



Postgres Enterprise Manager - Administration Guide

リリース *8.5.0*

EDB

2022 年 07 月 15 日

目次

第 1 章	Registering a Postgres server	3
1.1	データベースサーバーの手動登録	3
1.2	自動サーバー検出	14
1.3	pemworker ユーティリティを使用してサーバーを登録する	15
1.3.1	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	18
1.4	接続とバインドの確認	18
1.5	AWS での Postgres クラスターの登録	20
1.5.1	AWS EC2 で実行されている Postgres クラスターのモニタリング	20
1.5.2	AWS RDS で実行されている Postgres クラスターのモニタリング	21
第 2 章	Registering a PEM agent	23
2.1	PEM エージェントの登録解除	25
2.2	PEM エージェント構成パラメーターの設定	25
2.3	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	26
第 3 章	Managing a PEM server	29
3.1	PEM サーバーとエージェントの起動と停止	29
3.2	Linux での PEM サーバーまたは PEM エージェントの制御	30
3.3	Windows での PEM サーバーまたは PEM エージェントの制御	30
3.4	HTTPD サーバーの制御	31
3.5	PEM 認証の管理	32
3.6	PEM サーバー構成の編集	33
3.7	セキュリティの管理	34
3.7.1	ログインロール	34
3.7.2	グループロール	35
3.7.3	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	35
3.7.4	チームロールの使用	38
3.7.5	オブジェクトの権限	39
3.8	サーバー構成パラメーター - リファレンス	39
第 4 章	Managing Postgres servers	55
4.1	監視対象データベースサーバーの起動と停止	55
4.2	pg_hba.conf ファイルの変更	56
4.3	データベースとオブジェクトの作成と保守	57
4.4	テンプレート Linux サービススクリプト	58
第 5 章	Managing a PEM agent	59
5.1	ジョブ通知の管理	59
5.1.1	ジョブレベルでのジョブ通知の構成	59
5.1.2	エージェントレベルでのジョブ通知の構成	61
5.1.3	サーバーレベルでのジョブ通知の構成	62

5.2	PEM スケジュールされたジョブの管理	62
5.3	エージェント権限	72
5.4	エージェント構成	73
5.5	エージェントのプロパティ	78
第 6 章	Managing certificates	81
6.1	SSL 証明書の置き換え	81
6.2	エージェント SSL 証明書の更新	84
第 7 章	Managing configuration settings	87

Postgres Enterprise Manager (PEM) 管理ガイドへようこそ。このガイドでは、PEM を管理および管理するときに使用されるすべての機能について説明します。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 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。サーバーは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が、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。

1.1 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。

- 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コントロールにサーバーを表示できます。Create - Server ダイアログを開くには、Object メニュー（またはサーバーグループのコンテキストメニュー）の Create オプションをナビゲートし、Server…を選択します。

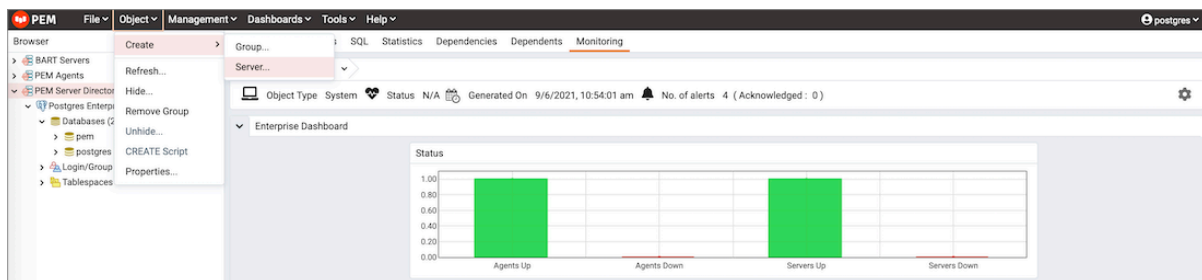


図 1: Accessing the Create – Server dialog

注釈: 接続を試みる前に、登録する 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 タブのフィールドを使用して、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、サーバーのわかりやすい名前を指定します。指定された名前は、PEM Browser ツリーコントロールでサーバーを識別します。
- グループを使用して、ツリーコントロールで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

図 2: The General tab of the Create – Server dialog

ブ、または LAN 固有のグループが必要な場合があります。Group ドロップダウンリストボックスを使用して、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、Postgres 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っている PEM ユーザーのみに、PEM にログオンしたときにこのサーバーが表示されます。このフィールドを空白のままにすると、すべての PEM ユーザーにサーバーが表示されます。
- **Background** カラーセクターを使用して、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 ツリーコントロールに表示される色を選択します。
- **Foreground** 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 ツリーコントロールのラベルのフォントの色を選択します。
- **Connect now?** の横にあるボックスをチェックして、[保存] ボタンをクリックしたときにサーバー接続を試行するように PEM に指示します。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、**Connect now?** をオフのままにします。
- **Comments** フィールドにサーバーに関するメモを入力します。

Connection tab のフィールドを使用して、サーバーの接続の詳細を指定します。

- **Host name/address** 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するか、PostgreSQL ソケットを含む代替パスに設定します。パスを入力する場合、パスは「/」で始まる必要があります。
- **Port** フィールドにホストのポート番号を指定します。
- **Maintenance database** 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 8.1 以降では、メンテナンス DB は通常 postgres と呼ばれます。

図 3: The Connection tab of the Create – Server dialog

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ために `postgres` データベースを作成することをお勧めします。

・監視対象サーバを Kerberos 認証を利用する場合、**Kerberos Authentication** ~Yes を設定します。デフォルトでは、監視対象サーバは PEM サーバーと同じ認証方法を使用します。監視対象サーバが Kerberos 認証を使用しない場合、`config_local.py` ファイルで `ALLOW_DATABASE_CONNECTION_WITHOUT_KERBEROS` を `TRUE` に設定する必要があります。

・ **Username** フィールドにサーバーで認証するとき使用する名前を指定します。

・ **Password** フィールドに指定されたユーザーに関連付けられたパスワードを入力します。

・ **Save password?** の横にあるボックスをオンにして、後で再利用できるように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するように PEM に指示します。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を切断してから、Object/Context メニューの **Clear Saved Password** メニュー項目を使用します。

・ **Role** フィールドを使用して、クライアントがサーバーに接続した後に使用する特権が割り当てられたロールの名前を指定します。これにより、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 タブのフィールドを使用して、SSL を構成します。

・ **SSL mode** フィールドのドロップダウンリストボックスを使用して、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、次の PostgreSQL ドキュメントを参照してください。

<https://www.postgresql.org/docs/current/static/libpq-ssl.html>

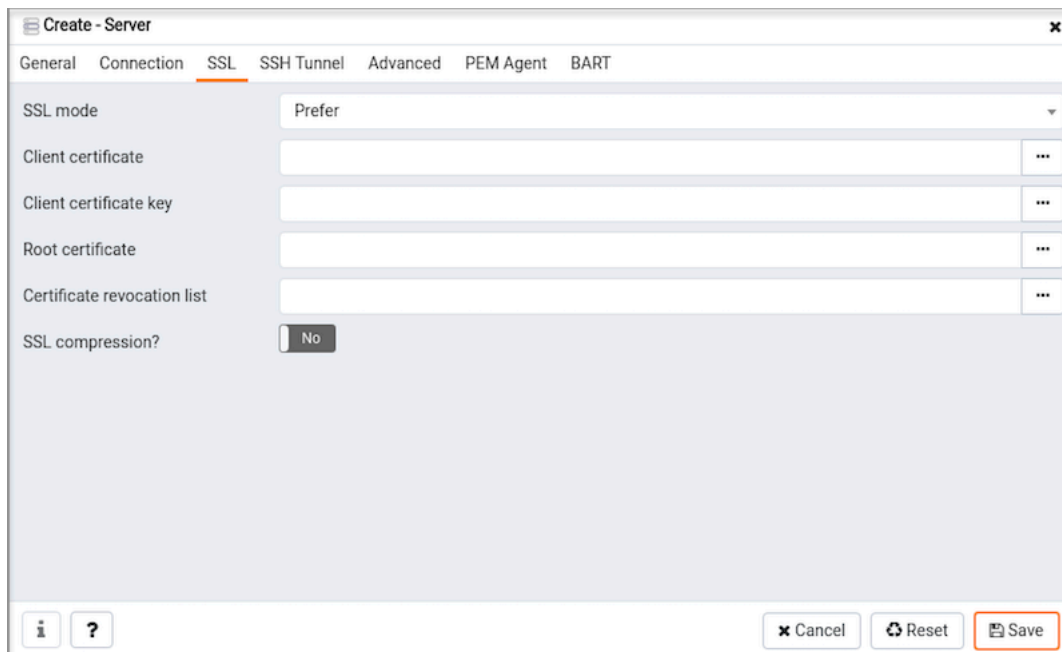


図 4: The SSL tab of the Create – Server dialog

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にアクセスするには、次の各フィールドの右側にあるアイコンをクリックします。

- **Client certificate** 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。
- **Client certificate** のキーフィールドで、クライアント証明書に使用する秘密鍵を格納した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。
- **Root certificate** 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。
- **Certificate revocation list** 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 compression?** を `True` に設定した場合、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、プライベートキー、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ストレージ領域に保存されます。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。この機能の使用を選択する前に、

適切な注意を払う必要があります。

図 5: The SSH Tunnel tab of the Create – Server dialog

SSH Tunnel タブのフィールドを使用して、SSH トンネリングを構成します。トンネルを使用して、データベースサーバーを（中間プロキシホストを介して）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。

- **Use SSH tunneling** を **Yes** に設定して、指定されたサーバーへの接続時に PEM が SSH トンネルを使用するように指定します。
- **Tunnel host** フィールドに SSH ホスト（クライアント接続が転送される）の名前または IP アドレスを指定します。
- **Tunnel port** フィールドに SSH ホスト（クライアント接続が転送される）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名を指定します。
- **Authentication** フィールドに SSH ホストに接続するときに使用される認証の種類を指定します。
- **Password** を選択して、PEM が SSH ホストへの認証にパスワードを使用することを指定します。これがデフォルトです。
- **Identity file** を選択して、接続時に PEM が秘密鍵ファイルを使用することを指定します。
- 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、**Identity file** 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、**Password** フィールドを使用してパスワードを指定するか、ID ファイルを使用している場合はパスフレーズ。

Advanced タブのフィールドを使用して、サーバーの管理に使用される詳細を指定します。

- **Host Address** フィールドにサーバホストの IP アドレスを指定します。

図 6: The Advanced tab of the Create – Server dialog

- **DB restriction** フィールドを使用して、ツリーコントロールに表示されるデータベースを制限するために `pg_database` テーブルに対して使用される SQL 制限を指定します。たとえば、次のように入力します。'live_db'、'test_db' live_db および test_db データベースのみを表示するように PEM ブラウザーに指示します。

`pg_namespace` に対する制限を入力することにより、データベースプロパティダイアログ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。

- **Password file** 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。PEM サーバーに存在する必要があります。詳細については、次の Postgres ドキュメントを参照してください。

<http://www.postgresql.org/docs/current/static/libpq-pgpass.html>

- **Service ID** フィールドを使用して、データベースサービスのプロセスを制御するための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、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、マシン名を先頭に付ける必要があります（例、PSE1\pgsql-8.0）。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、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、**EFM cluster name** 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** です。
- PEM を使用して Failover Manager クラスターのステータスを監視している場合、**EFM installation path** フィールドを使用して Failover Manager 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。

図 7: The PEM Agent tab of the Create – Server dialog

PEM Agent タブのフィールドを使用して、PEM エージェントの接続の詳細を指定します。

Connection Parameters タブ

- **Remote monitoring?** スライダーを **Yes** に移動して、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、チューニングウィザード)。
- **Bound agent** ラベルの右側にあるドロップダウンリストボックスを使用して、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。
- **Host** フィールドに、エージェントがデータベースサーバに接続するときに使用する 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 [**General**] 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、ソケットパスを指定できます。 `/tmp`。

- サーバーへの接続時にエージェントが使用する Port 番号を入力します。デフォルトでは、エージェントは [Properties] タブで定義されたポートを使用します。
- SSL フィールドのドロップダウンリストボックスを使用して、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 暗号化が利用可能な場合、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に接続で 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、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サーバーに要求する。
verify-full	SSL 暗号化を要求し、サーバーに `trusted` 認証局によって登録された証明書を使用した認証を要求します。

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。

- Database 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。BDR ノードを登録する場合は、このフィールドに BDR 対応のデータベース名を指定します。
- User name フィールドに、エージェントがサーバーに接続するときに使用するロールの名前を指定します。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ない場合、一部の機能は期待どおりに動作しないことに注意してください。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 `bdr_monitor` または `bdr_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。

- User name フィールドに、エージェントがサーバーに接続するときに使用するユーザーの名前を指定します。指定されたユーザーがデータベースのスーパーユーザーでない場合、一部の機能が期待どおりに動作しないことに注意してください。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。
- エージェントが Password フィールドにサーバーに接続するときに使用するパスワードを指定し、Confirm password フィールドに再度入力して確認します。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。このファイルは、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、アクセスする必要があります。

