



Postgres Enterprise Manager

リリース **8.0**

PEM Administrators Guide

2020 年 12 月 19 日

目次

第 1 章	PEM Overview	3
1.1	PEM Architecture	5
1.1.1	PEM サーバー	6
1.1.2	PEM エージェント	8
1.1.3	PEMWeb クライアント	10
1.1.4	SQL プロファイラープラグイン	10
1.2	Installing PEM - Overview	11
1.3	Using the PEM Web Interface	12
1.3.1	PEM メニューバー	14
1.3.2	チャート、グラフ、表の制御とカスタマイズ	22
1.3.3	オンラインヘルプとドキュメント	23
第 2 章	Registering a Server	25
2.1	サーバーの手動登録	25
2.2	自動サーバー検出	37
2.3	pemworker ユーティリティを使用してサーバーを登録する	40
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	43
2.4	接続とバインドの検証	44
第 3 章	Defining and Monitoring Postgres instances on AWS	47
3.1	AWSEC2 で実行されている Postgres インスタンスのモニタリング	47
3.2	AWSRDS で実行されている Postgres インスタンスのモニタリング	48
第 4 章	Managing Certificates	51
4.1	SSL 証明書の置き換え	52
4.2	エージェント SSL 証明書の更新	55
第 5 章	Managing Configuration Settings	57
第 6 章	Managing a PEM Server	59
6.1	PEM サーバーとエージェントの起動と停止	59
6.2	監視対象サーバーのリモート起動と停止	60
6.3	Linux での PEM サーバーまたは PEM エージェントの制御	61
6.4	Windows での PEM サーバーまたは PEM エージェントの制御	62

6.5	HTTPD サーバーの制御	63
6.6	pg_hba.conf ファイルの変更	64
6.7	データベースとオブジェクトの作成と保守	66
6.8	PEM 認証の管理	68
6.9	PEM サーバー構成の編集	69
6.10	セキュリティの管理	70
6.10.1	ログインロール	70
6.10.2	グループの役割	71
6.10.3	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	72
6.10.4	チームロールを使用する	75
6.10.5	オブジェクト権限	76
6.11	ジョブ通知の管理	77
6.11.1	ジョブレベルでのジョブ通知の構成	77
6.11.2	エージェントレベルでのジョブ通知の構成	78
6.11.3	サーバーレベルでのジョブ通知の構成	80
6.12	PEM スケジュール済みジョブの管理	81
第 7 章	Managing a PEM Agent	91
7.1	エージェント権限	91
7.2	エージェント構成	94
7.3	エージェントのプロパティ	101
第 8 章	Conclusion	105
索引		107

このドキュメントでは、PostgresEnterpriseManager™ (PEM) の概要を説明します。PostgresEnterpriseManager (PEM) は、PostgreSQL および EDBAdvancedServer データベースサーバーの管理、監視、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です。PEM は、数個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監視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の詳細については、次の EDBWeb サイトにアクセスしてください。

<https://www.enterprisedb.com/services-support/edb-supported-products-and-platforms#pem>

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

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。

- **Management en Masse Design**。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多数のサーバーの管理に取り組むために構築されています。グローバルダッシュボードは、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。
- **Distributed Architecture**。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ように設計されています。
- **Graphical Administration**。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ーフェイスを介して PEM クライアントで実行できます。サーバーの起動とシャットダウン、構成管理、ストレージとセキュリティの制御、オブジェクトの作成、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。
- **Full SQL IDE**。PEM には、アドホック SQL クエリ、ストアドプロシージャ/関数開発、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。
- **Enterprise Performance Monitoring**。PEM は、すべての管理対象データベースサーバーのエンタープライズクラスのパフォーマンス監視を提供します。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と各マシンの基盤となるオペレーティングシステムのすべての側面を監視し、インターフェース内で簡単にナビゲートされるパフォーマンスページに詳細な統計情報を返します。
- **Proactive Alert Management**。PEM には、24 時間監視される各主要メトリック（メモリ、ストレージなど）のパフォーマンスしきい値を作成する機能が付属しています。しきい値違反が発生すると、アラートが集中ダッシュボードに送信され、問題の性質と、サーバーの全体的なパフォーマンスが危険にさらされるのを防ぐために必要なアクションが通知されます。
- **Simplified Capacity Planning**。主要なパフォーマンス関連の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。**CapacityManager** ユーティリティを使用すると、さまざまな統計情報を選択し、時間の経過とともに傾向分析を実行して、ピーク負荷期間、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。

- **Audit Manager.** AuditManager は、AdvancedServer インスタンスの監査ログを設定します。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実行される 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表示できます。
- **Log Manager.** LogManager 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、サーバーのロギングパラメーターを構成します。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 Workload Profiling.** 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 AdvancedServer の IndexAdvisor 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ため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。
- **Expert Database Analysis.** PEM には PostgresExpert ユーティリティが含まれています。PostgresExpert 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専門家のヘルプとともに報告されます。
- **Streaming Replication Monitoring.** ストリーミングレプリケーションダッシュボードを監視するか、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昇格させることができます。
- **Secure Client Connectivity.** PEM は、暗号化された SSH トンネルを介した安全な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。
- **Wide Platform Support.** PEM は、ほとんどの主要な Linux および Windows プラットフォームをサポートしています。

1.1 PEM Architecture

PostgresEnterpriseManager (PEM) は、単一の GUI インターフェイスを介して複数の Postgres サーバーを監視および管理するために設計されたツールです。PEM は、インフラストラクチャの次の領域を監視できます。

注:Postgres という用語は、PostgreSQL または EDBPostgresAdvancedServer を指します。

- **Hosts** -1 つ以上のサーバー（物理または仮想）およびそれらのオペレーティングシステム。
- **Servers** -ホストで実行されている PostgreSQL または EDBPostgresAdvancedServer の 1 つ以上のインスタンス。
- **Databases** -1 つ以上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 Server** -PEM サーバーは、データを監視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、監視データを保存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 Agent** -PEM エージェントは、タスクを実行し、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 サーバーに統計を報告します。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 の複数のインストール済みインスタンスを監視できます。
- **PEM Web Client** -PEMWeb インターフェースにより、Postgres サーバーを管理および監視し、PEM 拡張機能を利用できます。Web インターフェイスソフトウェアは 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** -SQLProfiler は、PEM の SQLProfiler ツールで分析される監視データとクエリプランを記録する 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 AdvancedServer（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、「PEMSQL プロファイラー構成ガイド <<https://www.enterprisedb.com/edb-docs/p/edb-postgres-enterprise-manager>>`_」を参照してください。

PEM architecture

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。

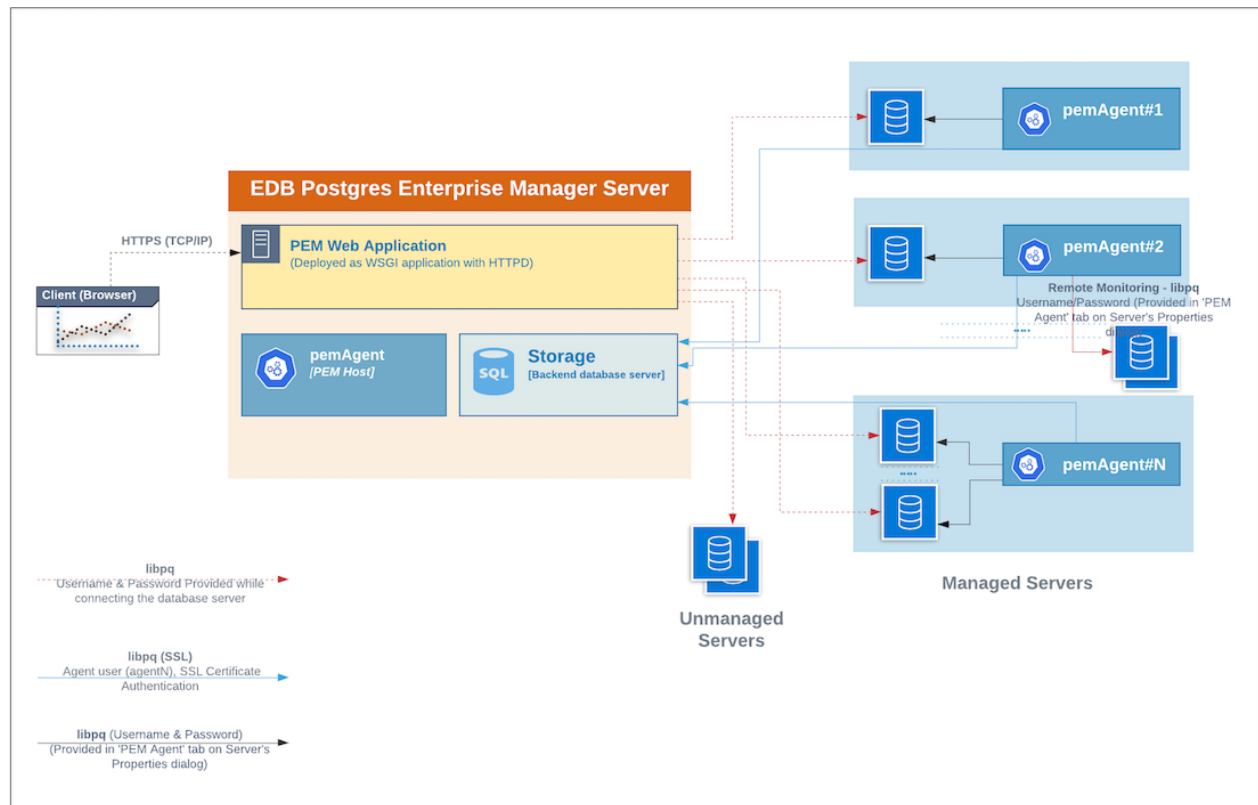


図 1 PEM アーキテクチャ

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 ApacheWeb 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、サーバー側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ー）のインスタンスと ApacheWeb サーバー（HTTPD）のインスタンスは、同じホスト上にあっても、別々のホスト上にあってもかまいません。

- **Postgres Instance (Database server)**-これはバックエンドデータベースサーバーです。PEM サーバーのリポジトリとして機能する****Postgres Instance (Database server)****という名前のデータベースをホストします**** Postgres Instance (Database server)**** データベースには、監視対象の各ホスト、サーバー、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。これには、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スナップショットが格納されます。
 - **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。
- **Apache Web Server (HTTPD)** -PEMWeb アプリケーションは、HTTPD を使用して WSGI アプリケーション

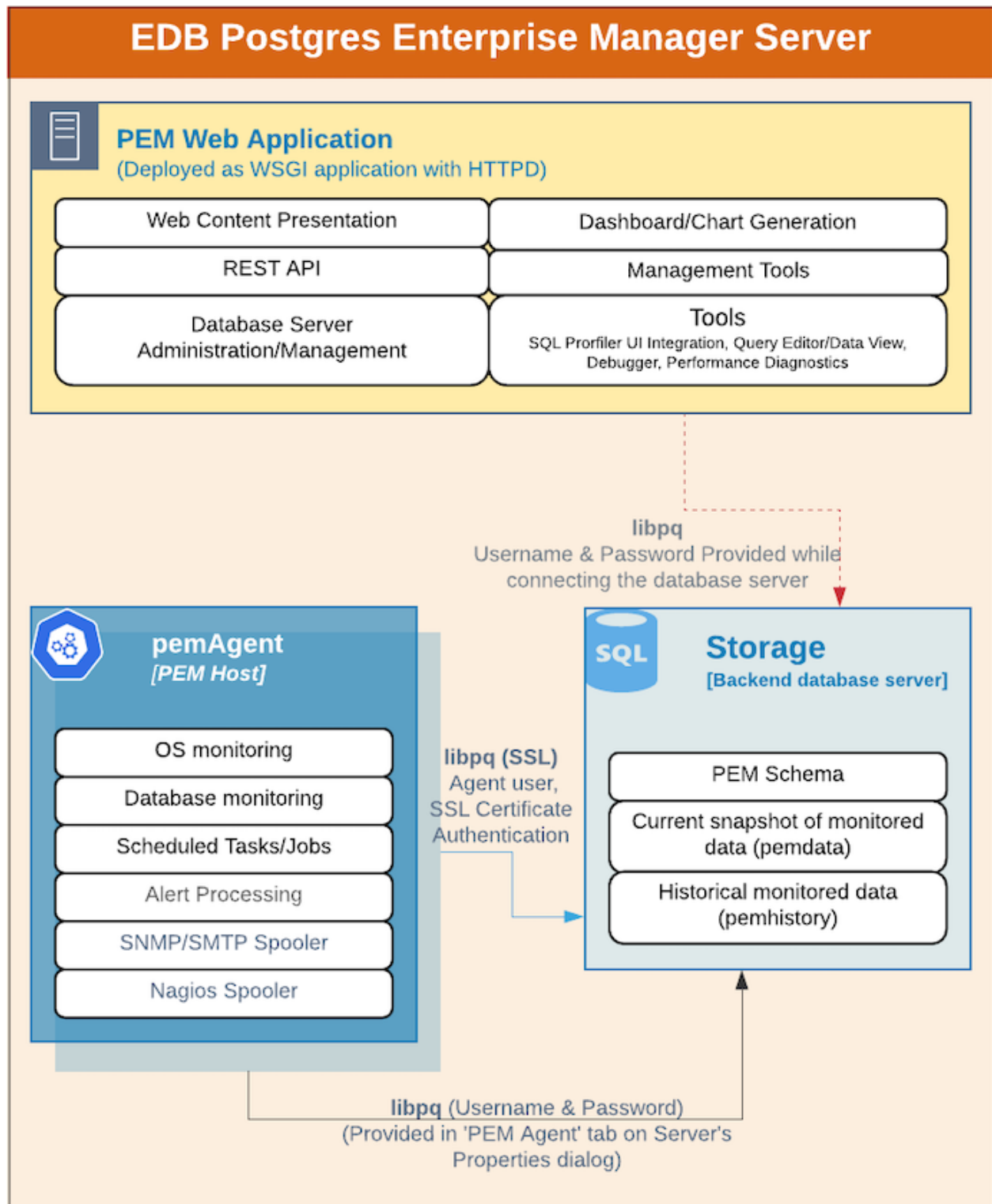


図 2 PEM サーバー

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 content presentation** -プレゼンテーションレイヤーは、Web アプリケーション（たとえば、ブラウザ、ログインページなど）によって作成されます。
- **Rest API** -RESTAPI は、他のアプリやサービスとの統合を可能にします。
- **Database Server Administration/Management** -CREATE、ALTER、DROP などのデータベースサーバーの管理および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。
- **Dashboard/Chart generation** -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。
- **Management Tools** -AuditManager、CapacityManager、LogManager、PostgresExpert、PostgresLogAnalysisExpert、および TuningWizard が Web アプリケーションで使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。
 - * **SQL Profiler UI Integration** -SQLProfiler 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。
 - * **Query Editor/Data View** -クエリエディタを使用すると、データをクエリ、編集、および表示できます。
 - * **Debugger** -デバッガーはクエリのデバッグに役立ちます。
 - * **Performance Diagnostics** -パフォーマンス診断は、AdvancedServer 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実稼働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受ける場合があります。

1.1.2 PEM エージェント

PEM エージェントは、マシンとオペレーティングシステム、およびそれらがバインドされている各 Postgres インスタンスからの監視データの収集を担当します。各 PEM エージェントは、1 台の物理マシンまたは仮想マシンを監視でき、複数のデータベースサーバーをローカルに（同じシステムにインストールされているか、リモートに）他のシステムにインストールされているかを監視できます。また、ユーザー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 プロファイラートレース、ユーザー定義ジョブなど）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。別途、PEM を使用して監視する Postgres インスタンスをホスト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。

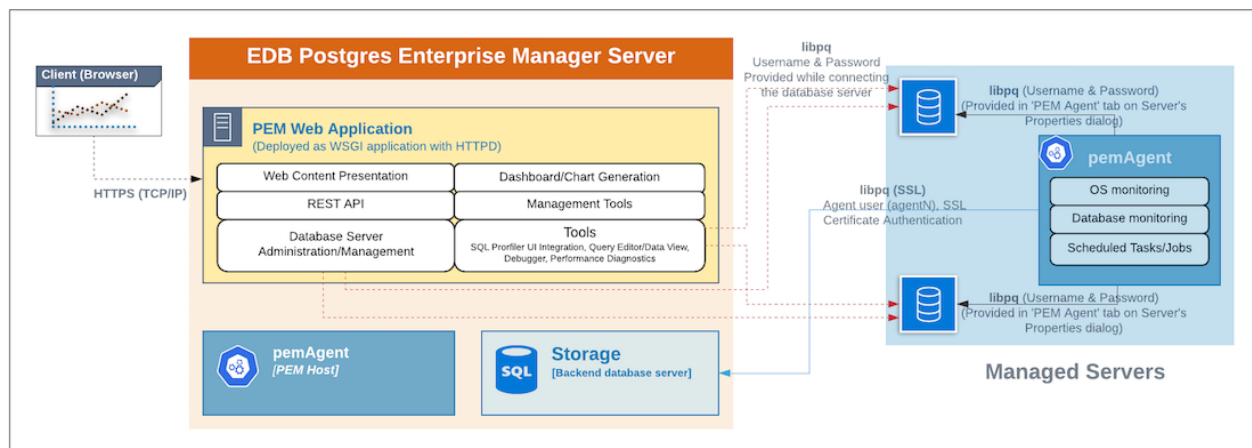


図 3 PEM エージェント

ローカルで監視するからリモートで監視するにかかわらず、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ーと Linux の pemworkerCLI 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。

