



EDB

PEM Agent User Guide

リリース 8.4.0

EDB

2022 年 04 月 14 日

目次

第 1 章	PEM architecture	3
1.1	PEM サーバー	4
1.2	PEM エージェント	6
1.3	PEM Web クライアント	7
1.4	SQL プロファイラープラグイン	8
第 2 章	Registering an agent	9
2.1	PEM エージェント構成パラメーターの設定	11
2.2	ルートユーザアカウントを使用して PEM エージェントを登録する	11
第 3 章	Managing a PEM agent	13
3.1	エージェント権限	13
3.2	エージェントの構成	15
3.3	エージェントのプロパティ	19
第 4 章	PEM agent troubleshooting	23
4.1	削除された PEM エージェントの復元	23
4.2	コマンドラインを使用して、ダウンまたは不明なステータスの PEM エージェントを削除する	23

PEM は、PEM サーバー、PEM エージェント、および PEM Web インタフェースの 3 つのプライマリコンポーネントで構成されています。PEM エージェントは、各管理対象マシンでタスクを実行し、データベースサーバーとオペレーティングシステムの統計を収集します。

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、Platform Compatibility を参照してください。

このドキュメントでは、PEM エージェントを使用するために必要な情報を提供します。このガイドでは、エージェントの基本的な登録、構成、および管理について説明します。このガイドは、次のコアセクションに分かれています。

- **PostgresEnterpriseManagerManager-概要**-このセクションでは、PEM アーキテクチャの概要を説明します。
- **PEM エージェントの登録**-このセクションでは、PEM エージェントの登録に関する情報を提供します。
- **PEM エージェントの管理**-このセクションでは、PEM エージェントの構成と管理に関する情報を提供します。
- **PEM エージェントのトラブルシューティング**-このセクションでは、PEM エージェントのトラブルシューティングに関する情報を提供します。

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

pem_architecture registering_agent Managing_pem_agent pem_agent_troubleshooting 結論

</ div>

第1章 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ー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、PEM SQL Profiler Configuration Guide 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。