Advanced タブ

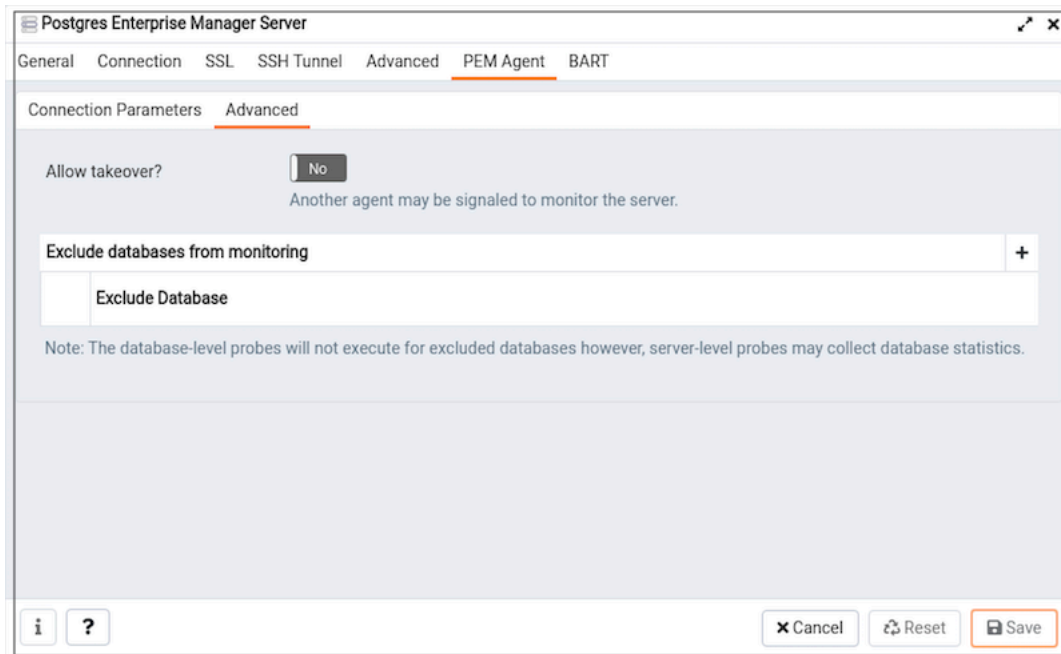


図 8: The PEM Agent Advanced tab of the Create – Server dialog

- **Allow takeover?** スライダーを **Yes** に設定して、サーバーを別のエージェントに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された場合などに、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。
- **+** 記号を使用して、PEM モニタリングから除外するデータベースを追加します。PEM Agent タブの **Connection Parameters** 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する場合があります。

BART タブの下の **General** タブのフィールドを使用して、PEM サーバーにマッピングされる BART サーバーの一般的なプロパティを説明します。

- **BART server** 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 サーバーがこのドロップダウンリストにリストされます。
- **Server name** 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。
- **Backup name** フィールドを使用して、データベースサーバ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。バックアップ名テンプレートを指定しない場合、バックアップは BART が割り当てた整数のバックアップ識別子によってのみ BART サブコマンドで参照できます。
- **Host address** 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の IP アドレスを指定します。
- **Port** 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

Postgres Enterprise Manager Server

General Connection SSL SSH Tunnel Advanced PEM Agent **BART**

General Misc

For BART configuration, you need to install a PEM agent on the database server if Remote Monitoring is disabled for the agent. BART supports database server version 9.5 and above.

BART server: Select from the list

Server name: Database server name that uniquely identifies an entry for database server in the server section of the configuration file

Backup name: Template for backup name (may include %year, %month, %day, %hour, %minute, and %second)

Host address: IP address of the database server to be configured for backup

Port:

User:

Password:

Cluster owner: Operating system user that owns the database cluster

Override archive command? ☐ No
Overrides the archive_command settings in the postgresql.conf file.

Archive command: Parameters for archive command (%p, %h, %a, %f). Ensure to bind a PEM agent and provide Service ID to reload or restart the database server.

Allow incremental backup? ☐ No

Setup passwordless SSH? ☐ No
Ensure to bind a PEM agent to setup passwordless SSH authentication between BART host and database server.

Cancel Reset **Save**

図 9: The Create Server dialog (BART - General tab)

のユーザーを指定します。このデータベースサーバ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。
- **Cluster Owner** 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 Advanced Server データベースクラスターの場合は **enterprisedb**、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 Advanced Server データベースクラスターの場合は **postgres** です。
- **Archive command** 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の `Postgresql.conf` ファイルを上書きします。サーバが追加されると、データベースサーバが再起動されるか、データベース構成がリロードされます。
- **Allow incremental backup?** スイッチを使用して、このデータベースサーバ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。
- **Setup passwordless SSH?** スイッチを使用して、データベースサーバと BART サーバー間のパスワード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、PEM エージェントをバインドしてください。パスワードなしの 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では機能しません。

BART タブの下 Misc タブのフィールドを使用して、BART Server のその他のプロパティを説明します。

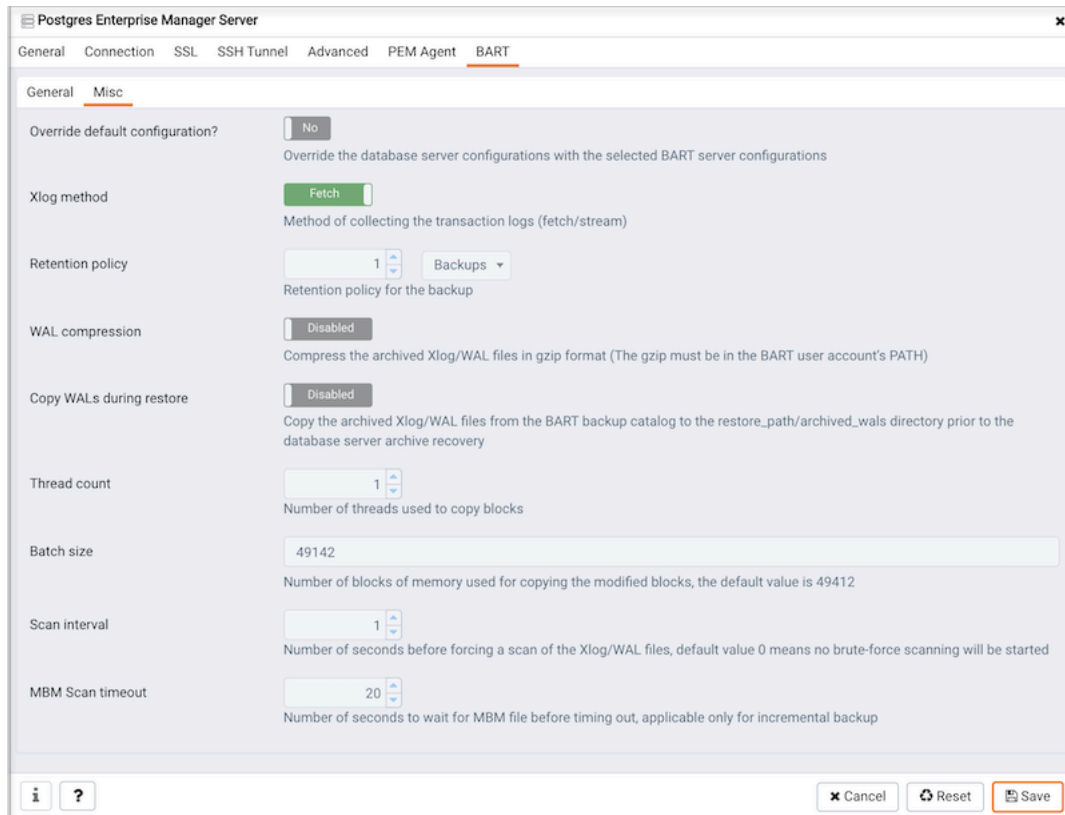


図 10: The Create - Server dialog (BART - Misc tab)

- **Override default configuration?** スイッチを使用して、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するかどうかを指定します。
- **Xlog メソッド**で `pg_basebackup` 実行時のトランザクションログの採取方法を指定します。
- **Retention policy** フィールドを使用して、バックアップの保持ポリシーを指定します。これにより、アクティブバックアップを古いものとしてマークする必要があるため、削除の候補にな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 compression** スイッチを使用して、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。増分バックアップを行う必要があるデータベースサーバーでは、`wal_compression` 設定を有効にしないでください。
- **Copy WALs during restore** 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。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するには、有効に設定します。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直接取得するには、無効に設定します。
- **Thread count** 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` ユーティリティでバックアップを作成する場合は、`thread count` を 1 に設定する必要があります。

- **Batch size** 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のブロック数を指定します。増分バックアップにのみ適用されます。
- **Scan interval** 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の秒数を指定します。
- **MBM scan timeout** 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。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コントロールでサーバー名前を右クリックし、コンテキストメニューから **Properties...** オプションを選択します。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 **Properties** ダイアログを開きます。

1.2 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、**Auto Discovery** ダイアログ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。**Manage Probes** タブにアクセスするには、PEM クライアントツリーコントロールで PEM エージェントの名前を強調表示し、**Management** メニューから **Manage Probes...** を選択します。**Manage Probes** タブが開いたら、**Enabled?** 列のスライダーコントロールが **Yes** に設定されていることを確認します。

Auto Discovery ダイアログを開くには、PEM クライアントツリーコントロールで PEM エージェントの名前を強調表示し、**Management** メニューから **Auto Discovery...** を選択します。

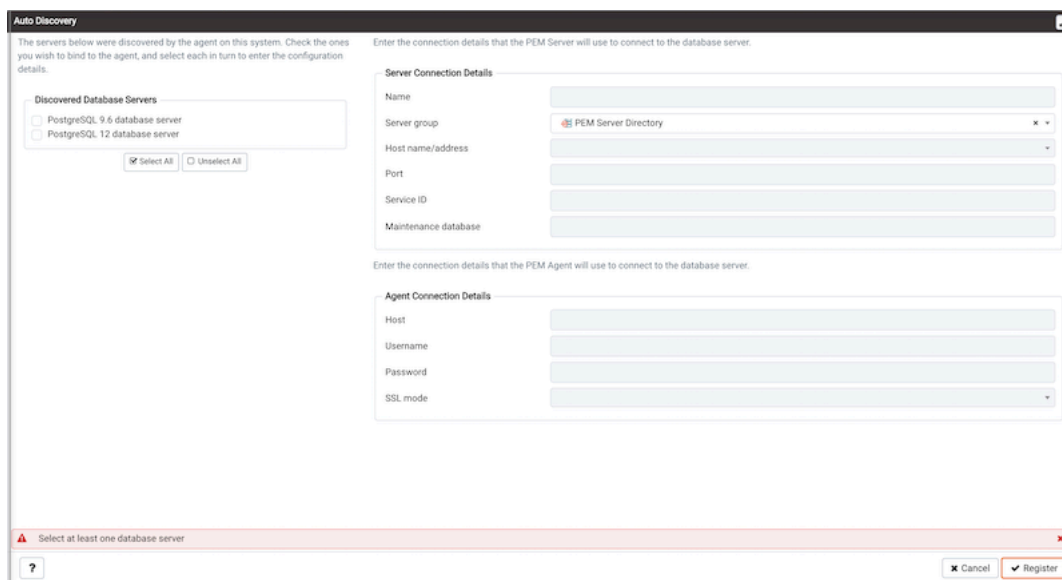


図 11: The PEM Auto Discovery dialog

Auto Discovery ダイアログが開くと、**Discovered Database Servers** 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チェックボックスをオンにして、サーバーに関する情報を **Server Connection Details** ボックスに表示し、エージェントの接続プロパティを **Agent Connection Details** ボックスに表示します。

Check All ボタンを使用して、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選択するか、Uncheck All を使用してサーバー名の左側にあるすべてのボックスの選択を解除します。

Server Connection Details 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** フィールドの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コントロールに表示されます。
- **Server group** ドロップダウンリストボックスを使用して、サーバーが PEM クライアントツリーコントロールに表示されるサーバーグループを選択します。
- **Host name/address** フィールドで監視対象サーバの IP アドレスを指定します。
- **Port** 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。
- **Service 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ることに注意してください。
- デフォルトでは、**Maintenance database** 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。**Maintenance database** フィールドの内容をインストールに合わせてカスタマイズします。

Agent Connection Details ボックスのフィールドは、サーバーへの接続時に PEM エージェントが使用するプロパティを指定します。

- **Host** フィールドには、PEM エージェントのバインドに使用される IP アドレスが表示されます。
- **User name** 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 mode** フィールドのドロップダウンリストボックスを使用して、SSL 接続設定を指定します。

監視用にバインドするサーバーの接続プロパティの指定が終了したら、OK ボタンをクリックしてサーバーを登録します。Cancel をクリックして、変更を保存せずに終了します。