PEM 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。ユーザーは、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 **metric** と呼ばれ、**metric** 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複数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ーのバージョン文字列
- データベースサーバー構成オプション（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 [こちら](https://www.enterprisedb.com/edb-docs/d/edb-postgres-enterprise-manager/reference/online-documentation/latest/pem_probes.html#pem-probes) <https://www.enterprisedb.com/edb-docs/d/edb-postgres-enterprise-manager/reference/online-documentation/latest/pem_probes.html#pem-probes> にあります。>_。

デフォルトでは、データベースサーバーにバインドされた PEM エージェントは OS/データベース監視統計を収集し、その特定のデータベースサーバーのスケジュールされたタスク/ジョブも実行し、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 サーバーの pem データベースに格納され、デフォルトで PEM ホストの 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することができます。

これらのパラメーターの詳細については、「サーバー構成<https://www.enterprisedb.com/edb-docs/d/edb-postgres-enterprise-manager/reference/online-documentation/latest/pem_config_options.html#pem->」を参照してください。config-options>`_。

1.1.3 PEMWeb 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 インターフェイスは 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、PostgresLogAnalysisExpert、CapacityManager 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、SQLProfiler 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 EDBPostgresAdvancedServer 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。SQLProfiler インストーラーは、EDBWeb サイト<<https://www.enterprisedb.com/downloads/edb-postgres-enterprise-manager>>`_から入手できます。

SQLProfiler 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、「PEMSQL プロファイラー構成ガイド<<https://www.enterprisedb.com/edb-docs/p/edb-postgres-enterprise-manager>>`_」を参照してください。

1.2 Installing PEM - Overview

PEM のインストールに関する詳細な手順については、次の場所にある*PEM インストールガイド*を参照してください。

<https://www.enterprisedb.com/edb-docs/p/edb-postgres-enterprise-manager>

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 サーバーソフトウェアとバックエンドデータベース（名前は `pem`）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もあります。

PEM サーバーインストーラーは、PEM サーバーのホストに PEM エージェントと PEM クライアントをインストールします。

2. PEM で管理する追加の各物理マシンまたは仮想マシンを登録します。便宜上、PEM はリモート監視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注意してください。各サーバー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。SQLProfiler インストーラーは、Postgres のインストール場所の入力を求め、必要なソフトウェアをそのディレクトリに配置します。SQLProfiler プラグインは AdvancedServer 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

PEMWeb インターフェイスは、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、次の場所に移動して、選択したブラウザで PEM インターフェイ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。PostgresEnterpriseManager ログインウィンドウが開きます。



図 4 PEM ログインページ

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ーの名前を入力します。最初に接続するユーザーの場合、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。
- Password フィールドにユーザーに関連付けられたパスワードを入力します。

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。

PEMWeb インターフェイスが開き、Global Overview ダッシュボードが表示されます。

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。

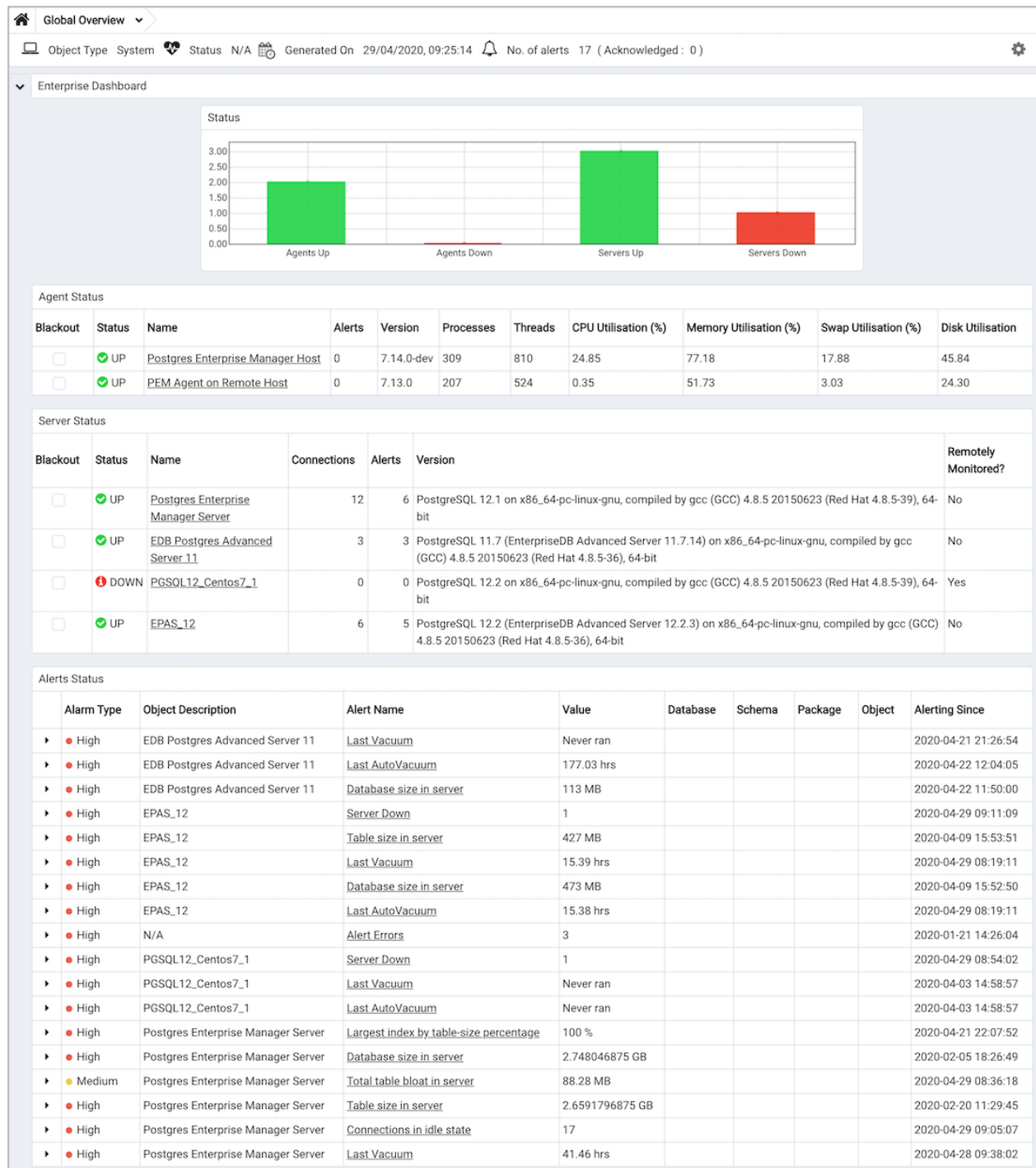


図 5 PEMGlobalOverview ダッシュボード

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ーは、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（設計上）有効でない場合があります。
- サーバーへの接続に使用した役割には、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。

PEMWeb インターフェイ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。

The File Menu

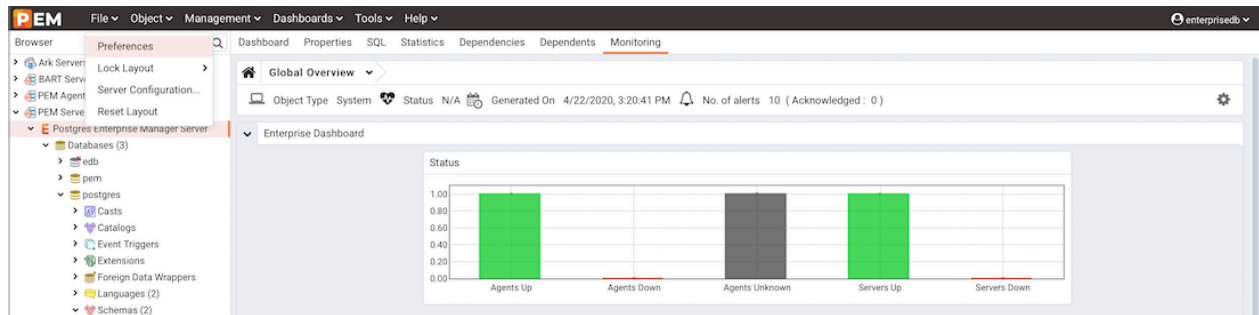


図 6 ファイルメニュー

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。

メニューオプション	アクション
プリファレンス	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。
レイアウトをロック	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。
サーバー構成	[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。
レイアウトをリセット	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。

The Object Menu

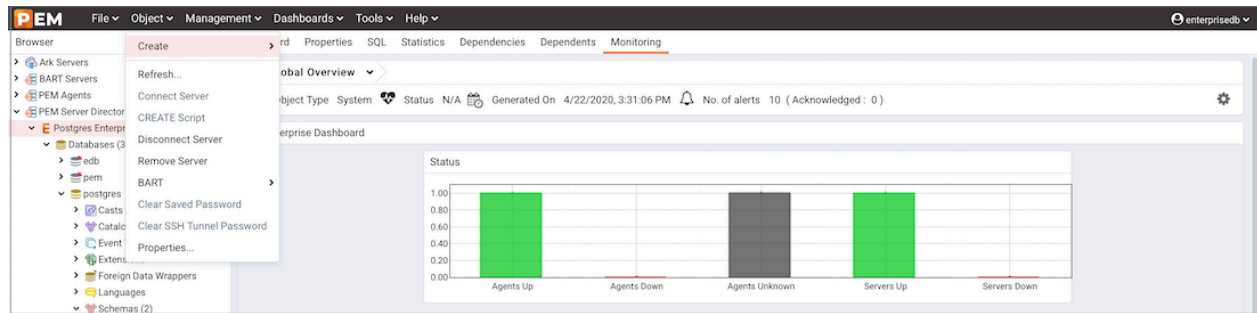


図 7 オブジェクトメニュー

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。

メニューオプション	アクション
作成する	[作成] をクリックして、状況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。
更新...	クリックすると、現在選択されているオブジェクトが更新されます。
削除/ドロップ	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。
接続サーバー	[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 スクリプト	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表示します。
サーバーの切断	クリックすると、現在選択されているオブジェクトが更新されます。
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。
保存したパスワードをクリア	データベースサーバー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。
SSH トンネルパスワードをクリア	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。
ドロップカスケード	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。
隠す	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表示するには、[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。
プロパティ...	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
トリガー	クリックして、現在選択されているテーブルのトリガーを*無効*または*有効* にします。
切り捨て	クリックすると、テーブルからすべての行が削除されます（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が削除されます（カスケードの切り捨て）。
データを見る	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。

The Management Menu

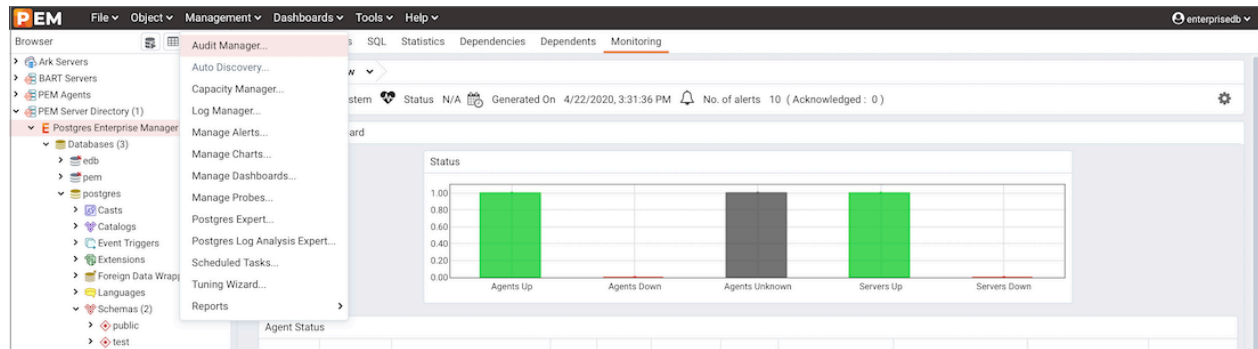


図 8 管理メニュー

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

メニューオプション	アクション
監査マネージャー...	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。
自動検出...	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。
キャパシティマネージャ...	クリックして CapacityManager 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。
ログマネージャー...	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。
アラートを管理...	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。
チャートを管理...	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。
ダッシュボードを管理...	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。
プローブの管理...	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 エキスパート...	クリックして PostgresExpert 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 ログ分析エキスパート...	クリックして PostgresLogAnalysisExpert 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。
スケジュールされたタスク	[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。
チューニングウィザード...	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。
報告書	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。

The Dashboards Menu

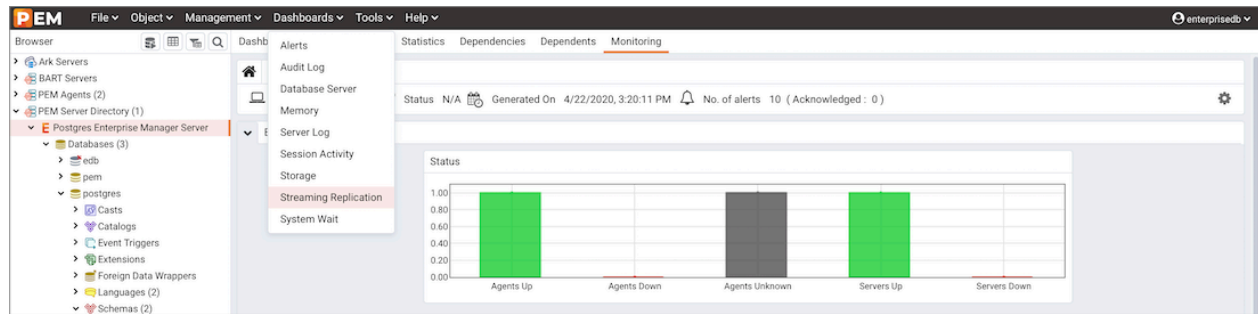


図9 ダッシュボードメニュー

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。

オプション	アクション
アラート	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。
監査ログ	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
データベースサーバー	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 分析	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。
記憶	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
オブジェクトアクティビティ	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。
オペレーティングシステム	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。
プローブログ	クリックして、選択したノードのプローブログ分析ダッシュボードを開きます。
サーバーログ	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。
セッション活動	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。
保管	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。
ストリーミングレプリケーション	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。
システム待機	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。

The Tools Menu

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。

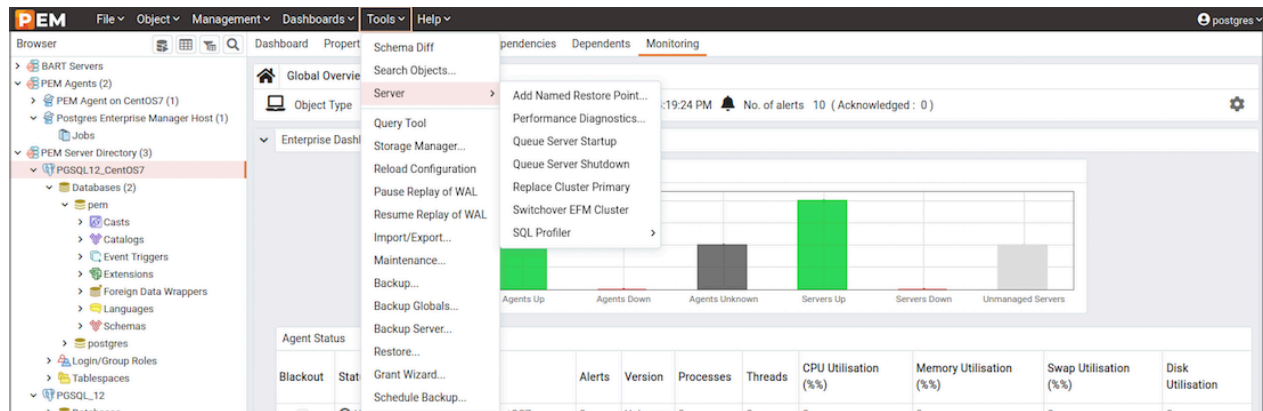


図 10 [ツール] メニュー

オプション	アクション
スキーマ差分	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。
オブジェクトを検索する	クリックして [オブジェクトの検索] ダイアログを開き、データベース内のデータベースオブジェクトを検索します。
サーバー	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。
クエリツール	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。
設定の再読み込み	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。
WAL のリプレイを一時停止	クリックして、WAL ログの再生を一時停止します。
WAL のリプレイを再開する	クリックして、WAL ログの再生を再開します。
インポート/エクスポート...	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。
メンテナンス...	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。
バックアップ...	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。
バックアップグローバル...	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。
バックアップサーバー...	[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。
復元...	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。
グラントウィザード...	クリックして、GrantWizard ツールにアクセスします。
バックアップのスケジュール	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。
20 ラージュマネージャー	クリックして StorageManager を開き、バックアップのアップロード、削除、またはダウンロードします。

