



PEM Administrator's Guide

リリース *8.4.0*

EDB

2022 年 04 月 14 日

目次

第 1 章	PEM overview	3
1.1	PEM architecture	4
1.1.1	PEM サーバー	5
1.1.2	PEM エージェント	8
1.1.3	PEM Web クライアント	9
1.1.4	SQL プロファイラープラグイン	9
1.2	Installing PEM - overview	10
1.3	Using the PEM web interface	10
1.3.1	PEM メニューバー	12
1.3.2	チャート、グラフ、テーブルの制御とカスタマイズ	18
1.3.3	オンラインヘルプと文書	18
第 2 章	Registering a server	21
2.1	サーバーを手動で登録する	21
2.2	自動サーバー検出	32
2.3	pemworker ユーティリティを使用してサーバーを登録する	33
2.3.1	pemworker ユーティリティを使用してサーバーの登録を解除する	36
2.4	接続とバインディングの検証	36
第 3 章	Remote monitoring	39
第 4 章	Defining and monitoring Postgres clusters on AWS	41
4.1	AWS EC2 で実行されている Postgres クラスターのモニタリング	41
4.2	AWS RDS で実行されている Postgres クラスターのモニタリング	42
第 5 章	Managing certificates	43
5.1	SSL 証明書の置き換え	43
5.2	エージェント SSL 証明書の更新	46
第 6 章	Managing configuration settings	49
第 7 章	Managing a PEM server	51
7.1	PEM サーバーとエージェントの起動と停止	51
7.2	監視対象サーバーのリモート起動と停止	52
7.3	Linux での PEM サーバーまたは PEM エージェントの制御	52
7.4	Windows での PEM サーバーまたは PEM エージェントの制御	53
7.5	HTTPD サーバーの制御	54
7.6	pg_hba.conf ファイルの変更	55
7.7	データベースとオブジェクトの作成と保守	56
7.8	PEM 認証の管理	57
7.9	PEM サーバー構成の編集	58

7.10	セキュリティの管理	58
7.10.1	ログインロール	58
7.10.2	グループの役割	60
7.10.3	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	60
7.10.4	チームロールを使用する	62
7.10.5	オブジェクト権限	62
7.11	ジョブ通知の管理	63
7.11.1	ジョブレベルでジョブ通知を設定する	63
7.11.2	エージェントレベルでジョブ通知を設定する	64
7.11.3	サーバーレベルでジョブ通知を設定する	65
7.12	PEM スケジュール済みジョブの管理	65
第 8 章	Managing a PEM agent	75
8.1	エージェント権限	75
8.2	エージェントの構成	77
8.3	エージェントのプロパティ	81

このドキュメントでは、Postgres Enterprise Manager (PEM) の概要を説明します。Postgres Enterprise Manager (PEM) は、PostgreSQL および EDB Advanced Server データベースサーバーの管理、モニタリング、およびチューニングにおいて、データベース管理者、システム設計者、およびパフォーマンスアナリストを支援するために設計されたエンタープライズ管理ツール。PEM は、ほんの一握りから数百台のサーバーまで、単一のコンソールからどこでも管理およびモニタできるように設計されており、データベースのあらゆる側面を完全かつリモートで制御できます。

PEM がサポートするプラットフォームとバージョンについては、edb_lk_asis_1 を参照してください。

このドキュメントでは、PostgreSQL または EDB Postgres Advanced Server データベースを意味する * Postgres *を使用しています。

概 要 registering_server pem_define_aws_instance_connection Managing_certificates Manag-
ing_configuration_settings Managing_pem_server Managing_pem_agent 結論

</ div>

第1章 PEM overview

PEM には、他の PostgreSQL 管理ツールにはない多くの利点があります。

- **一括設計の管理**-PEM はエンタープライズデータベース管理用に設計されており、地理的な境界を越えてラージのサーバーの管理に取り組むように構築されています。グローバルダッシュボードでは、すべてのサーバーのアップ/ダウン/パフォーマンスステータスを一目で把握できます。
- **分散アーキテクチャ**-PEM は、オペレーティングシステムプラットフォームに関係なく、統計情報を収集し、マシン上でリモートで操作を実行する能力を最大化する方法で設計されています。
- **グラフィカル管理**-データベース管理のすべての側面は、グラフィカルインタフェースを介して PEM クライアントで実行できます。サーバーのスタートアップとシャットダウン、構成管理、ストレージとセキュリティ制御、オブジェクト作成、パフォーマンス管理などを単一のコンソールから処理できます。
- **完全な SQLIDE** -PEM には、アドホック SQL クエリ、ストアドプロシージャ/ファンクション開発、およびグラフィカルデバッガを提供する堅牢な SQL 統合開発環境 (IDE) が含まれています。
- **エンタープライズパフォーマンスモニタリング**-PEM は、すべての管理対象データベースサーバーに対してエンタープライズクラスのパフォーマンスモニタリングを提供します。軽量で効率的なエージェントは、各データベースサーバーの動作のすべての側面と各マシンの基になるオペレーティングシステムをモニタし、詳細な統計情報をインタフェース内の簡単にナビゲートされるパフォーマンスページに返します。
- **プロアクティブなアラート管理**-PEM には、24 時間監視されるキーメトリック（メモリ、ストレージなど）ごとにパフォーマンスのしきい値を作成する機能が付属しています。しきい値違反があると、問題の性質と、シチュエーションがサーバーの全体的なパフォーマンスを危険にさらさないようにするために必要なアクションを伝えるアラートが集中ダッシュボードに送信されます。
- **簡素化されたキャパシティプランニング**-パフォーマンス関連のキー統計はすべて自動的に収集され、指定された期間 PEM のリポジトリに保持されます。Capacity Manager ユーティリティを使用すると、さまざまな統計を選択し、時間の経過に伴う傾向分析を実行して、ピークロード期間、ストレージ消費傾向などを把握できます。ツールの予測メカニズムにより、将来のリソース使用量を予測し、それに応じて計画/予算を立てることができます。
- **監査マネージャー** -監査マネージャーは、Advanced Server インスタンスの監査ロギングを構成します。データベースへの接続、データベースからの切断、データベースに対してデータベースされる SQL ステートメントなどのアクティビティをログに記録できます。その後、監査ログダッシュボードを使用して、ログをフィルタリングおよびビューます。
- **ログマネージャー**-ログマネージャーウィザードは、中央のテーブルへの（オプションの）ログ収集を使用してサーバーロギングパラメータを構成します。ウィザードを使用して、ログファイルのローテーション、ログの宛先、エラーメッセージの重大度などのロギング動作の設定を指定します。

サーバーログダッシュボードを使用して、収集されたサーバーログエントリをフィルタリングおよび確認します。

- **SQL ワークロードプロファイリング-PEM** には、1 つ以上のサーバーに対して実行される SQL ステートメントをトレースできる SQL プロファイリングユーティリティが含まれています。SQL プロファイリングは、アドホックまたはスケジュールされた方法で実行できます。キャプチャーされた SQL ステートメントはフィルターに掛けられるため、実行が不十分な SQL ステートメントを簡単に識別して調整できます。SQL ステートメントを Advanced Server の Index Advisor に入力して、各ステートメントを分析し、パフォーマンスを向上させるヘルプに作成する必要がある新しいインデックスに関する推奨事項を作成することもできます。
- **エキスパートデータベース分析-PEM** には Postgres Expert ユーティリティが含まれています。Postgres Expert は、ベストプラクティスの実施目的で選択されたデータベースを分析します。一般的な構成、セキュリティのセットアップなどの領域が調べられます。推奨されるベストプラクティスからの逸脱は、特定の各問題の説明と、物事を正しくするために何をすべきかについてのエキスパートのヘルプとともに報告されます。
- **ストリーミングレプリケーションの監視**-ストリーミングレプリケーションダッシュボードをモニターするか、PEM クライアントのオプションを使用して、レプリカノードをプライマリノードにプロモートさせることができます。
- **セキュアクライアント接続**-PEM は、暗号化された SSH トンネルを介したセキュアクライアント接続をサポートします。フル機能の PEM クライアントには、安全な接続のための接続情報を提供できる SSH トンネル定義ダイアログが含まれています。
- **ワイドプラットフォームサポート**-PEM は、ほとんどのメジャー Linux および Windows プラットフォームをサポートしています。

pem_architecture installed_pem_overview using_pem_web_interface

</div>

1.1 PEM architecture

Postgres Enterprise Manager (PEM) は、単一の GUI インタフェースを介してマルチプルの Postgres サーバーをモニターおよび管理するために設計されたツール。PEM は、インフラストラクチャの次の領域をモニターリングできます。

注釈: 用語 Postgres は、PostgreSQL または EDB Postgres Advanced Server または EDB Postgres Extended のいずれかを指します。

- **ホスト**-1 つまたは複数のサーバー（物理的または仮想）およびそれらのオペレーティングシステム。
- **サーバー** -ホストで実行されている PostgreSQL または EDB Postgres Advanced Server または EDB Postgres Extended（旧称 2ndQPostgres）の 1 つ以上のインスタンス。
- **データベース**-1 つまたは複数のデータベースと、その中のスキーマオブジェクト（テーブル、インデックスなど）。

PEM は、多数の個々のソフトウェアコンポーネントで構成されています。個々のコンポーネントについて以下に説明します。

- **PEM サーバー**-PEM サーバーは、データをモニタリングするためのデータリポジトリとして、およびエージェントとクライアントの両方が接続するサーバーとして使用されます。PEM サーバーは、PostgreSQL のインスタンスと、モニタリングデータをストレージするための関連データベース、および Web サービスを提供するサーバーで構成されます。
- **PEM エージェント**-PEM エージェントは、タスクの実行と、エージェントホストおよび監視対象の Postgres インスタンスから PEM サーバーへの統計のレポートを担当します。単一の PEM エージェントは、1 つ以上のホストに存在する Postgres のマルチプルのインストール済みインスタンスをモニタできます。
- **PEM Web クライアント** -PEM Web インタフェースにより、Postgres サーバーを管理およびモニタし、PEM 拡張機能を利用できます。Web インタフェースソフトウェアは PEM サーバーとともにインストールされ、サポートされている Web ブラウザーを介してアクセスされます。
- **SQL Profiler** - SQL Profiler は、PEM の SQL Profiler ツールで分析されるモニタリングデータとクエリプランをレコードする Postgres サーバープラグインです。これは PEM のオプションコンポーネントですが、SQL プロファイラーツールを使用する Postgres の各インスタンスにプラグインをインストールする必要があります。SQL プロファイラーは、PostgreSQL サーバーまたは Advanced Server (PEM サーバーで管理されているものだけでなく) の EDB ディストリビューションのサポートされているバージョンで使用できます。詳細およびサポートされているバージョンについては、`edb_lk_asis_1` を参照してください。

PEM アーキテクチャ

次のアーキテクチャ図は、PEM サーバー、クライアント、管理対象および非管理対象の Postgres サーバー間の関係を示しています。

1.1.1 PEM サーバー

PEM サーバーは、Postgres のインスタンス、クライアントに Web サービスを提供する Apache Web サーバーのインスタンス、および PEM エージェントで構成されます。PEM は、サーバサイドの暗号化プラグインを使用して認証証明書を生成します。

Postgres (データベースサーバ) のインスタンスと Apache Web サーバー (HTTPD) のインスタンスは、同じホスト上にある場合も、別々のホスト上にある場合もあります。

- **Postgres インスタンス (データベースサーバー)** -これはバックエンドデータベースサーバです。PEM サーバーのリポジトリとして機能する **pem** 名前付けのデータベースをホストします。 **pem** データベースには、監視対象の各ホスト、サーバー、およびデータベースから収集されたメトリックデータを格納するいくつかのスキーマが含まれています。
- **pem** -このスキーマは、PEM アプリケーションのコアです。アプリケーションに必要な構成関数、テーブル、またはビューの定義が含まれています。
- **pemdata** -このスキーマには、監視対象データの現在の snapshot が格納されます。
- **pemhistory** -このスキーマには、監視対象の履歴データが格納されます。

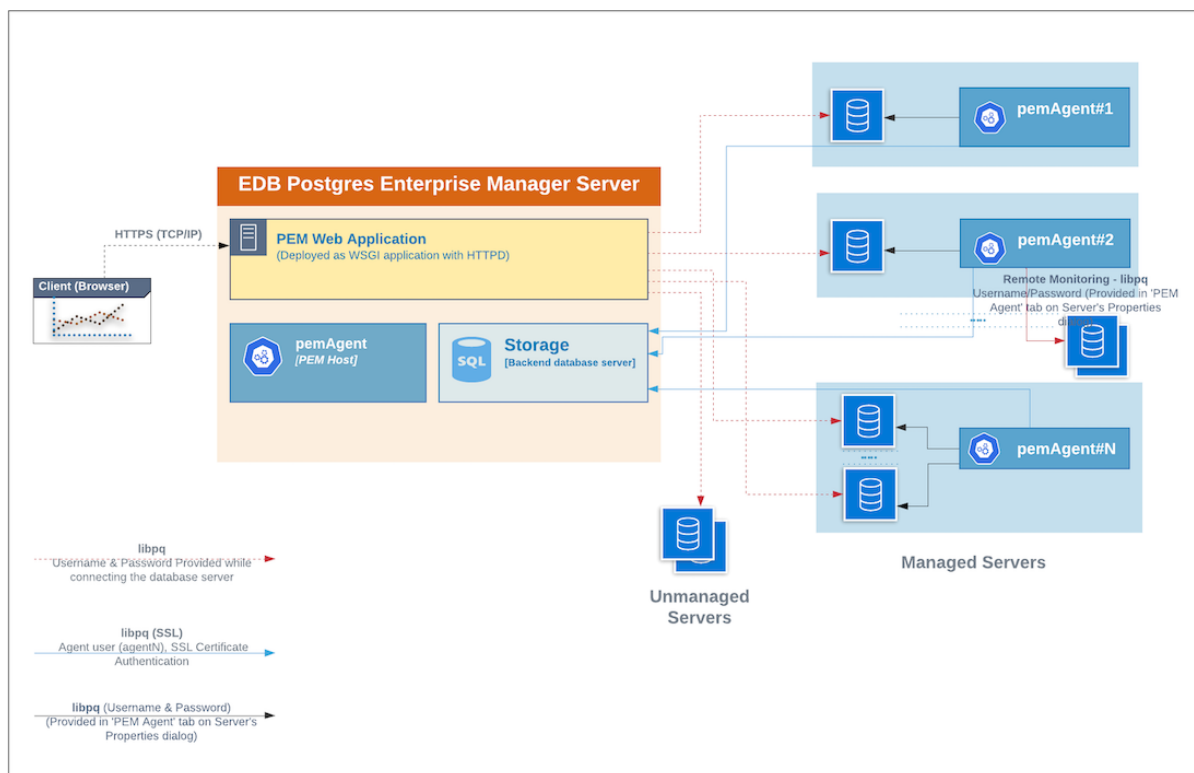


図 1: PEM Architecture

- **ApacheWebServer (httpd)** -PEM Web アプリケーションは、HTTPD を使用して WSGI アプリケーションとしてデプロイされ、クライアントに Web サービスを提供します。次のもので構成されます。
- **Web コンテンツプレゼンテーション**-プレゼンテーションレイヤーは Web アプリケーションによって作成されます（例、ブラウザ、ログインページ、..）。
- **RESTAPI** -REST API を使用すると、他のアプリやサービスと統合できます。
- **データベースサーバーの管理/管理**-データベースサーバーの管理と、CREATE、ALTER、DROP などの管理アクティビティは、管理対象サーバーおよび非管理対象サーバーに対して実行できます。
- **ダッシュボード/チャート生成**-内部的に、Web アプリケーションにはダッシュボードとチャートを生成する機能が含まれています。
- **管理ツール** -Audit Manager、Capacity Manager、Log Manager、Postgres Expert、Postgres Log Analysis Expert、Tuning Wizard が Web アプリケーションで利用可能になりました。-その他のツールは、管理対象サーバーまたは非管理対象サーバーで機能を提供します。
- **SQLProfilerUI の統合** -SQL Profiler は、簡単に分析されたセッションコンテンツのトレースを生成します。
- **クエリエディタ/データビュー**-クエリエディタを使用すると、データをクエリー、編集、およびビューます。
- **デバッガ** -デバッガはクエリのデバッグに役立ちます。
- **パフォーマンス診断** -パフォーマンス診断は、Advanced Server のパフォーマンス分析に役立ちます。