OK ボタンをクリックすると、新しく登録されたサーバーが PEM ツリーコントロールに表示され、PEM サーバーによって監視されます。

1.3 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。

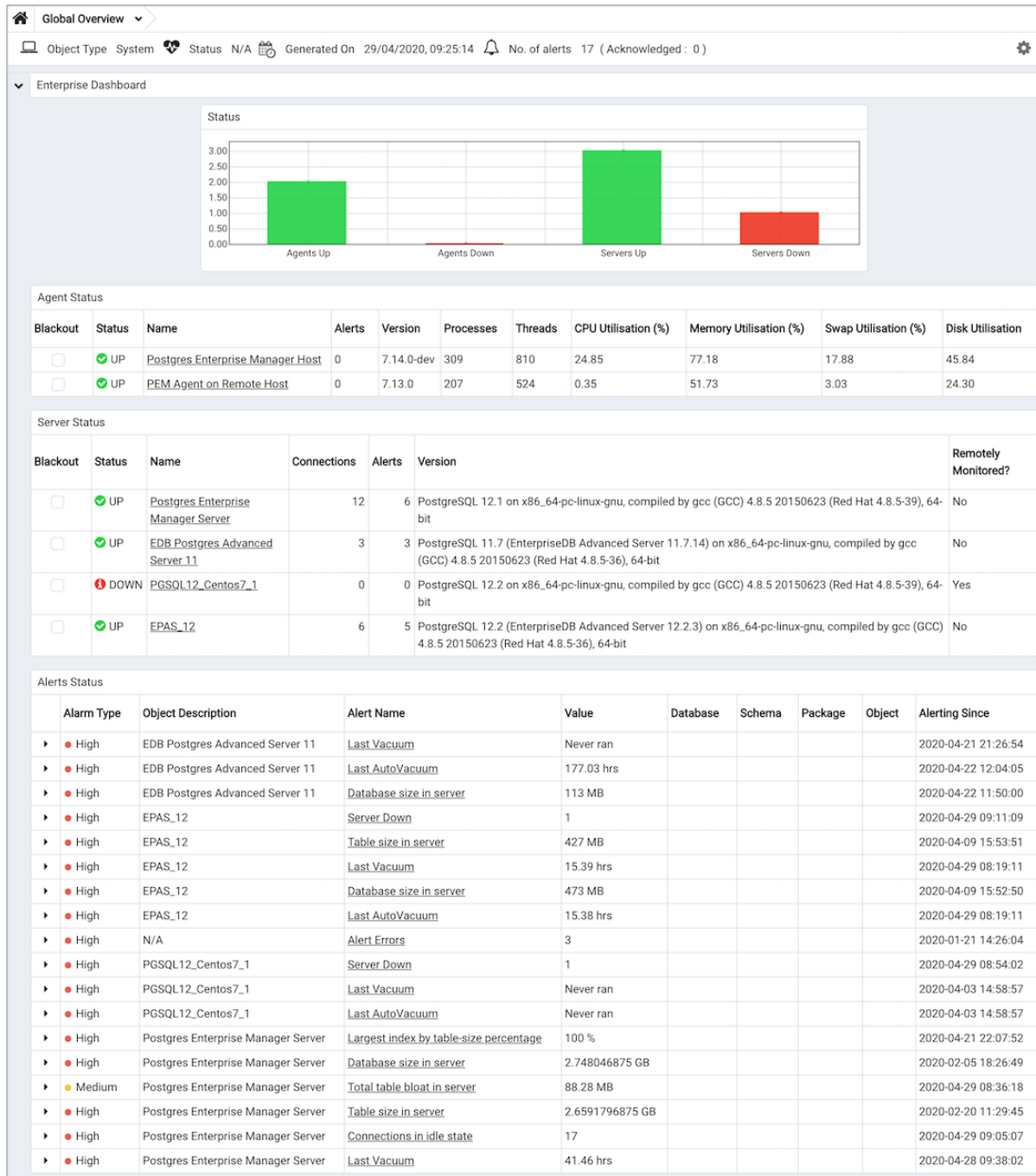


図 12: The registered server

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は、対応する値が続く必要があります。

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するか、PostgreSQL ソケットを含む代替パスに設定します。パスを入力する場合、パスは/で始まる必要があります。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>	登録されるサーバー上の操作 (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。オプション。
<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 (はい) 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 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>	フェールオーバーマネージャークラスター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 (該当する場合)。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エージェントが使用する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>	接続に使用される 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。prefer、require、disable、verify-CA、verify-full。
<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を許可す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 <name></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>"<pemworker path>/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。

1.3.1 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は、対応する値が続く必要があります。

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するか、PostgreSQL ソケットを含む代替パスに設定します。パスを入力する場合、パスは/で始まる必要があります。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>"<pemworker path>/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。

1.4 接続とバインドの確認

登録すると、新しいサーバーが PEM Browser ツリーコントロールに追加され、Global Overview 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、Global Overview ダッシュボードには、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。サーバーを認識する前に、バインドされたエージェントはサーバーを検査するために多数のプロ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完了までに数分かかる場合があります。

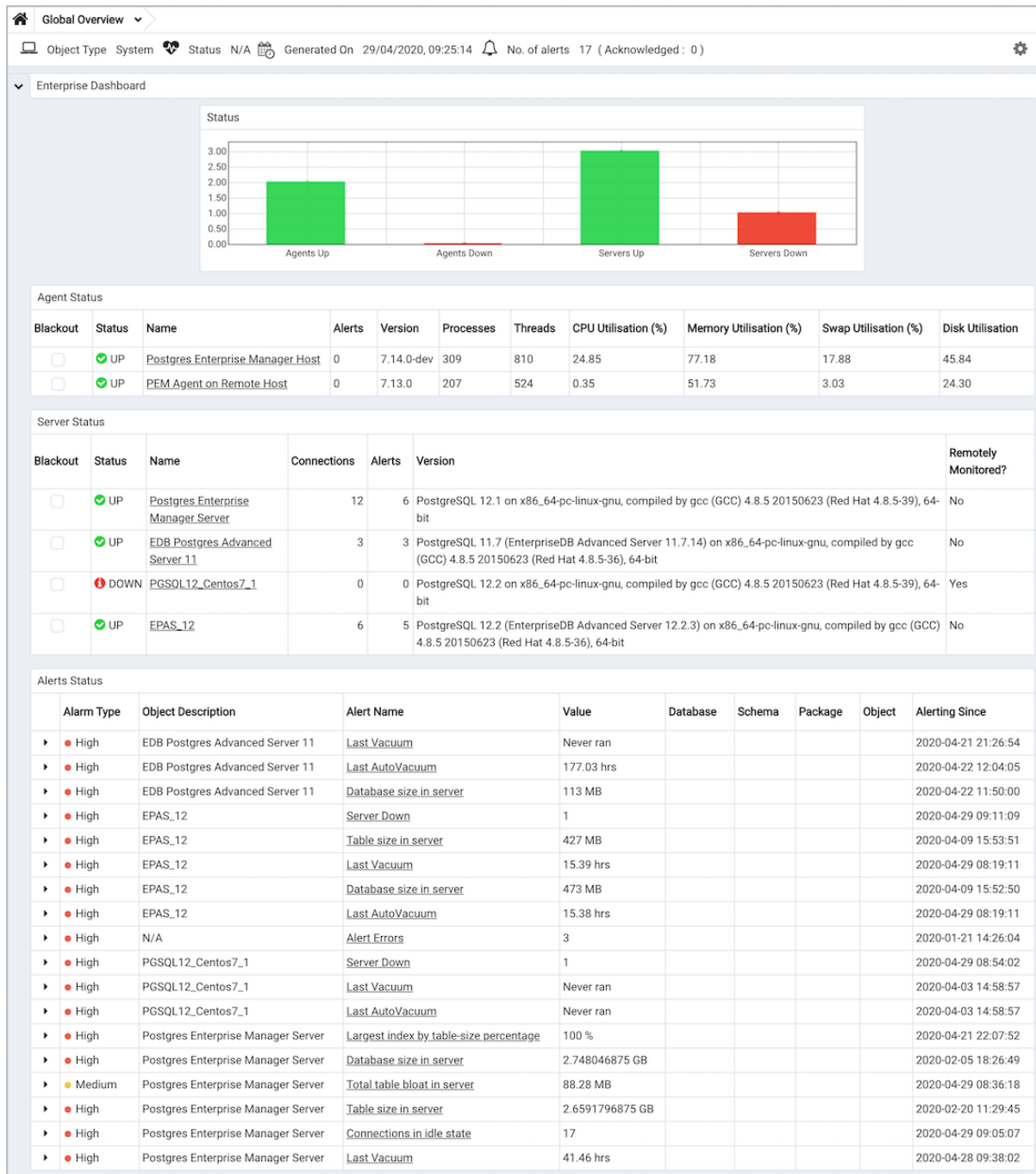


図 13: The Global Overview dashboard

数分以内に、Global Overview 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、新しいサーバーが Postgres Server Status リストに含まれることが示されます。

5 分後にも Global Overview 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を確認してエラーを確認する必要があります。ツリーコントロール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の Dashboards サブメニューから Probe Log Analysis オプションを選択します。

1.5 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

1.5.1 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 で VPN を構成して、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 をセットアップしてください。

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 エージェントを PEM サーバーに登録します。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。

PEM エージェントを PEM サーバーに登録し、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た後、PEM を使用して Postgres クラスターを監視できます。

1.5.2 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、Engine options 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2章 Registering a PEM agent