The Help Menu

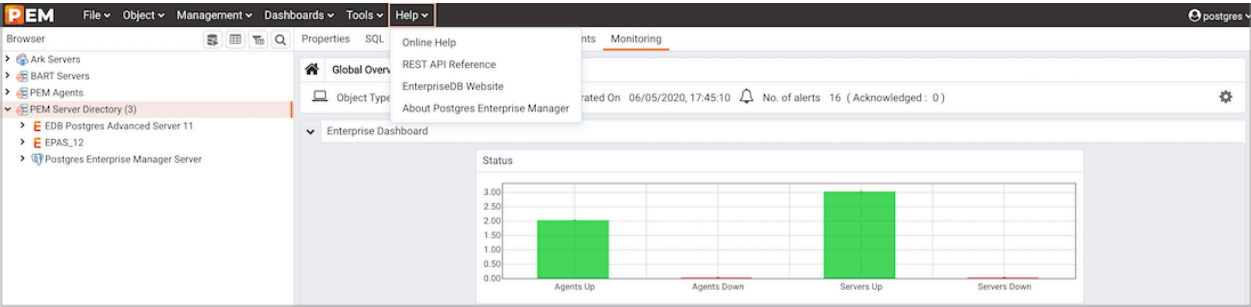


図 11 ヘルプメニュー

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。

オプション	アクション
オンラインヘルプ	クリックして PostgresEnterpriseManager のドキュメントを開きます。
RESTAPI リファレンス	クリックして RESTAPI リファレンスを開きます。
EDB ウェブサイト	クリックして、ブラウザウィンドウで EDBWeb サイトを開きます。
PostgresEnterpriseManager について	クリックして、PostgresEnterpriseManager のバージョン情報とユーザー情報を見つけます。

1.3.2 チャート、グラフ、表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、現在のユーザーセッション用に 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。

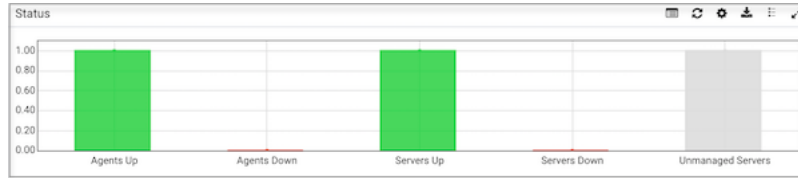


図 12 PEM クライアントチャートコントロールアイコン

アイコンを選択して:

	チャート、グラフ、またはテーブルに関する情報を表示します。
	チャート、グラフ、または表のコンテンツを更新します。
	現在のユーザーのチャート、グラフ、またはテーブル設定をカスタマイズします。
	チャートまたはグラフの画像をダウンロードします。
	チャート、グラフ、またはテーブルで使用されている凡例を表示します。
	チャートまたはグラフをフルスクリーンに展開します。

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ドキュメント

PEM には、ツールの使用を支援する組み込み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。

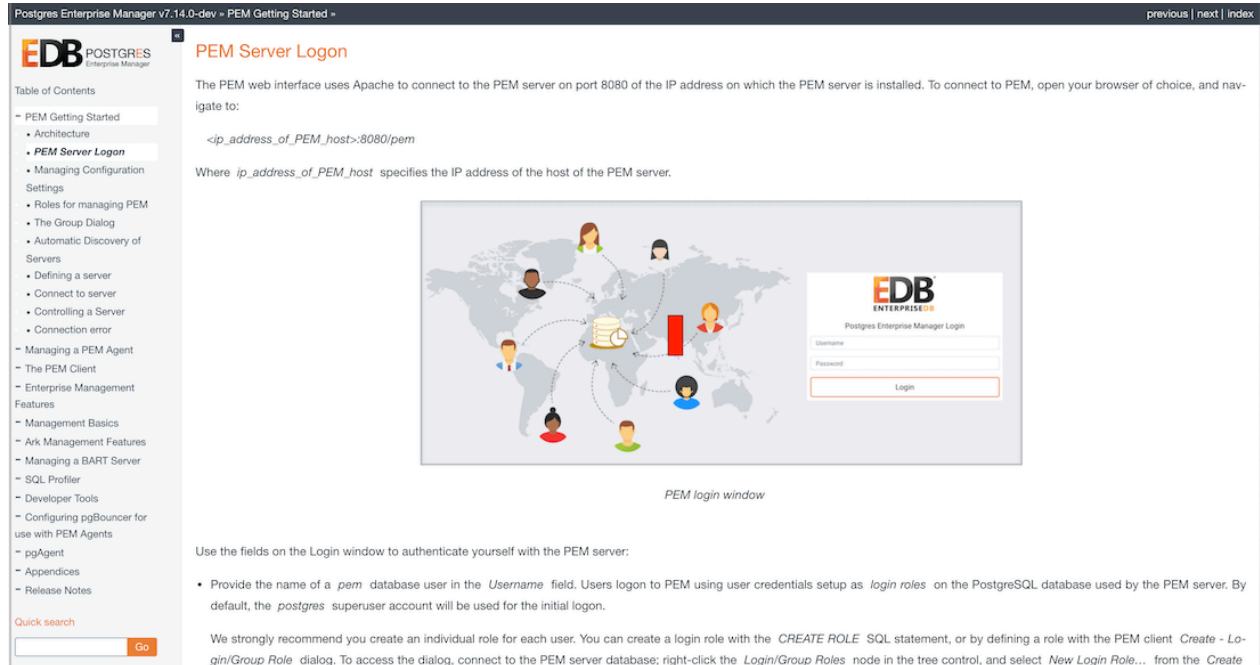


図 13 PEM オンラインヘルプ

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

第 2 章

Registering a Server

PEM でサーバーを管理または監視する前に、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（別のホストに存在するエージェント）にバインドできますが、エージェントが同じホストに存在しない場合、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 サーバーの手動登録

PEM で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。

- AdvancedServer 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。

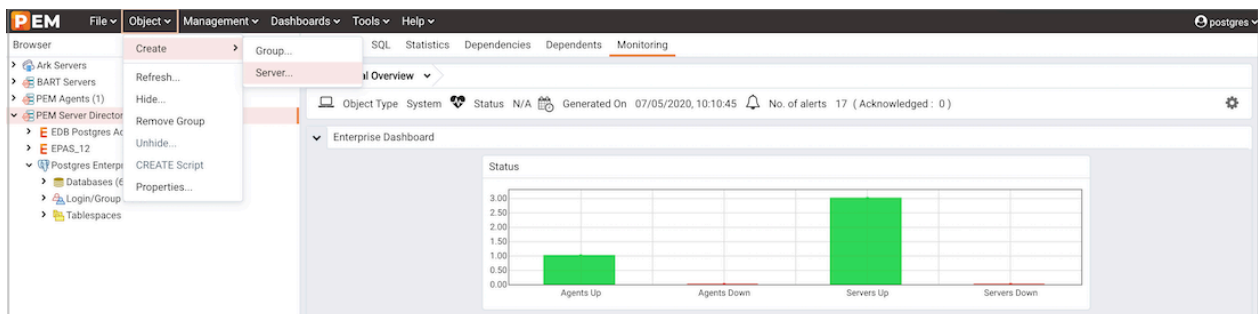


図 1 [作成-サーバー] ダイアログにアクセスする

注釈: 登録する PostgreSQL サーバーの `pg_hba.conf` ファイルが、接続を試みる前に 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確認する必要があります。

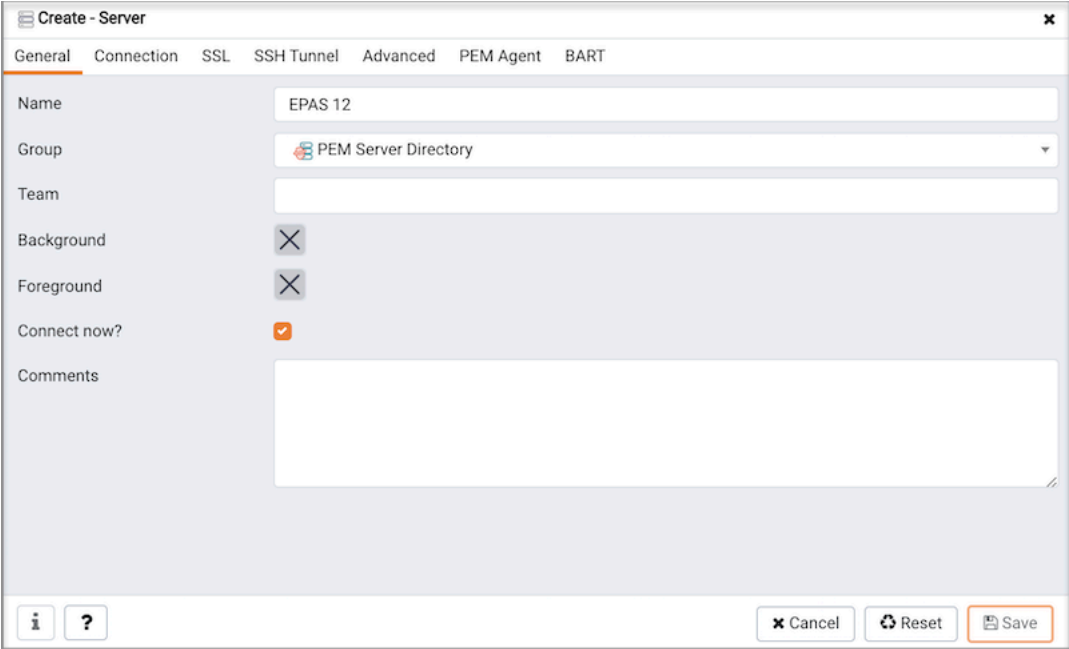


図2 [作成-サーバー] ダイアログの [全般] タブ

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、サーバーとエージェントをツリーコントロールで整理できます。グループを使用すると、多数のサーバーをより簡単に管理できます。たとえば、実稼働グループ、テストグループ、または LAN 固有のグループ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 ロール名を指定します。このロールのメンバーである、サーバーを最初に作成した、または PEM サーバーでスーパーユーザー特権を持つ PEM ユーザーのみが、PEM にログオンしたときにこのサーバーを表示します。このフィールドを空白のままにすると、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。
- [1] 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。

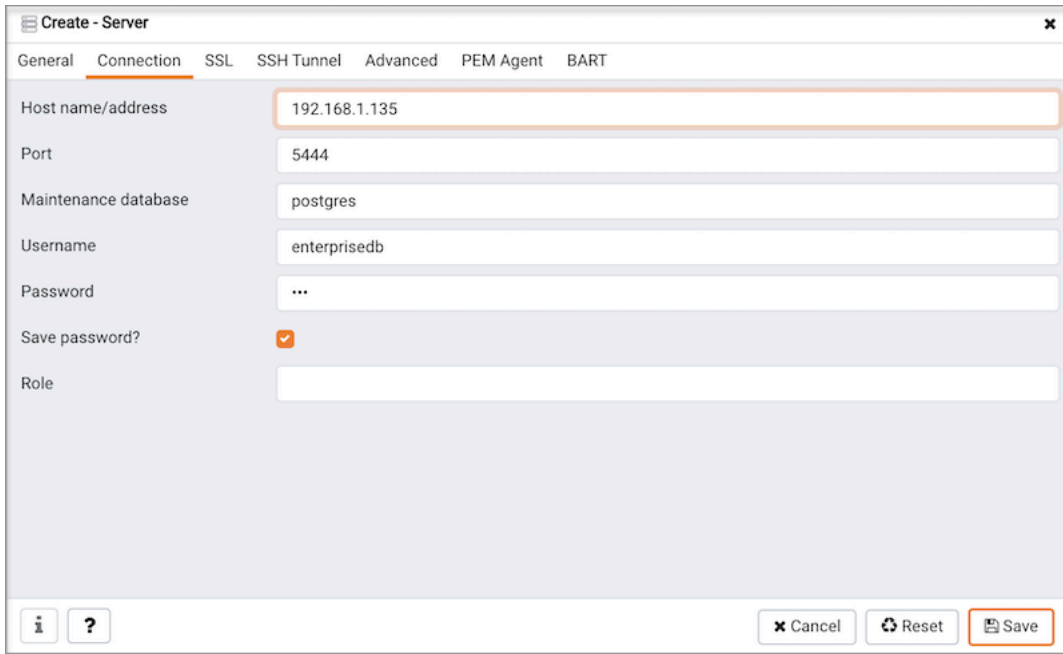


図3 [作成-サーバー] ダイアログの [接続] タブ

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、ローカルマシンでデフォルトの PostgreSQLUnix ドメインソケットを使用するためにアドレスフィールドを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。このデータベースには、pgAgent スキーマと adminpack オブジェクトがインストールされます (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。
- 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。
- Password フィールドに指定し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? の横にあるチェック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形式で PEM バックエンドデータベースに保存するよう 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、次回データベースサーバーに接続します。保存したパスワードを削除するには、最

初にデータベースサーバーを切断してから、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、サーバーへの接続後にクライアントが使用する必要があ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 つの役割として接続し、接続が確立されたときに別の役割（このフィールドで指定した役割）の許可を引き受けることができます。接続する役割は、指定された役割のメンバーでなければなりません。

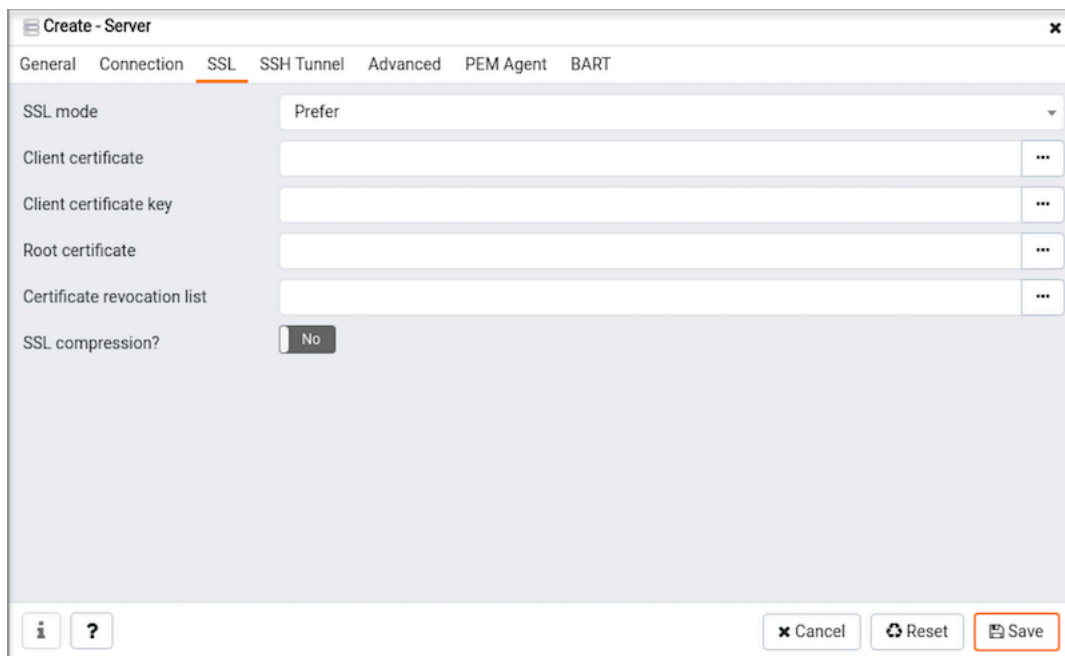


図 4 [作成-サーバー] ダイアログの [SSL] タブ

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>

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` を置き換えます。SSL 接続が確立されない場合、このパラメーターは無視されます。

- 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。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、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、`<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 接続が確立されない場合、このパラメーターは無視されます。
- 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、このパラメーターは無視されます。

警告: 証明書、秘密鍵、および失効リストは、PEM サーバ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。