PEM バックエンドデータベースの稼動インスタンスをホストするには、専用のマシンを使用することをお

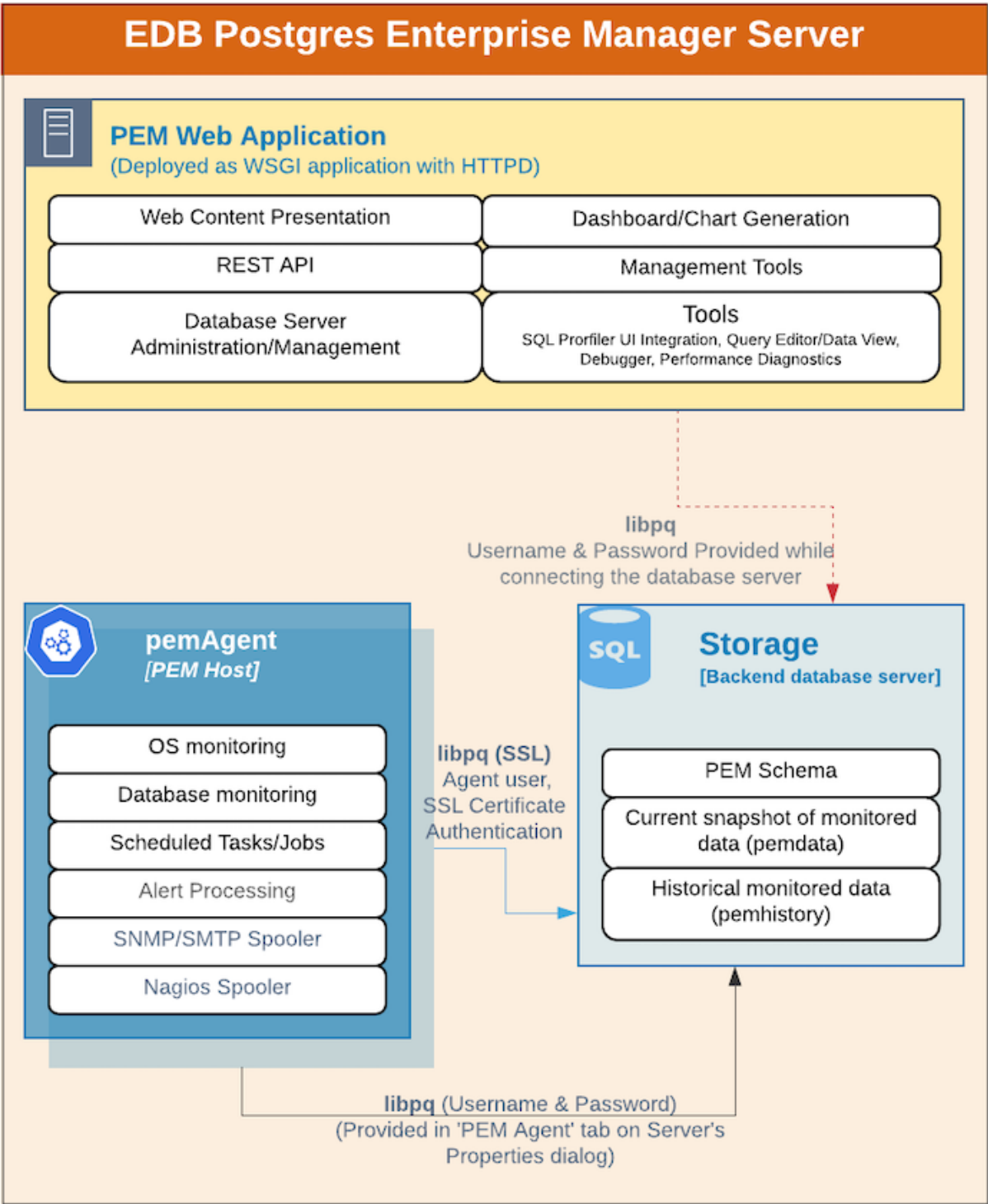


図 2: PEM Server

勧めします。ホストは、監視されているデータベースサーバーの数とサーバーが処理しているワークロードに応じて、高レベルのデータスループットの影響をサブジェクト場合があります。

1.1.2 PEM エージェント

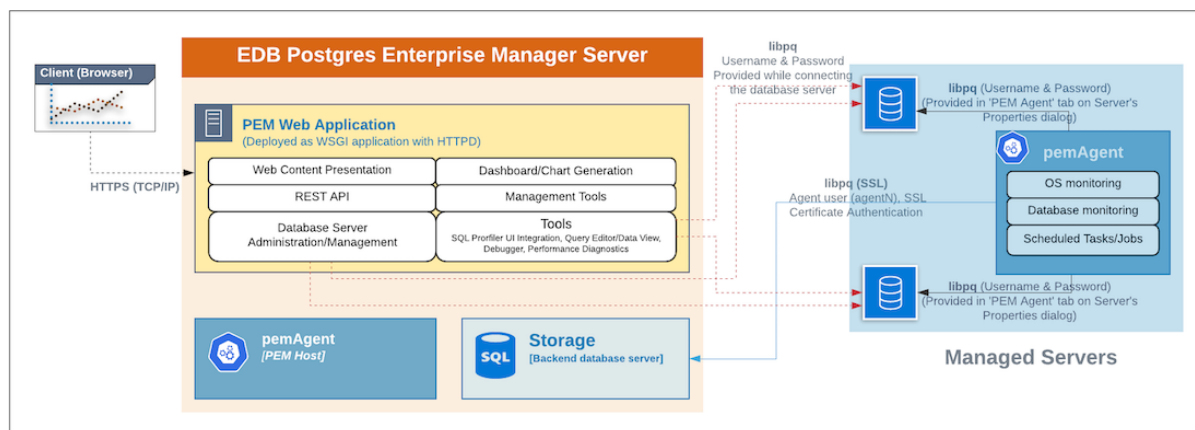


図 3: PEM Agent

PEM エージェントは、マシンとオペレーティングシステム、およびバインドされている各 Postgres インスタンスからのモニタリングデータの収集を担当します。各 PEM エージェントは、1 台の物理的マシンまたは仮想マシンをモニタでき、マルチプルのデータベースサーバーをローカル（同じシステムにインストール、またはリモートにインストール）にモニタリングできます。また、ユーザがスケジュールする他のタスク（サーバーのシャットダウン、SQL プロファイラーのトレース、ユーザー定義のジョブ例）の実行も担当します。

PEM エージェントは、PEM サーバーのインストールとともに PEM サーバーにデフォルトでインストールされます。通常、PEM ホスト上の PEM エージェントと呼ばれます。別に、PEM を使用して監視する Postgres インスタンスをホスティングする他のサーバーに PEM エージェントをインストールすることもできます。

ローカルでモニタリングかリモートで監視するかにかかわらず、PEM エージェントは、SSL 証明書ベースの認証を使用して、PostgreSQL の libpq を使用して PEM サーバーに接続します。Windows の PEM エージェントインストーラと Linux の pemworker CLI は、各エージェントを PEM サーバーに登録し、必要な証明書を生成およびインストールします。

ノートエージェントと PEM サーバーの間には一方方向のトラフィックしかありません。PEM エージェントは常に PEM サーバーに接続します。

PEM エージェントは、監視する各データベースサーバーに接続する必要があります。この接続は TCP/IP 接続（またはオプションで Unix ホストの Unix ドメインソケット）を介して行われ、オプションで SSL を使用できます。ユーザは、監視対象サーバーへの接続と認証を構成する必要があります。

構成が完了すると、各エージェントは、ホストと、監視対象の各データベースサーバーとデータベースに関する統計情報やその他の情報を収集します。各情報は**メトリック**と呼ばれ、**プローブ**によって収集されます。ほとんどのプローブは、効率のために一度にマルチプルのメトリックを収集します。収集されるメトリックの例は次のとおりです。

-ディスク I/O 統計 -ネットワーク統計 -データベースサーババージョン文字列-データベースサーバ構成オプション (GUC) 値 -テーブルアクセス統計-テーブルとインデックスのサイズ

PEM プローブのリストは edb_lk_asis_2 にあります。

デフォルトでは、データベースサーバにバインドされた PEM エージェントは OS /データベースモニタリング統計を収集し、その特定のデータベースサーバのスケジュールされたタスク/ジョブを実行し、PEM サーバーの pem データベースにデータを保存します。

アラート処理、SNMP /SMTP スプーラー、および Nagios スプーラーデータは、PEM サーバー上の pem データベースに格納され、デフォルトで PEM ホスト上の PEM エージェントによって処理されます。ただし、PEM エージェントの SNMP /SMTP および Nagios パラメーターを調整することにより、他の PEM エージェントによる処理を有効にできます。

これらのパラメーターの詳細については、edb_lk_asis_3 を参照してください。

1.1.3 PEM Web クライアント

PEM クライアントは、サポートされているブラウザで実行される Web ベースのアプリケーションです。クライアントの Web インタフェースは PEM サーバーに接続し、管理対象サーバーまたは管理対象外サーバー、およびそれらに存在するデータベースとスキーマを直接管理できます。

クライアントでは、ダッシュボード、Postgres Log Analysis Expert、Capacity Manager などの機能を介してサーバーに記録されたデータを利用する PEM 機能を使用できます。

1.1.4 SQL プロファイラープラグイン

すべてのサーバーに SQL プロファイラープラグインをインストールする必要はありませんが、SQL プロファイラーを使用する各サーバーにプラグインをインストールして構成する必要があります。監視されていない開発サーバーに SQL プロファイラーをインストールして構成することもできます。アドホック使用の場合も、SQL Profiler プラグインを一時的にインストールできます。

プラグインは EDB Postgres Advanced Server ディストリビューションとともにインストールされますが、PostgreSQL で使用するには別途インストールする必要があります。SQL Profiler インストーラは edb_lk_asis_4 から入手できます。

SQL Profiler は、PEM で管理されていないサーバーで使用できますが、スケジュールされたトレースを実行するには、サーバーにプラグインをインストールし、インストールおよび構成済みの PEM エージェントで管理する必要があります。

SQL プロファイラーの使用の詳細については、edb_lk_asis_5 を参照してください。

1.2 Installing PEM - overview

PEM のインストールの詳細な手順については、以下を参照してください。

-edb_lk_asis_1 -edb_lk_asis_2

PEM インストールプロセスに含まれる基本的な手順は次のとおりです。

1. PEM サーバーコンポーネントをインストールします。PEM サーバソフトウェアとバックエンドデータベース (pem 名前付け) は、サポートする httpd サーバーと同じホストに存在する場合も、別のホストに存在する場合があります。PEM サーバーインストーラは、PEM サーバーのホストに PEM エージェントと PEM Web クライアントをインストールします。
2. PEM で管理する追加の物理的マシンまたは仮想マシンをそれぞれ登録します。便宜上、PEM はリモートモニタリングをサポートしています (監視するサーバーと同じホストにエージェントを配置する必要はありません)。リモートエージェントは、監視対象サーバーから利用可能なすべての情報を取得できないことにノートしてください。各サーバホストにエージェントをインストールできます。
3. SQL プロファイラコンポーネントを、SQL キャプチャおよび分析を実行する各 Postgres インスタンスにインストールします。SQL Profiler インストーラは、Postgres のインストール場所のプロンプトを求め、必要なソフトウェアをそのディレクトリに配置します。SQL Profiler プラグインは Advanced Server インスタンスに既にインストールされており、プロファイリングを有効にするための設定のみが必要です。

1.3 Using the PEM web interface

PEM Web インタフェースは、PEM サーバーとともにインストールされます。サーバーのインストールが完了したら、次の場所へ移動して、選択したブラウザで PEM インタフェースを開くことができます。

`https://<ip_address_of_PEM_host>:8443/pem`

`ip_address_of_PEM_host` は、PEM サーバーのホストの IP アドレスを指定します。Postgres Enterprise Manager ログインウィンドウが開きます。

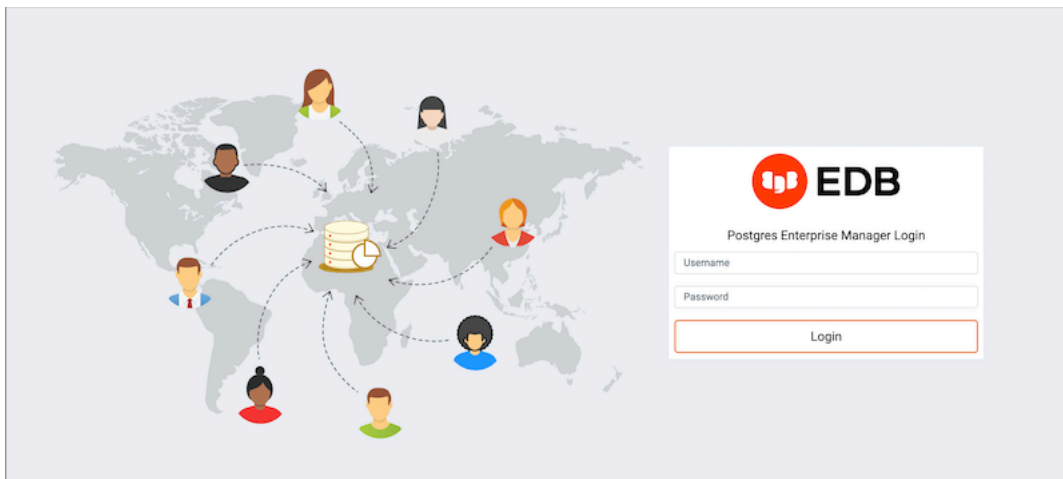


図 4: The PEM Login

Postgres Enterprise Manager Login ウィンドウのフィールドを使用して、PEM サーバーで自分自身を認証します。

- **Username** フィールドに pem データベースユーザの名前を入力します。最初に接続するユーザの場合、これは PEM サーバーのインストール時に指定された名前になります。

-ユーザに関連付けられたパスワードを **Password** フィールドに入力します。

資格情報を入力したら、**Login** をクリックして PEM に接続します。

PEM Web インタフェースが開き、**Global Overview** ダッシュボードが表示されます。

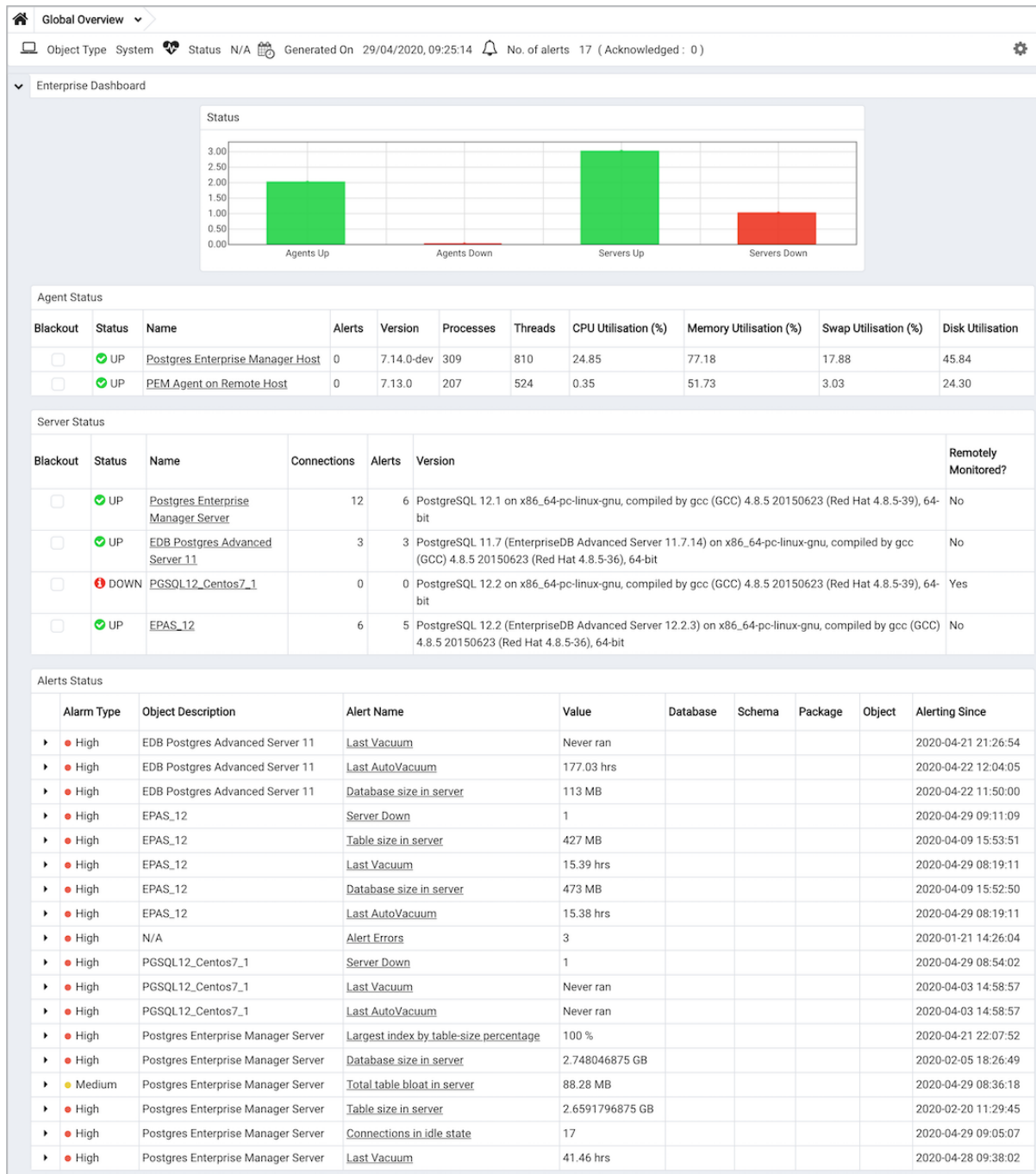


図 5: The Global Overview

Browser ペインには、各サーバーにあるデータベースオブジェクトに関する情報へのアクセスを提供するツリーコントロールが表示されます。ツリーコントロールが展開され、PEM サーバーによって監視されているサーバーとオブジェクトの階層ビューが表示されます。

1.3.1 PEM メニューバー

PEM メニューバーを使用すると、データベースサーバーとそれらのサーバーにあるオブジェクトを管理するために使用できるコマンドと機能にアクセスできます。オプションが無効になっている場合：

- 現在接続しているデータベースサーバは、選択した機能をサポートしていない可能性があります。
- 選択されたメニューオプションは、現在のオブジェクトに対して有効でない場合があります（デザイン）。
- サーバーへの接続に使用したロールには、選択したオブジェクトを変更するための十分な権限がない可能性があります。