別のホスト（PEM サーバーホストとは異なる）にインストールされた各 PEM エージェントは、PEM サーバーに登録する必要があります。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。エージェントが Linux ホスト上にある場合は、`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、登録の詳細とともに `--register-agent` キーワードを含めます。

```
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は、対応する値が続く必要があります。

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 アドレスを指定します。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を指定します。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データベースユーザー（スーパーユーザー権限を持つ）の名前を指定します。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を指定します。
<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を指定します。パスを指定しない場合、証明書は次の場所に作成されます。Linux の場合、 <code>~/.pem</code> Windows の場合、 <code>%APPDATA%/pem</code>
<code>--config-dir</code>	構成ファイルがあるディレクトリパスを指定します。デフォルトは <code><pemworker path>/etc</code> です。
<code>--display-name</code>	PEM ブラウザーツリーコントロールに表示されるエージェントのわかりやすい名前を指定します。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、指定された引数でエージェントを登録するように PEM サーバーに指示します。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 [はい] 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へのアクセスに必要な 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上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 True 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 False 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を指定します。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 False です。
<code>--enable-smtp</code>	<code>enable-smtp</code> パラメーターを有効にして、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 False です。
<code>--enable-snmp</code>	<code>enable-snmp</code> パラメーターを有効にして、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 False 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、またはチューニングウィザードなど) がデータベースサーバー `restart` を必要とする 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 7.11 以降のバージョン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` パラメーターの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侵害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` パラメーター値を `'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。管理パスワードを指定するには 3 つの方法があります。次のことができます。

- 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。その他の必須だが省略された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。

2.1 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は、対応する値が続く必要があります。

Option	Description
<code>--pem-user <username></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（ <code>pem_admin</code> 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>	構成ファイルがあるディレクトリパスを指定します。デフォルトは <code>"<pem-worker path>/etc"</code> です。

2.2 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 プログラムは実際のエージェント構成ファイル（`agent.cfg` という名前）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 ファイル（認証局）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す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、Global Overview 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コントロールに表示されます。

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、

Registering database server を参照してください。

2.3 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、非 root ユーザー（たとえば、edb）の ID を引き継ぎ、次の手順を実行します。

1. edb ユーザーでログインします。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/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。

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ないため、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。
`edb_notranlate_6`

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。

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3章 Managing a PEM server

以下のセクションでは、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、変更など、PEM サーバーに関連するタスクに関する情報を提供します `pg_hba.conf` ファイル、プロキシサーバーを使用するように PEM を変更

3.1 PEM サーバーとエージェントの起動と停止

PEM サーバーは、それが常駐する 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、ポイントアンドクリック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。**Services** ユーティリティは、Windows Control Panel からアクセスできます。ユーティリティが開いたら、スクロールバーを使用してリストされたサービスを移動し、サービス名を強調表示します。

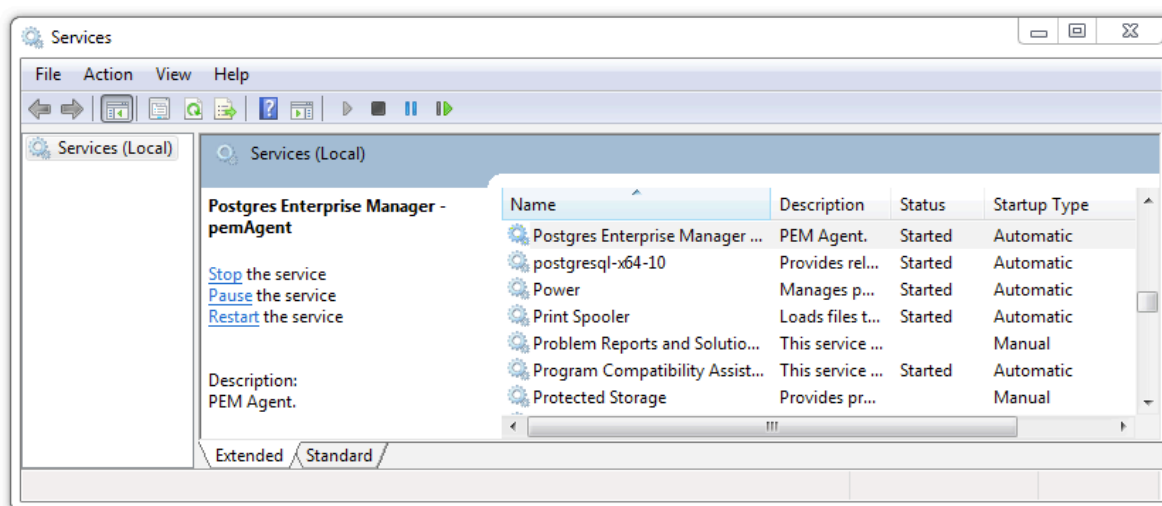


図 1: The PEM service in the Windows Services window

Stop、Pause、Start、または Restart ボタンを使用して、サービスの状態を制御します。

サービスを停止すると、Postgres サーバーに接続しているユーザー（またはクライアントアプリケーション）が突然切断されることに注意してください。サービスの制御の詳細については、EDB Web サイトから

入手可能な [*EDB Postgres Advanced Server Installation Guide*](#) を参照してください。

3.2 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。

- Advanced Server 上の PEM サーバーは `edb-as-<x>` または `ppas-<x>`
- 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、コマンドラインを開き、スーパーユーザ権限を引き受けて、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。

- サービスを開始する `start`。
- `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。

3.3 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、ポイントアンドクリック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 上の PEM サーバーホストは `edb-as-<x>`、または `ppas-<x>`
- 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 Control Panel をナビゲートして、**Services** ユーティリティを開きます。ユーティリティが開いたら、スクロールバーを使用してサービスのリストを参照します。

サービスを停止するには、**Stop the service** オプションを使用します。サービスを停止すると、サーバーに接続しているユーザー（またはクライアントアプリケーション）は突然切断されます。

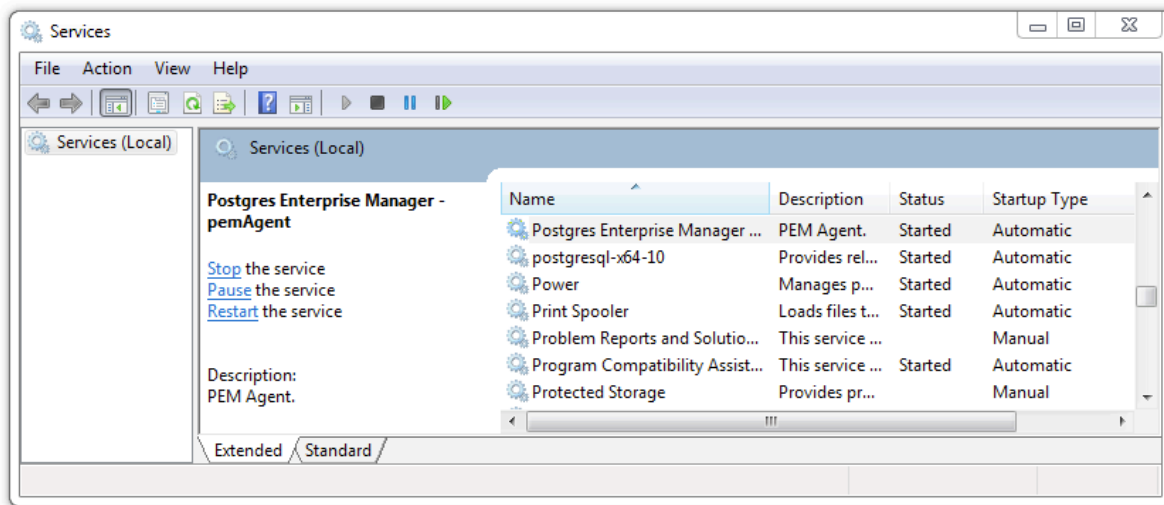


図 2: The Windows Services window

Pause the service オプションを使用して、サービスの構成パラメーターをリロードするように Postgres に指示します。Pause the service オプションは、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。

Start the service オプションを使用してサービスを開始します。

3.4 HTTPD サーバーの制御

Linux では、コマンドラインを開き、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 が次の例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、Services 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、リストをスクロールして PEM HTTPD サービスを見つけます。

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 リンクをクリックして PEM HTTPD を起動します。

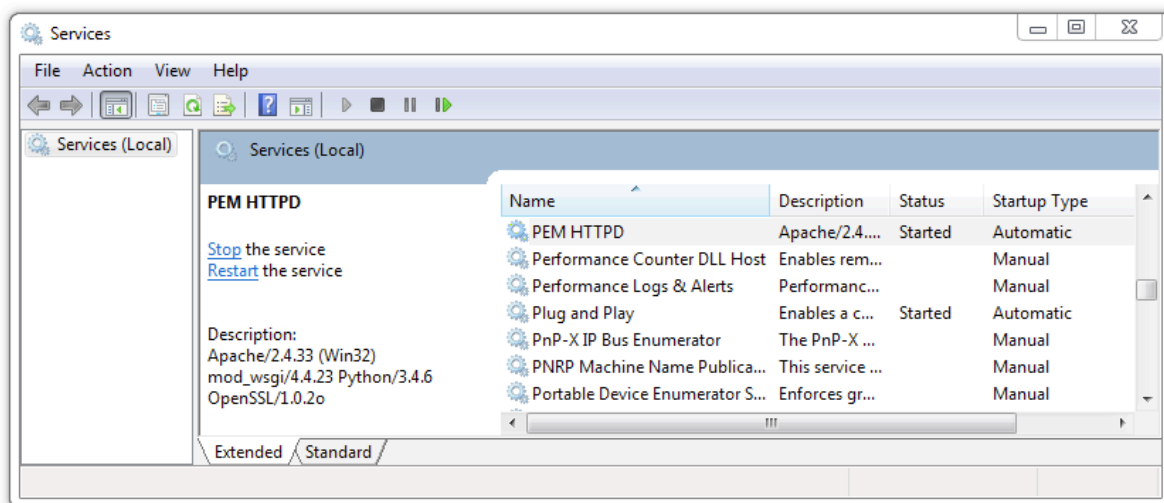


図 3: The PEM HTTPD Windows service

3.5 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 (および PEM) 認証は、`pg_hba.conf` 構成ファイルによって制御されます。構成ファイル内のエントリは、特定のデータベースに接続できるユーザーと、そのユーザーに接続を許可する前に必要な認証の種類を指定します。

`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は次のとおりです。

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

システムの構成によっては、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して、エージェントがサーバーの認証要求に適切に応答できるようにする必要があります。1safepwd というパスワードを持つ `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は次の形式になります。

localhost:5432:*:postgres:1safepwd

パスワードファイルの名前は通常、Linux システムの場合は `~root/.pgpass`、Windows の場合は `%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](#) をご覧ください。

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。

3.6 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グラフィカルに管理し、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 ダイアログを開くには、Management メニューから Server Configuration… を選択します。

Parameter	Value	Unit
audit_log_retention_time	30	days
auto_create_agent_alerts	True	t/f
auto_create_server_alerts	True	t/f
bart_log_retention_time	30	days
cm_data_points_per_report	50	
cm_max_end_date_in_years	5	years
dash_alerts_timeout	60	seconds
dash_db_comrol_span	168	hours
dash_db_comrol_timeout	1800	seconds
dash_db_connovervw_timeout	300	seconds
dash_db_eventlag_span	7	days
dash_db_eventlag_timeout	1800	seconds
dash_db_hottable_rows	25	rows
dash_db_hottable_timeout	300	seconds
dash_db_io_span	168	hours
dash db io timeout	1800	seconds

図 4: The Server Configuration dialog

パラメーターの値を変更する場合は、パラメーター名の右側の Value フィールドに表示される内容を編集します。Save ボタンをクリックして変更を保存するか、Close ボタンをクリックして変更を適用せずにダイアログを終了します。Reset ボタンを使用して、パラメーターを元の値に戻します。

3.7 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グラフィカル方法を提供します。

3.7.1 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 インターフェース **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 ダイアログを使用してロールを作成するには、PEM ツリーコントロールでロールが存在するサーバーのノードを展開し、**Login/Group Roles** ノードを右クリックしてコンテキストメニューにアクセスします。次に、**Create** メニューから **Login/Group Role...** を選択します。

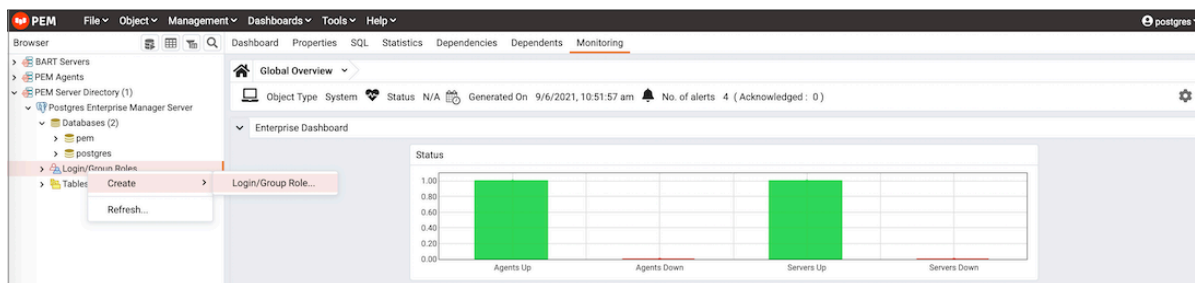


図 5: The context menu of the Login Roles node

Create - Login/Group Role ダイアログのタブのフィールドを使用して、ロールを定義します。ブラウザータブに PEM オンラインヘルプを表示するには、ダイアログの左下隅にあるヘルプ (?) ボタンをクリックします。

新しいロールの定義が終了したら、**Save** をクリック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、ツリーコントロール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 **Properties** 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。

図 6: Create Login Role - General tab

3.7.2 グループロール

グループロールは、システム権限（データベースの作成など）とオブジェクト権限（特定のテーブルへのデータの挿入など）を分配するために使用されるコンテナとして機能できます。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より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。

Group Roles ノード（PEM ツリーコントロールの登録済みサーバーの名前の下）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にアクセスできます。ロール

[here](#) の作成に関する詳細を見つけることができます。

3.7.3 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 ダイアログを使用して、権限が制限されたロールが監査マネージャー、Capacity Manager、SQL プロファイラーなどの PEM 機能にアクセスできるようにします。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、**Membership** 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、ドロップダウン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、**Save** をクリック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。

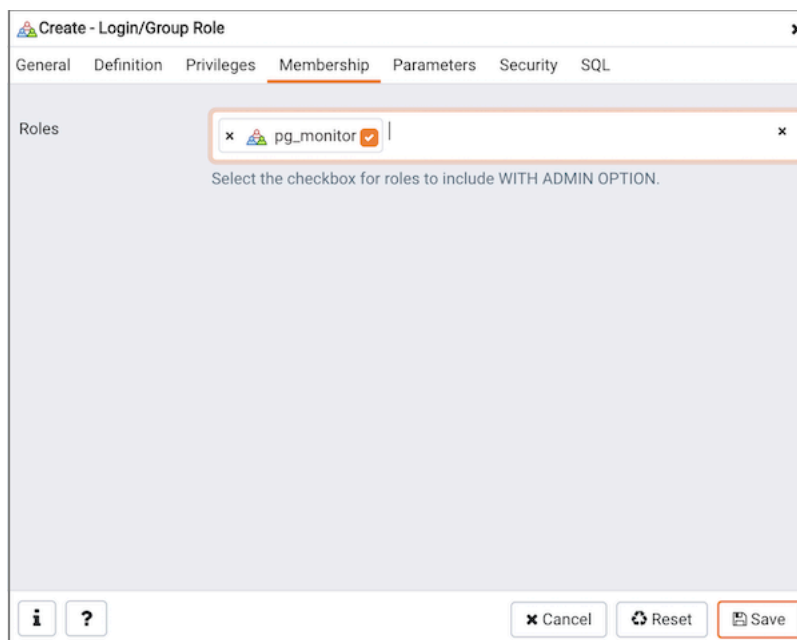


図 7: Create Login Role - Membership tab

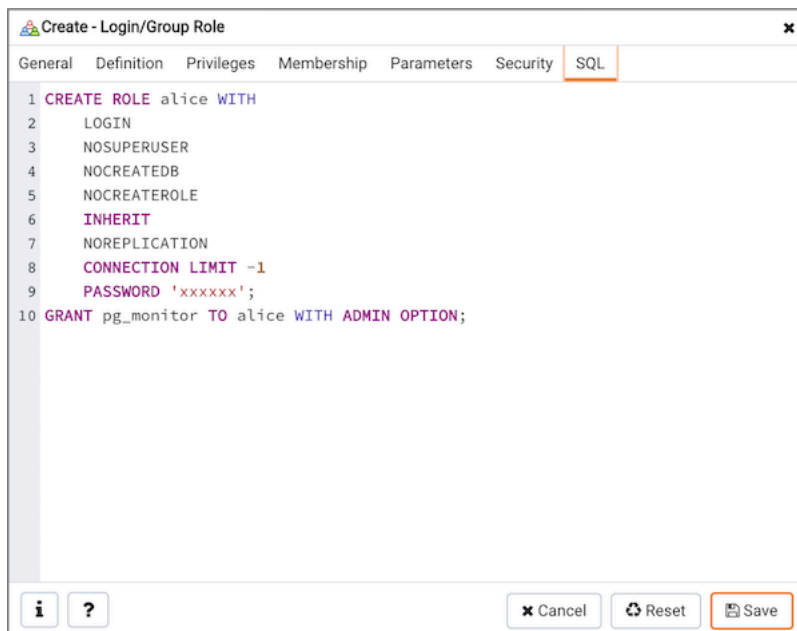


図 8: Create Login Role - SQL tab

上記の例では、`acctg_clerk` という名前のログインロールを作成します。ロールは、任意の時点でサーバーへの無制限の接続を作成できます。

PEM の事前定義されたロールを使用して、以下の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
<code>pem_super_admin</code>		Postgres Enterprise Manager 上のすべてを管理/構成するロール。
<code>pem_admin</code>	<code>pem_super_admin</code>	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/構成のためのロール。
<code>pem_config</code>	<code>pem_admin</code>	Postgres Enterprise Manager の構成管理のためのロール。
<code>pem_component</code>	<code>pem_admin</code>	すべてのウィザード/ダイアログベースのコンポーネントを実行/実行するロール。
<code>pem_rest_api</code>	<code>pem_admin</code>	REST API にアクセスするロール。
<code>pem_server_service_management</code>	<code>pem_admin</code>	監視対象データベースサーバーの再起動/リロードを許可するためのロール（サーバー ID が提供されている場合）。
<code>pem_manage_schedule_task</code>	<code>pem_admin</code>	スケジュールタスクを設定するロール。
<code>pem_manage_alert</code>	<code>pem_admin</code>	アラートを管理/設定するロールとそのテンプレート。
<code>pem_config_alert</code>	<code>pem_config</code> , <code>pem_manage_alert</code>	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。
<code>pem_manage_probe</code>	<code>pem_admin</code>	カスタムプローブを作成、更新、削除、およびカスタムプローブ構成を変更するロール。
<code>pem_config_probe</code>	<code>pem_config</code> , <code>pem_manage_probe</code>	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。
<code>pem_database_server_registration</code>	<code>pem_admin</code>	データベースサーバを登録するロール。
<code>pem_comp_postgres_expert</code>	<code>pem_component</code>	Postgres 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_auto_discovery	pem_component	データベースサーバー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。
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 機能を管理するロール。
pem_comp_capacity_manager	pem_component	Capacity Manager を実行するロール。
pem_comp_log_manager	pem_component	Log Manager 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	チューニングウィザードを実行するロール。

3.7.4 チームロールの使用

PEM による監視のためにサーバーを登録するとき、サーバーに関連付けられる**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、指定されたチームを持つサーバーを次のユーザーに表示します。

- 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、PEM ツリーコントロールでロールが存在するサーバーのノードを展開し、Login/Group Roles ノードを右クリックしてコンテキストメニューにアクセスします。次に、Create メニューから Login/Group Role… を選択します。Create - Login/Group Role ダイアログが開いたら、提供された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。