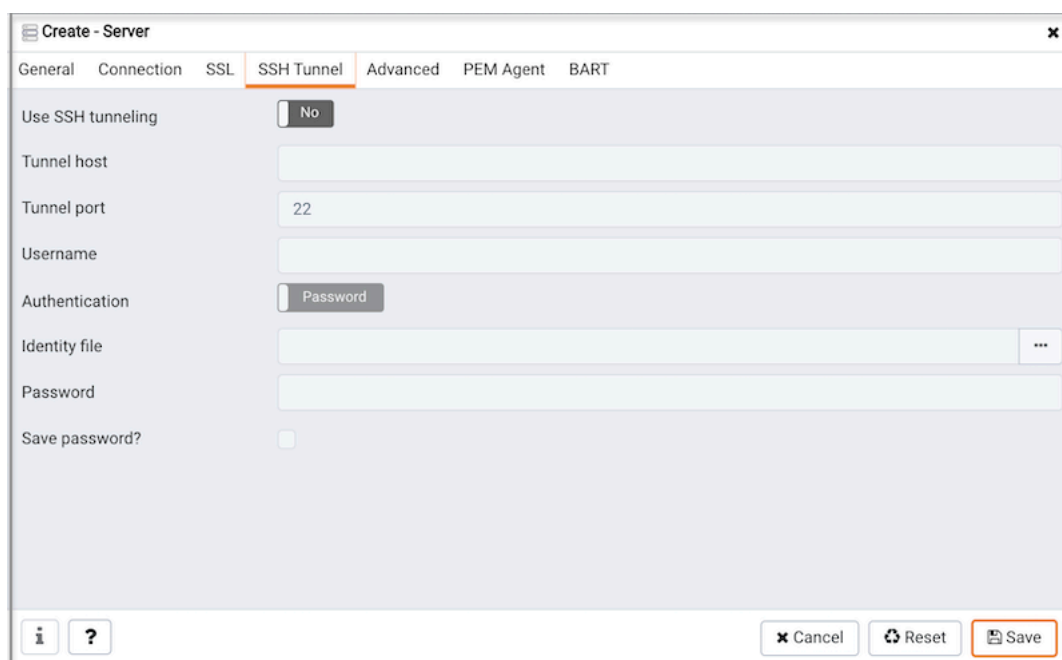


図 5 [作成-サーバー] ダイアログの [SSH トンネル] タブ

SSH Tunnel タブのフィールドを使用して、SSH トンネリングを設定します。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。
- 「クライアント接続が転送される」SSH ホストの名前または IP アドレスを Tunnel host フィールドに指定します。
- 「クライアント接続が転送される」SSH ホストのポートを Tunnel port フィールドに指定します。
- Username フィールドに SSH ホストのログイン特権を持つユーザーの名前を指定します。
- Authentication フィールドで、SSH ホストに接続するときに使用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。これがデフォルトです。
- Identity file を選択して、PEM が接続時に秘密鍵ファイルを使用することを指定します。
- SSH ホストが認証用の秘密キーファイルを予期している場合、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フレーズを指定します。

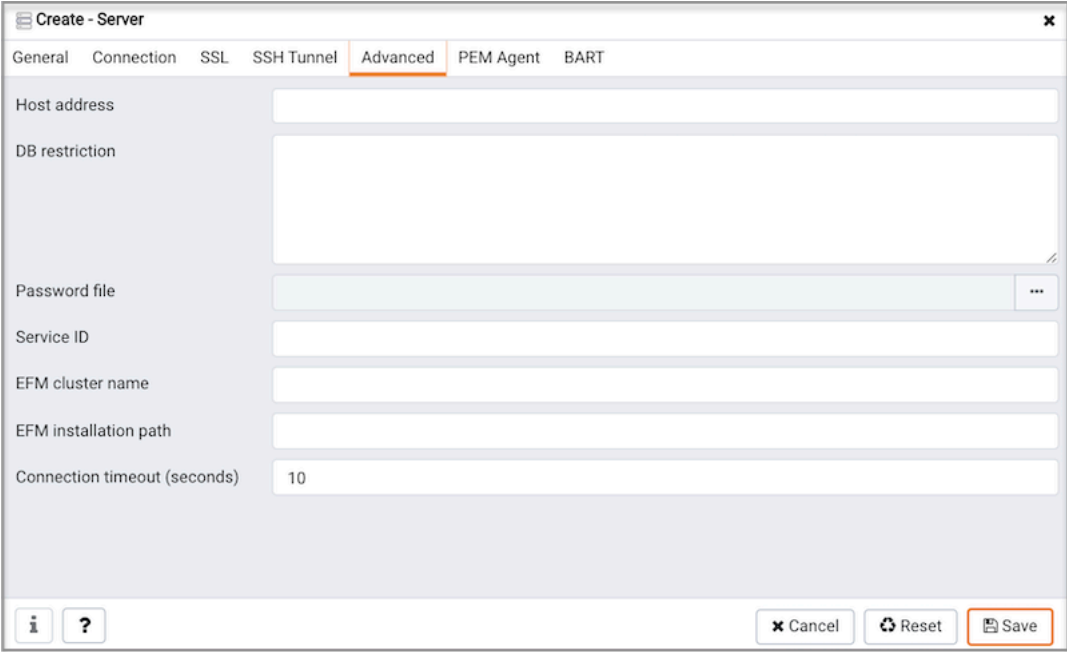


図 6 [作成-サーバー] ダイアログの [詳細設定] タブ

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ーホストの IP アドレスを指定します。
- DB restriction フィールドを使用して、`pg_database`<<http://www.postgresql.org/docs/current/interactive/catalog-pg-database.html>>`\_\_テーブルに対して使用される SQL 制限を指定し、ツリーコントロールに表示されるデータベース。たとえば、`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に指示できます。
`pg\_namespace`<<http://www.postgresql.org/docs/current/interactive/catalog-pg-namespace.html>>`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の場所を指定します (.pgpass)。 .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 の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>

注釈: パスワードファイルの使用は、PEM が libpqv10.0 以降を使用してサーバーに接続している場合にのみサポートされます。

- 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。EnterpriseManager 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、サーバーの起動に使用される init スクリプトの名前は /etc/init.d で、サーバーを起動する systemd スクリプトの名前は systemctl です。たとえば、AdvancedServer11 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。リモートサーバーの場合は、マシン名を先頭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/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。 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 FailoverManager 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交換できます。PEM がフェールオーバーマネージャ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、FailoverManager 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 を使用してフェールオーバーマネージャークラスターの状態を監視している場合は、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バイナリファイルの場所を指定します。デフォルトでは、FailoverManager 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 FailoverManager のバージョンを指定します。

図7 [作成-サーバー] ダイアログの [PEM エージェント] タブ

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。

- 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、EnterpriseManager 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。
- 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およびダッシュボードで利用できず、リモートサーバーは一部の 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、PostgresExpert など）からアクセスできません（チューニングウィザード）。
- 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 Host フィールドに入力します。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`。
- サーバーに接続するとき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 動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。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.enterprisedb.com/edb-docs/d/postgresql/reference/manual/12.1/libpq-ssl.html>

- 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データベースの名前を指定します。
- 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のスーパーユーザーで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。

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、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、以下を参照してください。

<https://www.postgresql.org/docs/current/default-roles.html>

- Password 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パスワードを指定し、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は、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上に存在し、アクセス可能でなければなりません。
- 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、たとえば、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、エージェントはデータベースサーバーの監視を行うことができます。

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 コンソールで設定されたすべての 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ー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ーの 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ーを指定します。このデータベースサーバーの増分バックアップを有効にする場合、ユーザーはスーパーユーザーである必要があります。

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.

Buttons: Cancel, Reset, Save

図 8 [サーバーの作成] ダイアログ (BART-[一般] タブ)

- 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 AdvancedServer データベースクラスターの場合は `enterprisedb`、PostgreSQL データベース互換モードでインストールされた PostgreSQL データベースクラスターおよび AdvancedServer 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 コマンドに入力すると、データベースサーバーの `Postgresql.conf` ファイルが上書きされます。サーバーが追加されると、データベースサーバーが再起動されるか、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ー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ーと 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 認証をセットアップする前に、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ーでは機能しません。

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

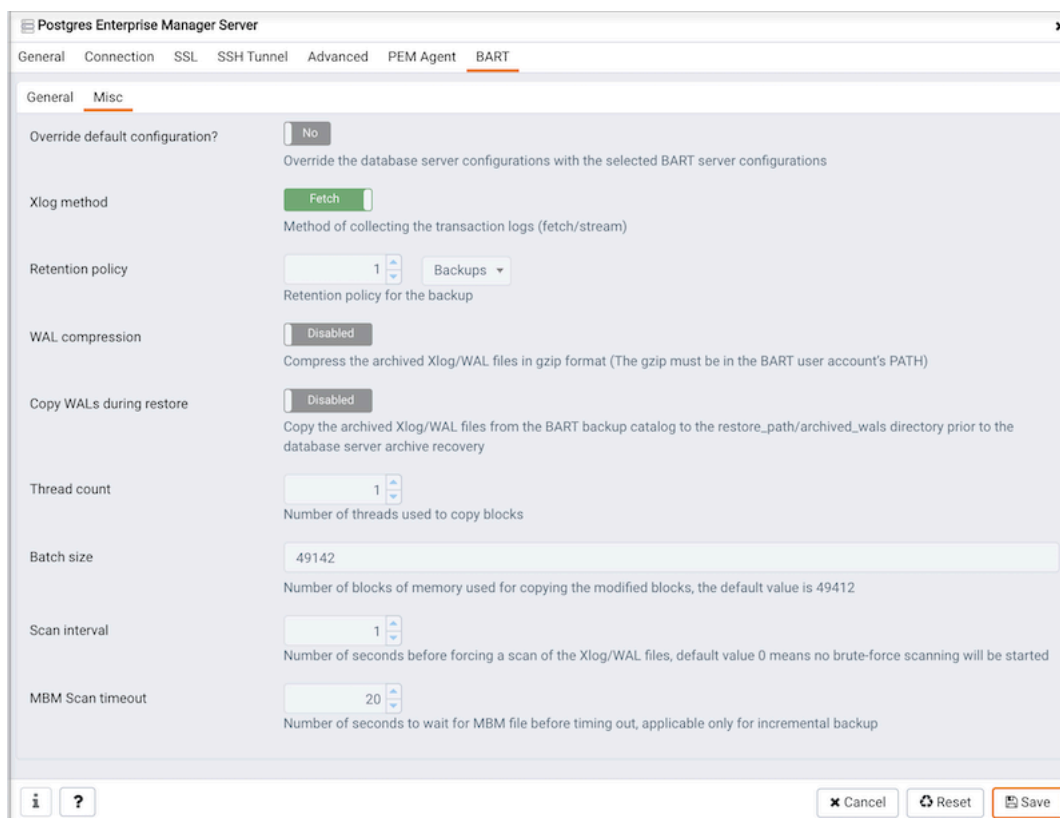


図9 [作成-サーバー] ダイアログ (BART-[その他] タブ)

- `Override default configuration?` スイッチを使用して、特定のデータベースサーバー構成で 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。
- `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` 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、`gzip` 圧縮プログラムが BART ユーザーアカウントの `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` 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。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 BART バックアップカタログからアーカイブされた `WAL` ファイルを直接取得します。

- 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取得する場合は、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、MBM ファイルがタイムアウト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

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 プロンプト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。

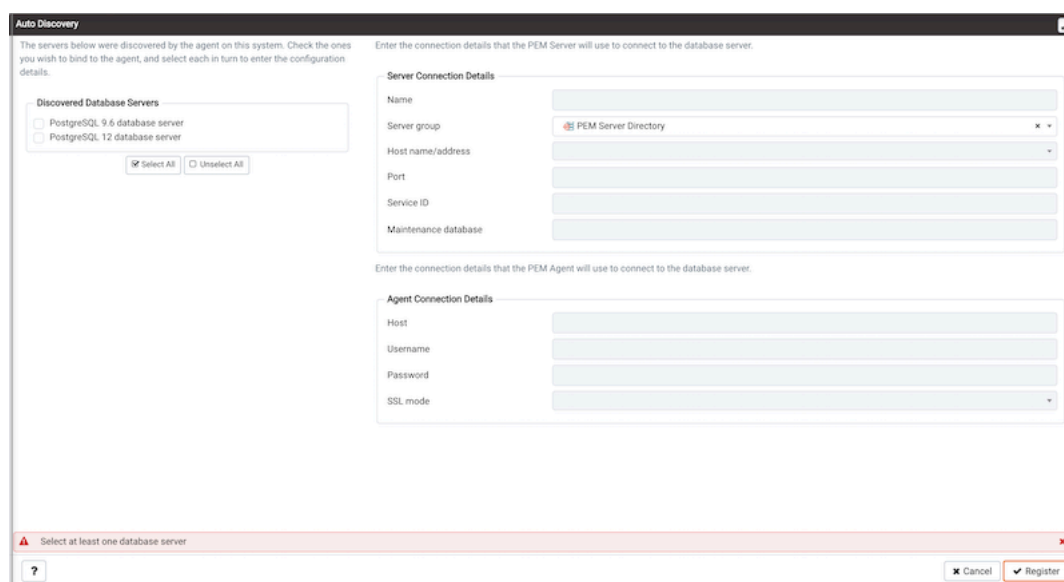


図 10 PEM 自動検出ダイアログ

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 が監視するサーバーに関する情報を提供します。

- Name フィールドで監視対象サーバーの名前を受け入れるか、変更します。指定した名前は、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、サーバーが PEM クライアントツリーコン

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。インストールの Maintenance database フィールドの内容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 エージェントバインディングに使用される IP アドレスが表示されます。
- 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。

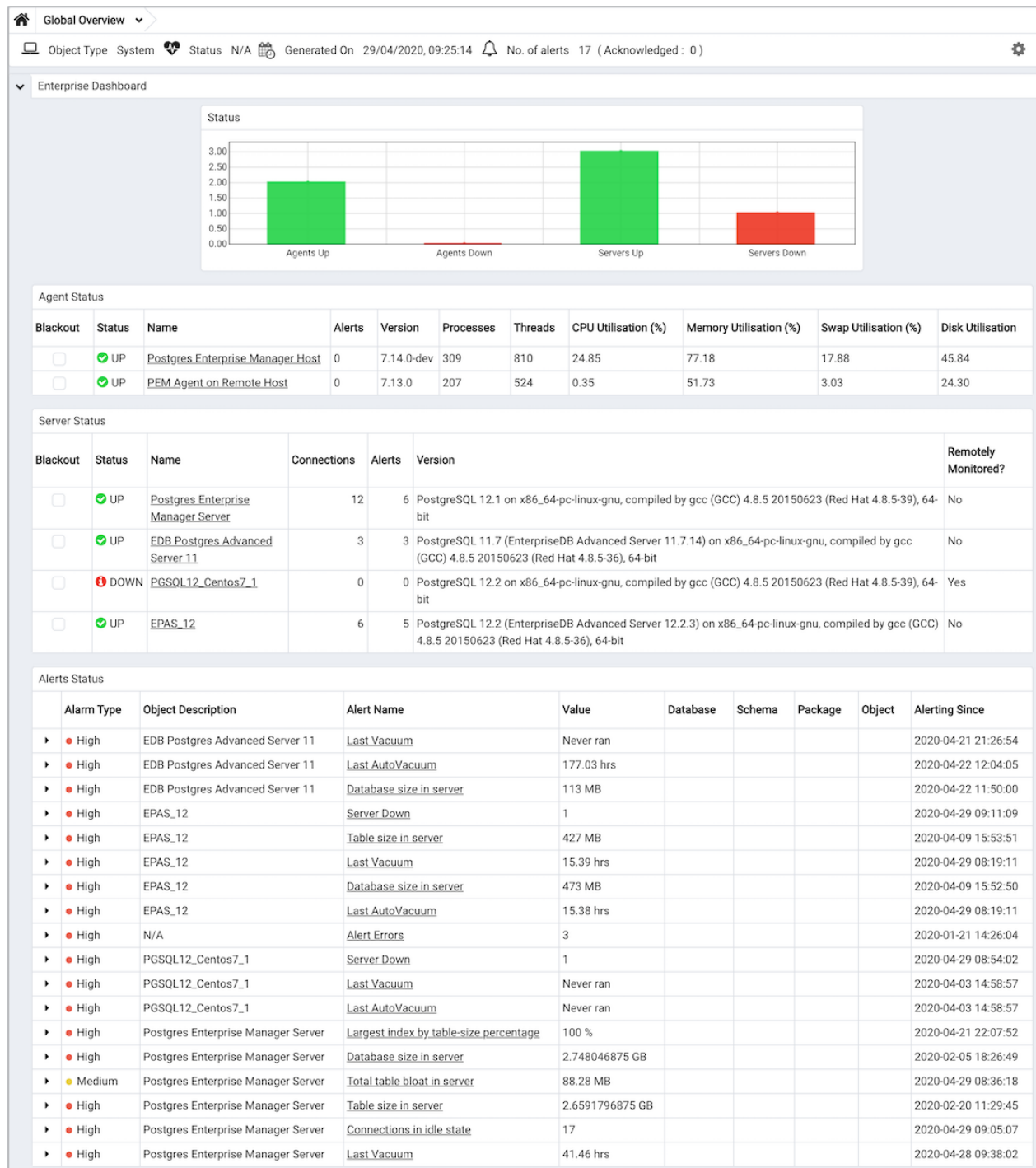


図 11 登録済みサーバー

2.3 pemworker ユーティリティを使用してサーバーを登録する

pemworker ユーティリティを使用して、PEM サーバーによる監視用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、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、コマンドラインオプションをコマンド文字列に追加します。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。

オプション	説明
<code>--pem-user</code>	PEM 管理ユーザーの名前を指定します。必須です。
<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、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アドレスフィールドを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>	サーバーのモニター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。必須です。
<code>--server-service-name</code>	登録されるサーバーでの操作を制御するデータベースサービスの名前を指定します (STOP、START、RESTART など)。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 (はい) の場合、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、キャパシティマネージャー、ログマネージャーなど) からリモートサーバーにアクセスできません。PostgresExpertandTuningWizard)。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>	データベースで認証するとき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役割の名前を指定します。