PEM Web インタフェースの上部にあるコンテキスト依存メニューにより、環境をカスタマイズし、PEM のエンタープライズ管理機能へのアクセスを提供できます。

[ファイル] メニュー

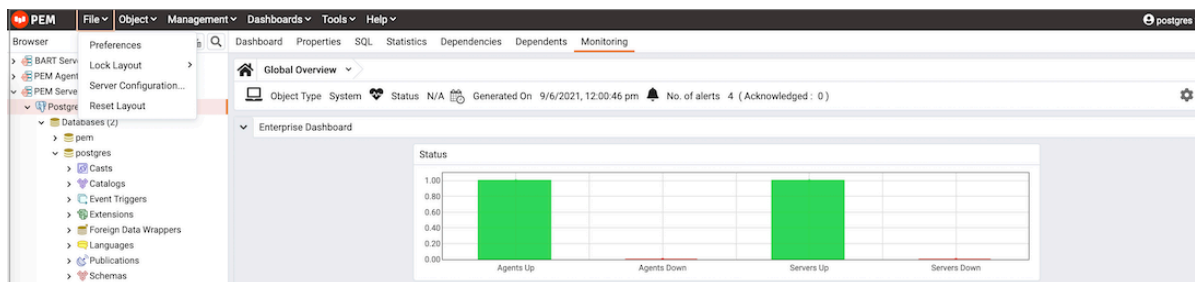


図 6: The File Menu

File メニューを使用して、次のオプションにアクセスします。

Menu Option	Action
Preferences	クリックして [設定] ダイアログを開き、PEM クライアント設定をカスタマイズします。
Lock Layout	クリックしてサブメニューを開き、UI レイアウトをロックするレベルを選択します。
Server Configuration	[サーバー構成] ダイアログをクリックして開き、PEM サーバーの構成設定を更新します。
Reset Layout	ワークスペースパネルが誤ってまたは意図的にポップアウトされた場合、[レイアウトのリセット] を使用してデフォルトにリセットできます。

オブジェクトメニュー

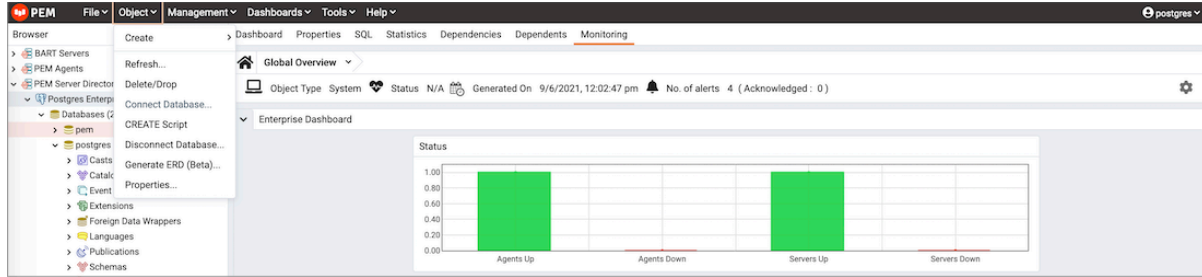


図 7: The Object Menu

Object メニューは状況依存です。Object メニューを使用して、次のオプションにアクセスします。

Menu Option	Action
Create	`Create`をクリックして、コンテキスト依存の選択を提供するコンテキストメニューにアクセスします。
Refresh...	クリックすると、現在選択されているオブジェクトがリフレッシュされます。
Delete/Drop	クリックすると、現在選択されているオブジェクトがサーバーから削除されます。
Connect Server	[サーバーに接続] ダイアログをクリックして開き、サーバーとの接続を確立します。
CREATE Script	クリックしてクエリツールを開き、選択したスクリプトを編集またはビューします。
Disconnect Server	クリックすると、現在選択されているオブジェクトがリフレッシュされます。
BART	クリックすると、BART 設定の削除、BART バックアップの取得、または BART 設定の再検証のオプションを提供するコンテキストメニューにアクセスできます。
Clear Saved Password	データベースサーバのパスワードを保存した場合は、クリックして保存したパスワードをクリアします。パスワードが保存された後にのみ有効になります。
Clear SSH Tunnel Password	ssh トンネルパスワードを保存した場合は、クリックして保存したパスワードをクリアします。パスワードが保存された後にのみ有効になります。
Drop Cascade	クリックすると、現在選択されているオブジェクトとすべての依存オブジェクトがサーバーから削除されます。
Hide	クリックすると、現在選択されているグループが非表示になります。非表示のグループをビューするには、環境設定の「非表示のグループを表示」オプションを有効にします。
Properties...	クリックして、現在選択されているオブジェクトのプロパティを確認または変更します。
Trigger(s)	現在選択されているテーブルの `Disable` または `Enable` トリガーをクリックします。
Truncate	クリックして、テーブルからすべての行を削除する（切り捨て）、またはテーブルとその子テーブルからすべての行を削除する（カスケードを切り捨て）。

次のページに続く

表 2 – 前のページからの続き

Menu Option	Action
View Data	クリックして、データを表示するためのいくつかのオプションを提供するコンテキストメニューにアクセスします。

管理メニュー

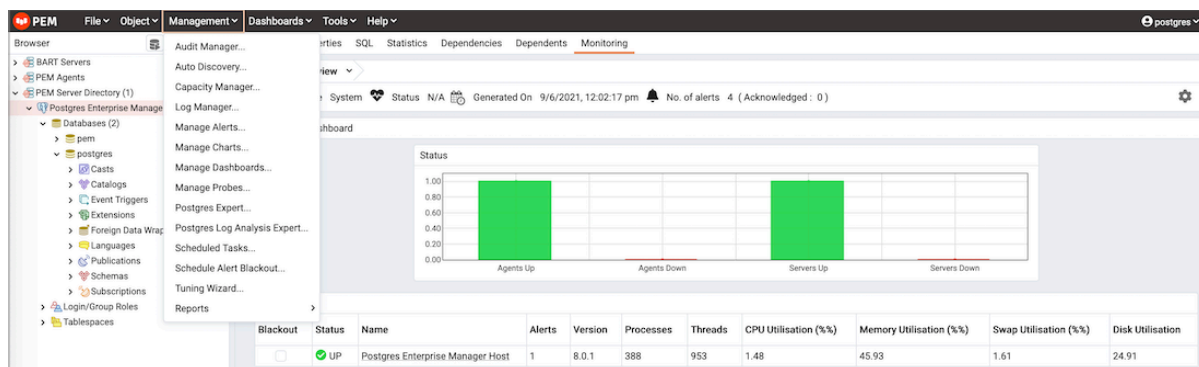


図 8: The Management Menu

Management メニューを使用して、次の PEM 機能にアクセスします。

Menu Option	Action
Audit Manager...	クリックして監査マネージャーを開き、監視対象サーバーで監査を構成します。
Auto Discovery...	クリックして自動検出ダイアログを開き、PEM エージェントに監視対象のデータベースサーバーを見つけてバインドするよう指示します。
Capacity Manager...	クリックして Capacity Manager ダイアログを開き、履歴またはプロジェクトの将来のリソース使用を分析します。
Log Manager...	クリックして [ログマネージャー] ダイアログを開き、サーバーのログ収集を構成します。
Manage Alerts...	クリックして [アラートの管理] タブにアクセスし、アラートの動作を作成または変更します。
Manage Charts...	クリックして [チャートの管理] タブを開き、PEM チャートを作成または変更します。
Manage Dashboards...	クリックして、ダッシュボードの管理ダイアログを開き、VACUUM、ANALYZE、REINDEX、または CLUSTER を実行します。
Manage Probes...	クリックして、プローブの管理ダイアログを開き、VACUUM、ANALYZE、REINDEX、または CLUSTER を実行します。
Postgres Expert...	クリックして Postgres Expert ウィザードを開き、サーバーとデータベースの静的分析を実行します。
Postgres Log Analysis Expert...	クリックして Postgres Log Analysis Expert ダイアログにアクセスし、使用傾向のログファイルの内容を分析します。
Scheduled Tasks	[スケジュールされたタスク] タブをクリックして開き、保留中または最近完了したタスクを確認します。

次のページに続く

表 3 – 前のページからの続き

Menu Option	Action
Tuning Wizard...	クリックして [チューニングウィザード] ダイアログを開き、サーバーの一連のチューニング推奨事項を生成します。
Reports	クリックしてレポートダイアログを開き、サーバーのシステム構成レポートとコア使用率レポートを生成します。

ダッシュボードメニュー

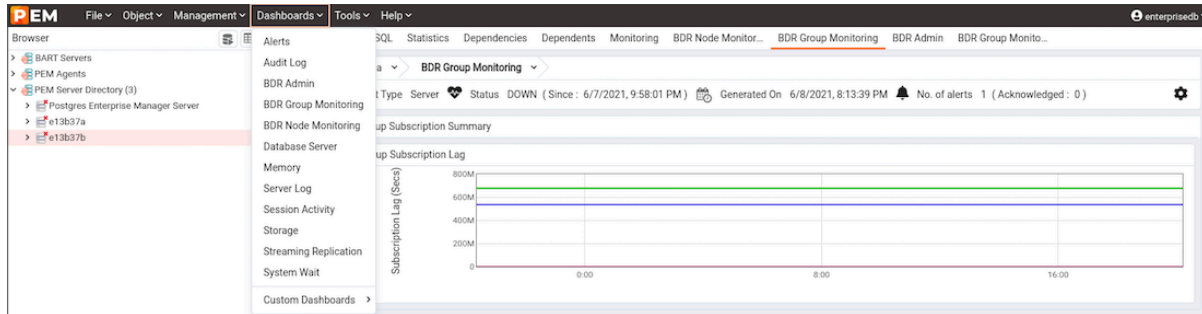


図 9: The Dashboards menu

状況依存の Dashboards メニューを使用して、ダッシュボードにアクセスします。

Option	Action
Alerts	クリックして、選択したノードのアラートダッシュボードを開きます。
Audit Log	クリックして、選択したノードの監査ログ分析ダッシュボードを開きます。
BDR Admin	クリックして、選択したノードの BDR 管理ダッシュボードを開きます。
BDR Group Monitoring	クリックして、選択したノードの BDR グループ監視ダッシュボードを開きます。
BDR Node Monitoring	クリックして、選択したノードの BDR ノード監視ダッシュボードを開きます。
Database Server	クリックして、選択したノードのデータベース分析ダッシュボードを開きます。
I/O Analysis	クリックして、選択したノードの I/O 分析ダッシュボードを開きます。
Memory	クリックして、選択したノードのメモリ分析ダッシュボードを開きます。
Object Activity	クリックして、選択したノードのオブジェクトアクティビティ分析ダッシュボードを開きます。
Operating System	クリックして、選択したノードのオペレーティングシステム分析ダッシュボードを開きます。
Probe Log	クリックして、選択したノードのプロブログ分析ダッシュボードを開きます。
Server Log	クリックして、選択したノードのサーバーログ分析ダッシュボードを開きます。
Session Activity	クリックして、選択したノードのセッションアクティビティ分析ダッシュボードを開きます。

次のページに続く

表 4 – 前のページからの続き

Option	Action
Storage	クリックして、選択したノードのストレージ分析ダッシュボードを開きます。
Streaming Replication	クリックして、選択したノードのストリーミングレプリケーション分析ダッシュボードを開きます。
System Wait	クリックして、選択したノードのシステム待機分析ダッシュボードを開きます。

[ツール] メニュー

Tools メニューのオプションを使用して、次の機能にアクセスします。

Option	Action
Schema Diff	クリックして [スキーマの差分] ダイアログを開き、2つのデータベーススキーマ間でスキーマオブジェクトを比較します。
Search objects	データベース内のデータベースオブジェクトを検索するには、[オブジェクトの検索] ダイアログをクリックして開きます。
Server	クリックして、名前付き復元ポイントの追加、パフォーマンス診断、キューサーバーの起動、キューサーバーのシャットダウン、クラスタープライマリの交換、スイッチオーバー EFM クラスター、SQL プロファイラーなどのさまざまなサーバー関連ツールにアクセスします。
Query Tool	クリックすると、現在選択されているオブジェクトのクエリツールが開きます。
Reload Configuration	サーバーを再起動せずに構成ファイルを更新する場合にクリックします。
Pause replay of WAL	クリックして、WAL ログのリプレイを一時停止します。
Resume replay of WAL	クリックして、WAL ログのリプレイを再開します。
Import/Export...	クリックして [データのインポート/エクスポート] ダイアログを開き、テーブルからデータをインポートまたはエクスポートします。
Maintenance...	VACUUM、ANALYZE、REINDEX、または CLUSTER の [メンテナンス...] ダイアログをクリックして開きます。
Backup...	クリックして [バックアップ...] ダイアログを開き、データベースオブジェクトをバックアップします。
Backup Globals...	クリックして [グローバルのバックアップ...] ダイアログを開き、クラスターオブジェクトをバックアップします。
Backup Server...	[サーバーのバックアップ...] ダイアログをクリックして、サーバーをバックアップします。
Restore...	クリックして [復元] ダイアログにアクセスし、バックアップからデータベースファイルを復元します。
Grant Wizard...	クリックして、Grant Wizard ツールにアクセスします。
Schedule Backup	クリックすると、BART バックアップの [バックアップのスケジュール] ダイアログにアクセスできます。

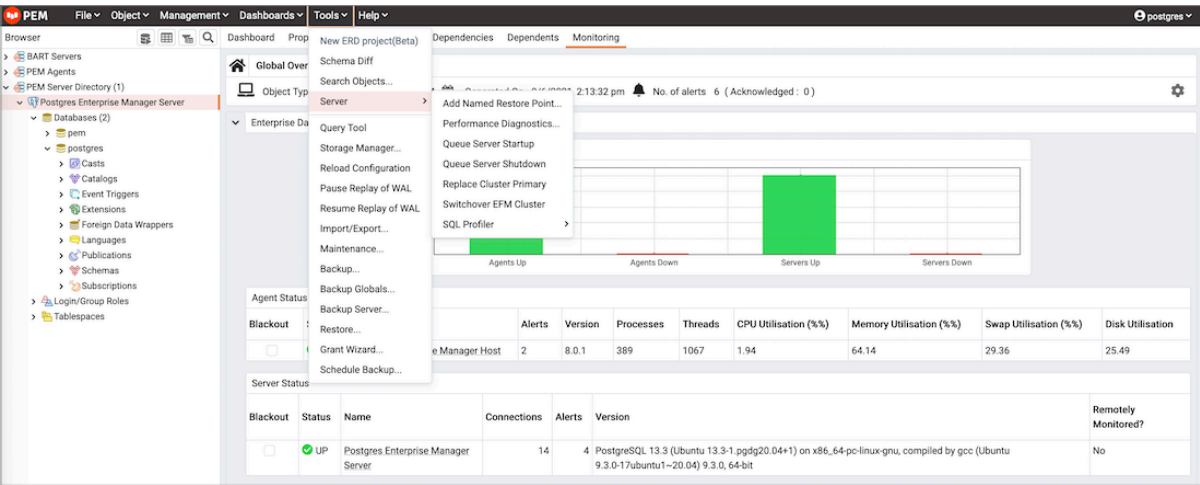


図 10: The Tools menu

ヘルプメニュー

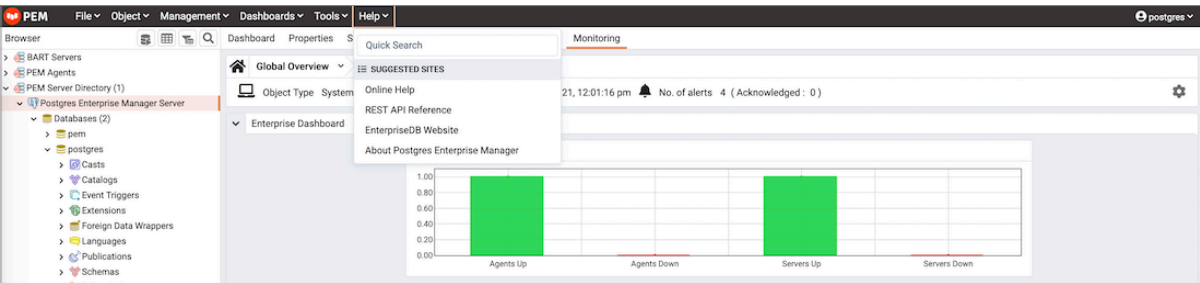


図 11: The Help menu

Help メニューのオプションを使用して、オンラインヘルプドキュメントにアクセスするか、PEM インストールに関する情報を確認します。

Option	Action
Online Help	クリックして Postgres Enterprise Manager の文書を開きます。
REST API Reference	クリックして REST API リファレンスを開きます。
EDB Website	ブラウザウィンドウで EDB ウェブサイトをクリックして開きます。
About Postgres Enterprise Manager	クリックして、Postgres Enterprise Manager のバージョン情報とユーザ情報を見つけます。

1.3.2 チャート、グラフ、テーブルの制御とカスタマイズ

ダッシュボードの各グラフィックの右上隅にあるアイコンを使用して、現在のユーザセッション用に PEM クライアントに表示されるチャート、グラフ、および表を制御およびカスタマイズします。

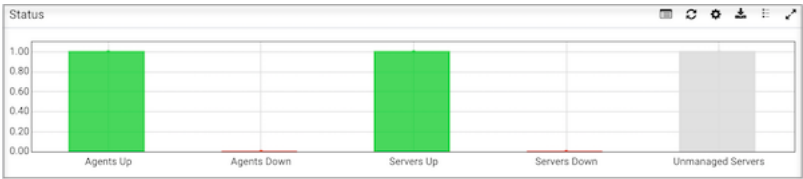


図 12: The PEM Client chart control icons

次の表に、アイコンに関する情報を示します。

Icons	Information
![icon6](../images/lginformation.png)	View information about the chart, graph, or table.
![icon1](../images/lgrefresh.png)	Refresh the content of a chart, graph or table.
![icon3](../images/lgperson.png)	Personalize the chart, graph, or table settings for the current user.
![icon4](../images/lgdownload.png)	Download an image of the chart or graph.
![icon5](../images/lginformation.png)	View the legends that are used in the chart, graph, or table.
![icon2](../images/lgfullscreen.png)	Expand the chart or graph to full-screen.