3.7.5 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 クライアントのグラフィカル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を介して管理されます。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、ツリーコントロール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 [Properties] オプションを選択します。[特権] タブに表示されるオプションを使用して、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 Grant Wizard（Tools メニューからアクセス）も含まれています。

3.8 サーバー構成パラメーター - リファレンス

以下のセクションは参照用です。次の表に記載されている事項は、変更される場合がありますので、ご了承ください。

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ることに注意してください。

Parameter name	Value and Unit	Description
audit_log_retention_time	30 days	監査ログが PEM サーバーに保持され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。

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
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自動更新されるまでの秒数を指定します。
dash_db_comrol_span	7 days	Database Analysis ダッシュボードおよび 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 ダッシュボードと 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自動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自動更新されるまでの秒数を指定します。
dash_db_eventlag_span	7 days	Database Analysis 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	スロニーレプリケーションのイベント数ラグ折れ線グラフが Database Analysis ダッシュボードで自動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。

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_hottable_timeout	300 seconds	Database Analysis ダッシュボードで Hot Tables テーブルが自動更新されるまでの秒数を指定します。
dash_db_io_span	7 days	Database Analysis ダッシュボードおよび I/O Analysis ダッシュボードの Database I/O Analysis グラフにプロットするデータの日数を指定します。
dash_db_io_timeout	1800 seconds	Database Analysis ダッシュボードおよび I/O Analysis ダッシュボードで Database I/O 折れ線グラフが自動更新されるまでの秒数を指定します。
dash_db_rowact_span	7 days	データベース分析ダッシュボード、I/O 分析ダッシュボード、およびサーバー分析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 ダッシュボード、I/O Analysis ダッシュボード、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自動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自動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。

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_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 ダッシュボードで自動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線グラフが自動更新されるまでの秒数を指定します。
dash_efm_timeout	300 seconds	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ストリーミングレプリケーションダッシュボードで自動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自動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自動更新されるまでの秒数を指定します。
dash_io_chkpt_span	7 days	I/O Analysis 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	チェックポイント折れ線グラフが I/O 分析ダッシュボードに自動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 棒グラフが自動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 棒グラフが自動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 ダッシュボードと 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 ダッシュボードと 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自動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 ダッシュボードおよび Object Activity Analysis ダッシュボードの [Object I/O Details] テーブルに表示される行数を指定します。
dash_io_objectio_timeout	300 seconds	I/O Analysis ダッシュボードおよび Object Activity Analysis ダッシュボードでオブジェクト I/O の詳細テーブルが自動更新されるまでの秒数を指定します。
dash_memory_hostmemory_act_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_hostmemory_act_timeout	300 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自動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_memory_hostmemc	300 seconds	Memory Analysis ダッシュボードおよび Server Analysis ダッシュボードでホストメモリ構成円グラフが自動更新されるまでの秒数を指定します。
dash_memory_servmemact	7 days	Memory Analysis ダッシュボードのサーバーの Memory Activity Analysis 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	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自動更新されるまでの秒数を指定します。
dash_memory_servmemc	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自動更新されるまでの秒数を指定します。
dash_objectact_objstorage	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	300 seconds	Object Activity Analysis ダッシュボードで Object Storage テーブルが自動更新されるまでの秒数を指定します。
dash_objectact_objtopindex	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自動更新されるまでの秒数を指定します。
dash_objectact_objtoptable	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自動更新されるまでの秒数を指定します。
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。

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_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 折れ線グラフが自動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自動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 折れ線グラフが自動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自動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自動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_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自動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自動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。
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 ラインチャートが自動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自動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自動更新されるまでの秒数を指定します
dash_replication_archivespan	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_archivestart	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自動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_replication_pagelag_7days	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_1800seconds	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自動更新されるまでの秒数を指定します。
dash_replication_segmentlag_7days	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_segmentlag_1800seconds	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自動更新されるまでの秒数を指定します。
dash_replication_timelag_7days	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_1800seconds	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自動更新されるまでの秒数を指定します。
dash_server_buffers_writes_68hours	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_3600seconds	3600 seconds	Server Analysis 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自動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_connovervw_300seconds	300 seconds	Server Analysis ダッシュボードで接続の概要の円グラフが自動更新されるまでの秒数を指定します。
dash_server_database_time300seconds	300 seconds	Server Analysis ダッシュボードで Databases テーブルが自動更新されるまでの秒数を指定します。
dash_server_dbsize_span 7 days	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_time1800seconds	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自動更新されるまでの秒数を指定します。
dash_server_disk_timeout1800seconds	1800 seconds	Server Analysis ダッシュボードでディスク折れ線グラフが自動更新されるまでの秒数を指定します。
dash_server_global_span 7 days	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	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_time1800seconds	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自動更新されるまでの秒数を指定します。
dash_server_tabspacesize_7days	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_time1800seconds	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自動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_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_time	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自動更新されるまでの秒数を指定します。
dash_sessact_lockact_time	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自動更新されるまでの秒数を指定します。
dash_sessact_workload_time	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自動更新されるまでの秒数を指定します。
dash_sess_waits_nowaits_time	300 seconds	セッションが待機数で待機するまでの秒数を指定します。
dash_sess_waits_timewait_time	300 seconds	Session Waits By Time Waited 円グラフが Session Waits Analysis ダッシュボードで自動更新されるまでの秒数を指定します。
dash_sess_waits_waitdtl_time	300 seconds	Session Waits Analysis ダッシュボードで Session Waits Details テーブルが自動更新されるまでの秒数を指定します。
dash_storage_dbdtls_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自動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自動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_hostdtls_timeouts	300 seconds	ホストの詳細テーブルが自動更新されるまでの秒数を指定します
dash_storage_hostoverview_timeouts	300 seconds	Storage Analysis ダッシュボードでホストの概要円グラフが自動更新されるまでの秒数を指定します。
dash_storage_tblspcdtls_timeouts	300 seconds	Storage Analysis ダッシュボードで Tablespace Details テーブルが自動更新されるまでの秒数を指定します。
dash_storage_tblspcoverview_timeouts	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自動更新されるまでの秒数を指定します。
dash_sys_waits_nowaits_timeouts	300 seconds	System Waits By Number Of Waits 円グラフが System Waits Analysis ダッシュボードで自動更新されるまでの秒数を指定します。
dash_sys_waits_timewait_timeouts	300 seconds	System Waits By Time Waited 円グラフが System Waits Analysis ダッシュボードで自動更新されるまでの秒数を指定します。
dash_sys_waits_waitdtl_timeouts	300 seconds	システム待機の詳細テーブルがシステム待機分析ダッシュボードで自動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 の場合があります。

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
flap-ping_detection_state_change	3	指定された間隔内に検出される状態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、 $[2 * N * \min(\text{probe_interval})]$ 分間にわたって ZERO の状態変化が発生すると終了します。
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付けられた日数を指定します
long_running_transaction	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>	パッシブサービスのチェック結果を送信する nagios コマンドファイルを指定します。
nagios_enabled	t	nagios にアラート通知を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。

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
re-minder_notification_interval	24 hours	アラートがクリアされない場合にリマインダー電子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される 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される 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され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される 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される 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、その可能な値は次のとおりです。SNMPv3 でのみ使用されます。
snmp_context_name		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名を指定します。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。

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_authentication_protocol	NONE	SNMP トラップの認証タイプを指定します。その可能な値は NONE、HMACMD5 または 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。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 コアドキュメントの場所を指定します。

第4章 Managing Postgres servers

4.1 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、サーバー登録ダイアログを完了し、データベースサーバーを PEM エージェント（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）に登録します。

- Properties ダイアログで Store on PEM Server オプションを指定します。
- Advanced タブの Service 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。
- 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、ツリーコントロールでサーバー名を強調表示し、Tools メニューから Queue Server Startup または Queue Server Shutdown を選択することにより、サーバーを起動または停止できます。

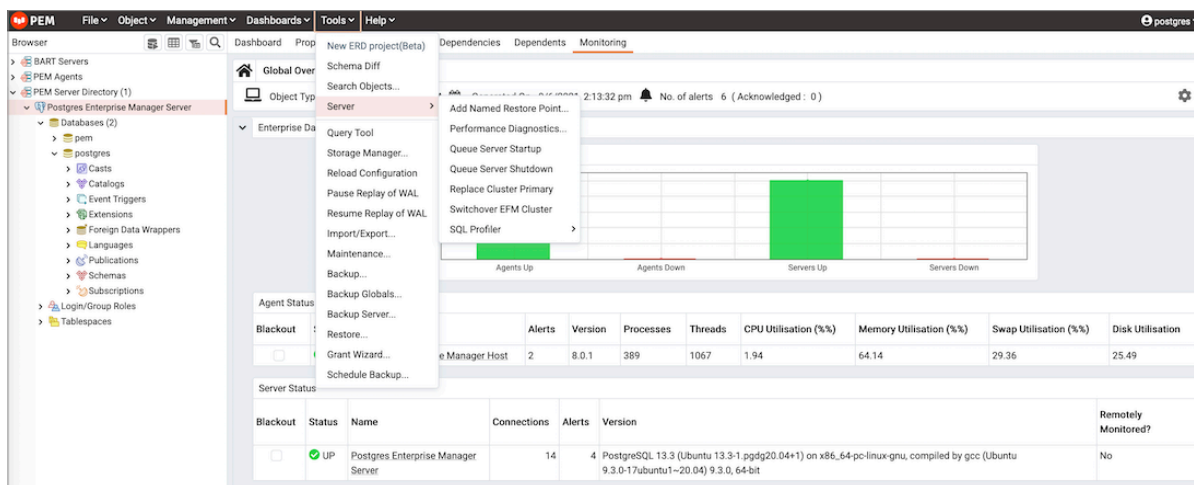


図 1: The Tools menu of a managed server

4.2 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中に、PEM によって監視されるホストの 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 バックアップデータベースの pg_hba.conf ファイルの先頭に追加されます。

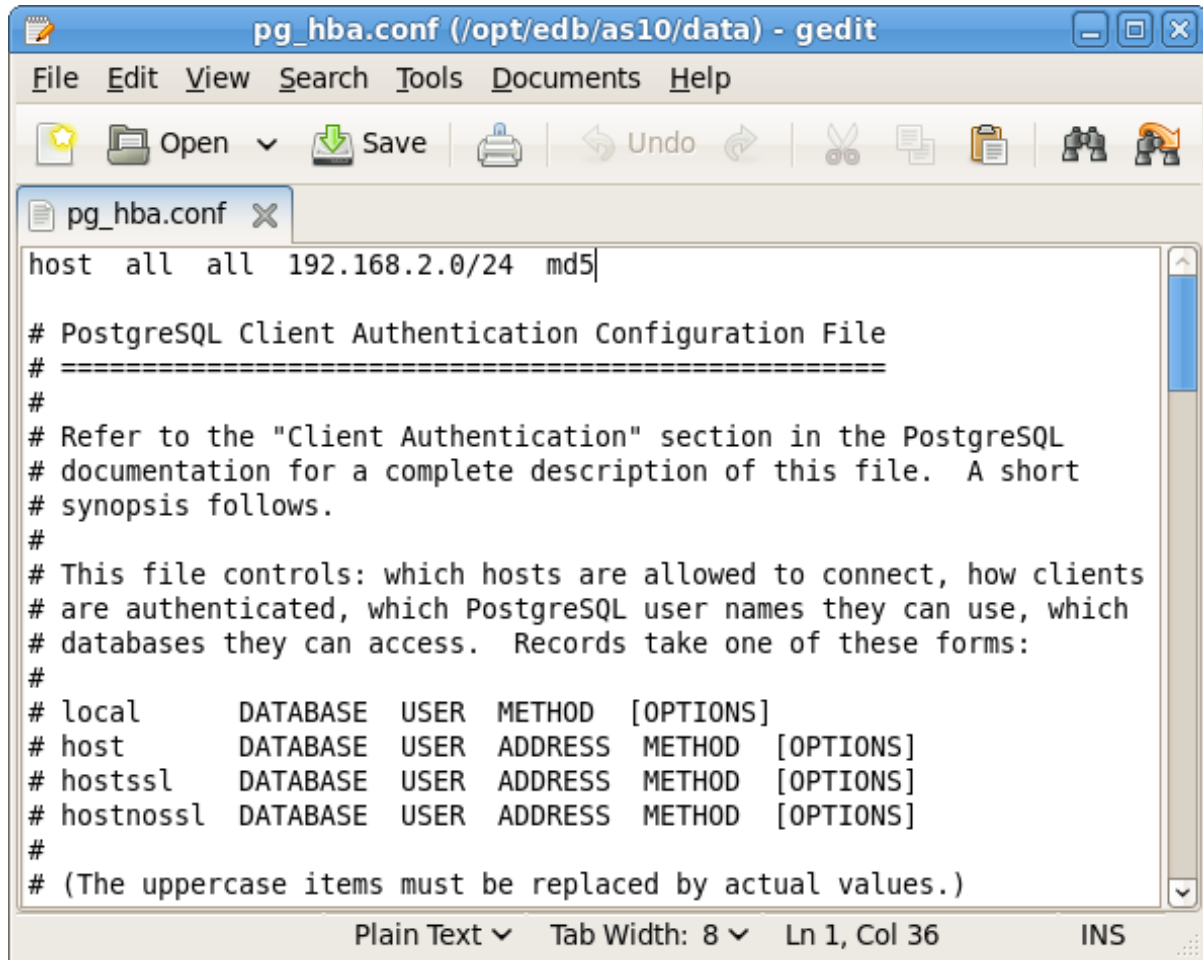


図 2: PEM entries in the pg_hba.conf file

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 インストールの下での data ディレクトリにあります。たとえば、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。

次の例は、postgres という名前のユーザーから pg_hba.conf ファイル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このエラーを受け取った場合は、pg_hba.conf ファイルを変更し、接続を許可するエントリーを追加します。

4.3 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 クライアントは、データベース、およびデータベースを構成するさまざまなオブジェクト（テーブル、インデックス、ストアドプロシージャなど）を作成および管理できるダイアログを提供します。

PEM でのデータベースの作成は簡単です。管理対象サーバーの **Databases** ノードを右クリックし、**Create** メニューから **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内にオブジェクトを作成できます。

たとえば、新しいテーブルを作成するには、**Tables** ノードを右クリックし、**Create** メニューから **Table...** を選択します。New Table ダイアログが開いたら、新しいテーブルの属性を指定します。

PEM は、他のデータベースオブジェクトの作成と管理のための同様のダイアログを提供します。

- 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

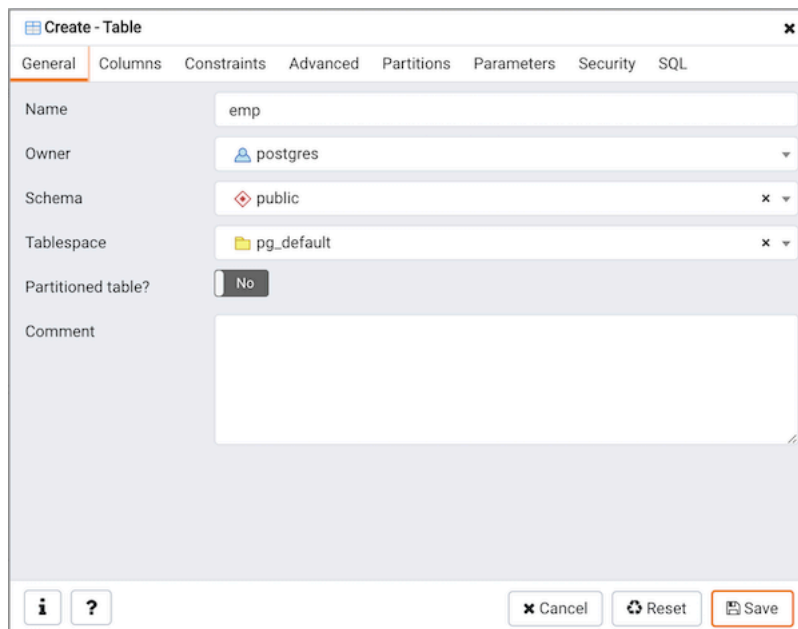


図 3: Create Table - General Tab

- ビュー
- 制約など

各オブジェクトタイプがツリーコントロールに表示されます。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 **Create** メニューにアクセスし、新しいオブジェクトを作成するか、名前付けたノードのコンテキストメニューから **Properties** を選択して、強調表示されたオブジェクトの管理タスクを実行します。