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

ベースサーバーのパスワードを指定します。

パスワードの入力に失敗すると、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ー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、コマンドラインオプションをコマンド文字列に追加します。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。

オプション	説明
--pem-user	PEM 管理ユーザーの名前を指定します。必須です。
--server-addr	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、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アドレスフィールドを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。
--server-port	ホストのポート番号を指定します。必須です。

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。

パスワードの入力に失敗すると、パスワード認証エラーが発生します。PEM サーバーは、サーバーが登録解除されたことを確認します。

2.4 接続とバインドの検証

登録されると、新しいサーバーが PEM の Browser ツリーコントロールに追加され、Global Overview に表示されます。

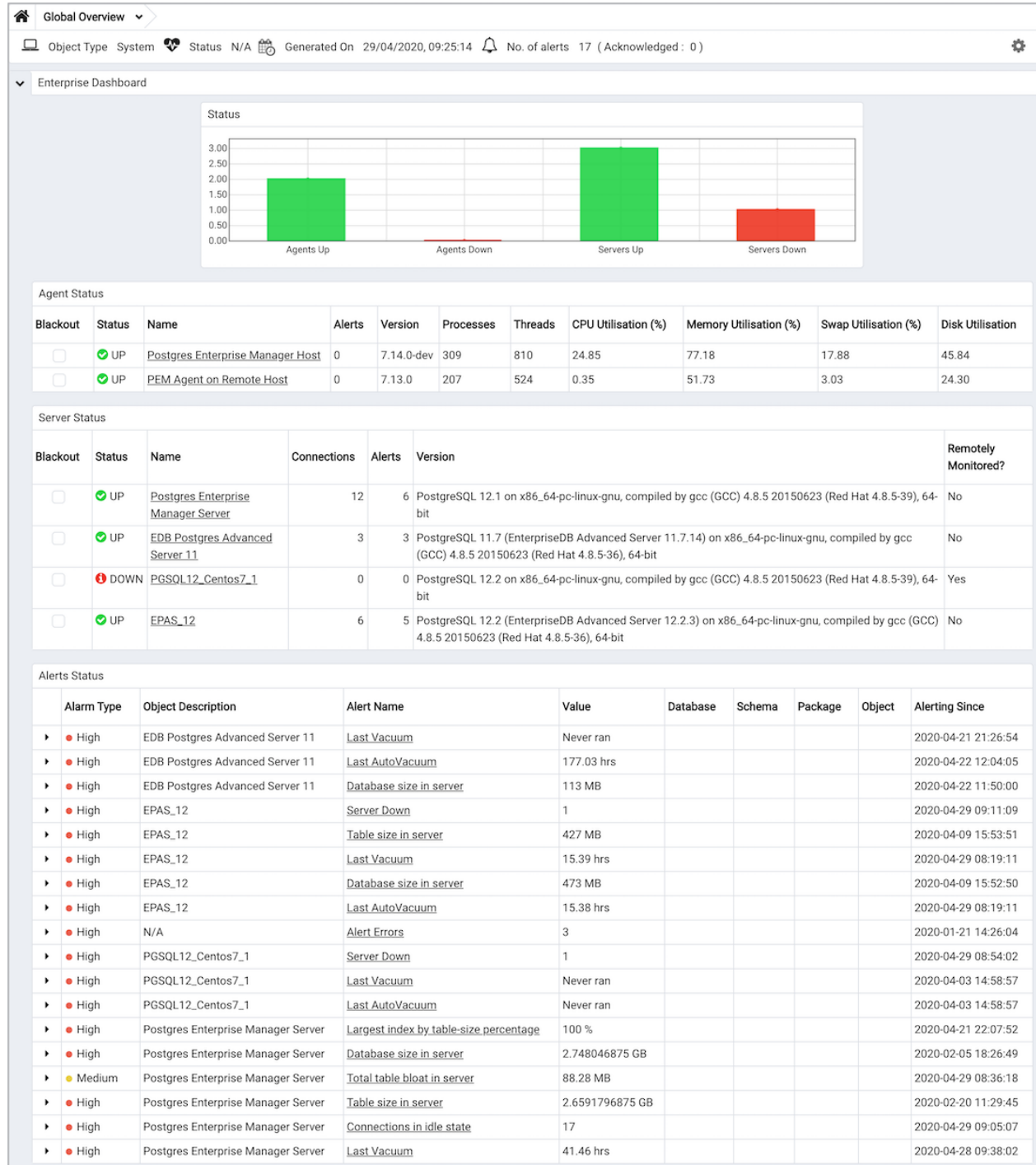


図 12 グローバル概要ダッシュボード

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、サーバーリストで「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ブを実行してサーバーを検査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 リストに追加されたことが示されます。

5 分経過しても Global Overview ダッシュボードに新しいサーバーが表示されない場合は、監視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。

第 3 章

Defining and Monitoring Postgres instances on AWS

PEM を使用して AWS ホスト上の Postgres インスタンスを監視できるシナリオは 2 つあります。以下を監視できます。

- AWSEC2 で実行されている Postgres インスタンス
- AWSRDS で実行されている Postgres インスタンス

3.1 AWSEC2 で実行されている Postgres インスタンスのモニタリング

AWSEC2 で Postgres インスタンスを作成した後、PEM サーバーを使用してインスタンスを登録および監視できます。現在、次のシナリオがサポートされています。

- 同じ AWSEC2 で実行されている Postgres インスタンスと 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 インスタンスと PEM エージェント、および AWSEC2 で実行されている PEM サーバー。
- 同じ AWSEC2 で実行されている Postgres インスタンスと PEM エージェント、および異なる AWSEC2 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、AWSEC2 で VPN を設定する必要があるため、AWSEC2 インスタンスは pem データベースにアクセスできます。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。

AWSEC2 またはローカルマシンで実行されている PEM エージェントを PEM サーバーに登録する必要があります。PEM エージェントを PEM サーバーに登録するときは、AWSEC2 インスタンスのホスト名を使用する必要があります。

あることに注意してください。PEM エージェントの登録の詳細については、[PEMSelfRegistration](#) を参照してください。

Create - Server ダイアログを使用して、PEM サーバー上の AWSEC2 で実行されている Postgres インスタンスを登録できます。Create - Server ダイアログを使用したサーバーの登録の詳細については、サーバーの登録<registering_server>を参照してください。`Create - Server` ダイアログの PEM Agent タブを使用して、登録済みの PEM エージェントを Postgres インスタンスにバインドします。

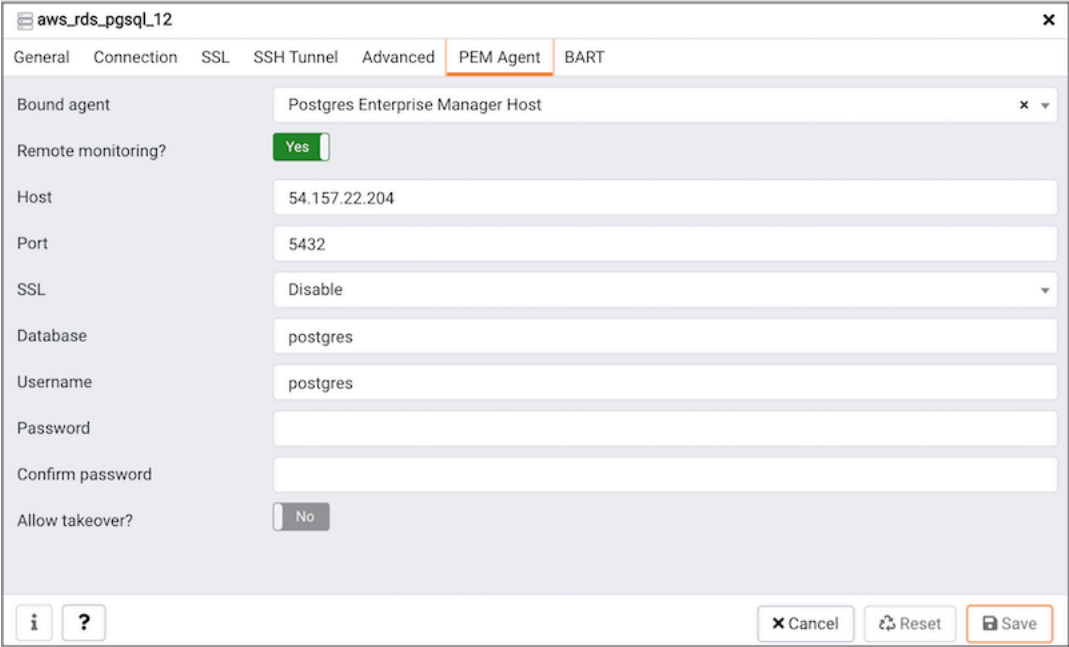
PEM エージェントが PEM サーバーに登録され、AWSEC2 で実行されている Postgres インスタンスが PEM サーバーに登録されている場合、PEM でインスタンスを監視できます。

3.2 AWSRDS で実行されている Postgres インスタンスのモニタリング

AWSRDS 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。AWS で Postgres (RDS) インスタンスを作成した後、Create - Server ダイアログを使用して Postgres (RDS) インスタンスを PEM サーバーに追加します。このダイアログを使用して、新しいサーバー接続を記述し、サーバーを PEM エージェントにバインドし、サーバーを PEM ブラウザーツリーコントロールに表示できます。

Create - Server ダイアログと各タブの設定の詳細については、[サーバーの登録](#) を参照してください。

PEM サーバーを使用して AWS インスタンスの Postgres (RDS) インスタンスを監視するには、Create - Server ダイアログの PEM Agent タブの Remote Monitoring フィールドを Yes に設定する必要があります。



General	Connection	SSL	SSH Tunnel	Advanced	PEM Agent	BART
Bound agent: Postgres Enterprise Manager Host						
Remote monitoring?: Yes						
Host: 54.157.22.204						
Port: 5432						
SSL: Disable						
Database: postgres						
Username: postgres						
Password:						
Confirm password:						
Allow takeover?: No						
[i] [?] [Cancel] [Reset] [Save]						

図 1 サーバーの作成ダイアログ-PEM エージェントタブ

PEM エージェントは Postgres (RDS) AWS インスタンスをリモートで監視するため、以下に説明するように機能が制限されます。

Feature Name	Works with re- mote PEM Agent	Comments
監査マネージャー	いや	
キャパシティマネージャー	限定	データベースサーバ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。
ログマネージャー	いや	
アラートを管理する	限定	データベースサーバーでアラートスクリプトを実行すると、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、実際のデータベースサーバーマシンでは実行されません。
チャートを管理する	はい	
ダッシュボードを管理する	限定	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。たとえば、データベースサーバーの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。
プローブを管理する	限定	PEM プローブの一部は情報を返さず、機能の一部が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 エージェントガイド< https://enterprisedb.com/edb-docs >」を参照してください。
Postgres エキスパート	限定	PostgresExpert 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。
Postgres ログ分析エキスパート	いや	PostgresLogAnalysisExpert は、ログマネージャーによってインポートされたログに依存しているため、分析を実行できません。これは、必要に応じて機能しません。
スケジュールされたタスク	限定	スケジュールされたタスクは、データベースサーバーに対してのみ機能します。スクリプトはリモートエージェントで実行されます。
チューニングウィザード	いや	
システムレポート	はい	
コア使用レポート	限定	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。たとえば、プラットフォーム、コアの数、合計 RAM は表示されません。
BART の管理	いや	BART では、データベースサーバーと BART がインストールされている 2 台のマシン間でパスワードなしの認証が必要です。AWSRDS インスタンスはホストアクセスの使用を許可しません。

第 4 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切れそうになると、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）に追加され、証明書を使用して PEM サーバーに接続できなくなります。

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。

4.1 SSL 証明書の置き換え

次の手順では、既存の PEM インストールでの 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. 実行中のすべての PEM エージェントを、最初にサーバーホストで停止し、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、スーパーユーザー権限を引き受けて、次のコマンドを入力します。

Linux で init.d を使用する場合（例:Centos6）

```
/etc/init.d/pemagent stop
```

Linux で systemd を使用している場合（例:Centos7）

```
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 `PostgresEnterpriseManagerAgent` 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`。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、各ファイルに拡張子を追加して、名前を一意にします。

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

たとえば、次のコマンド:

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

エントリに `old` という単語が追加された `ca_certificate` ファイルのバックアップを作成します。

3. `openssl_rsa_generate_key()` 関数を使用して `ca_key.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/ca_key.key
```

`ca_key.key` ファイルを作成した後、`cat` は `ca_certificate.crt` 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` に内容を追加します。

```
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/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()` 関数を使用して、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()` 関数を使用して `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` は `server.crt` 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` に内容を追加します。

```
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。 `server.crt` ファイルに証明書を保存します。簡単にするために、まず一意の名前を持つ一時ファイルに `SQL` クエリを配置します。

```
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。

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

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

`systemd` の Linux の場合; たとえば、Centos7 ホストの場合:

```
systemctl restart postgresql-x
```

4.2 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 テーブルに格納される一意の識別子があります。キーおよび証明書ファイルを、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（ステップ 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
```

* On a 32-bit Windows installation, check:

```
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、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
```

Modify the privileges of the ``agent.key`` file:

```
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 の Linux では、次のコマンドを使用してサービスを再起動します。

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

systemd の Linux では、次のコマンドを使用してサービスを再起動します。

```
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 5 章

Managing Configuration Settings

複数の設定ファイルは、起動時に `PostgresEnterpriseManager` 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、構成設定の参照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 `PostgresEnterpriseManager` の配布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 `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` の後に読み取られ、エンドユーザーがローカルの設定や標準に合わせて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、ユーザーが作成できます必要に応じて `config.py` と同じディレクトリに。

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。

第 6 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、プロキシサーバーなどを使用するように PEM を変更

6.1 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 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 ユーティリティには、Windows の 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、サービス名を強調表示します。

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）は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、EDBWeb サイトから入手できる*EDBPostgresAdvancedServer インストールガイド*を参照してください。

<https://www.enterprisedb.com/resources/product-documentation>

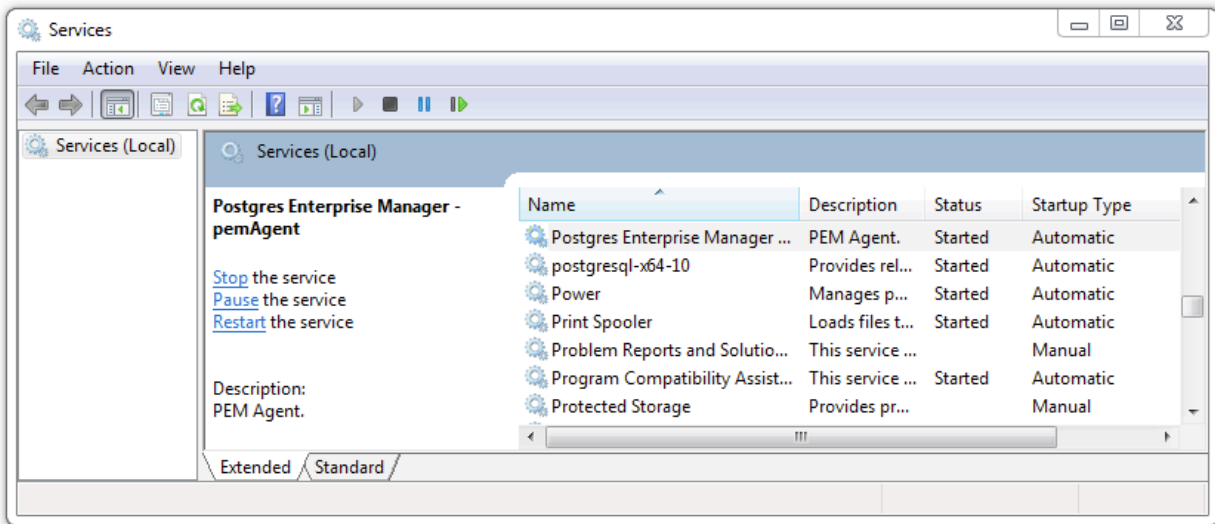


図 1 Windows サービスウィンドウの PEM サービス

6.2 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て、データベースサーバーを PEM エージェントに登録し、以下を実行します。

- 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。
 - AdvancedServer 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。