PEM ダッシュボードに表示されるグラフィックのカスタマイズの詳細については、PEM クライアントのオンラインヘルプを参照してください。

1.3.3 オンラインヘルプと文書

PEM には、ツールの使用を支援するビルトインヘルプが含まれています。PEM のオンラインヘルプにアクセスするには、Help メニューの Help オプションから Online Help を選択します。

Help メニューから EDB ウェブサイトにすばやくアクセスすることもできます。

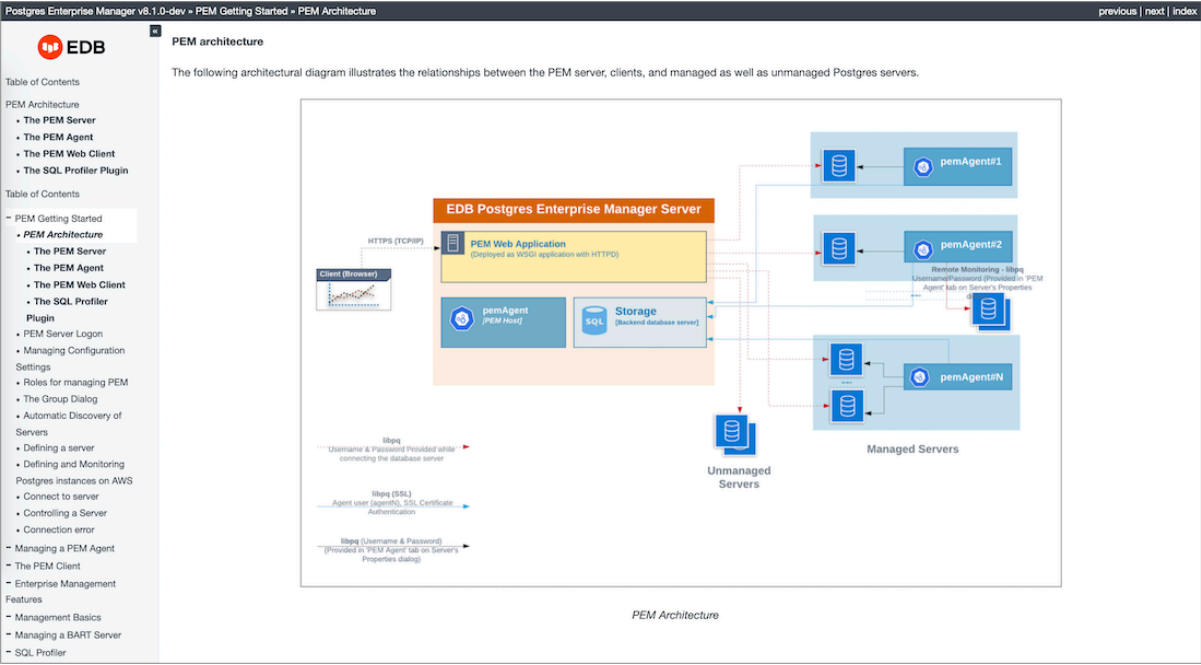


図 13: The PEM online help

第2章 Registering a 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（別のホストに常駐するエージェント）にバインドできますが、エージェントが同じホストに常駐していない場合、インスタンスに関するすべての統計情報にアクセスできません。

2.1 サーバーを手動で登録する

PEM でサーバーを管理またはモニタするには、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。-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。

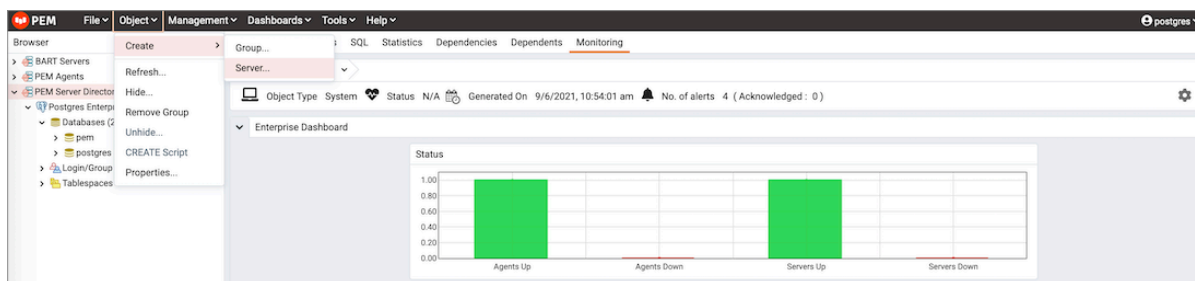


図 1: Accessing the Create – Server dialog

注釈: 接続しようとする前に、登録している Postgres サーバーの `pg_hba.conf` ファイルが PEM クライアントのホストからの接続を許可していることを保証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。

ユーザフレンドリ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、ラージのサーバーをより簡単に管理できヘルプ。例、稼動グループ、テストグループ、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、新しいサーバーが表示され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 ロール名前を指定します。このロールのメンバーである

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

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権限を持つ PEM ユーザーのみが、PEM にログインしたときにこのサーバーを表示します。このフィールドを空白のままにすると、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?**] の横にあるチェックボックスをオンにして、[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。PEM クライアントが、後の接続試行まで指定された接続パラメーターを検証しないようにする場合は、**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ホストの IP アドレス、または完全修飾ドメイン名前を指定します。Unix ベースのシステムでは、アドレスフィールドは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ために空白のままにするか、PostgreSQL ソケットを含む代替パスに設定できます。パスを入力する場合、パスは「/」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** オブジェクトが含まれます (両方ともオプション)。PostgreSQL 8.1 以降では、メンテナンス DB は通常 **postgres** と呼ばれます。以前のバージョンでは **template1** がよく使用されますが、**postgres** データベースを作成してテンプレートデータベースが乱雑にならないようにすることをお勧めします。-監視対象サーバーに Kerberos 認証を使用するには、**Kerberos Authentication** を **Yes** に設定します。デフォルトでは、監視対象サーバーは PEM サーバーと同じ認証メソッドを使用します。監視対象サーバーが Kerberos 認証を使用しない場合は、**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に使用される名前を指定します。

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

- **Password** フィールドに、指定したユーザに関連付けられたパスワードを入力します。
- **Save password?** の横のボックスをオンにして、後で再利用できるように PEM バックエンドデータベースに暗号化フォーマットでパスワードを保存するように PEM に指示します。各パスワードはユーザごと、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、次回データベースサーバに接続します。保存したパスワードを削除するには、まずデータベースサーバを切断し、次に **Object/Context** メニューの **Clear Saved Password** メニューアイテムを使用します。
- **Role** フィールドを使用して、サーバーへの接続後にクライアントが使用する特権が割り当てられている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もの）の許可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でなければなりません。

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 文書を参照してください。

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

プラットフォーム固有のファイルマネージャ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` を置き換えます。SSL 接続

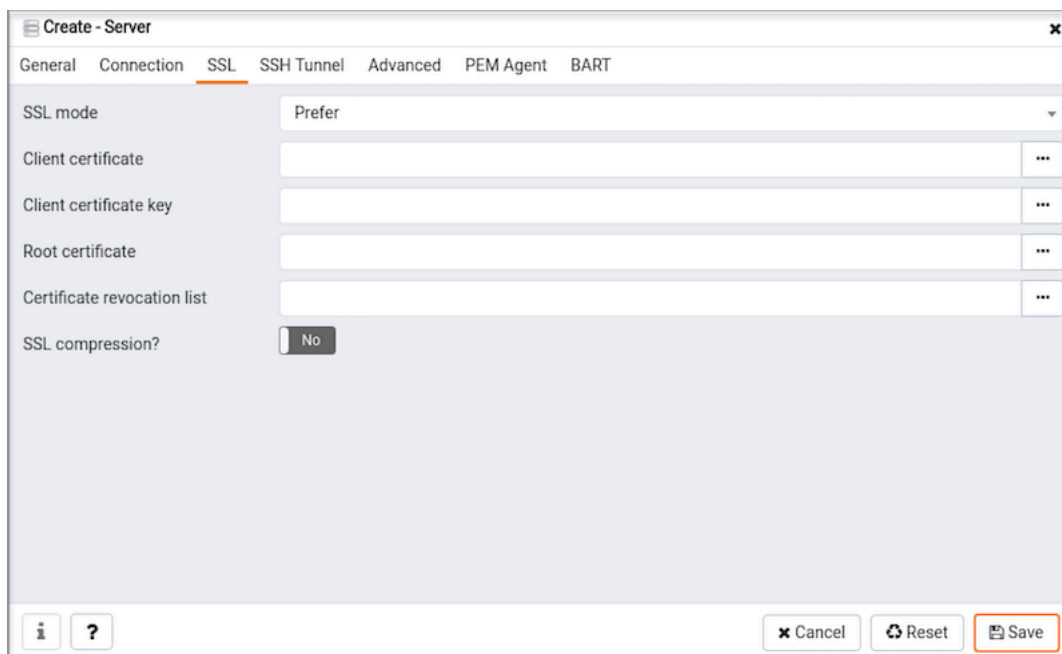


図 4: The SSL tab of the Create – Server dialog

が確立されない場合、このパラメータは無視されます。

- **Client certificate** キーフィールドを使用して、クライアント証明書に使用される秘密キーを含むファイルを指定します。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、このファイルはデフォルトの `~/.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 `<STORAGE_DIR>/<USERNAME>/.postgresql/postgresql.key` を置き換えます。SSL 接続が確立されない場合、このパラメータは無視されます。
- **Root certificate** フィールドを使用して、SSL 認証権限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 接続が確立されない場合、このパラメータは無視されます。
- **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 接続が確立されない場合、このパラメータは無視されます。
- **SSL compression?** が **True** に設定されている場合、SSL 接続を介して送信されるデータは圧縮されます。デフォルト値は **False** です（圧縮は無効です）。SSL 接続が確立されない場合、このパラメータは無視されます。

警告：証明書、プライベート鍵、および失効リストは、PEM サーバプロセスが実行されるユーザアカウントが所有するサーバー上のユーザーごとのファイル格納領域に格納されます。これは、サーバーの管理者がこれらのファイルにアクセスできることを意味します。この機能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。

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、指定したサーバーに接続するときに PEM が SSH トンネルを使用するように指定します。

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

- **Tunnel host** フィールドに、SSH ホストの名前または IP アドレス（クライアント接続が転送される）を指定します。
- **Tunnel port** フィールドで、SSH ホストのポート（クライアント接続が転送される）を指定します。
- **Username** フィールドに SSH ホストのログイン特権を持つユーザの名前を指定します。
- **Authentication** フィールドで SSH ホストに接続するときを使用される認証のタイプを指定します。
- **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。
- **Identity file** を選択して、接続時に PEM がプライベートキーファイルを使用することを指定します。-SSH ホストが認証にプライベートキーファイルを期待している場合、**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要求している場合、**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ワードファイルを使用します。

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 Address1** フィールドにサーバホストの IP アドレスを指定します。
- **DB restriction** フィールドを使用して、edb_lk_asis_1 テーブルに対して使用される SQL 制約を指定し、ツリーコントロールに表示されるデータベースを制限します。例、'live_db'、'test_db' と入力して、PEM ブラウザに live_db および test_db データベースのみを表示するよう指示します。edb_lk_asis_2 に対する制約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、ユーザは接続時にパスワードを入力せずにログイン。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 の文書を参照してください。

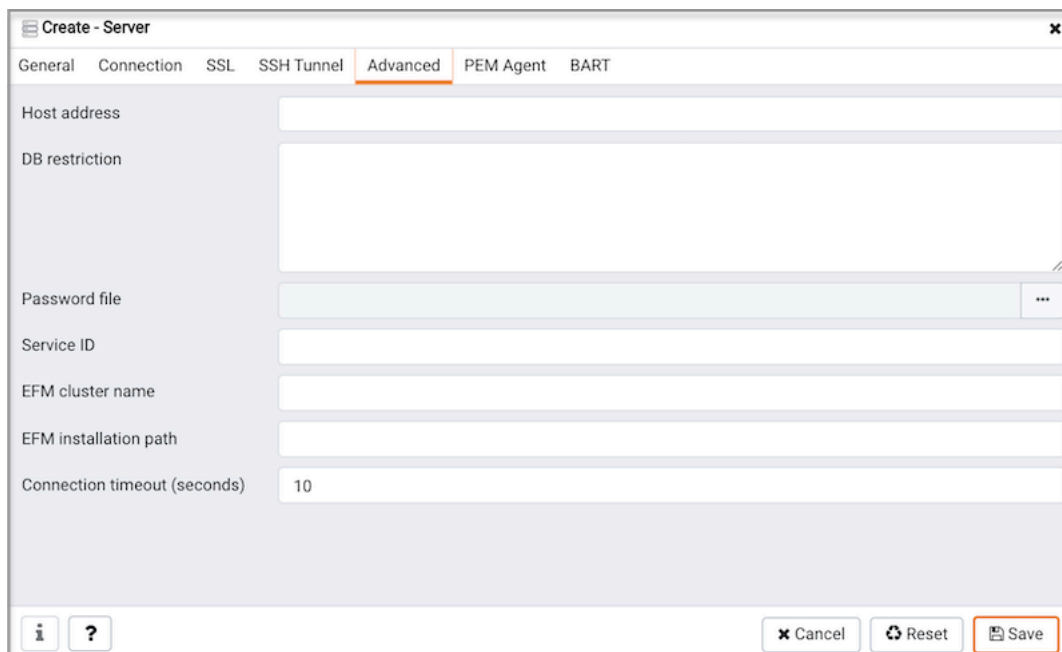


図 6: The Advanced tab of the Create – Server dialog

< <http://www.postgresql.org/docs/current/ 静的/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プロセスを制御するパラメーターを指定します。Enterprise Manager ディレクトリに格納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、サーバーのスタートに使用される init スクリプトの名前は `/etc/init.d` で、サーバーをスタートする `systemd` スクリプトの名前は `systemctl` です。例、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依存します。