4.4 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 サーバーは、構成管理、証明書管理、またはその他の管理タスクを実行するときに、必要に応じて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。
- 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。

CentOS または RHEL7.x の場合|**RockyLinux または RHEL8.x**、サービススクリプトは `/usr/lib/systemd/system` ディレクトリにあります。

- サービススクリプトはプラットフォーム固有です。
- Postgres サービスのカスタマイズについては、次の Web サイトにアクセスしてください。

<https://www.postgresql.org/docs/current/static/server-start.html>

第5章 Managing a PEM agent

以下のセクションでは、PEM エージェントの動作と管理に関する情報を提供します。

5.1 ジョブ通知の管理

システム生成のジョブ（スケジュールされたタスクの下にリストされます）またはカスタム定義のエージェント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ように、PEM コンソールで設定を構成できます。これらの電子メール通知設定を次の3つのレベル（優先順位）で構成して、指定したユーザグループに電子メール通知を送信できます。

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

5.1.1 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、カスタム定義のエージェントジョブに対してのみジョブレベルで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、ジョブ自分自身の作成中に **Create-Agent Job** ウィザードの **Notification** タブでメール通知設定を構成できます。
- 既存のカスタム定義ジョブの場合、ジョブのプロパティを編集し、通知設定を構成できます。

Notifications タブのフィールドを使用して、ジョブレベルでメール通知設定を構成します。

- **Send the notifications** フィールドを使用して、電子メール通知をいつ送信するかを指定します。
- **Email group** フィールドを使用して、メール通知を受信するメールグループを指定します。

The screenshot shows the 'Create - Agent Job' dialog box with the 'Notifications' tab selected. The dialog has four tabs: 'General', 'Steps', 'Schedules', and 'Notifications' (which is highlighted with an orange border). Below the tabs, there are two main sections: 'Send the notifications' and 'Email group'. The 'Send the notifications' section has a dropdown menu set to 'ALWAYS'. Below this dropdown, there is explanatory text: 'Determines when to send a notification for the job:'. This is followed by three options: 'ON FAILURE : Send a notification on the failure/interruption of the job.', 'ALWAYS : Send a notification on the completion of the job regardless of the result.', and 'NEVER : Do not send a notification for the job.'. Below these is the 'DEFAULT :' option with the text 'Use the agent/system level job notification configuration to determine whether, and when to send the notification.'. The 'Email group' section has a dropdown menu set to '<Default>'. Below this dropdown is the text 'Select the email-group to get the job/scheduled-task notification on completion.'. At the bottom of the dialog, there are three buttons: 'Cancel', 'Reset', and 'Save' (which is highlighted with an orange border). There are also information and help icons on the bottom left.

Create - Agent Job

General Steps Schedules **Notifications** SQL

Send the notifications

ALWAYS

Determines when to send a notification for the job:

ON FAILURE :
Send a notification on the failure/interruption of the job.

ALWAYS :
Send a notification on the completion of the job regardless of the result.

NEVER :
Do not send a notification for the job.

DEFAULT :
Use the agent/system level job notification configuration to determine whether, and when to send the notification.

Email group

<Default>

Select the email-group to get the job/scheduled-task notification on completion.

Cancel Reset Save

図 1: Configure Job Notification - Job level

5.1.2 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、右クリックして[プロパティ]を選択します。Properties ダイアログで、ジョブ通知タブを選択します。

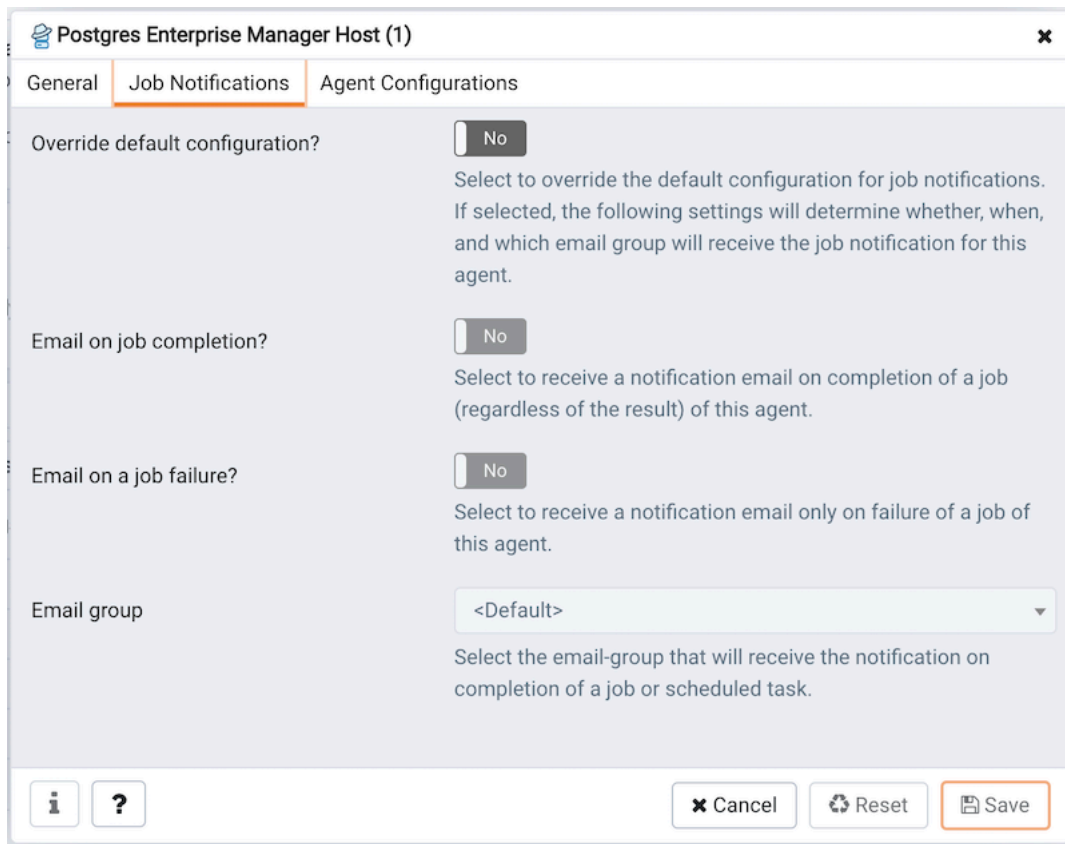


図 2: Configure Job Notification - Agent level

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

- **Override default configuration?** 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。このスイッチでは**はい**を選択した場合、このダイアログの残りの設定を使用して、ジョブ通知をいつ、誰に送信するかを定義できます。このダイアログの残りの設定は、**Override default configuration?** スイッチを有効にしている場合にのみ機能することに注意してください。
- **Email on job completion?** 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。
- **Email on a job failure?** 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。
- **Email group** 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。

5.1.3 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。サーバーの構成ダイアログを開くには、PEM クライアントの [Management] メニューから *Server Configuration*...を選択します。

The screenshot shows the 'Server Configuration' dialog box with a search bar at the top right labeled 'Search by parameter name'. Below the search bar is a table of configuration parameters. The 'job_status_change_notification' row is highlighted with a red border. At the bottom of the dialog are buttons for '?', 'Cancel', 'Reset', and 'Save'.

Parameter Name	Value	Unit
job_failure_notification	False	t/f
job_notification_email_group	default	
job_retention_time	30	days
job_status_change_notification	True	t/f
long_running_transaction_minutes	5	minutes
max_metrics_per_group_chart	16	
nagios_cmd_file_name	/usr/local/nagios/var/rw/nagios.cmd	
nagios_enabled	True	t/f
nagios_medium_alert_as_critical	False	t/f
nagios_spool_retention_time	7	days
probe_log_retention_time	30	days
reminder_notification_interval	24	hours

図 3: Configure Job Notification - Server level

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。

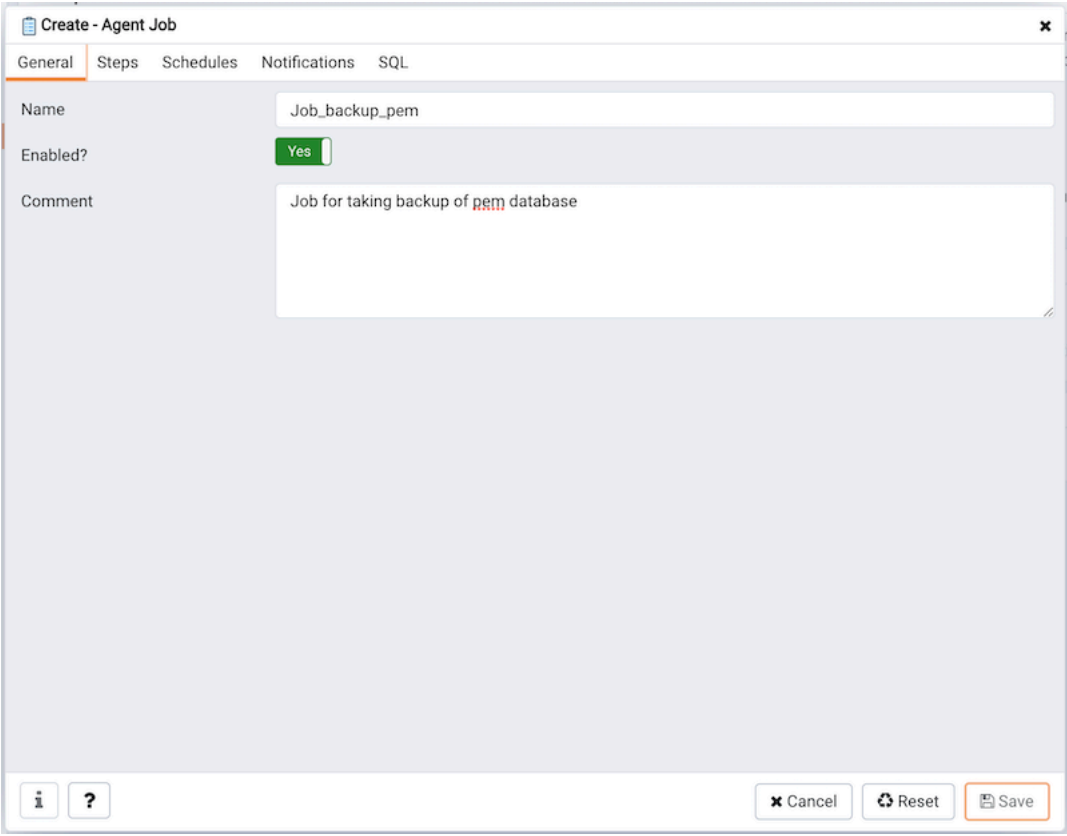
- **job_failure_notification** スイッチを使用して、ジョブが失敗するたびに電子メール通知を送信するかどうかを指定します。
- **job_notification_email_group** パラメーターを使用して、メール通知を受信するメールグループを指定します。
- **job_retention_time** parameter を使用して、繰り返しのないスケジュールされたタスクをシステムに保持する日数を指定します。
- **job_status_change_notification** スイッチを使用して、ステータスが失敗、成功、または中断に関係なく、各ジョブのステータスが変更された後に電子メール通知を送信するかどうかを指定します。

5.2 PEM スケジュールされたジョブの管理

指定されたシーケンスで一連のカスタム定義されたステップを実行する 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/シェルスクリプトが含まれる場合があります。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 ツリーコントロールを使用して、ジョブを作成する PEM エージェントを参照します。ツリーコントロールに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表示されます。新しい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 **Create Job...** を選択します。

Create - Agent Job ダイアログが開いたら、Create - Agent Job ダイアログのタブを使用して、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。



The screenshot shows the 'Create - Agent Job' dialog box with the 'General' tab selected. The 'Name' field contains 'Job_backup_pem', the 'Enabled?' checkbox is checked (labeled 'Yes'), and the 'Comment' field contains 'Job for taking backup of pem database'. The bottom of the dialog features 'Cancel', 'Reset', and 'Save' buttons.

図 4: Create Agent Job - General tab

General タブのフィールドを使用して、ジョブに関する一般情報を提供します。

- Name 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 の位置に移動してジョブを有効にするか、No を移動します。
- Comment 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。

Steps 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。Add アイコン (+) をクリックして、新しいステップを追加します。次に、(ヘッダーの左側にある) 作成アイコンをクリックしてステップ定義ダイアログを開きます。

ステップ定義ダイアログのフィールドを使用して、ステップを定義します。

- Name 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ることに注意してください。
- Enabled スイッチを使用して、ジョブ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、[Code] タブを使用してステップの SQL コードを提供します。- Batch を選択する場合、Code タブを使用して、ステップ中に実行されるバッチスクリプトを提供します。

The screenshot shows the 'Create - Agent Job' window with the 'Steps' tab active. A table lists the job steps. Below the table, the 'General' sub-tab is selected, displaying configuration details for the selected step.

Name	Enabled?	Kind	On error
Step1_backup	True	SQL	Success

General	
Name	Step1_backup
Enabled?	Yes
Kind	SQL
On error	Success
Server	Postgres Enterprise Manager Server (192.168.1.19:5432)
Database	pem
Comment	

図 5: Create Agent Job - Steps tab

- **On error** ドロップダウンを使用して、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。 - **Success** - 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。 - **Ignore** - 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、次の追加情報を提供します。
- **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ているサーバーを指定します。 - **Database** 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の **Code** タブの状況依存フィールドを使用して、ステップ中に実行される 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、**SQL query** 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、**Code** フィールドにスクリプトを提供します。Windows サーバーで実行している場合、標準のバッチファイル構文を使用する必要があります。Linux サーバーで実行する場合、最初の行で適切なインタープリターが指定されていれば、任意のシェルスクリプトを使用できます（例：`#!/bin/sh`）。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。

The screenshot shows a 'Create - Agent Job' dialog box with the 'General' tab selected. The dialog has a title bar with a close button. Below the title bar are tabs for 'General', 'Steps', 'Schedules', 'Notifications', and 'SQL'. The 'General' tab contains the following fields:

- Name:** A text input field containing 'Job_backup_pem_db'.
- Enabled?:** A checkbox with a green 'Yes' label and a checked state.
- Comment:** A text area containing 'Job for taking backup of pem database'.

At the bottom of the dialog, there are three buttons: 'Cancel' (with a close icon), 'Reset' (with a refresh icon), and 'Save' (with a save icon and an orange border).

図 6: Create Agent Job - Steps tab - General tab

Create - Agent Job

General **Steps** Schedules Notifications SQL

	Name	Enabled?	Kind	On error
☒	step1_backup	☒ True	☒ SQL	Success

General Code

Name: step1_backup

Enabled?: ☒ Yes

Kind: ☒ SQL

On error: Success

Server: Postgres Enterprise Manager Server (192.168.1.19:5432)

Database: pem

Comment:

図 7: Create Agent Job - Steps tab - Code tab

- 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、スクリプトの実行に使用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。

ステップに必要なすべての情報を入力したら、**Save** ボタンをクリックして保存してステップ定義ダイアログを閉じます。

追加アイコン (+) をクリックして各ステップを追加するか、**Schedules** タブを選択してジョブスケジュールを定義します。

追加アイコン (+) をクリックして、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) 作成アイコンをクリックして、スケジュール定義ダイアログを開きます。

The screenshot shows the 'Create - Agent Job' dialog box with the 'Schedules' tab selected. The 'General' sub-tab is active, showing details for the 'Backup_pem_db_schedule' job. The table lists the job name, its status (Enabled), and its start/end times. Below the table, the 'General' sub-tab provides a form to configure the job's name, status, start/end times, and a comment.

Name	Enabled?	Start	End
Backup_pem_db_schedule	True	2020-04-01 23:30 +05:30	2020-04-02 23:30 +05:30

General Repeat Exceptions

Name: Backup_pem_db_schedule

Enabled?: Yes

Start: 2020-04-01 23:30 +05:30

End: 2020-04-02 23:30 +05:30

Comment:

Buttons: Cancel, Reset, Save

図 8: Create Agent Job - Schedules tab - General tab

Schedules definition タブのフィールドを使用して、ジョブを実行する日時を指定します。

- Name** 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 (Yes) を使用するか、スケジュールを無効

にする (No)。

- **Start** フィールドのカレンダーセレクトを使用して、スケジュールの開始日時を指定します。
- **End** フィールドのカレンダーセレクトを使用して、スケジュールの終了日時を指定します。
- **Comment** 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。

Repeat タブを選択して、スケジュールを実行する日を定義します。

Create - Agent Job

General Steps **Schedules** Notifications SQL

Name	Enabled?	Start	End
Backup_pem_db_schedule	True	2020-04-01 23:30 +05:30	2020-04-02 23:30 +05:30

General **Repeat** Exceptions

Schedules are specified using a **cron-style** format.
 For each selected time or date element, the schedule will execute.
 e.g. To execute at 5 minutes past every hour, simply select '05' in the Minutes list box.
 Values from more than one field may be specified in order to further control the schedule.
 e.g. To execute at 12:05 and 14:05 every Monday and Thursday, you would click minute 05, hours 12 and 14, and weekdays Monday and Thursday.
 For additional flexibility, the Month Days check list includes an extra Last Day option. This matches the last day of the month, whether it is a weekday or not.

Days

Week Days: Select the weekdays...