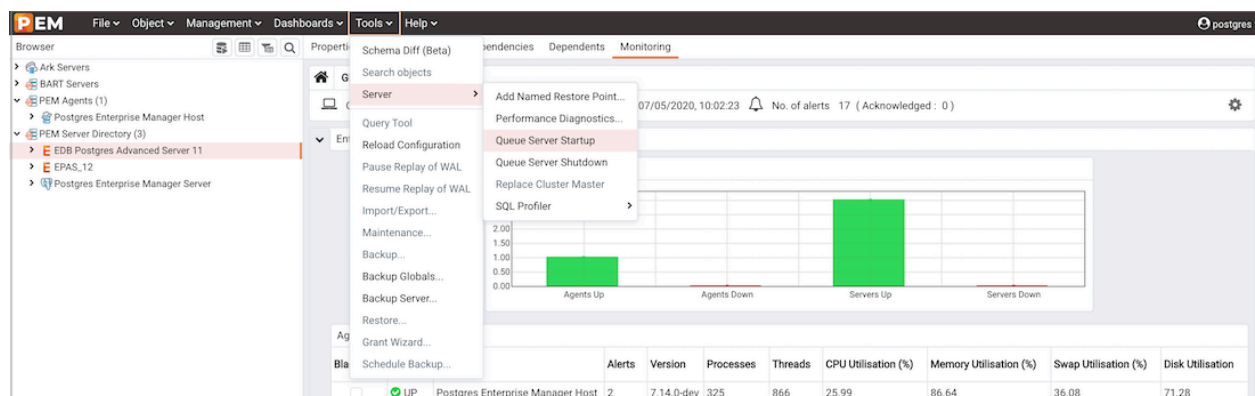


図 2 管理対象サーバーの [ツール] メニュー

6.3 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:

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

x はサーバーのバージョン番号を示します。

サービスを制御するには、サービススクリプトを使用できます。

- 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、スーパーユーザー権限を引き受けて、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` でサービスを開始します。
- `stop` はサービスを停止します。
- `restart` でサービスを停止してから開始します。
- `status` は、サービスのステータスを確認します。

6.4 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
- AdvancedServer 上の PEM サーバーホストが edb-as-<x> または ppas-<x>
- PEM エージェントは Postgres Enterprise Manager - pemAgent です

ここで、*x* はサーバーのバージョン番号を示します。

Windows の 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。

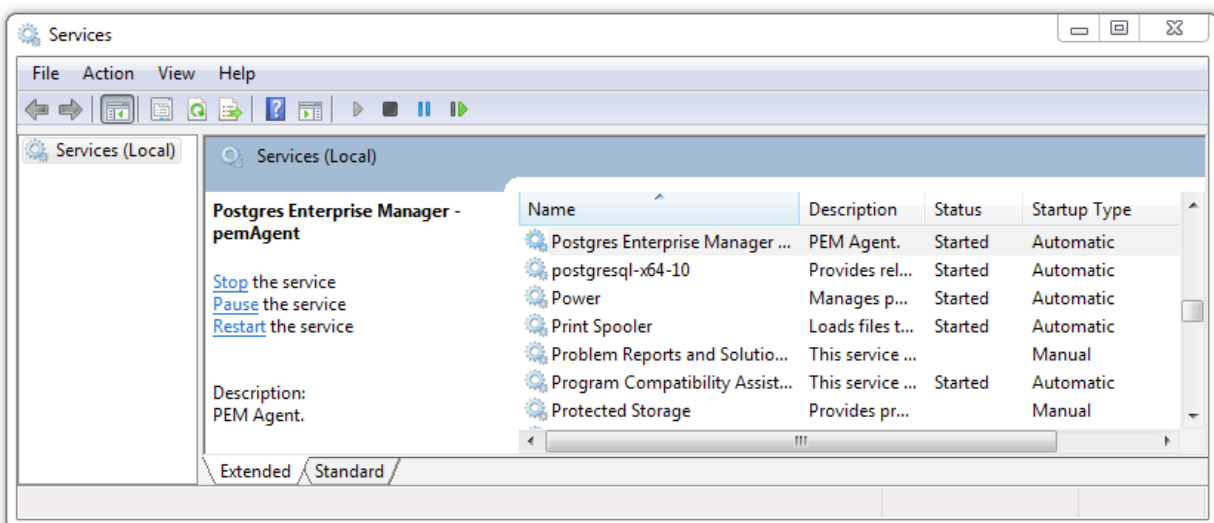


図3 Windows サービスウィンドウ

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ー（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再ロードするように 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することなく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。

6.5 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する場合、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 RHEL7.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 では、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。

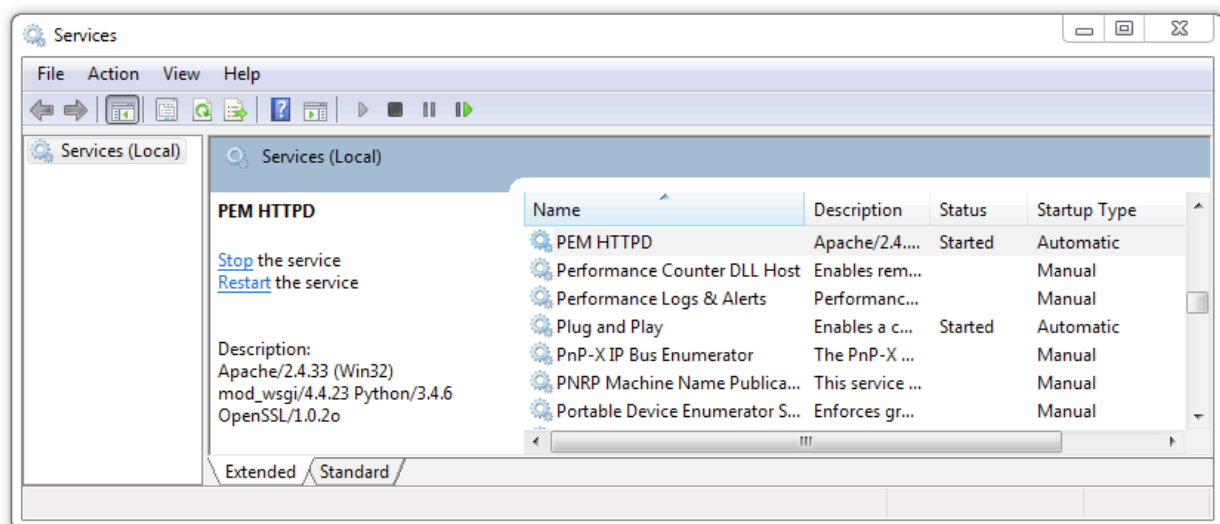


図 4 PEMHTTPDWindows サービス

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 を開始します。

6.6 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 バックエンドデータベースの pg_hba.conf ファイルの先頭に追加されます。

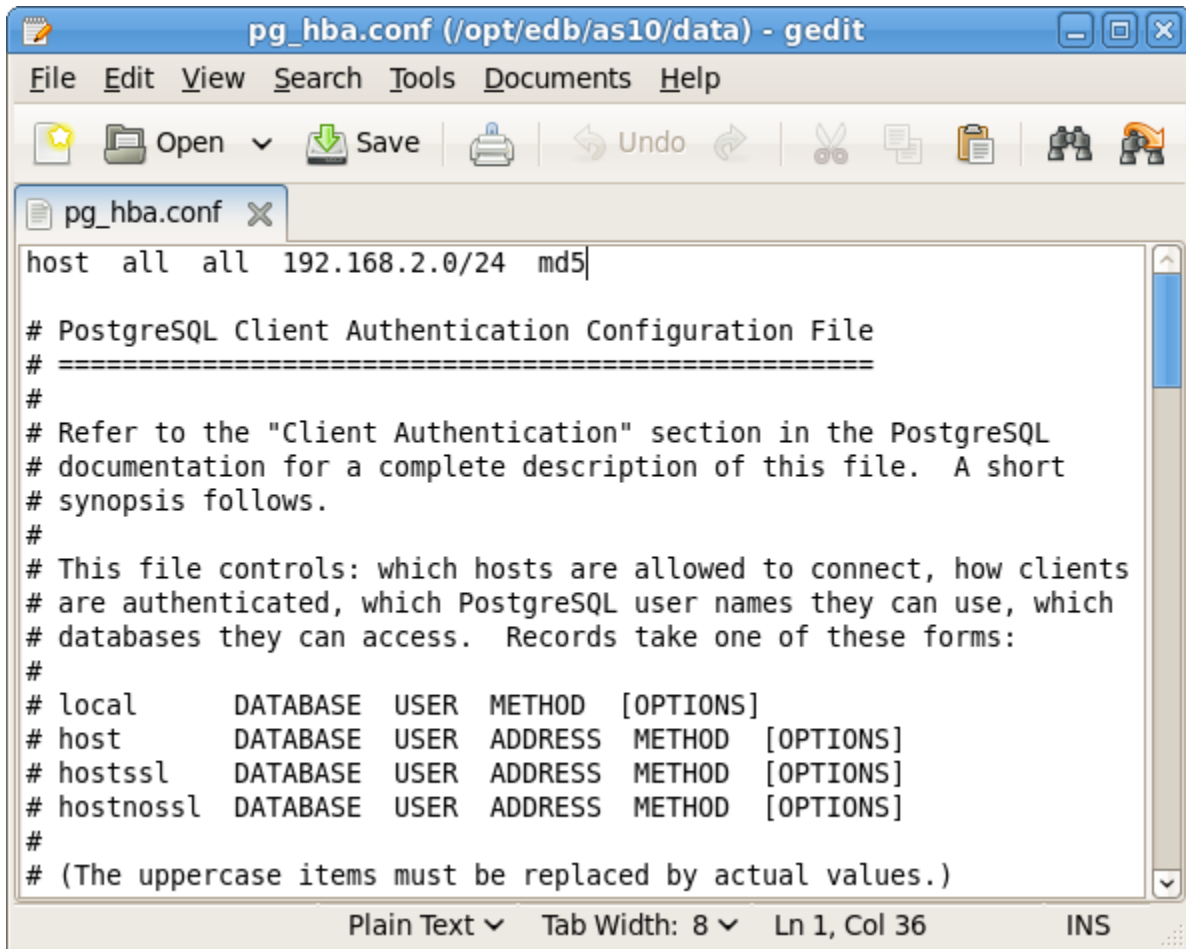


図 5 pg_hba.conf ファイルの PEM エントリ

また、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。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、PEM-HTTPD がサーバーに接続できるように PEM サーバーホストの pg_hba.conf ファイルを変更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。たとえば、AdvancedServer10 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、アドレスが `192.168.10.23`、`192.168.10.76`、`192.168.10.184` のサーバーからの接続を許可する場合は、`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` ファイルエントリのフォーマットの詳細については、次の PostgreSQL コアドキュメントを参照してください。

<http://www.postgresql.org/docs/current/auth-pg-hba-conf.html>

PEM で Postgres サーバーに接続する前に、両方のサーバーの `pg_hba.conf` ファイルが接続を許可し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を追加します。

6.7 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。

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。

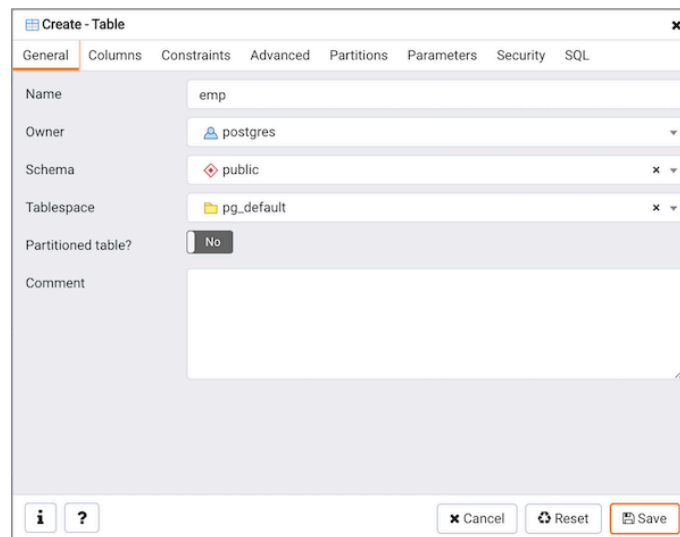


図 6 テーブルの作成-一般タブ

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
- 景色
- 制約など

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、名前付き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。

6.8 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
- 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 という名前のユーザーが、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、次の形式を取ります。

```
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 `lsafePWD` である postgres という名前のユーザーのパスワードファイルのエントリは、次の形式になります。