-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、ポストマスターサービスを制御できます。サービスの名前を入力します。リモートサーバの場合は、マシン名前の前に追加する必要があります（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 / スタート / 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 x
```

は PostgreSQL データベースサーバのバージョンです。

-サーバーが Failover Manager クラスターのメンバである場合、PEM を使用してクラスターの状態をモニタし、必要に応じてプライマリノードを交換できます。PEM でフェールオーバーマネージャーをモニタ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前を指定します。クラスター名前は、Failover Manager 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。例、クラスタープロパティファイルの名前付けが `efm.properties` の場合、クラスター名前は `efm` です。

-PEM を使用してフェールオーバーマネージャークラスターのステータスをモニタしている場合は、**EFM installation path** フィールドを使用して、フェールオーバーマネージャーバイナリファイルの場所を指定します。デフォルトでは、Failover Manager 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 Failover Manager 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上にないことを示します。リモートモニタリング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およびダッシュボードで使用できず、一部の PEM ユーティリティ（監査マネージャー、容量マネージャー、ログマネージャー、Postgres Expert など）からリモートサーバーにアクセスできません（チューニングウィザード）。
- **Bound agent label** 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マルチプル of Postgres サーバーをモニタできます。
- **Host** フィールドに、データベースサーバーに接続するときにエージェントが使用する 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、`/tmp` などのソケットパスを指定できます。

-サーバーへの接続時に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 暗号化を要求するには。
prefer	SSL 暗号化が使用可能な場合に、サーバーとエージェントの間で SSL 暗号化を使用します。

次のページに続く

表 1 – 前のページからの続き

Mode	Description
allow	サーバーが必要とする場合、接続が 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、認証権限によって登録された証明書を使用してサーバーに認証を要求するため。
verify-full	SSL 暗号化を要求し、サーバーが `trusted` 認証権限によって登録された証明書を使用して認証することを要求するには。

SSL 暗号化の使用の詳細については、edb_lk_asis_3 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前を指定します。
- **User name** フィールドで、サーバーに接続するときにエージェントが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スーパーユーザでない場合、一部の機能が期待どおりに機能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スーパーユーザのスーパーユーザでない場合に機能しない機能のリストについては、edb_lk_asis_4 を参照してください。Kerberos 認証メソッド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を指定します。

Postgres バージョン 10 以降を使用している場合は、**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、edb_lk_asis_5 を参照してください。

- **User name** フィールドに、サーバーに接続するときにエージェントが使用するユーザの名前を指定します。指定されたユーザがデータベーススーパーユーザではない場合、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。Postgres バージョン 10 以上を使用している場合、**pg_monitor** ロールを使用して、スーパーユーザー以外に必要な権限を付与できます。**pg_monitor** ロールの詳細については、edb_lk_asis_6 を参照してください。

-エージェントがサーバーに接続するときに使用するパスワードを **Password** フィールドに指定し、**Confirm password** フィールドにもう一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。例、**.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上に存在し、アクセス可能である必要があります。

Advanced タブで

- **Allow takeover?** スライダーを **Yes** に設定して、サーバーが別のエージェントに引き継がれるように指定します。この機能により、例、サーバーが高可用性フェイルオーバープロセスのパートとして別のホストに移動された場合、エージェントはデータベースサーバのモニタリング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: 除外されたデータベースに対してデータベースレベルのプロローブは実行されませんが、サーバーレベルのプロローブはデータベース統計を収集する場合があります。

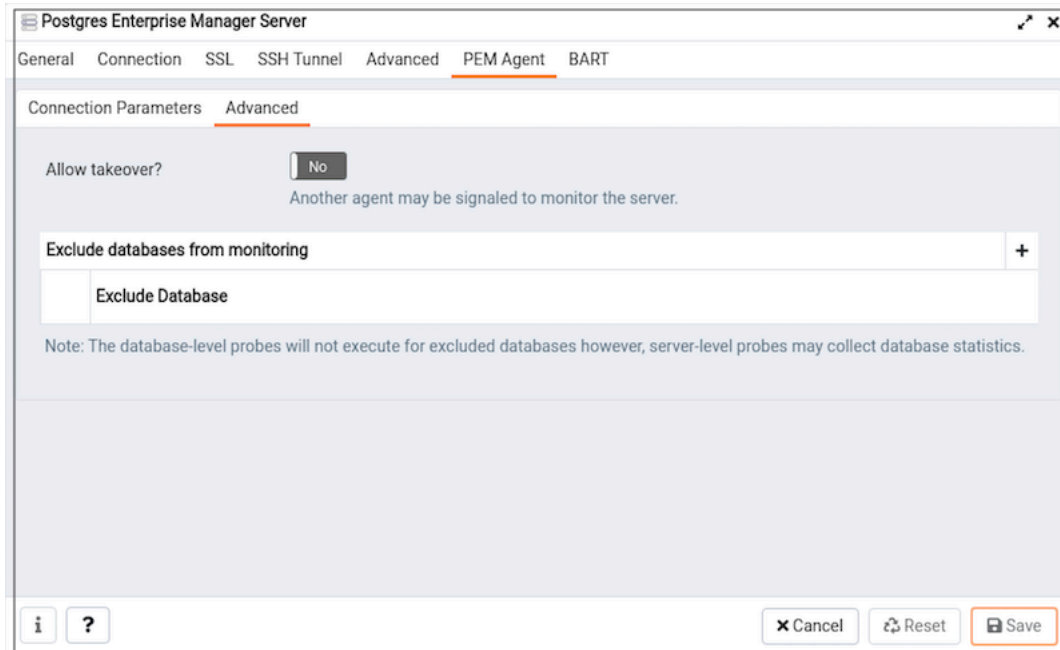


図 8: The PEM Agent Advanced tab of the Create – Server dialog

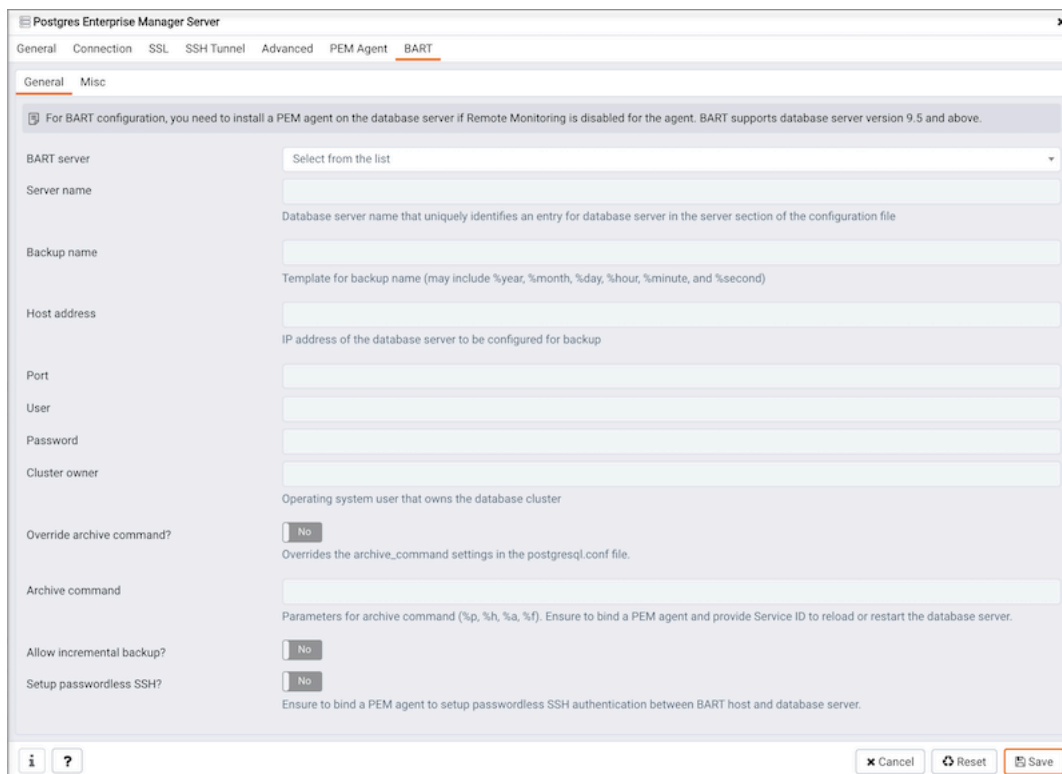


図 9: The Create Server dialog (BART - General tab)

BART タブの下 General タブのフィールドを使用して、PEM サーバーにマップされる 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 コンソールで設定されたすべての BART サーバーは、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名のテンプレートを指定します。バックアップ名前テンプレートを指定しない場合、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 サブコマンドで被参照できます。
- **Host address** フィールドを使用して、バックアップ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のユーザを指定します。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、ユーザはスーパーユーザ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ユーザアカウントを指定します。これは通常、オラクル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フォーマットを指定します。Archive コマンドに入力すると、データベースサーバの **Postgresql.conf** ファイルが上書きされます。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再読み込みされます。
- **Allow incremental backup?** スイッチを使用して、このデータベースサーバで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レス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。パスワードレス 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。

- **Override default configuration?** スイッチを使用して、特定のデータベースサーバ構成で BART サーバー構成を上書きするかどうかを指定します。
- **Xlog** メソッドを使用して、**pg_basebackup** の実行中にトランザクションログを収集する方法を指定します。

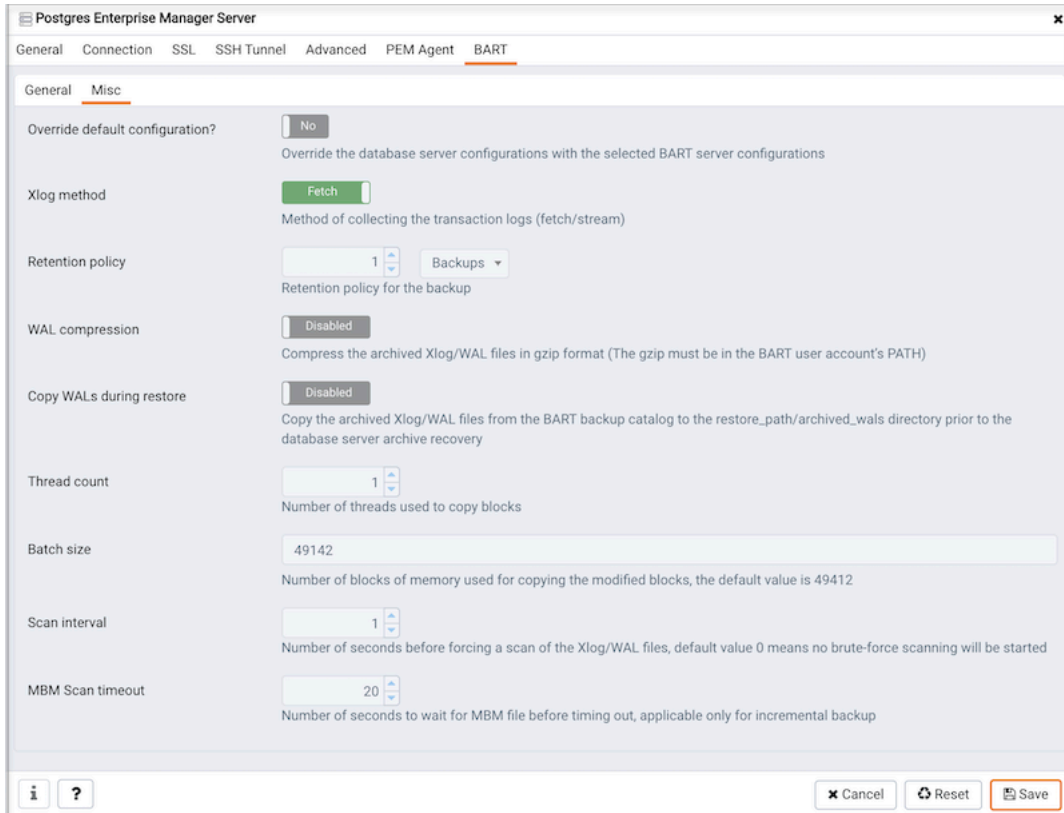


図 10: The Create - Server dialog (BART - Misc tab)

- **Retention policy** フィールドを使用して、バックアップの保存ポリシーを指定します。これにより、アクティブなバックアップが廃止としてマークされ、したがって削除の候補となるタイミングが決定されます。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 フォーマット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、gzip 圧縮プログラムが BART ユーザアカウントの PATH に存在する必要があります。増分バックアップを取得する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 オペレーション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。無効に設定すると、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 ファイルが BART バックアップカタログから直接取得されます。
- **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 ユーティリティを使用して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。これは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、MBM ファイルがタイミングするまで待機する秒数を指定します。これは増分バックアップにのみ適用されます。

サーバーのプロパティをビューするには、**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、**Properties** ダイアログを開く前にサーバーから切断します。

2.2 自動サーバー検出

モニタするサーバーがモニタリング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。

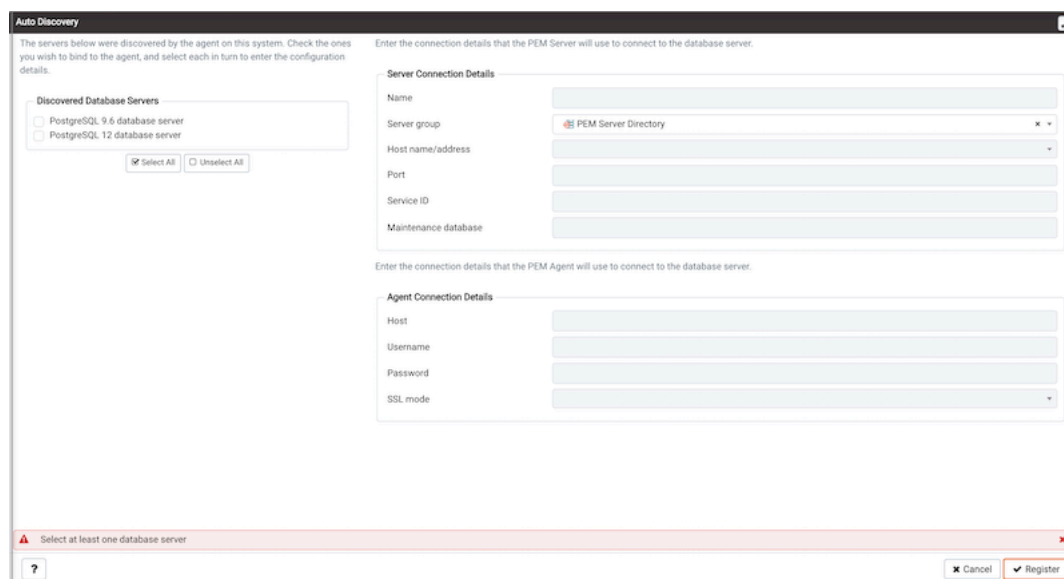


図 11: The PEM Auto Discovery dialog

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。サーバー名前の横にあるボックスをオンにして、**Server Connection Details** ボックスにサーバーに関する情報を表示し、**Agent Connection Details** ボックスにエージェントの接続プロパティを表示します。

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 がモニタするサーバーに関する情報を提供します。

- **Name** フィールドで監視対象サーバーの名前を受け入れるか変更します。指定した名前は、PEM クライアントのツリーコントロールに表示されます。
- **Server group** ドロップダウンリストボックスを使用して、PEM クライアントツリーコントロールでサーバーが表示されるサーバーグループを選択します。
- **Host name/address** フィールドを使用して、監視対象サーバーの IP アドレスを指定します。
- **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、サービス名前を指定する必要があることにノートしてください。-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。インストールの **Maintenance database** フィールドのコンテンツを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、指定したユーザ名前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。

モニタリングにバインドしているサーバーの接続プロパティの指定が終了したら、**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。

2.3 pemworker ユーティリティを使用してサーバーを登録する

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

Linux ホストでは、次のコマンドを使用します。

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

Windows ホストでは、次のコマンドを使用します。

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

pemworker ユーティリティを呼び出すときに、コマンド文字列にコマンドラインオプションを追加します。各オプションの後には、対応する値が続く必要があります。

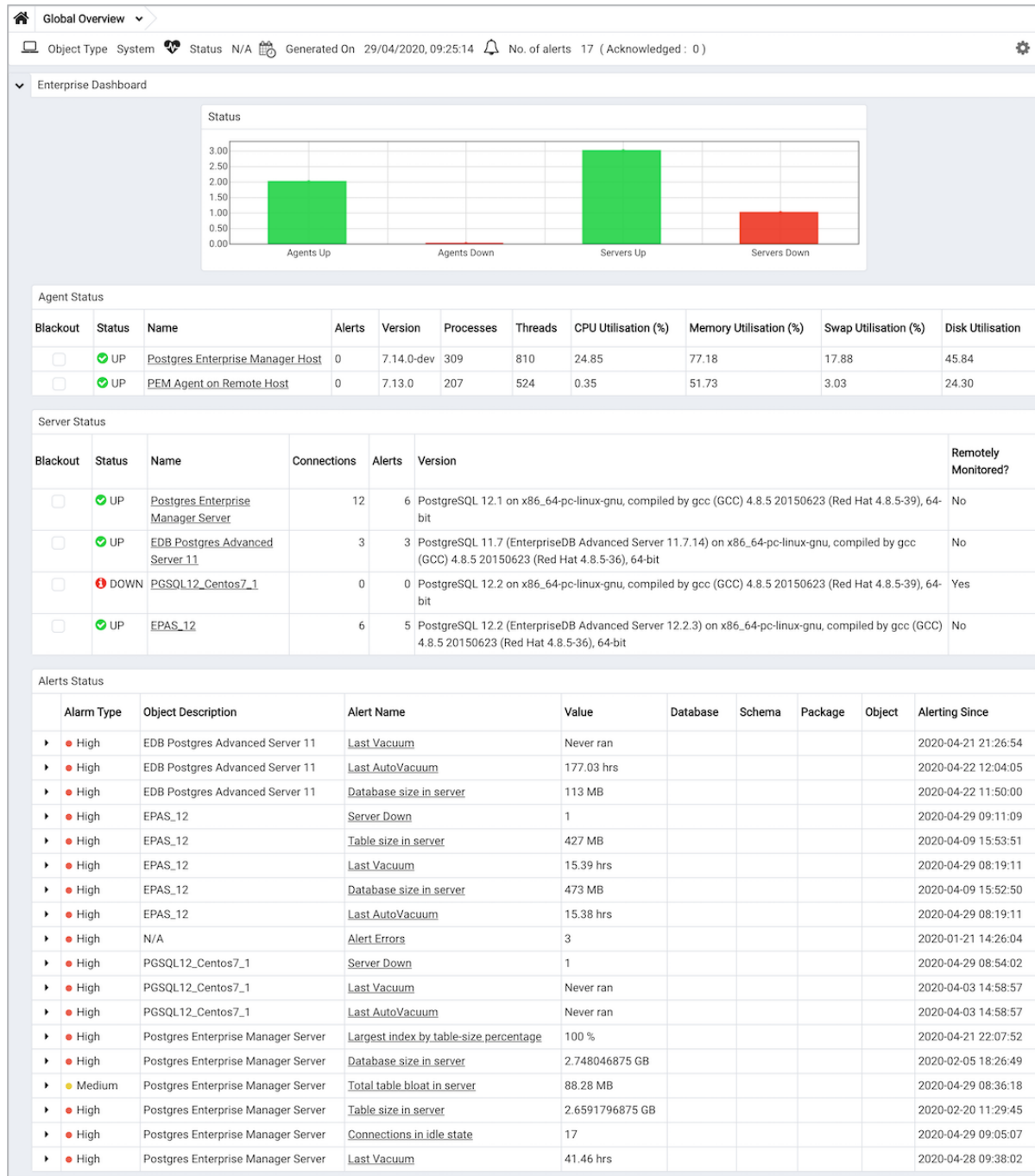


図 12: The registered server

Option	Description
<code>--pem-user</code>	PEM 管理ユーザの名前を指定します。必須です。
<code>--server-addr</code>	サーバホストの IP アドレス、または完全修飾ドメイン名前を指定します。Unix ベースのシステムでは、アドレスフィールドは、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ために空白のままにするか、PostgreSQL ソケットを含む代替パスに設定できます。パスを入力する場合、パスは/で始まる必要があります。必須です。
<code>--server-port</code>	ホストのポート番号を指定します。必須です。
<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。必須です。
<code>--server-user</code>	サーバーのモニタリング時にエージェントが使用するユーザの名前を指定します。必須です。
<code>--server-service-name</code>	登録されるサーバーでの操作 (STOP、START、RESTART など) を制御するデータベースサービスの名前を指定します。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。リモートモニタリングが有効 (yes) の場合、監視対象リモートサーバのエージェントレベルの統計はカスタムチャートおよび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 and Tuning Wizard)。必須です。
<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しているホストの名前を指定します。
<code>--asb-host-port</code>	データベースに接続するとき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データベースユーザ名前を指定します。
<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。サポートされる値には、 <code>prefer</code> 、 <code>require</code> 、 <code>disable</code> 、 <code>verify-CA</code> 、 <code>verify-full</code> が含まれます。
<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されるグループロールの名前を指定します。
<code>--owner</code>	監視対象サーバーを所有するロールの名前を指定します。
<code>--display-name <name></code>	監視対象データベースサーバの表示名前を指定します。デフォルトはシステムのホスト名です。
<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。

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

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

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

Linux ホストでは、次のコマンドを使用します。

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

Windows ホストでは、次のコマンドを使用します。

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

pemworker ユーティリティを呼び出すときに、コマンド文字列にコマンドラインオプションを追加します。各オプションの後には、対応する値が続く必要があります。

Option	Description
<code>--pem-user</code>	PEM 管理ユーザの名前を指定します。必須です。
<code>--server-addr</code>	サーバホストの IP アドレス、または完全修飾ドメイン名前を指定します。Unix ベースのシステムでは、アドレスフィールドは、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ために空白のままにするか、PostgreSQL ソケットを含む代替パスに設定できます。パスを入力する場合、パスは/で始まる必要があります。必須です。
<code>--server-port</code>	ホストのポート番号を指定します。必須です。
<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** を使用して PEM サーバーのパスワードを提供し、**pemworker** が PEM 管理ユーザとして接続できるようにします。

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

2.4 接続とバインディングの検証

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き、**Global Overview** ダッシュボードは、サーバーリストに「不明」のステータスで新しいサーバーを表示する場合があります。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多数のプロープを実行してサーバーを検査する必要がありますが、ネットワークの可用性によっては完了するまでに数分かかる場合があります。

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 分経っても **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

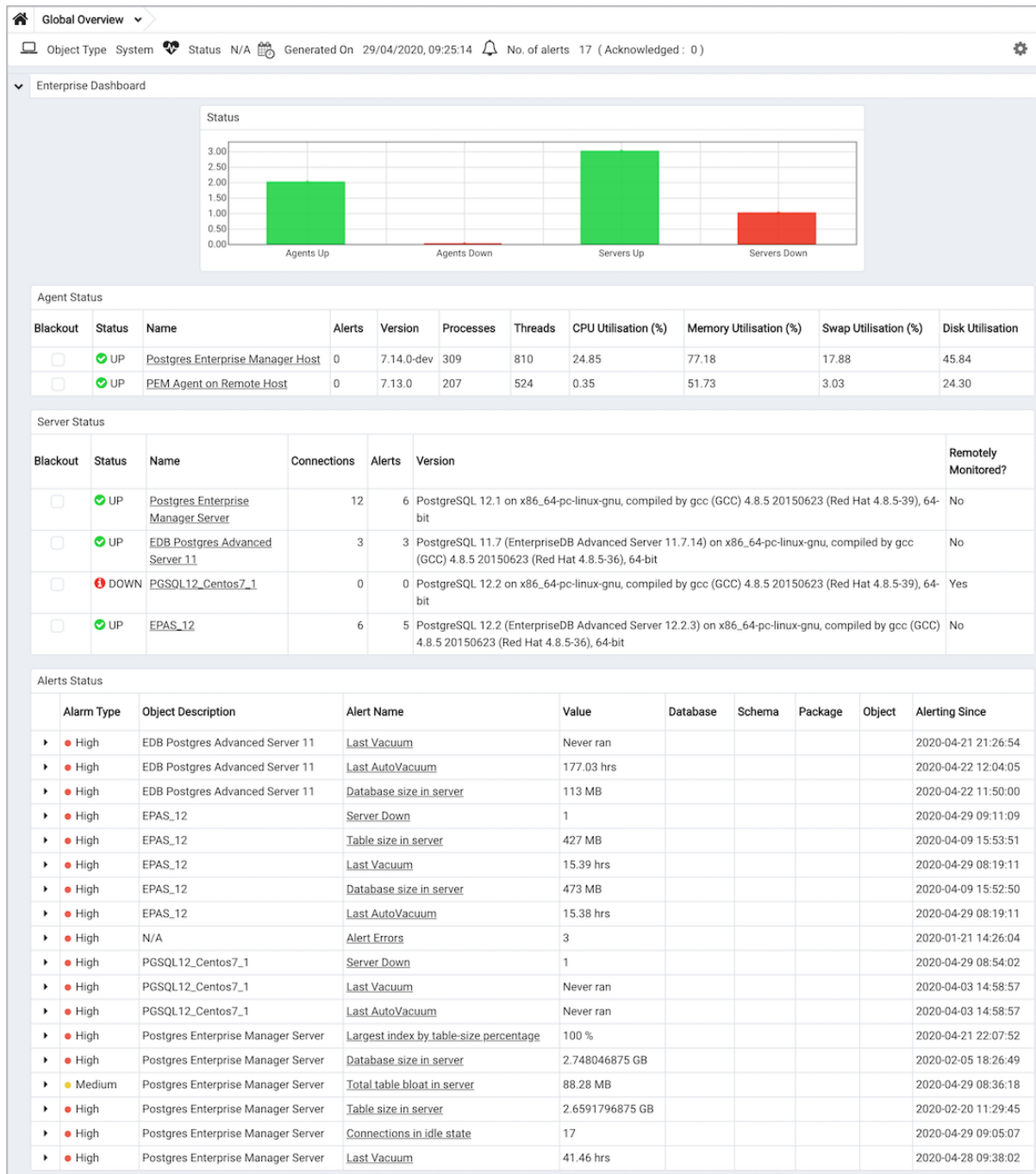


図 13: The Global Overview dashboard

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。

第3章 Remote monitoring

リモートモニタリングは、異なるホストにある PEM エージェントを使用して Postgres クラスターをモニタリングします。

PEM で Postgres クラスターをリモートでモニタするには、クラスターを PEM に登録し、PEM エージェントをバインドする必要があります。詳細については、`edb_ik_asis_1` を参照してください。

次のシナリオでは、PEM を使用したリモートモニタリングが必要です。

`-edb_ik_asis_2 -edb_ik_asis_3`

PEM リモートモニタリングは以下をサポートします。

Feature Name	Remote Monitoring Supported?	Comments
[Manage Charts](../pem_ent_feat/05_performance_monitoring_and_management/#using-the-manage-charts-tab)	Yes	NA
[System Reports](../pem_ent_feat/12_reports/#system-configuration-report)	Yes	NA
[Capacity Manager](../pem_ent_feat/06_capacity_manager)	Limited	Postgres クラスターとオペレーティングシステムのメトリック間に相関関係はありません。
[Manage Alerts](../pem_ent_feat/05_performance_monitoring_and_management/#alerts)	Limited	Postgres クラスターでアラートスクリプトを実行するのではなく、バインドされた PEM エージェントが実行されているマシンで実行されます。
[Manage Dashboards](../pem_ent_feat/05_performance_monitoring_and_management/#dashboards-to-view-performance-information)	Limited	一部のダッシュボードでは、完全なデータを表示できない場合があります。例、Postgres クラスターが実行されているオペレーティングシステム情報は利用できないため表示されません。
[Manage Probes](../pem_ent_feat/05_performance_monitoring_and_management/#probes)	Limited	PEM プローブの一部は情報を結果ず、一部の機能を影響を受ける可能性があります。プローブ機能の詳細については、[PEM Agent Privileges](../pem_agent/03_managing_pem_agent/#agent-privileges) を参照してください。
[Postgres Expert](../pem_ent_feat/11_postgres_expert/)	Ex-Limited	Postgres Expert は、オペレーティングシステム情報が利用できないため、部分的な情報を提供します。

次のページに続く

表 1 – 前のページからの続き

Feature Name	Remote Monitoring Supported?	Comments
[Scheduled Tasks](../pem_online_help/04_toc_pem_features/15_pem_features/scheduled_tasks/)	Limited	スケジュールされたタスクは、Postgres クラスターでのみ機能します。タスクはリモートエージェントで実行されます。
[Core Usage Reports](../pem_ent_feat/12_reports/#core-usage-report)	Limited	コア使用状況レポートには完全な情報は表示されません。例、プラットフォーム、コアの数、合計 RAM は表示されません。
[Audit Manager](../pem_ent_feat/07_audit_manager)	No	NA
[Log Manager](../pem_ent_feat/08_log_manager)	No	NA
[Postgres Log Analysis Expert](../pem_ent_feat/08_log_manager/#postgres-log-analysis-expert)	No	NA
[Tuning Wizard](../pem_ent_feat/10_tuning_wizard/)	No	NA

第4章 Defining and monitoring Postgres clusters on AWS

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シナリオは2つあります。以下をモニタできます。

-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

4.1 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後、PEM サーバーを使用してクラスターを登録およびモニタできます。現在、次のシナリオがサポートされています。

Postgres クラスターと PEM エージェントは同じ AWS EC2 で実行され、PEM サーバーはローカルマシンで実行されます。-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、および異なる AWS EC2 で実行されている PEM サーバー

注釈: 最初の2つのシナリオでは、AWS EC2 インスタンスが`pem`データベースにアクセスできるように、AWS EC2 で VPN を設定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 サーバーとは異なるホスト上にあるため、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に、AWS EC2 インスタンスのホスト名を使用 make ようにしてください。PEM エージェントの登録の詳細については、`edb_ik_asis_1` 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 サーバーに追加しながら PEM エージェントを Postgres クラスターにバインドしたら、PEM を使用して Postgres クラスターをモニタできます。

4.2 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 のプロンプトが表示されたら、PostgreSQL を選択します。次の手順については、edb_lk_asis_2 を参照してください。

第5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間もなくエキスパイアと、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` ファイルに保持され保証) に追加され、証明書を使用して PEM サーバーに接続できないようになります。

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。

5.1 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、次にエージェントインストーラー）を呼び出します。次に：

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

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

systemd を使用する Linux の場合、たとえば：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付けは Postgres Enterprise Manager Agent です。Services ダイアログでサービス名前を強調表示し、**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 `data` ディレクトリに保存されます。例、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 データベースバージョンです。

次のファイルのコピーを作成し、名前を一意に `make` に各ファイルに拡張子を追加します。

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

例、コマンド：

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

エントリにワード `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` ファイルを生成し、`ca_key.key` ファイルの特権を変更するときに使用する変数 `CA_KEY` の内容を `cat` しファイル。