Month Days: × 1st ×

Months: Select the months...

Times

Hours: Select the hours...

Minutes: Select the minutes...

Buttons: ? i Cancel Reset Save

図 9: Create Agent Job - Schedules tab - Repeat tab

Repeat タブのフィールドを使用して、スケジュールに関する詳細を cron 形式で指定します。ジョブは、[Repeat] 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。

フィールド内をクリックして、そのフィールドの有効な値のリストを開きます。特定の値をクリックして、その値を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 をクリックします。

- **Days** ボックス内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。
 - **Week Days** 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。
 - **Month Days** フィールドを使用して、ジョブを実行する数値を選択します。 **Last Day** を指定して、日付に関係なく、月末にジョブを実行することを示します。
 - **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** ボックス内のフィールドを使用して、ジョブを実行する時間を指定します。
 - **Hours** フィールドを使用して、ジョブを実行する時間を選択します。
 - **Minutes** フィールドを使用して、ジョブを実行する分を選択します。

行する分を選択します。

Exceptions タブを選択して、スケジュールが **not** を実行する日を指定します。

The screenshot shows the 'Create - Agent Job' dialog box with the 'Schedules' tab selected. Within this tab, the 'Exceptions' sub-tab is active. The main table lists the schedule 'Backup_pem_db_schedule' as 'Enabled?'. Below this, a sub-table for exceptions shows a single entry for the date '2020-04-01' at time '00:00'. The 'Save' button at the bottom right is highlighted in orange.

図 10: Create Agent Job - Schedules tab - Exceptions tab

Exceptions タブのフィールドを使用して、ジョブを実行しない日を指定します。たとえば、祝日にジョブを実行しないようにすることができます。

追加アイコン (+) をクリックして例外表に行を追加し、次のことを行います。

- **Date** 列内をクリックしてカレンダーセクターを開き、ジョブを実行しない日付を選択します。**Date** 列に<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。
- **Time** 列内をクリックして時間セクターを開き、ジョブを実行しない時間を指定します。**Time** 列に<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。

Notifications タブのフィールドを使用して、ジョブのメール通知設定を構成します。

- **Send the notifications** フィールドを使用して、電子メール通知をいつ送信するかを指定します。
- **Email group** フィールドを使用して、メール通知を受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終了したら、[**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。

The screenshot shows the 'Create - Agent Job' dialog box with the 'Notifications' tab selected. The dialog has a title bar with a close button (X) and a tab bar with 'General', 'Steps', 'Schedules', 'Notifications' (active), and 'SQL'. The main content area is divided into two sections: 'Send the notifications' and 'Email group'. The 'Send the notifications' section has a dropdown menu set to 'ALWAYS'. Below the dropdown, there is explanatory text: 'Determines when to send a notification for the job:'. It lists three options: 'ON FAILURE : Send a notification on the failure/interruption of the job.', 'ALWAYS : Send a notification on the completion of the job regardless of the result.', and 'NEVER : Do not send a notification for the job.' It also mentions 'DEFAULT : Use the agent/system level job notification configuration to determine whether, and when to send the notification.' The 'Email group' section has a dropdown menu set to '<Default>'. Below it, there is text: 'Select the email-group to get the job/scheduled-task notification on completion.' At the bottom of the dialog, there are three buttons: 'Cancel', 'Reset', and 'Save' (highlighted with an orange border). There are also information (i) and help (?) icons on the bottom left.

Create - Agent Job

General Steps Schedules **Notifications** SQL

Send the notifications

ALWAYS

Determines when to send a notification for the job:

ON FAILURE :
Send a notification on the failure/interruption of the job.

ALWAYS :
Send a notification on the completion of the job regardless of the result.

NEVER :
Do not send a notification for the job.

DEFAULT :
Use the agent/system level job notification configuration to determine whether, and when to send the notification.

Email group

<Default>

Select the email-group to get the job/scheduled-task notification on completion.

Cancel Reset Save

図 11: Create Agent Job - Notifications tab

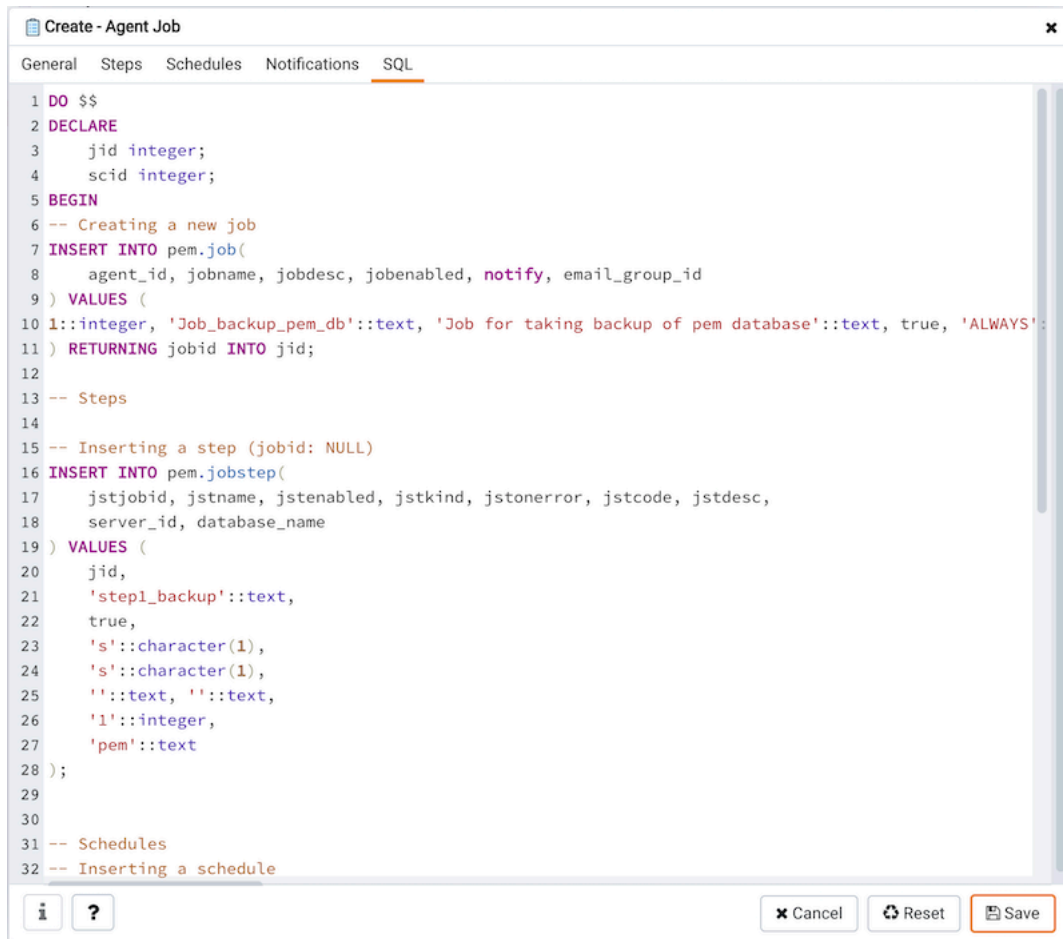


図 12: Create Agent Job - SQL tab

Save ボタンをクリックしてジョブ定義を保存するか、Cancel ボタンをクリックして保存せずにジョブを終了します。Reset ボタンを使用して、未保存のエントリをダイアログから削除します。

ジョブを保存すると、ジョブは定義されたサーバーの PEM ツリーコントロールの Jobs ノードの下にリストされます。PEM コンソールの Properties タブには、選択したジョブの高レベルの概要が表示され、[Statistics] 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 Properties を選択します。

5.3 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 特権とデータベースサーバーのスーパーユーザー特権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 機能が低下することに注意してください。完全な機能を実現するには、PEM エージェントを root として実行する必要があります。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 プロブは情報を結果ず、次の機能が影響を受ける可能性があります。

Probe or Action	Operating System	PEM Functionality Affected
Data And Logfile Analysis	Linux/ Windows	Postgres Expert は完全な情報にアクセスできません。
Session Information	Linux	プロセスごとの統計は不完全です。
PG HBA	Linux/ Windows	Postgres Expert は完全な情報にアクセスできません。
Service restart functionality	Linux/ Windows	Audit Log Manager、Server Log Manager Log Analysis Expert、および PEM は、要求された変更を適用できない場合があります。
Package Deployment	Linux/ Windows	PEM はダウンロードしたインストールモジュールを実行できません。
Batch Task	Windows	PEM は Windows でスケジュールされたバッチジョブを実行できません。
Collect data from server (root access required)	Linux/ Windows	スワップ使用量、CPU 使用率、IO 読み取り、IO 書き込みなどの列は、セッションアクティビティダッシュボードに 0 として表示されます。

注釈: 上記のリストは包括的なものではありませんが、制限される機能の種類の概要を提供する必要があります。

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 は PEM データベースから空のデータを受け取ります。
Server Log Collection	Linux/Windows	PEM はサーバーログ情報を収集できません。
Database Statistics	Linux/Windows	データベース/サーバー分析ダッシュボードには不完全な情報が含まれます。
Session Waits/System Waits	Linux/Windows	セッション/システム待機ダッシュボードには、不完全な情報が含まれます。
Locks Information	Linux/Windows	データベース/サーバー分析ダッシュボードには不完全な情報が含まれます。
Streaming Replication	Linux/Windows	ストリーミングレプリケーションダッシュボードには情報が表示されません。
Slony Replication	Linux/Windows	Database Analysis ダッシュボードの Slony 関連のグラフには情報が表示されません。
Tablespace Size	Linux/Windows	Server Analysis ダッシュボードには完全な情報が表示されません。
xDB Replication	Linux/Windows	PEM は xDB アラートとトラップを送信できません。

プローブが十分な権限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。

プローブが不十分な権限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、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。

Probe Log ダッシュボードでサーバーで発生したプローブ関連のエラーを表示したり、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、ログ情報は **Event Viewer**。

5.4 エージェント構成

多数のユーザー設定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する必要があります。

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、以下にリストされている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_crt</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まれる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 エージェントの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 PEM エージェントが待機する最大時間 (秒)。	30 seconds
<code>short_wait</code>	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 (実行を待機する) までの最小時間 (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 以下) 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、送信するメールの重複が少なくなります。	true for PEM server host; false for all others.
<code>enable_snmp</code>	複数の PEM エージェント (7.13 以下) を true に設定すると、より多くの重複トラップ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、送信するトラップは少なくなります。	true for PEM server host; false for all others.
<code>enable_nagios</code>	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.

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_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秒単位の最大時間（10 進の整数文字列）。	Not set by default; set to 0 to indicate the agent should 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応じて、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 に設定すると、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す。	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される 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。	
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

レジストリには次の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	ログレベルは、PEM ログファイルに書き込まれる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 PEM エージェントが待機する最大時間（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（実行を待機する）までの最小時間（秒）。	10 seconds
AlertThreads	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.
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.
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 will wait indefinitely.

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
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応じて、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 に設定すると、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す。	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	スクリプトファイル (アラート用) が保存され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される 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。	
WebhookSSLCrl	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。	
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

5.5 エージェントのプロパティ

PEM エージェント Properties ダイアログは、ダイアログが開かれた PEM エージェントに関する情報を提供します。ダイアログを開くには、PEM クライアントツリーコントロールでエージェント名前を右クリックし、コンテキストメニューから Properties を選択します。

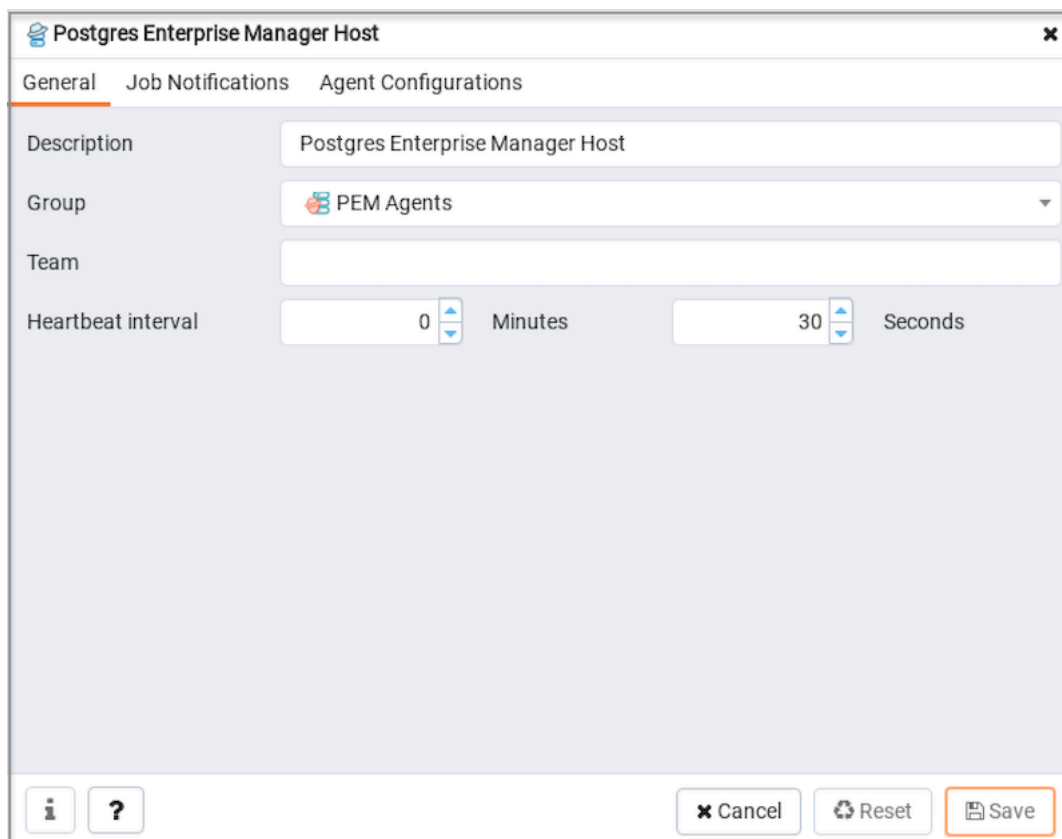


図 13: PEM Agent Properties dialog - General tab

PEM エージェント Properties ダイアログのフィールドを使用して、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コントロールに表示されます。
- グループを使用して、PEM クライアントツリーコントロールでサーバーとエージェントを整理できます。Group ドロップダウンリストボックスを使用して、エージェントを表示するグループを選択します。
- **Team** 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ようにす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の PEM クライアントツリーコントロールに表示されます。これは便利な機能であることに注意してください。チームフィールドは真の分離を提供しないため、セキュリティ目的で使わないでください。
- **Heartbeat interval** フィールドには、PEM エージェントから PEM サーバーへの経過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。

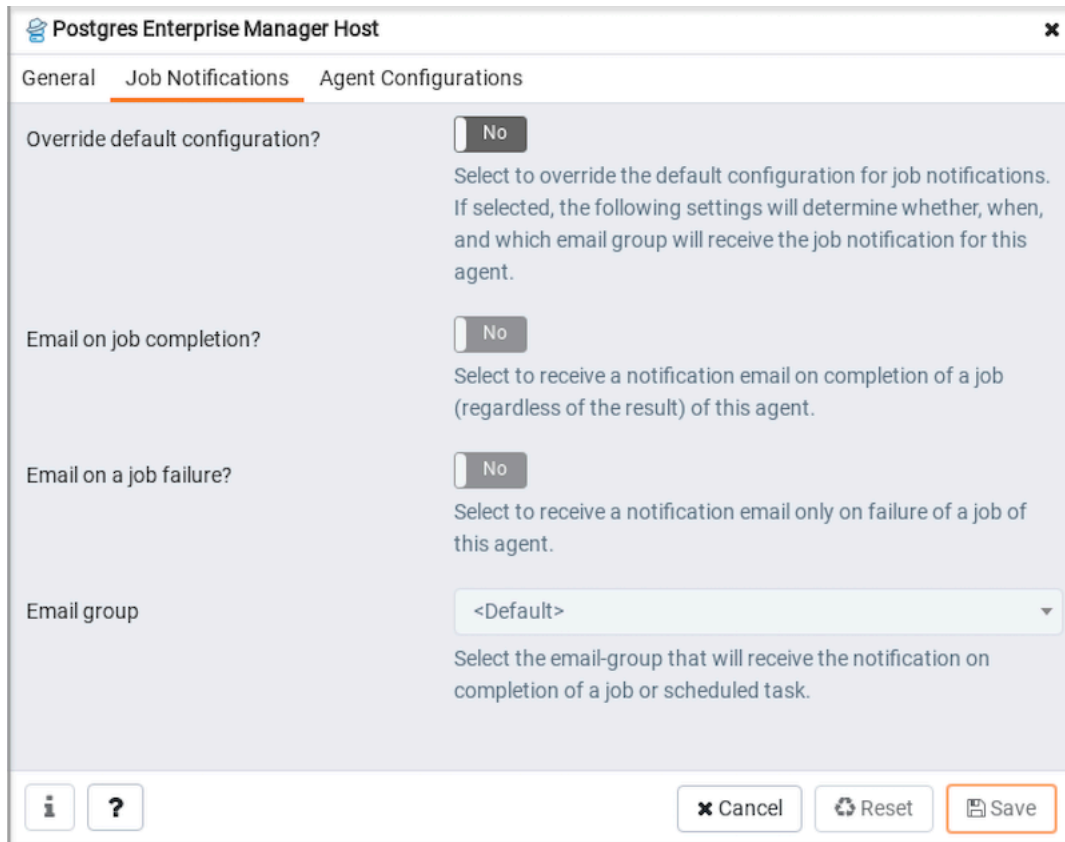


図 14: PEM Agent Properties dialog - Job Notifications tab

Job Notifications タブのフィールドを使用して、エージェントレベルでメール通知設定を構成します。

- **Override default configuration?** 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。このスイッチに **Yes** を選択した場合、このダイアログの残りの設定を使用して、ジョブ通知をいつ誰に送信するかを定義できます。このダイアログの残りの設定は、**Override default configuration?** スイッチを有効にしている場合にのみ機能することに注意してください。
 - **Email on job completion?** 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。
 - **Email on a job failure?** 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。
 - **Email group** 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。

Agent Configurations 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。

- **Parameter** 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。
 configuration または capability のいずれかです。

Postgres Enterprise Manager Host

General

Job Notifications

Agent Configurations

Parameter	Value	Category
Agent Id	1	configuration
Running as root?	true	capability
Running as User	root	capability
Platform	"CentOS Linux 7 (Core)"	capability
Architecture	x64	capability
PEM host	127.0.0.1	configuration
PEM port	5444	configuration
Log level	debug1	configuration
Agent SSL key path	/root/.pem//agent1.key	configuration
Agent SSL crt path	/root/.pem//agent1.crt	configuration
Long wait	30	configuration
Short wait	10	configuration
Alert threads	1	configuration

i

?

✕ Cancel

🔄 Reset

💾 Save

図 15: PEM Agent Properties dialog - Agent Configurations tab

第6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利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証明書失効リスト (`root.crl` ファイルで保持) 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ことが保証されます。

以下のセクションには、証明書ファイルの手動置換に関する詳細情報が含まれています。

6.1 SSL 証明書の置き換え

次の手順では、既存の PEM インストールの SSL 証明書の置換について詳しく説明します。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. 最初にサーバーホストで、次に監視対象ノードで、実行中のすべての PEM エージェントを停止します。

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。

Linux で systemd を使用している場合：Centos 7 または 8