```
localhost:5432:*:postgres:lsafePWD
```

パスワードファイルの名前は通常、Linux システムでは `~root/.pgpass`、Windows では `%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、次の PostgreSQLWeb サイトにアクセスしてください。

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

Postgres がサポートする認証方法の詳細については、次の PostgreSQL コードドキュメントを参照してください。

<http://www.postgresql.org/docs/current/client-authentication.html>

6.9 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

Buttons: ? Cancel Reset Save

図 7 サーバー構成ダイアログ

パラメーター値を変更するには、パラメーター名の右側にある 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、パラメータを元の値に戻します。

6.10 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な方法を提供します。

6.10.1 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は postgres スーパーユーザーアカウントが使用されますが、接続ユーザーごとに個別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のため）。PEM クエリツール、PEMWeb インターフェイス 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。

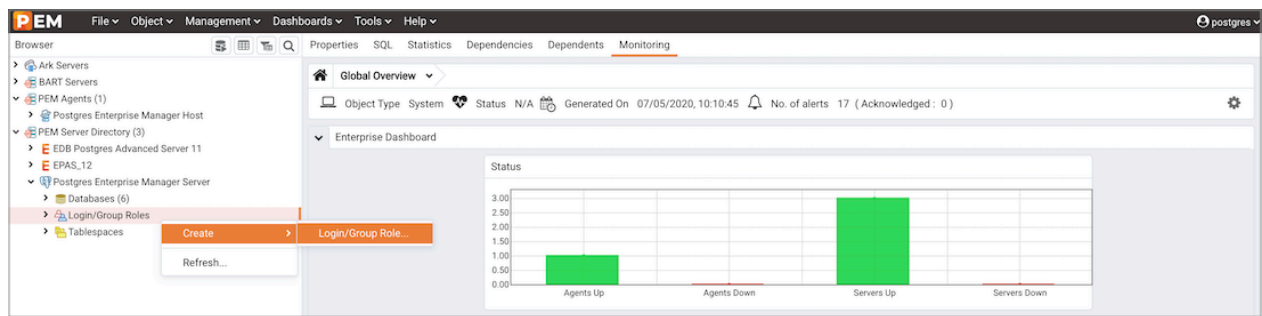


図 8 ログインロールノードのコンテキストメニュー

Create - Login/Group ロール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。

新しい役割の定義が完了したら、Save をクリックして役割を作成します。

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 オンラインドキュメントを参照してください。

<http://www.postgresql.org/docs/current/sql-createrole.html>

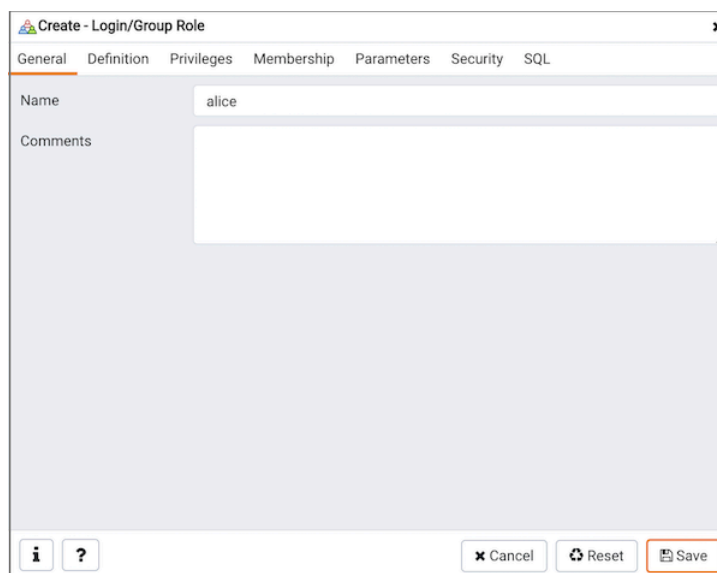


図 9 ログインロールの作成-[全般] タブ

6.10.2 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。グループロールの主な目的は、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簡単にすることです。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。

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る）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。ロールの作成に関する詳細については、次を参照してください。

<http://www.postgresql.org/docs/current/sql-createrole.html>

6.10.3 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、キャパシティマネージャ、SQL プロファイラなどの 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。

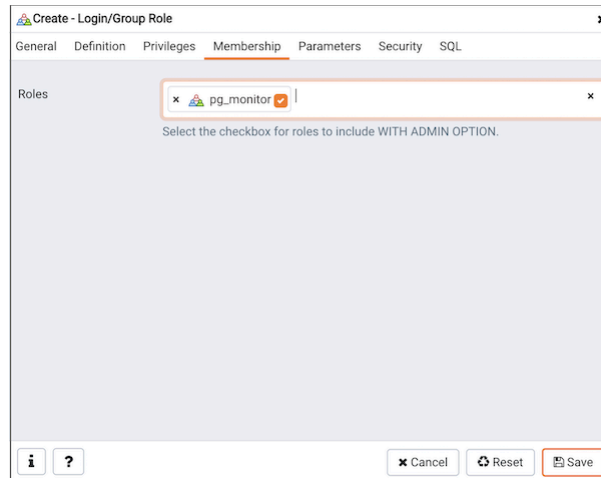


図 10 ログインロールの作成-[メンバーシップ] タブ

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、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。

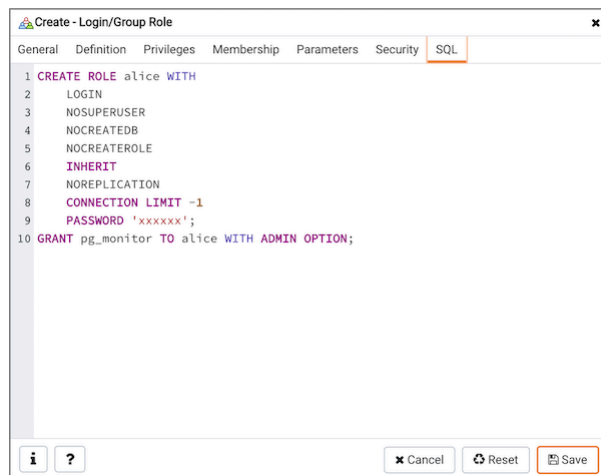


図 11 ログインロールの作成-[SQL] タブ

上記の例は、Audit Manager にアクセスできる `acctg_clerk` という名前のログインロールを作成します。役割は、いつでもサーバーに無制限に接続できます。

PEM の事前定義されたロールを使用して、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	説明
pem_super_admin		PostgresEnterpriseManager ですべてを管理/設定する役割。
pem_admin	pem_super_admin	すべての表示可能なエージェント/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	PostgresEnterpriseManager 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/実行する役割。
pem_rest_api	pem_admin	RESTAPI 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ーの再起動/再読み込みを許可するロール (server-id が提供されている場合)。
pem_manage_schedule_task	pem_admin	スケジュールタスクを構成する役割。
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。
pem_config_probe	pem_config、pem_manage_probe	表示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 (履歴保持、実行頻度、プローブの有効化/無効化) の役割。
pem_database_server_registration	pem_admin	データベースサーバーを登録するロール。
pem_comp_postgres_expert	pem_component	PostgresExpert を実行する役割。
pem_comp_auto_discovery	pem_component	データベースサーバーの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	CapacityManager を実行する役割。
pem_comp_log_manager	pem_component	LogManager 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。

6.10.4 チームロールを使用する

PEM による監視のためにサーバーを登録するときに、サーバーに関連付けられる*Team*を指定できます。チームは、1 つまたは複数の監視対象サーバーへのアクセスを、制限された役割メンバーの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役割です。PEM クライアントは、指定されたチームを持つサーバーのみを以下のユーザーに表示します。

- チームの役割のメンバー
- サーバーを作成した役割
- PEM サーバーでのスーパーユーザー特権を持つ役割。

チームの役割を作成するには、PEM ツリーコントロールで役割を配置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。

6.10.5 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権限を割り当てるには、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 [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。

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

6.11 ジョブ通知の管理

システムが生成したジョブ（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。これらの電子メール通知設定は、指定されたユーザーグループに電子メール通知を送信するために、次の 3 つのレベルで（優先順に）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）

6.11.1 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、ジョブレベルで電子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、ジョブ自体の作成中に 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て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、ジョブレベルで電子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、電子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.

Buttons: Cancel, Reset, Save

図 12 ジョブ通知の設定-ジョブレベル

6.11.2 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。

[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電子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。

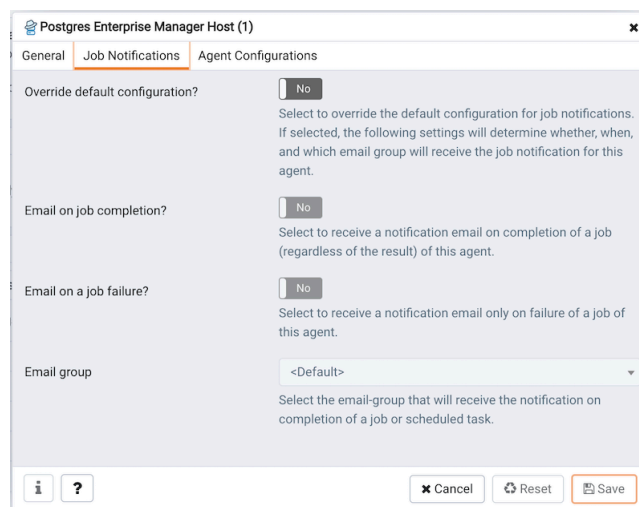


図 13 ジョブ通知の設定-エージェントレベル

- Email group フィールドを使用して、ジョブ通知の送信先の電子メールグループを指定します。

6.11.3 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。

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 three buttons: a help button with a question mark, a 'Cancel' button, a 'Reset' button, and a 'Save' button.

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 14 ジョブ通知の設定-サーバーレベル

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、電子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、ステータスが失敗、成功、または中断に関係なく、各ジョブステータスの変更後に電子メール通知を送信するかどうかを指定します。

6.12 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、指定した順序でカスタム定義の一連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。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取るジョブを作成し、毎月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。

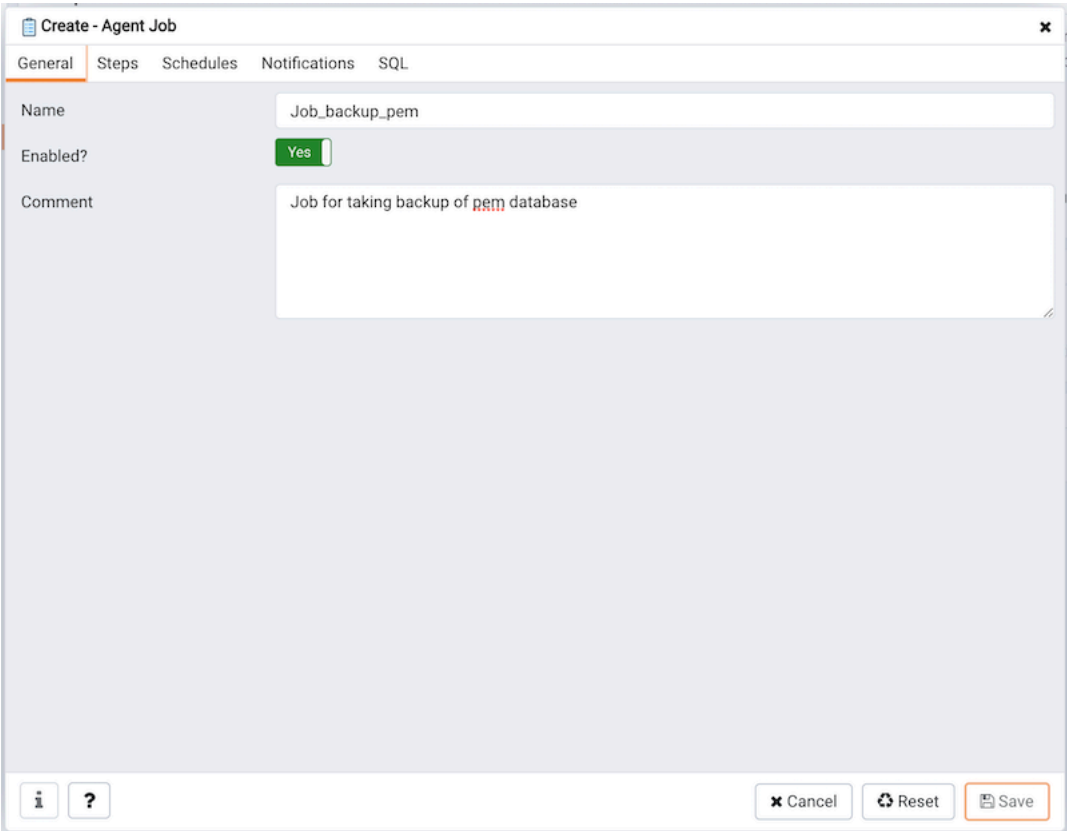
The image shows a screenshot of the 'Create - Agent Job' dialog box in the Postgres Enterprise Manager interface. The dialog has a title bar with a close button. Below the title bar are five tabs: 'General', 'Steps', 'Schedules', 'Notifications', and 'SQL'. The 'General' tab is selected and highlighted. The 'General' tab contains three main fields: 'Name' with the value 'Job_backup_pem', 'Enabled?' with a green 'Yes' button, and 'Comment' with the text 'Job for taking backup of pem database'. At the bottom of the dialog are three buttons: 'Cancel', 'Reset', and 'Save'. There are also information and help icons on the bottom left.

図 15 エージェントジョブの作成-[全般] タブ

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。
- ジョブを有効にするには Enabled スイッチを Yes の位置に移動し、ジョブを無効にするには No の位置

に移動します。

- 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。

The screenshot shows the 'Create - Agent Job' dialog box with the 'Steps' tab selected. A table lists the steps, with 'Step1_backup' being the first and only entry. Below the table, the 'General' sub-tab is active, displaying configuration fields for the selected step. The fields include Name (Step1_backup), Enabled? (Yes), Kind (SQL), On error (Success), Server (Postgres Enterprise Manager Server (192.168.1.19:5432)), Database (pem), and a large text area for Comment. At the bottom right are buttons for Cancel, Reset, and Save.

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 16 エージェントジョブの作成-[ステップ] タブ

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。手順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、ジョブの実行時にステップを含めるか (True)、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 を選択した場合は、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。

図 17 エージェントジョブの作成-[ステップ] タブ-[全般] タブ

- 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してい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

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 エージェントジョブの作成-[ステップ]タブ-[コード]タブ

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は、Code フィールドにスクリプトを指定します。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`）。定義済みのインラインコードに加えて、ファイルシステム上のバッチスクリプト、シェルスクリプト、または SQL ファイルのパスを指定することもできます。

Linux システムでスクリプトを呼び出すには、`agent.cfg` ファイルの `batch_script_user` パラメーターのエントリを変更し、スクリプト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提供したら、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。

The screenshot shows the 'Create - Agent Job' dialog box with the 'Schedules' tab selected. A table lists the schedule 'Backup_pem_db_schedule' with 'Enabled?' set to 'True' and a specific time range. Below the table, the 'General' sub-tab is active, displaying the same schedule details in a form layout, including a 'Yes' toggle for 'Enabled?' and a 'Save' button at the bottom right.

図 19 エージェントジョブの作成-[スケジュール] タブ-[全般] タブ

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 がスケジュールを使用する (Yes) か、スケジュールを無効にする (No) ことを示します。

- 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

Days

Week Days

Month Days

Months

Times

Hours

Minutes

図 20 エージェントジョブの作成-[スケジュール] タブ-[繰り返し] タブ

Repeat タブのフィールドを使用して、cron スタイル形式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間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。

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. It contains a table with the following data:

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

Below the main table, the 'Exceptions' sub-tab is active, showing a table with one exception:

Date	Time
2020-04-01	00:00

At the bottom of the dialog, there are buttons for 'Cancel', 'Reset', and 'Save'.

図 21 エージェントジョブの作成-[スケジュール] タブ-[例外] タブ

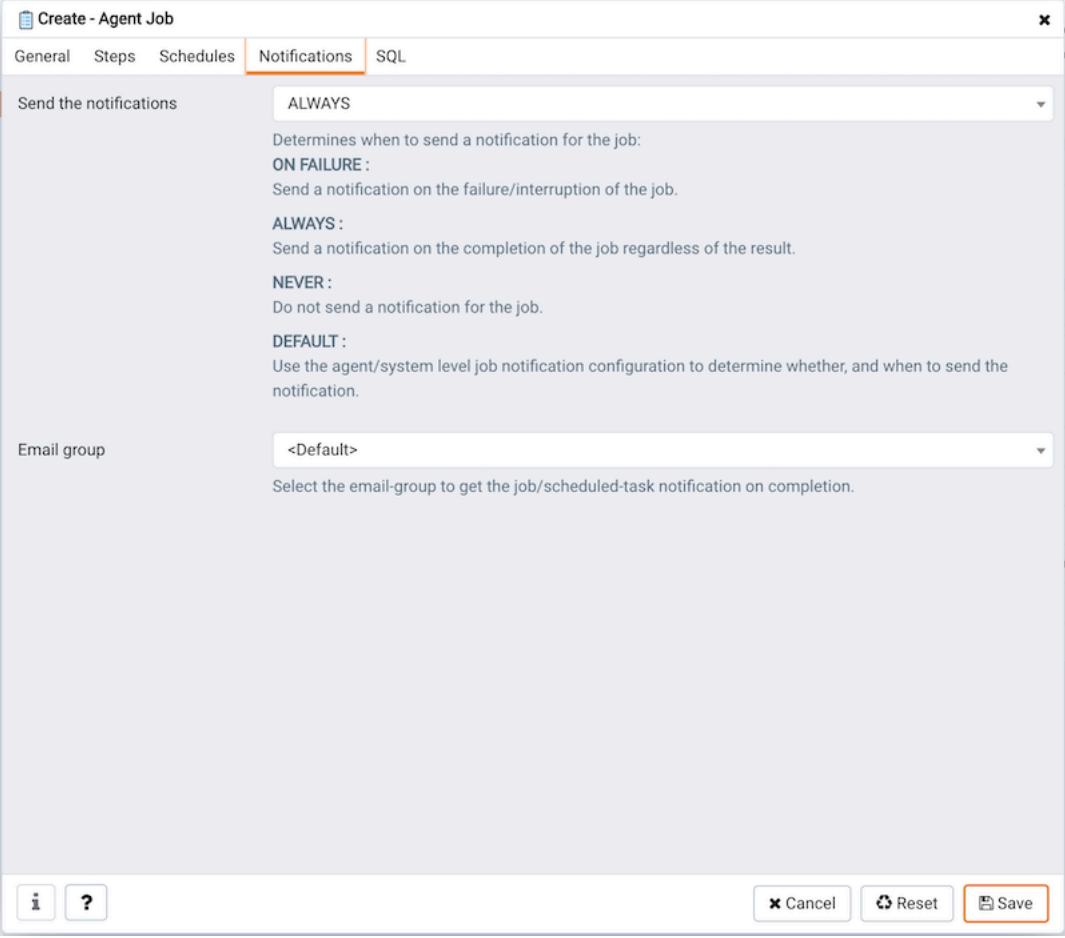
Exceptions タブのフィールドを使用して、ジョブを実行しない日を指定します。たとえば、国民の祝日にはジョブを実行しないようにすることができます。

[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レクトを開き、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、ジョブレベルで電子メール通知設定を構成します。



The screenshot shows the 'Create - Agent Job' dialog box with the 'Notifications' tab selected. The 'Send the notifications' dropdown menu is set to 'ALWAYS'. Below this, there is explanatory text: 'Determines when to send a notification for the job:'. Under 'ON FAILURE:', it says 'Send a notification on the failure/interruption of the job.' Under 'ALWAYS:', it says 'Send a notification on the completion of the job regardless of the result.' Under 'NEVER:', it says 'Do not send a notification for the job.' Under 'DEFAULT:', it says 'Use the agent/system level job notification configuration to determine whether, and when to send the notification.' The 'Email group' dropdown menu is set to '<Default>'. Below this, it says 'Select the email-group to get the job/scheduled-task notification on completion.' At the bottom right, the 'Save' button is highlighted with an orange border.