```
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 クエリーを一意の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` ファイルの権限を変更し、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ファイル。

```
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 ファイルを生成し、server.key ファイルの特権を変更するときに使用する変数 SSL_KEY の内容を cat しファイル。

```
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、最初に一意の名前を持つ一時ファイルに SQL クエリーを配置します。

```
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 で `init.d` を使用する場合; 例、Centos6 ホストの場合：

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

Linux で `systemd` を使用する場合; 例、Centos7 ホストの場合：

```
systemctl restart postgresql-x
```

5.2 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、以下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。-エージェントをリスタートします。

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

PEM エージェントキーファイルペアを生成するには：

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

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

-Linux では、`/etc/postgres-reg.ini` ファイルの `PEMagent` セクションで、キーと証明書のエージェント識別子と場所を見つけることもできます。

Windows では、情報はレジストリに保存されます：

-64 ビット Windows インストールでは、次を確認します。

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

-32 ビット Windows インストールでは、次を確認します。

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

2. キーファイルを必要とするエージェントを特定した後、各エージェントに対して `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
```

ここで、*\$ID*はエージェントのエージェント番号です（psql コマンドラインから取得）。

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で置き換えます。

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 エージェントをスタートできます。PEM エージェントサービスの名前付けは Postgres Enterprise Manager Agent です。Services ダイアログでサービス名前を強調表示し、Start the service をクリックします。

第6章 Managing configuration settings

複数の構成ファイルは、スタートアップに 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。これは、構成設定のリファレンスとして使用できます。これは、次のファイルのいずれかでオーバーライドできます。
- **config_distro.py** : このファイルは **config.py** の後に読み込まれ、パッケージャーが Postgres Enterprise Manager のディストリビューションに必要な設定を変更するためのものです。これには通常、特定の経路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 **config.py** と同じディレクトリにパッケージャーが作成できます。
- **config_local.py** : このファイルは **config_distro.py** の後に読み込まれ、エンドユーザーがローカルの設定や標準を満たすために調整するデフォルトまたはパッケージ固有の設定を変更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、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、リファレンス用に PEM オンラインヘルプに含まれています。

第7章 Managing a PEM server

</div>

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、プロキシサーバーなどを使用するように PEM を変更します。

7.1 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 サーバインスタンスが起動、停止、または再起動したときに起動、停止、および再起動します。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、プラットフォームおよび OS バージョンによって異なります。

PEM エージェントは、`pemagent` 名前付け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ナビゲートし、サービス名前を強調表示します。

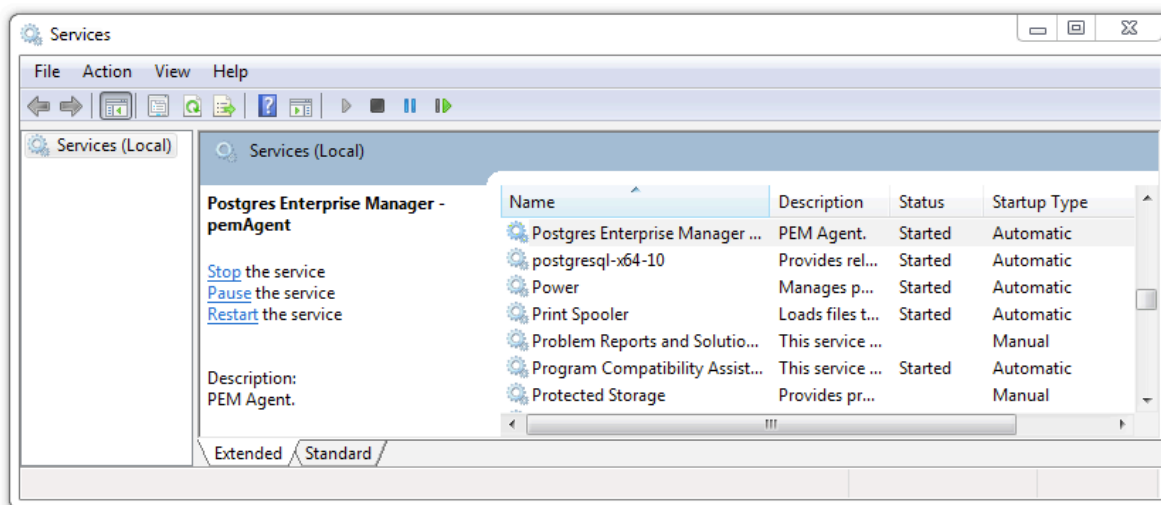


図 1: The PEM service in the Windows Services window

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（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ノートしてください。サービスの制御の詳細については、EDB ウェブサイトから入手可能な edb_ik_asis_1 を参照してください。

7.2 監視対象サーバーのリモート起動と停止

PEM では、PEM クライアントを使用して管理対象サーバーインスタンスをスタートアップ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を PEM 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 Advanced Server の場合、サービス名前は edb-as-<x> または ppas-<x> です。PostgreSQL の場合、サービス名前は postgresql-<x> です。

x はサーバ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前をハイライトし、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、サーバーをスタートまたは停止できます。