```
systemctl stop pemagent
```

Windows ホストでは、**Services** 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** ダイアログでサービス名を強調表示し、**Stop the service** をクリックします。

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。

次のファイルのコピーを作成します。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

例、次のコマンド：

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

エントリーにワード old を追加して、ca_certificate ファイルのバックアップを作成します。

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後、cat ca_certificate.crt ファイルの生成時に使用する変数 CA_KEY に内容を追加し、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_crt(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ーを実行し、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。

Linux の場合 `init.d`; たとえば、Centos6 ホストでは：

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

Linux の場合 `systemd`; たとえば、Centos7 ホストでは：

```
systemctl restart postgresql-x
```

6.2 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、`agent.key` および `agent.crt` ファイル（手順 2 および 3 で生成した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ることに注意してください。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. キーファイルが必要な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 ダイアログでサービス名を強調表示し、`Start the service` をクリックします。

第7章 Managing configuration settings

複数の構成ファイルは、起動時に Postgres Enterprise Manager によって読み取られます。ファイルは次のとおりです。

- **config.py** : これはメインの構成ファイルであり、変更しないでください。構成設定のリファレンスとして使用でき、次のファイルのいずれかでオーバーライドされます。
- **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あり、必要に応じてパッケージによって **config.py** と同じディレクトリに作成されます。
- **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固有の設定を変更して、ローカルの設定または標準を満たすように調整す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あり、同じユーザーが作成できます必要に応じて **config.py** としてディレクトリ。

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。