図 22 エージェントジョブの作成-[通知] タブ

Notifications タブのフィールドを使用して、ジョブの電子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、電子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。

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

Save 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、ダイアログから未保存のエントリを削除します。

ジョブを保存すると、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 [2] 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。

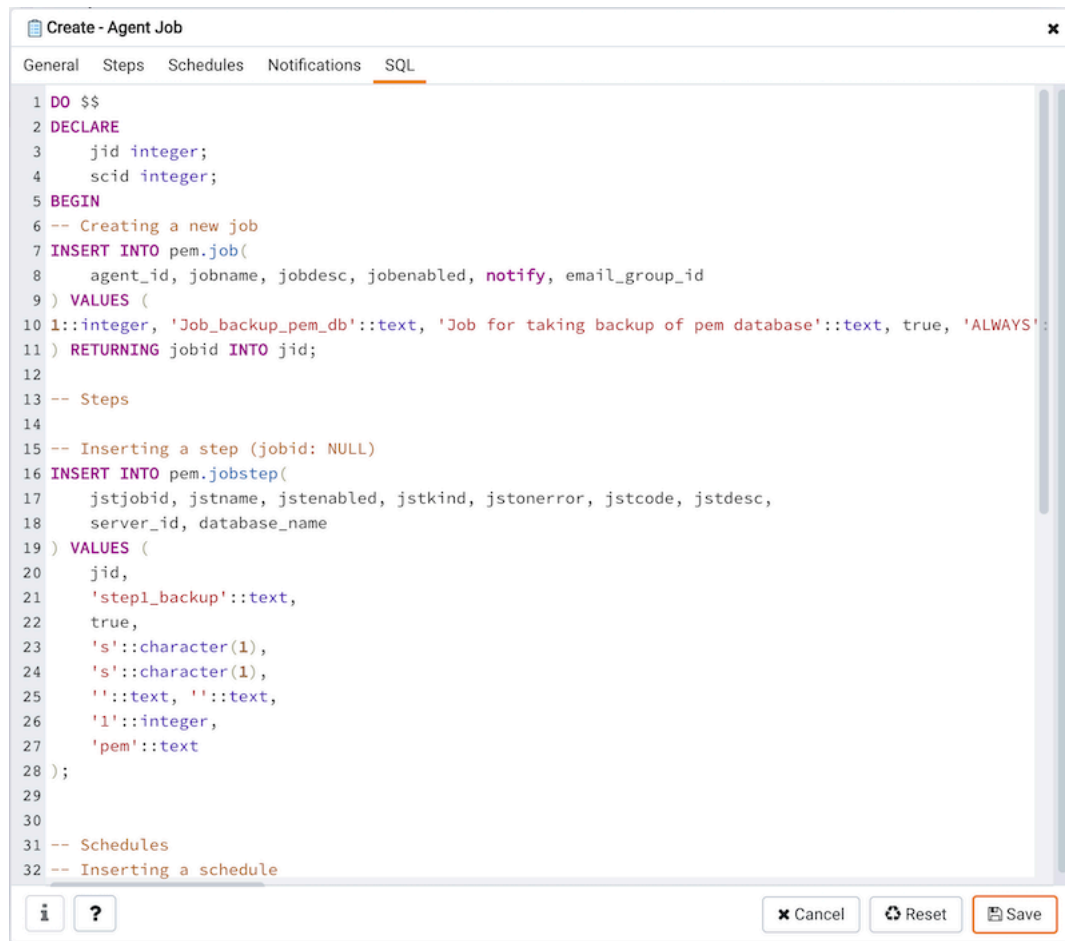


図 23 エージェントジョブの作成-[SQL] タブ

第 7 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。

7.1 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` 権限とデータベースサーバーのスーパーユーザー権限でインストールされます。これらの特権により、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ープを呼び出し、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 プロープ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 プロープは情報を返さず、次の機能が影響を受ける可能性があります。

プローブまたはアクション	オペレーティングシステム	影響を受ける PEM 機能
データとログファイルの分析	Linux/Windows	PostgresExpert は完全な情報にアクセスできません。
セッション情報	Linux	プロセスごとの統計は不完全です。
PGHBA	Linux/Windows	PostgresExpert は完全な情報にアクセスできません。
サービス再起動機能	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。
パッケージ展開	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。
バッチタスク	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。
サーバーからデータを収集する（ルートアクセスが必要）	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。

プローブ	オペレーティングシステム	影響を受ける PEM 機能
監査ログの収集	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。
サーバーログ収集	Linux/Windows	PEM はサーバーログ情報を収集できません。
データベース統計	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。
セッション待機/システム待機	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。
ロック情報	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。
ストリーミングレプリケーション	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 レプリケーション	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。
表領域サイズ	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 レプリケーション	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特権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。

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

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

Windows では、ログ情報は Event Viewer で利用できます。

7.2 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 エージェント設定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 パラメーターを除いて、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せずに、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 設定オプションは /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ます。

パラメータ名	説明	デフォルト値
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。	ポート 5432。
pem_agent	PEM エージェントに割り当てられた一意の識別子。	最初のエージェントは、2 番目のエージェントは などを。
agent_ssl_key	PEM エージェントのキーファイルへの完全なパス。	/root/.pem/agent.key
agent_ssl_crt	PEM エージェントの証明書ファイルへの完全なパス。	/root/.pem/agent.crt
agent_flag_dir	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内のファイルの形式で行われます。	デフォルトでは設定されていません。
log_level	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。	警告
log_location	PEM ワーカーログファイルの場所を指定します。	127.0.0.1。
agent_log_location	PEM エージェントログファイルの場所を指定します。	/var/log/pem/agent.log

次のページに続く

表 1 – 前のページからの続き

パラメータ名	説明	デフォルト値
long_wait	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時間（秒）。	30 秒
short_wait	PEM エージェントがキュー内に次にある（実行を待機している）プローブをチェックする前に待機する最小時間（秒単位）。	10 秒
alert_threads	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。	PEM サーバーのホストに常駐するエージェントに対して 1 に設定します。他のすべてのエージェントの場合は 0。
enable_smtp	複数の PEM エージェント（7.13 以下）および PEM バックエンドデータベース（9.4 以下）に対して true に設定すると、より多くの重複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）および PEM バックエンドデータベース（9.5 以降）の場合、送信する重複メールの数は少なくなります。	PEM サーバーホストの場合は true。他のすべての場合は false。
enable_snmp	複数の PEM エージェント（7.13 以下）および PEM バックエンドデータベース（9.4 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）および PEM バックエンドデータベース（9.5 以降）の場合、送信する重複トラップが少なくなる場合があります。	PEM サーバーホストの場合は true。他のすべての場合は false。
enable_nagios	true に設定すると、Nagios アラートが有効になります。	PEM サーバーホストの場合は true。他のすべての場合は false。
enable_webhook	true に設定すると、Webhook アラートが有効になります。	PEM サーバーホストの場合は true。他のすべての場合は false。

次のページに続く

表 1 – 前のページからの続き

パラメータ名	説明	デフォルト値
max_webhook_retries	失敗時に pemAgent が webhook の呼び出しを再試行する最大回数を設定します。	デフォルト 3。
connect_timeout	エージェントが接続を待機する最大時間（秒）（10 進整数文字列）。	デフォルトでは設定されていません。0 に設定すると、エージェントは無期限に待機する必要があります。
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。	偽
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（無制限の数）
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（別のデータベースへの）新しい接続が必要な場合、このパラメータは無視されます。	デフォルトでは、0 に設定されます（接続は、エージェントの処理ループの後にアイドル状態になると切断されます）。
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。	偽
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。	偽
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。	デフォルトでは設定されていません。

次のページに続く

表 1 – 前のページからの続き

パラメータ名	説明	デフォルト値
ca_file	CA 証明書が存在するパスを指定します。	デフォルトでは設定されていません。
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。	なし
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_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 SSLca 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せるようになります。	偽

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。

パラメータ名	説明	デフォルト値
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。	ポート 5432。
AgentID	PEM エージェントに割り当てられた一意の識別子。	最初のエージェントは、2 番目のエージェントは などを。
AgentKeyPath	PEM エージェントのキーファイルへの完全なパス。	% APPDATA %//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。	% APPDATA %//Roaming//pem//agent.crt

次のページに続く

表 2 – 前のページからの続き

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内のファイルの形式で行われます。	デフォルトでは設定されていません。
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。	警告
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時間（秒）。	30 秒
shortWait	PEM エージェントがキュー内に次にある（実行を待機している）プローブをチェックする前に待機する最小時間（秒単位）。	10 秒
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。	PEM サーバーのホストに常駐するエージェントに対して 1 に設定します。他のすべてのエージェントの場合は 0。
EnableSMTP	true に設定すると、SMTP 電子メール機能が有効になります。	PEM サーバーホストの場合は true。他のすべての場合は false。
EnableSNMP	true に設定すると、SNMP トラップ機能が有効になります。	PEM サーバーホストの場合は true。他のすべての場合は false。
EnableWebhook	true に設定すると、Webhook アラートが有効になります。	PEM サーバーホストの場合は true。他のすべての場合は false。
MaxWebhookRetries	失敗時に pemAgent が webhook の呼び出しを再試行する最大回数を設定します。	デフォルト 3。
ConnectTimeout	エージェントが接続を待機する最大時間（秒）（10 進整数文字列）。	デフォルトでは設定されていません。0 に設定すると、エージェントは無期限に待機します。

次のページに続く

表 2 – 前のページからの続き

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。	本当
MaxConnections	接続スロットルで使用するプローブ接続の最大数。	0（無制限の数）
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（別のデータベースへの）新しい接続が必要な場合、このパラメータは無視されます。	デフォルトでは、0 に設定されます（接続は、エージェントの処理ループの後にアイドル状態になると切断されます）。
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。	偽
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。	偽
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
ConnectionCustomSetup	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。	デフォルトでは設定されていません。
ca_file	CA 証明書が存在するパスを指定します。	デフォルトでは設定されていません。
AllowBatchJobSteps	true に設定すると、バッチ/シェルスクリプトは管理者ユーザーアカウントを使用して実行されます。	なし
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。	

次のページに続く

表 2 – 前のページからの続き

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 SSLca 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せるようになります。	偽

7.3 エージェントのプロパティ

PEM エージェントの Properties ダイアログは、ダイアログを開いた 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。

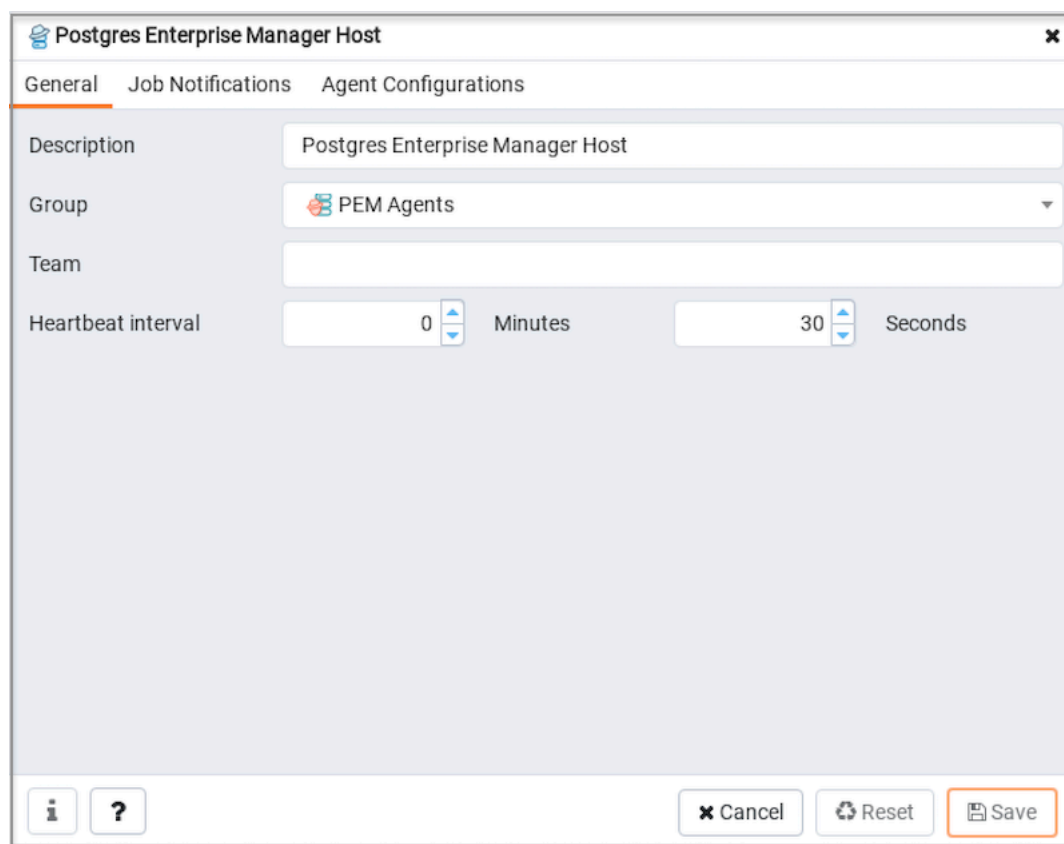


図 1 PEM エージェントプロパティダイアログ-[全般] タブ

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、接続されているチームメンバーの 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。Minutes または Seconds フィールドの隣のセクターを使用して、間隔を変更します。

Postgres Enterprise Manager Host

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.

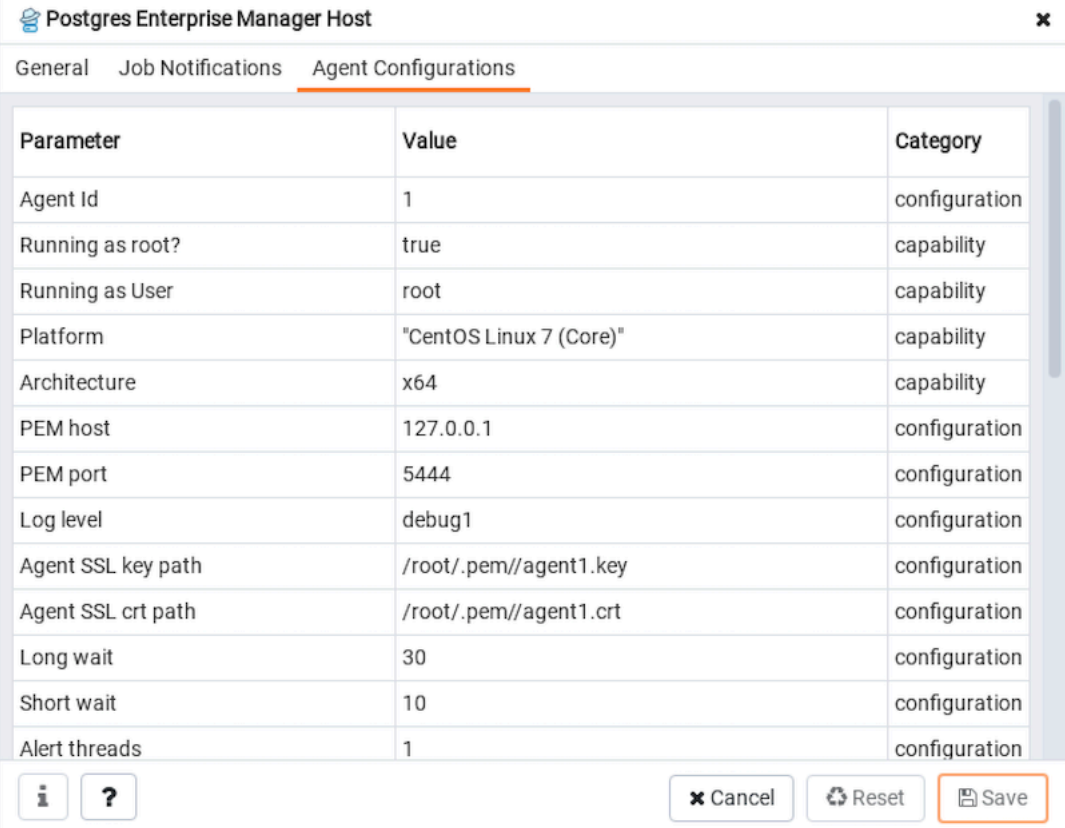
Email group
Select the email-group that will receive the notification on completion of a job or scheduled task.

Cancel Reset Save

図2 PEM エージェントプロパティダイアログ-ジョブ通知タブ

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電子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電子メールグループを指定します。



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

Buttons: ⓘ ⓘ × Cancel ↺ Reset 💾 Save

図3 PEM エージェントプロパティダイアログ-エージェント設定タブ

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。

第 8 章

Conclusion

PostgresEnterpriseManager 入門ガイド

Copyright©2013-2020EnterpriseDBCorporation。無断複写・転載を禁じます。

EnterpriseDB®Corporation34CrosbyDrive、Suite201、Bedford、MA01730、USA

T+17813573390F+19784671307Einfo@enterprisedb.comwww.enterprisedb.com

- EDB および PostgresEnterpriseManager は、EnterpriseDBCorporation の登録商標です。EDB および EDB-Postgres は、EnterpriseDBCorporation の商標です。Oracle は、Oracle、Inc の登録商標です。その他の商標は、それぞれの所有者の商標です。
- EDB は、PostgresEnterpriseManager が存在する場合、ログオン UI の設計、コーディングのベストプラクティスの確立、レビュー、入力検証を検証します。EDB は追加の入力コンポーネントに対して同じアプローチに従いますが、製品の性質上、限定された検証が可能な信頼できるユーザーによって入力および送信されるフリーフォーム SQL、WMI またはその他の文字列を受け入れる必要がある場合があります。このような場合、ユーザーが誤った入力や危険な入力を禁止することはできません。
- EDB は、将来、認証された信頼できるユーザーからのフリーフォーム SQL、WMI またはその他の潜在的に危険な入力を受け入れる製品に機能を追加する権利を留保しますが、そのような機能はすべて、可能な限り最小のリスクを提供するように設計およびテストされます可能であれば、スーパーユーザーまたは同等の特権が必要です。
- EDB は、すべての潜在的な脅威を予測できることを保証しません。したがって、すべての潜在的な脆弱性が対処または検討されたことをプロセスが完全に保証することはできません。

索引

Agent Configuration, 94
Agent Privileges, 91
Agent Properties, 101
agent ssl certificate, 55

Conclusion, 105
control server remotely, 59
Control service on Linux, 60
Control service on Windows, 61
Creating PEM scheduled jobs, 81

Defining and Monitoring Postgres instances on AWS, 47

Group Roles, 70

HTTPD server, 62

Installing PEM - Overview, 11

Login Roles, 70

Managing a PEM Agent, 91
Managing a PEM Server, 59
Managing Certificates, 51
Managing Configuration Settings, 57

PEM
 architecture, 5
 installlation, 10
 overview, 1
PEM Architecture, 5
PEM Overview, 3
pg_hba.conf file, 63

Registering a Server, 25
replacing, 51
replacing ssl certificates, 51
restart PEM agent, 59
restart PEM service, 59

Security, 69
Sending email notifications for a job, 77
ssl certificates, 51
starting service, 59

update agent ssl certificate, 55
Using the PEM Web Interface, 12