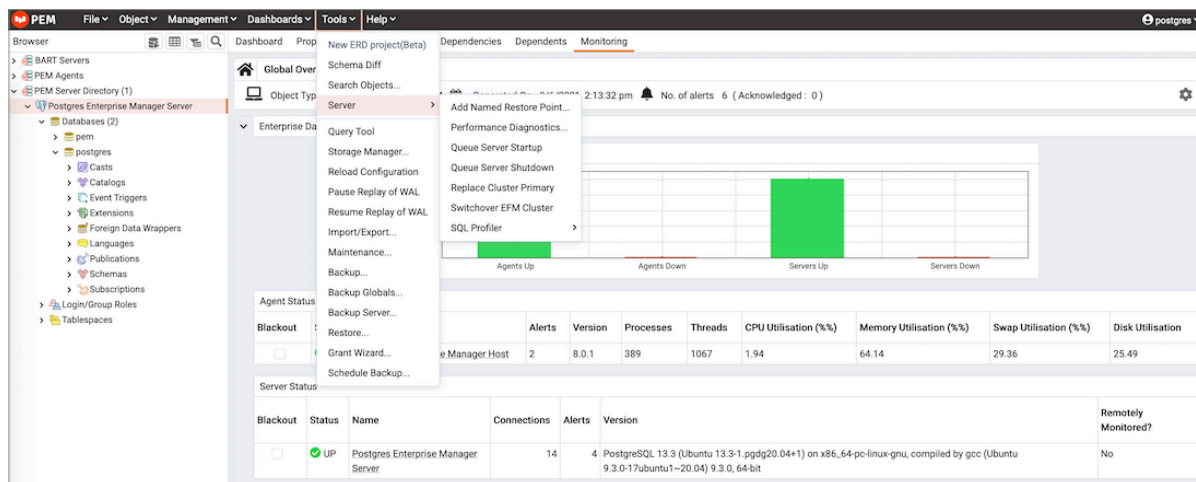


図 2: The Tools menu of a managed server

7.3 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、次を制御するサービススクリプトの名前：

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

x はサーバ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>
```

どこで：

`service_name` はサービスの名前です。

`action` は、サービスによって実行されるアクションを指定します。指定してください：

- `start` はサービスをスタートします。-サービスを停止するには、`stop`。
- `restart`。サービスを停止してからスタートしサービス。
- `status`。サービスのステータスを確認します。

7.4 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が含まれています。以下を制御するサービスの登録名前：

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

`x` はサーバ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。

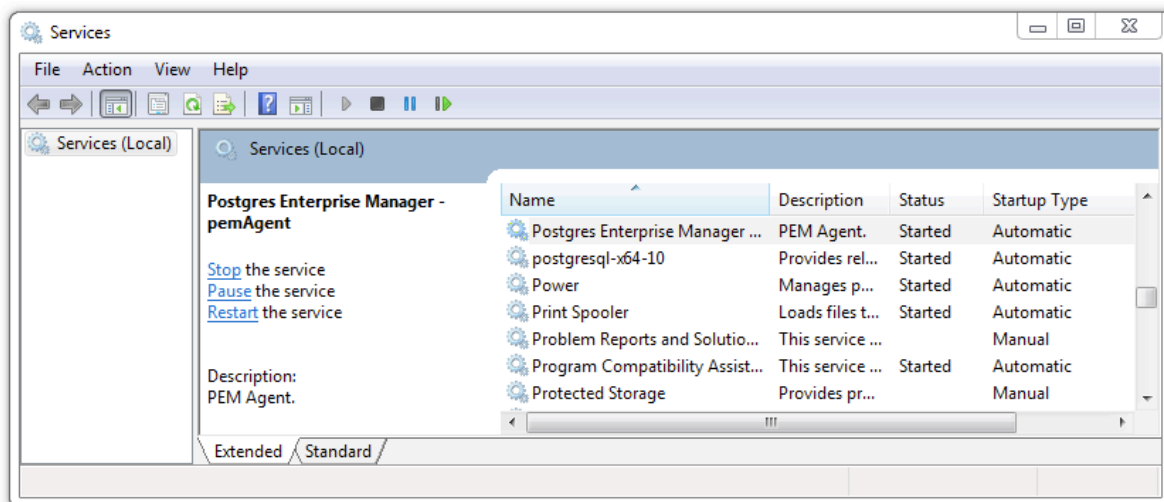


図 3: The Windows Services window

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

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、サービスをスタートします。

7.5 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スタートするには、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後、リストをスクロールして PEM HTTPD サービスを見つけます。

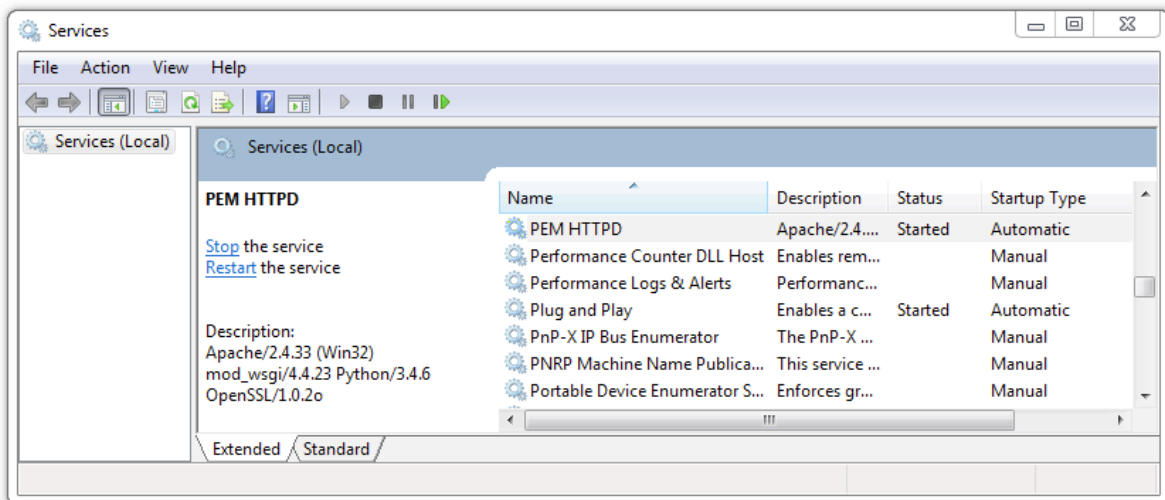


図 4: The PEM HTTPD Windows service

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 をスタートします。

7.6 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 バッキングデータベースの pg_hba.conf ファイルの先頭に追加されます。

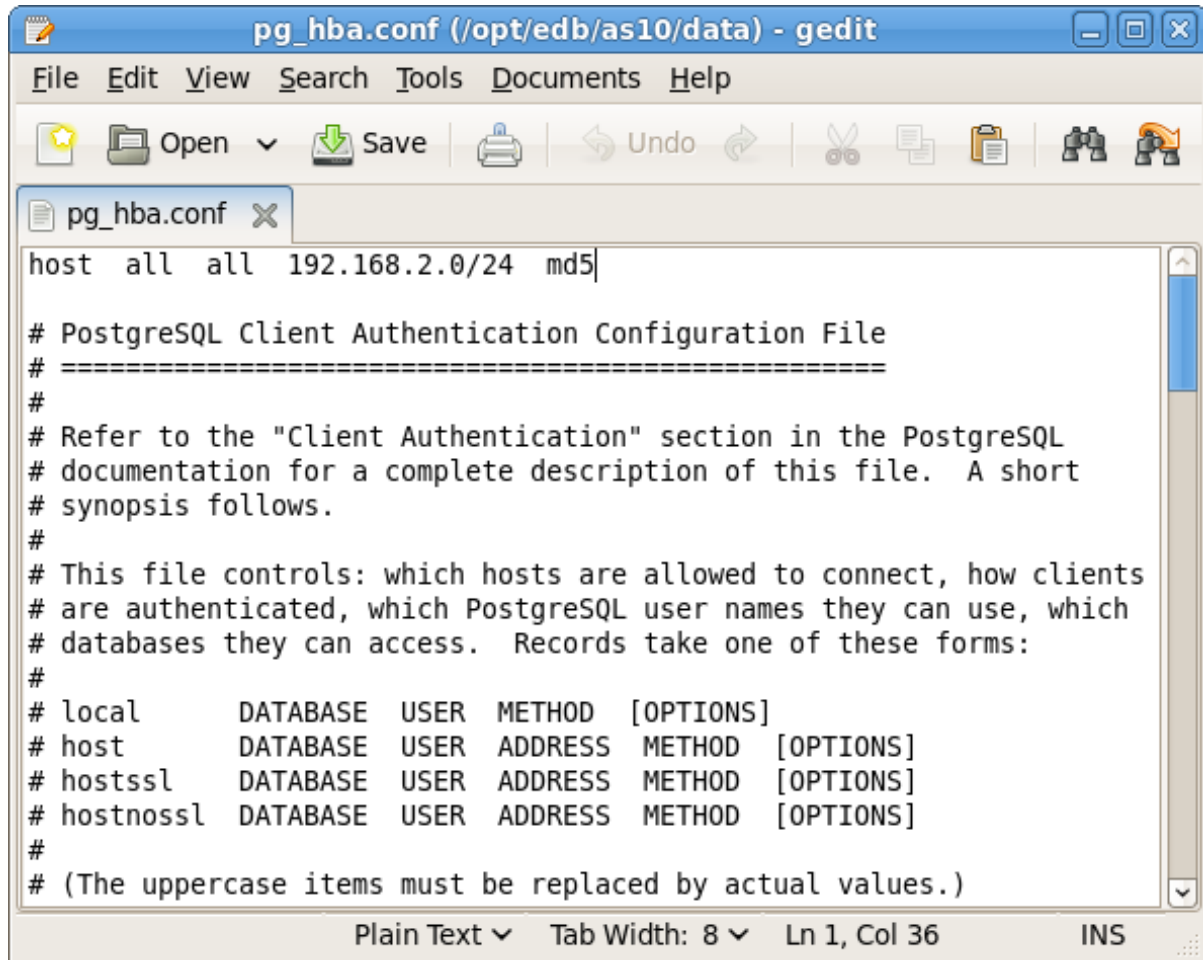


図 5: PEM entries in the pg_hba.conf file

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

デフォルトでは、pg_hba.conf ファイルは Postgres インストールの下にデータディレクトリにあります。例、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、サーバーをリスタートして変更を有効にします。

次の例は、postgres 名前付けのユーザから 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レンジを指定できます。例、192.168.10.23、192.168.10.76、および 192.168.10.184 のアドレスを持つ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、edb_lk_asis_2 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保証必要があります。

データベースサーバー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。

7.7 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。各ユーザは、データベースに含まれる情報にアクセス前に、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）をすべて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…を選択するだけです。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。

例、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 -ビュー -コンストレインなど

各オブジェクトタイプは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。

図 6: Create Table - General Tab

7.8 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 Kerberos -識別 -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、そのユーザーが接続を許可される前に必要な認証の種類を指定します。

`postgres` 名前付けのユーザーが、セキュアパスワード (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、次のフォームを取ります。

```
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。パスワードが `1safepwd` である `postgres` 名前付けのユーザーのパスワードファイルのエントリは、次のフォームになります。

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

パスワードファイルの名前付けは通常、Linux システムでは `~root/.pgpass`、Windows `%APPDATA%\postgresql\pgpass.conf` です。パスワードファイルの設定の詳細については、`edb_lk_asis_3` 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、`edb_lk_asis_4` を参照してください。

7.9 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 7: The Server Configuration dialog

パラメータ値を変更するには、パラメータ名前の右側にある Value フィールドに表示されるコンテンツを編集します。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、パラメーターをオリジナルの値に結果ます。

7.10 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。

7.10.1 ログインロール

PEM サーバーに接続するとき、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資格情報を提供する必要があります。デフォルトでは、サーバーへの最初の接続に postgres スーパーユーザアカウントが使用されますが、接続ユーザごとに個別のロールを作成することを（セキュリティと監査の両方の目的で）強くお勧めします。PEM クエリツール、PEM Web インタフェース 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。

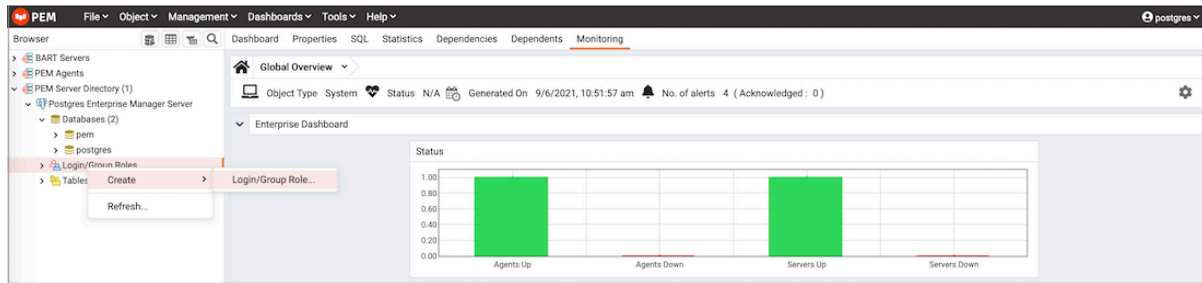


図 8: The context menu of the Login Roles node

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。

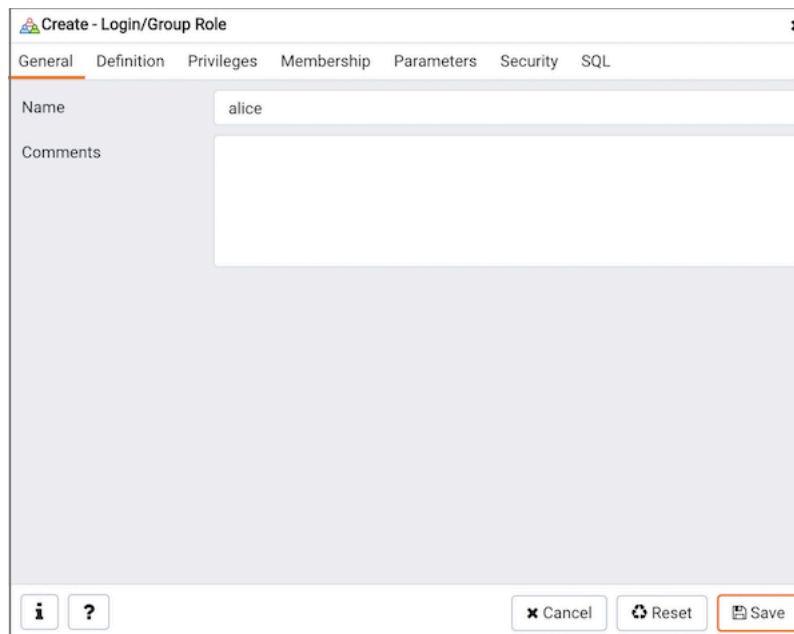


図 9: Create Login Role - General tab

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、edb_lk_asis_5 を参照してください。

7.10.2 グループの役割

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）およびオブジェクト権限（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。グループロールのプライマリ目的は、DBA にとってシステムとオブジェクトのパーミッションの一括管理をはるかに **make** にすることです。多くの異なるログインアカウントにわたって個別に特権を割り当てたり変更したりするのではなく、単一のロールの特権を割り当てまたは変更してから、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。

グループロールを作成および管理するには、**Group Roles** ノード（PEM ツリーコントロールで登録されている各サーバーの名前の下にあります）を使用します。コンテキストメニューのオプションを使用すると、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 `edb_ik_asis_6` の作成に関する詳細情報を見つけることができます。

7.10.3 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、制限された権限を持つロールが監査マネージャー、キャパシティマネージャー、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することができます。PEM の事前定義されたロールにより、PEM 機能にアクセスできます。これらのロールのメンバーシップが割り当てられたロールは、関連する機能にアクセスできます。

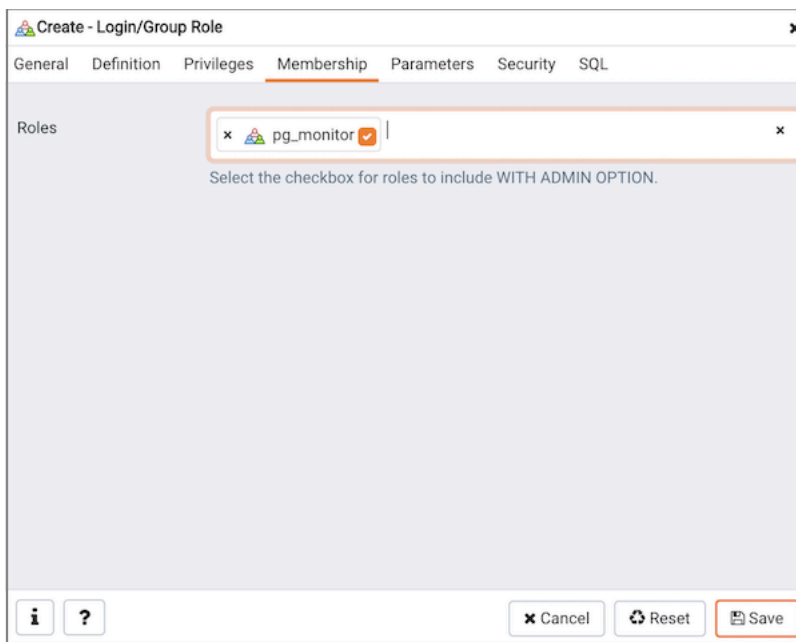


図 10: Create Login Role - Membership tab

ユーザを定義するときは、**Membership** タブを使用して、新しいユーザがメンバであるロールを指定します。新しいユーザは、メンバである各ロールに関連付けられた特権を共有します。ユーザが PEM 拡張機能にアクセスするには、ロールは `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 SQL コマンドが表示されます。

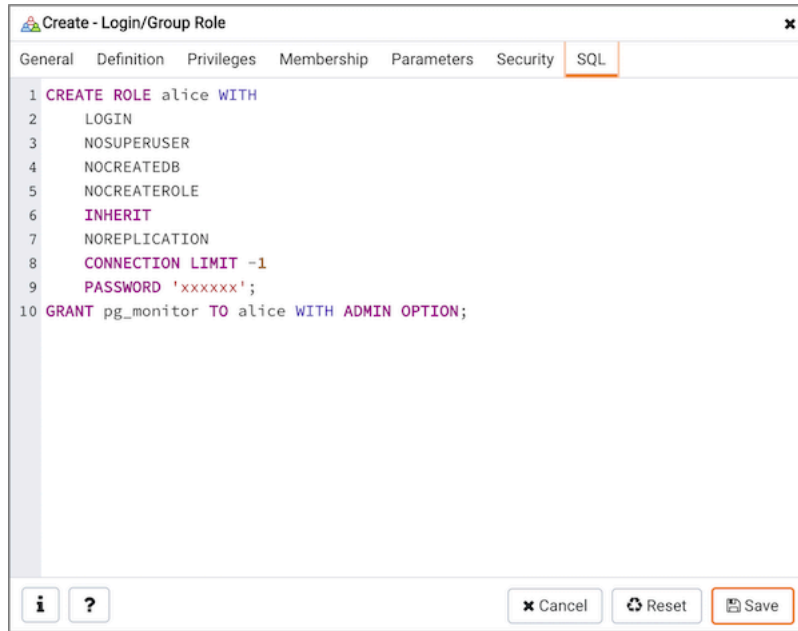


図 11: Create Login Role - SQL tab

上記の例では、Audit Manager にアクセスする acctg_clerk 名前付けのログインロールを作成します。ロールは、いつでもサーバーに無制限に接続でき make。

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
pem_super_admin		Postgres Enterprise Manager ですべてを管理/設定する役割。
pem_admin	pem_super_admin	すべての可視エージェント/サーバー、および監視対象オブジェクトの管理/管理/構成の役割。
pem_config	pem_admin	Postgres Enterprise Manager の構成管理の役割。
pem_component	pem_admin	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/実行役割。
pem_rest_api	pem_admin	REST API にアクセスするロール。
pem_server_service_manager	pem_admin	監視対象データベースサーバのリスタート/リロードを許可するロール（サーバー ID が指定されている場合）。
pem_manage_schedule_task	pem_admin	スケジュールタスクを構成する役割。
pem_manage_alert	pem_admin	アラートとそのテンプレートを管理/構成するためのロール。
pem_config_alert	pem_config, pem_manage_alert	監視対象オブジェクトのアラートを設定するためのロール。
pem_manage_probe	pem_admin	カスタムプローブを作成、更新、削除し、カスタムプローブ設定を変更する役割。

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_config_probe	pem_config, pem_manage_probe	可視されているすべての監視対象オブジェクトのプローブ設定の役割（履歴保持、実行頻度、プローブの有効化/無効化）。
pem_database_server_registration	pem_admin	データベースサーバを登録するロール。
pem_comp_postgres_expert	pem_component	Postgres Expert を実行する役割。
pem_comp_auto_discovery	pem_component	データベースサーバダイアログの自動検出を実行するロール。
pem_comp_log_analysis_expert	pem_component	ログ分析エキスパートを実行する役割。
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	フェールオーバーマネージャーの機能を管理する役割。
pem_comp_capacity_manager	pem_component	Capacity Manager を実行する役割。
pem_comp_log_manager	pem_component	Log Manager を実行するロール。
pem_comp_audit_manager	pem_component	監査マネージャーを実行する役割。
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。

7.10.4 チームロールを使用する

PEM によるモニタリングのためにサーバを登録するときに、サーバに関連付けられる* Team *を指定できます。チームは、1 つ以上の監視対象サーバへのアクセスを、ロールメンバの限定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のみを以下のユーザーに表示します。

-チームロールのメンバ-サーバを作成したロール-PEM サーバでのスーパーユーザ特権を持つロール。

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。

7.10.5 オブジェクト権限

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、ロールに十分な特権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フェースを使用して、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を介して管理されます。例、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前を右クリックし、コンテキストメニューから [プロパティ] 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。

また、PEM クライアントには、Grant Wizard（Tools メニューからアクセス）が含まれており、一度に多くのオブジェクト権限を管理できます。

7.11 ジョブ通知の管理

システムが生成しジョブ（スケジュールされたタスクの下にリストされている）またはカスタム定義のエージェントジョブの成功または失敗時に 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。これらのメール通知設定は、指定されたユーザグループにメール通知を送信するために、次の 3 つのレベルで（優先オーダーに）構成できます。

-職務レベル -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）

7.11.1 ジョブレベルでジョブ通知を設定する

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。

The screenshot shows the 'Create - Agent Job' wizard with the 'Notifications' tab selected. The 'Send the notifications' dropdown is set to 'ALWAYS'. Below this, the text explains that 'ALWAYS' means sending a notification on the completion of the job regardless of the result. The 'Email group' dropdown is set to '<Default>'. The 'Save' button is highlighted with an orange border.