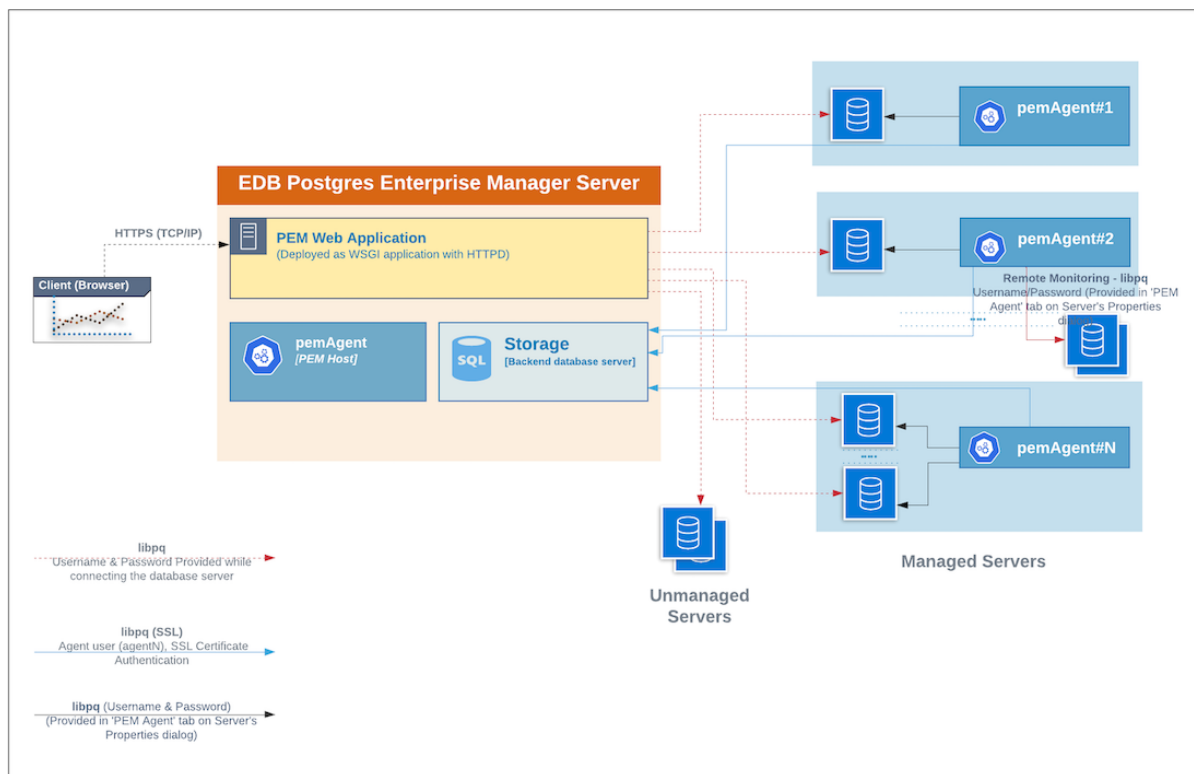


Figure 1: PEM Architecture

PEM アーキテクチャ

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。

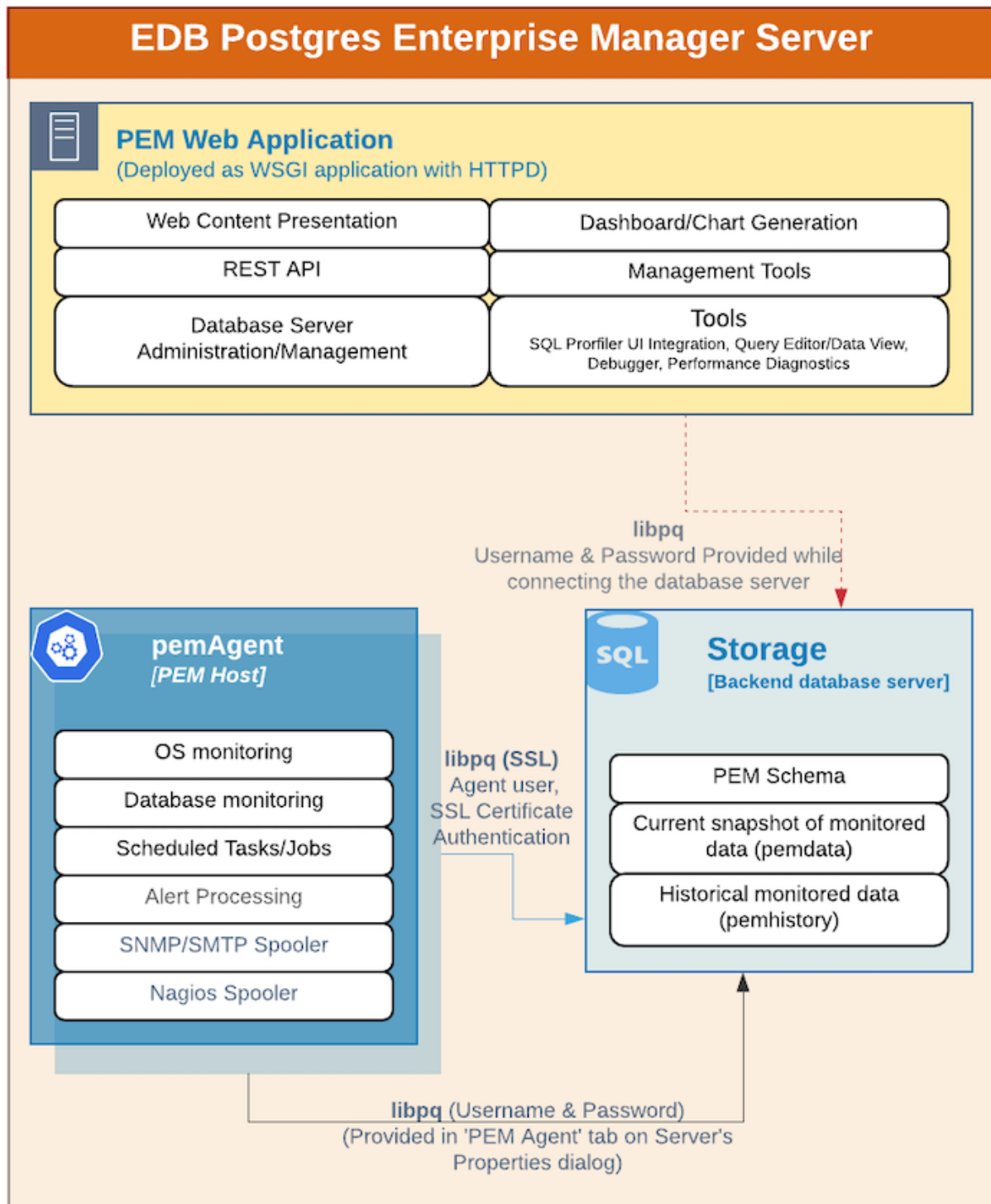


図 2: PEM Server

- **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、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ヘルプます。

