crt /var/lib/pgsql/x/data/root.crt
```

root.crt ファイルのアクセス許可を変更します。

```
chmod 600 /var/lib/pgsql/x/data/root.crt
```

6. openssl\_rsa\_generate\_crl() ファンクションを使用して、証明書失効リスト (root.crl) を作成します。

```
/usr/pgsql-x/bin/psql -U postgres -d pem --no-psqlrc -t -A -c
"SELECT openssl_rsa_generate_crl(/var/lib/pgsql/x/data/ca_certificate.crt, /var/lib/
→pgsql/x/data/ca_key.key)" > /var/lib/pgsql/x/data/root.crl
```

root.crl ファイルのアクセス許可を変更します。

```
chmod 600 /var/lib/pgsql/x/data/root.crl
```

7. openssl\_rsa\_generate\_key() ファンクションを使用して、server.key ファイルを生成します。

```
/usr/pgsql-x/bin/psql -U postgres -d pem --no-psqlrc -t -A -c "SELECT public.openssl_
→rsa_generate_key(1024)" >> /var/lib/pgsql/x/data/server.key
```

server.key ファイルを作成した後、cat server.crt ファイルの生成時に使用する変数 SSL\_KEY に内容を追加し、server.key ファイルの特権を変更します。

```
SSL_KEY=$(cat /var/lib/pgsql/x/data/server.key)
```

```
chmod 600 /var/lib/pgsql/x/data/server.key
```

8. SSL\_KEY を使用してサーバー証明書を生成します。証明書を server.crt ファイルに保存します。簡単にするために、最初に SQL クエリを一意的な名前で一時ファイルに配置します。

```
echo "SELECT openssl_csr_to_crt(openssl_rsa_key_to_csr(${SSL_KEY},
PEM,US, MA, Bedford, Postgres Enterprise Manager,
support@enterprisedb.com),
/var/lib/pgsql/x/data/ca_certificate.crt,
/var/lib/pgsql/x/data/ca_key.key)" > /tmp/_random.$$

/usr/pgsql-x/bin/psql -U postgres -d pem --no-psqlrc -t -A -f /tmp/_random.$$ >> /
→var/lib/pgsql/x/data/server.crt
```

9. server.crt ファイルの特権を変更し、一時ファイルを削除します。

```
chmod 600 /var/lib/pgsql/x/data/server.crt

rm -f /tmp/_random.$$
```

10. Postgres サーバーを再起動します。

たとえば、Centos6 ホスト上の `init.d` を使用する Linux の場合：

```
/etc/init.d/postgresql-x restart
```

たとえば、Centos7 ホスト上の `systemd` を使用する Linux の場合：

```
systemctl restart postgresql-x
```

## 7.2 エージェント SSL 証明書の更新

PEM サーバーと対話するエージェントごとに、次のことを行う必要があります。

- rsa キーと証明書を生成します。
- キーと証明書をエージェントにコピーします。
- エージェントを再起動します。

各エージェントには、`pem` データベースの `pem.agent` テーブルに保存されている一意の識別子があります。キーと証明書ファイルを、エージェントの識別子に対応するキーまたは証明書に置き換える必要があります。次のキーファイルペアを生成する前に、手順 2 と 3 で生成した `agent.key` および `agent.crt` ファイルをそれぞれの PEM エージェントホストに移動する必要があります。後続のコマンドは、以前に生成されたファイルを上書きします。

PEM エージェントキーファイルペアを生成するには：

1. `psql` を使用して、エージェントの数とそれに対応する識別子を見つけます。

```
/usr/pgsql-x/bin/psql -U postgres -d pem --no-psqlrc -t -A -c "SELECT ID FROM pem.
↪agent"
```

- Linux では、`/etc/postgres-reg.ini` ファイルの `PEMAgent` セクションで、キーと証明書のエージェント識別子と場所を見つけることもできます。
- Windows では、情報はレジストリに保存されます：
- 64 ビット Windows インストールでは、次を確認します。

```
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\EnterpriseDB\PEM\agent
```

- 32 ビット Windows インストールでは、次を確認します。

```
HKEY_LOCAL_MACHINE\SOFTWARE\EnterpriseDB\PEM\agent
```

2. キーファイルを必要とするエージェントを特定したら、エージェントごとに `agent.key` を生成します。キーを生成するには、次のコマンドを実行して、出力をファイルにキャプチャします。

```
/usr/pgsql-x/bin/psql -U postgres -d pem --no-psqlrc -t -A -c "SELECT openssl_rsa_
↳generate_key(1024)" > agent.key
```

`agent.key` ファイルの特権を変更します。

```
chmod 600 agent.key
```

3. 各エージェントの証明書を生成します。証明書を生成するには、次のコマンドを実行して、証明書ファイルに出力をキャプチャします。

```
/usr/pgsql-x/bin/psql -U postgres -d pem --no-psqlrc -t -A -c
"SELECT openssl_csr_to_cert(openssl_rsa_key_to_csr($(cat agent.key),
agent<$ID>, US, MA, Bedford, Postgres Enterprise Manager,
support@enterprisedb.com),
/var/lib/pgsql/x/data/ca_certificate.crt,
/var/lib/pgsql/x/data/ca_key.key)" > agent.crt
```

`$ID` はエージェントのエージェント番号です (psql コマンドラインで取得)。

4. `agent.crt` ファイルの特権を変更します。

```
chmod 600 agent.crt
```

5. PEM エージェントサービスを再起動する前に、各エージェントのキーと証明書ファイルを新しく生成されたファイルに置き換えます。

`init.d` を使用する Linux では、次のコマンドを使用してサービスをリスタートします。

```
/etc/init.d/pemagent start
```

`systemd` を使用する Linux では、次のコマンドを使用してサービスをリスタートします。

```
systemctl start pemagent
```

Windows ホストでは、サービスアプレットを使用して PEM エージェントを起動できます。PEM エージェントサービスの名前は Postgres Enterprise Manager Agent です。[ Services ] ダイアログボックスでサービス名を選択し、**Starttheservice** を選択します。



## 第8章 Managing configuration settings

Postgres Enterprise Manager は起動時に複数の構成ファイルを読み取ります。

- **config.py** —これがメインの設定ファイルです。変更しないでください。次のいずれかのファイルがオーバーライドされる構成設定のリファレンスとして使用できます。
- **config\_distro.py** —このファイルは **config.py** の後に読み取られ、パッケージャが Postgres Enterprise Manager ディストリビューションに必要な設定を変更することを目的としています。これには通常、特定のパスとファイルの場所が含まれます。このファイルはオプションです。パッケージャは、必要に応じて **config.py** と同じディレクトリに作成できます。
- **config\_local.py** —このファイルは **config\_distro.py** の後に読み取られ、エンドユーザーがデフォルトまたはパッケージング固有の設定を変更して、ローカルの設定または標準に合わせることを目的としています。このファイルはオプションです。必要に応じて **config.py** と同じディレクトリに作成します。

デフォルトの **config.py** ファイルのコピーは、参照用に PEM オンラインヘルプに含まれています。