図 12: Configure Job Notification - Job level

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、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電子メールグループを指定します。

7.11.2 エージェントレベルでジョブ通知を設定する

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。

Postgres Enterprise Manager Host (1)

General Job Notifications Agent Configurations

Override default configuration? ☐ No
Select to override the default configuration for job notifications. If selected, the following settings will determine whether, when, and which email group will receive the job notification for this agent.

Email on job completion? ☐ No
Select to receive a notification email on completion of a job (regardless of the result) of this agent.

Email on a job failure? ☐ No
Select to receive a notification email only on failure of a job of this agent.

Email group
Select the email-group that will receive the notification on completion of a job or scheduled task.

Buttons: [i] [?] [Cancel] [Reset] [Save]

図 13: 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上書きするかどうかを指定します。このスイッチで[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た場合にのみ機能することにノートしてください。
- **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、ジョブの失敗時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、ジョブ通知の送信先のメールグループを指定します。

7.11.3 サーバーレベルでジョブ通知を設定する

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メール通知構成に関する情報を提供できます。[サーバー構成] ダイアログを開くには、PEM クライアントの [管理] メニューから [サーバー構成…] を選択します。

Parameter	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 14: 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** パラメータを使用して、メール通知を受信する電子メールグループを指定します。
- **job_retention_time** parameter を使用して、非定期的なスケジュール済み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、失敗、成功、中断などのステータスに関係なく、各ジョブステータスの変更後にメール通知を送信するかどうかを指定します。

7.12 PEM スケジュール済みジョブの管理

PEM スケジュールジョブを作成して、指定したシーケンスで一連のカスタム定義のステップを実行できます。これらの手順には、エージェントにバインドされている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ている場合があります。ビジネス要件に合わせてこれらのジョブをスケジュールできます。例、特定のデータベースサーバのバックアップを取得するジョブを作成し、毎月特定の日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は [ジョブ] ノードが表示され、その下に現在定義されているジョブが表示されます。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 スケジュールジョブを make 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 15: 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 をジョブを無効にします。
- 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オーダーに実行されることにノートしてください。
- Enabled スイッチを使用して、ジョブの実行時にステップを含める (True) か、ステップを無効にします (False)。
- Kind スイッチを使用して、ジョブステップが SQL コード (SQL) を呼び出すか、バッチスクリプト (Batch) を呼び出すかを示します。
- SQL を選択した場合は、Code タブを使用して、ステップの SQL コードを提供します。- Batch を選択した場合、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' dialog box with the 'Steps' tab selected. A table lists the steps, with 'Step1_backup' being the only one. Below the table, the 'General' sub-tab is active, displaying configuration fields for the selected step. The 'Code' sub-tab is also visible.

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:

Buttons: Cancel, Reset, Save

図 16: 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。以下から選択してください：

-失敗-このステップの処理中にエラーが発生した場合、ジョブを停止します。-成功-ステップを正常に完了したとしてマークし、続行します。-無視-エラーを無視して続行します。

- **Kind** スイッチの入力として SQL を選択した場合、以下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 ステートメントを指定します。

-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は、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、最初の行に適切なインタプリタが指定されていれば（たとえば、`#!/bin/sh`）、任意のシェルスクリプトを使用できます。定義済みの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。

Create - Agent Job

General Steps Schedules Notifications SQL

Name Job_backup_pem_db

Enabled? ☒ Yes

Comment Job for taking backup of pem database

図 17: 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:

Cancel Reset **Save**

図 18: 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を指定する必要があります。このパラメータには、非ルートユーザまたはルートを指定できます。ユーザが存在しない場合、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 エージェントをリスタートします。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、ステップ定義ダイアログを保存して閉じます。

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。

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:

Cancel
Reset
Save

図 19: 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) を使用することを示すか、スケジュールを無効にする (No) します。

- **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

Month Days

Months

Times

Hours

Minutes

図 20: Create Agent Job - Schedules tab - Repeat tab

Repeat タブのフィールドを使用して、cron スタイルフォーマットでスケジュールに関する詳細を指定します。ジョブは、Repeat タブで選択された各日付または時刻要素で実行されます。

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、ジョブを実行日数を選択します。Last Day を指定して、日付に関係なく、ジョブが月の最後の日に実行されることを示します。
- **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 を実行日を指定します。

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

Date	Time
2020-04-01	00:00

図 21: 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。例、国民の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、例外テーブルに行を追加してから、次の操作を行います。

- Date 列内をクリックしてカレンダーセクターを開き、ジョブが実行されない日付を選択します。Date 列に <Any> を指定して、選択した時刻のどの日にも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、ジョブが実行されない時間を指定します。Time 列に <Any> を指定して、選択した日のどの時点でも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、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電子メールグループを指定します。

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.

図 22: Create Agent Job - Notifications tab

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。

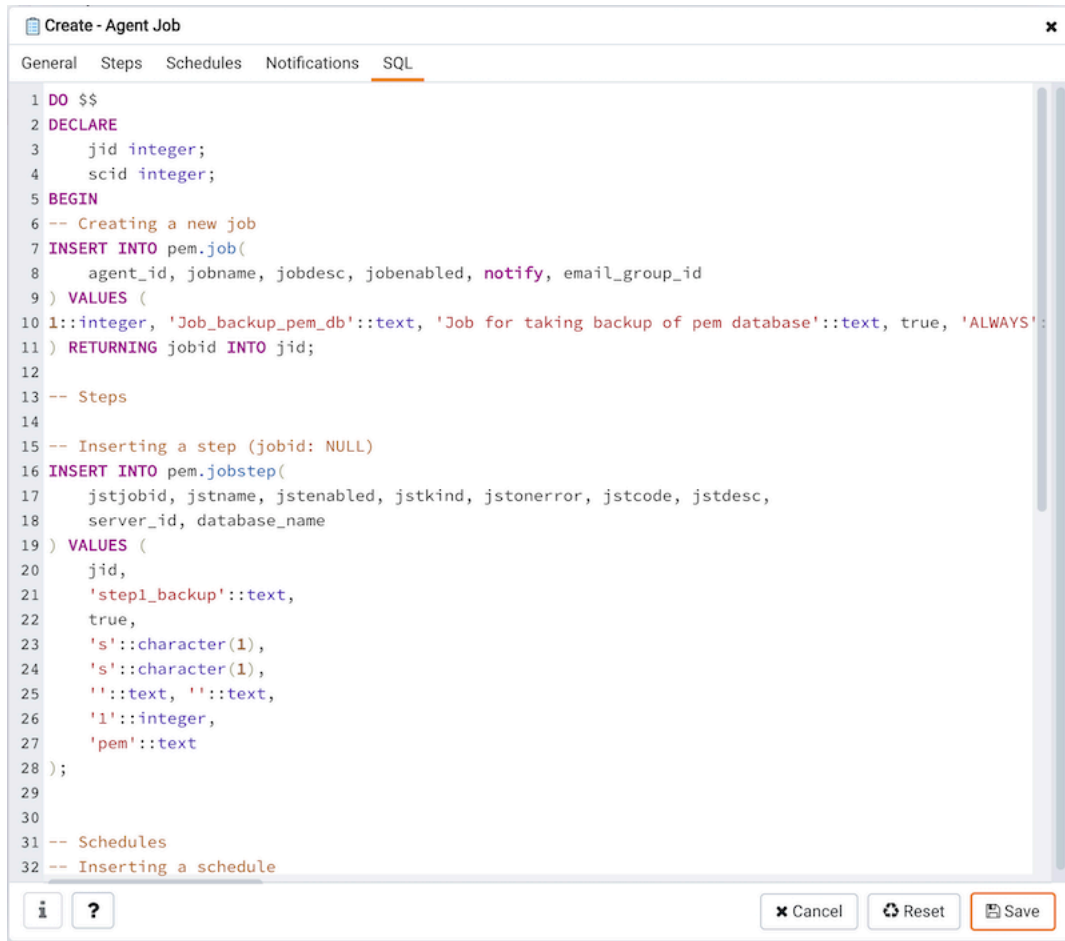


図 23: Create Agent Job - SQL tab

Save ボタンをクリックしてジョブ定義を保存するか、Cancel 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、ダイアログから未保存のエントリを削除します。

ジョブは、それが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、選択したジョブの概要が表示され、[統計] タブにはジョブの各実行の詳細が表示されます。既存のジョブを変更するか、ジョブに関する詳細情報を確認するには、ジョブ名前を右クリックして、コンテキストメニューから Properties を選択します。

第8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。

8.1 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のスーパーユーザ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で無制限のプローブを呼び出し、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ノートしてください。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のサービスアカウントで実行されている場合、PEM プロー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 プローブは情報を結果ず、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	監査ログマネージャー、サーバーログマネージャー、ログ分析エキスパート、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	スワップ使用率、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 Sys-tem	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は表示されません。
Slony Replication	Linux/Windows	データベース分析ダッシュボードの Slony 関連のチャートには情報が表示されません。
Tablespace Size	Linux/Windows	サーバー分析ダッシュボードには完全な情報は表示されません。
xDB Replication	Linux/Windows	PEM は xDB アラートとトラップを送信できません。

プローブが不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** エラーを結果場合があります。

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** エラーを結果か、返されたデータを空の値として PEM チャートまたはグラフに表示する場合があります。

プローブが失敗すると、プローブの名前、プローブが失敗した理由、および問題の解決にヘルプヒントを含むエントリー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ビューしたり、PEM ワーカーログファイルで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** で利用可能です。

8.2 エージェントの構成

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、変更を適用するには PEM エージェントをリスタートする必要があります。

PEM_MAXCONN パラメータを例外、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。別のサーバーのモニタリング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内のファイルのフォームで行われます。	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タイプを指定します。	warning
<code>log_location</code>	PEM ワーカーログファイルの場所を指定します。	127.0.0.1.
<code>agent_log_location</code>	PEM エージェントログファイルの場所を指定します。	<code>/var/log/pem/agent.log</code>
<code>long_wait</code>	最初の接続試行が失敗した場合に、PEM エージェントが PEM サーバーへの接続を試行する前に待機する最大長さ（秒）。	30 seconds
<code>short_wait</code>	PEM エージェントがキュー内の次のプローブ（実行待ち）をチェックする前に待機する最小長さ（秒単位）。	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>	マルチプルの PEM エージェント（7.13 以下）に対して <code>true</code> に設定すると、より多くの重複した電子メールを送信する場合があります。一方、PEM エージェント（7.14 以降）の場合、重複する電子メールの送信回数が少なくなる場合があります。	true for PEM server host; false for all others.

次のページに続く

表 3 – 前のページからの続き

Parameter Name	Description	Default Value
enable_snmp	マルチプルの PEM エージェント（7.13 以下）に対して 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）の場合は、より少ない重複トラップを送信する場合があります。	true for PEM server host; false for all others.
enable_nagios	true に設定すると、Nagios アラートが有効になります。	true for PEM server host; false for all others.
enable_webhook	true に設定すると、Webhook アラートが有効になります。	true for PEM server host; false for all others.
max_webhook_retries	失敗時に pemAgent が webhook の呼び出しをリトライする最大回数を設定します。	Default 3.
connect_timeout	エージェントが接続を待機する最大秒数（10 進数整数文字列）。	Not set by default; set to 0 to indicate the agent should wait indefinitely.
allow_server_restart	真に設定されている場合、エージェントは監視するデータベースサーバをリスタートできます。このパラメータの値に応じて、一部の PEM 機能を有効/無効にすることができます。	False
max_connections	接続スロットルで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が保持される最小秒数を指定します。MaxConnections で指定された値に達し、待機中の要求を満たすために（異なるデータベースへの）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	真に設定すると、ユーザ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	真に設定すると、ハートビートの送信に専用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の名前を指定します。	None
webhook_ssl_key	webhook の SSL クライアントキーファイルへの完全なパス。	

次のページに続く

表 3 – 前のページからの続き

Parameter Name	Description	Default Value
webhook_ssl_cert	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_cert	webhook の SSL ca 証明書ファイルへの完全なパス。	
allow_insecure_webhooks	true に設定されている場合、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.
AgentCertPath	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。別のサーバーのモニタリング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内のファイルのフォームで行われます。	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タイプを指定します。	warning
LongWait	最初の接続試行が失敗した場合に、PEM エージェントが PEM サーバーへの接続を試行する前に待機する最大長さ（秒）。	30 seconds
shortWait	PEM エージェントがキュー内の次のプローブ（実行待ち）をチェックする前に待機する最小長さ（秒単位）。	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.

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
EnableSMTP	true に設定すると、SMTP 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	失敗時に pemAgent が webhook の呼び出しをリトライ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最大秒数（10 進数整数文字列）。	Not set by default; if set to 0, the agent will wait indefinitely.
AllowServerRestart	真に設定されている場合、エージェントは監視するデータベースサーバをリスタートできます。このパラメータの値に応じて、一部の PEM 機能を有効/無効にすることができます。	true
MaxConnections	接続スロットルで使用す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が保持される最小秒数を指定します。MaxConnections で指定された値に達し、待機中の要求を満たすために（異なるデータベースへの）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	真に設定すると、ユーザ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	真に設定すると、ハートビートの送信に専用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、バッチ/シェルスクリプトは管理ユーザアカウントを使用して実行されます。	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。	

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
WebhookSSLCaCrt	webhook の SSL ca 証明書ファイルへの完全なパス。	
AllowInsecureWebhooks	true に設定されている場合、webhook が安全でないフラグで呼び出すことを許可します。	false

8.3 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。

The screenshot shows the 'Postgres Enterprise Manager Host' dialog box with the 'General' tab selected. The 'Description' field is filled with 'Postgres Enterprise Manager Host'. The 'Group' dropdown menu is set to 'PEM Agents'. The 'Team' field is empty. The 'Heartbeat interval' is set to 0 Minutes and 30 Seconds. At the bottom, there are buttons for 'Cancel', 'Reset', and 'Save'.

図 1: 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、エージェントが表示され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ノートしてください。[チーム] フィールドは真の分離を提供しないため、セキュリティの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 サーバーへのレポート間で経過する時間の長さが表示されます。インターバルを変更するには、**Minutes** または **Seconds** フィールドの横にあるセクターを使用します。

The screenshot shows the 'Postgres Enterprise Manager Host' dialog box with the 'Job Notifications' tab selected. The dialog contains the following settings:

- Override default configuration?**: A toggle switch set to 'No'. Description: 'Select to override the default configuration for job notifications. If selected, the following settings will determine whether, when, and which email group will receive the job notification for this agent.'
- Email on job completion?**: A toggle switch set to 'No'. Description: 'Select to receive a notification email on completion of a job (regardless of the result) of this agent.'
- Email on a job failure?**: A toggle switch set to 'No'. Description: 'Select to receive a notification email only on failure of a job of this agent.'
- Email group**: A dropdown menu set to '<Default>'. Description: 'Select the email-group that will receive the notification on completion of a job or scheduled task.'

At the bottom of the dialog are three buttons: 'Cancel', 'Reset', and 'Save'.

図 2: 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上書き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、ジョブ通知を送信するタイミングと宛先を定義できます。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た場合にのみ機能することにノートしてください。
- **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、ジョブの失敗時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、ジョブ通知の送信先の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 3: PEM Agent Properties dialog - Agent Configurations tab