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。

1.2 PEM エージェント

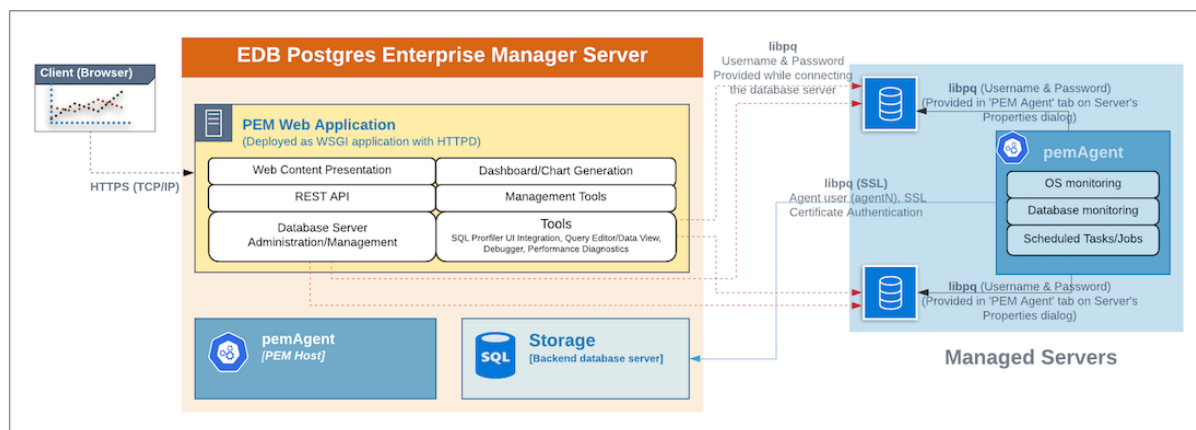


図 3: PEM Agent

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 [here](#) 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、Server Configuration を参照してください。

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ーで実行される 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。

1.4 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 website 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 プロファイラの使用の詳細については、PEM SQL Profiler Guide を参照してください

第2章 Registering an agent

各 PEM エージェントは、PEM サーバーに登録する必要があります。登録プロセスは、エージェントと通信するためにニーズ情報を PEM サーバーに提供します。Windows 用の PEM エージェントグラフィカルインストーラは、エージェントの自己登録をサポートしています。エージェントが Linux ホスト上にある場合、**pemworker** ユーティリティを使用してエージェントを登録する必要があります。

RPM インストーラは、PEM エージェントを `/usr/edb/pem/agent/bin` ディレクトリに配置します。エージェントを登録するには、**pemworker** ユーティリティを呼び出すときに、登録の詳細とともに `--register-agent` キーワードを含めます。

pemworker --register-agent

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

Option	Description
<code>--pem-server</code>	PEM バックエンドデータベースサーバの IP アドレスを指定します。このパラメータは必須です。
<code>--pem-port</code>	PEM バックエンドデータベースサーバのポートを指定します。デフォルト値は 5432 です。
<code>--pem-user</code>	PEM バックエンドデータベースサーバのデータベースユーザ（スーパーユーザ特権を持つ）の名前を指定します。このパラメータは必須です。
<code>--pem-agent-user</code>	PEM サーバーバックエンドデータベースサーバに接続するエージェントユーザを指定します。
<code>--cert-path</code>	証明書が作成されるディレクトリへの完全なパスを指定します。パスを指定しない場合、証明書は次の場所に作成されます：Linux では <code>~/.pem</code> Windows では <code>% APPDATA %/ pem</code>
<code>--config-dir</code>	構成ファイルを見つけることができるディレクトリパスを指定します。デフォルトは <code>< pemworker パス> /.. / etc</code> です。
<code>--display-name</code>	PEM ブラウザツリーコントロールに表示されるエージェントのわかりやすい名前を指定しユーザフレンドリ。デフォルトはシステムのホスト名です。
<code>--force-registration</code>	<code>force_registration</code> 句を含めて、指定された引数でエージェントを登録するように PEM サーバーに指示します。この句は、既存のエージェント構成をオーバーライドする場合に役立ちます。デフォルト値は「はい」です。
<code>--group</code>	エージェントが表示されるグループの名前。
<code>--team</code>	監視対象のデータベースサーバにアクセスする必要がある、PEM バックエンドデータベースサーバ上のデータベースロールの名前。
<code>--owner</code>	エージェントを所有する PEM バックエンドデータベースサーバ上のデータベースユーザの名前。

次のページに続く

表 1 – 前のページからの続き

Option	Description
<code>--allow_server_restart</code>	<code>allow-server_restart</code> パラメータを有効にして、PEM が監視対象サーバーをリスタートできるようにします。デフォルト値は <code>True</code> です。
<code>--allow_batch_probes</code>	<code>allow-batch-probes</code> パラメータを有効にして、PEM がこのエージェントでバッチプローブを実行できるようにします。デフォルト値は <code>False</code> です。
<code>--batch-script-user</code>	バッチ/シェルスクリプトの実行に使用するオペレーティングシステムユーザを指定します。デフォルト値は <code>none</code> です。このパラメータを空白のままにした場合、または指定したユーザが存在しない場合、スクリプトは実行されません。
<code>--enable-heartbeat-connection</code>	<code>enable-heartbeat-connection</code> パラメータを有効にして、PEM エージェントとサーバー間に専用のハートビート接続を作成し、アクティブステータスを更新します。デフォルト値は <code>False</code> です。
<code>--enable-smtp</code>	<code>enable-smtp</code> パラメータを有効にして、PEM エージェントが PEM サーバーに代わってメールを送信できるようにします。デフォルト値は <code>False</code> です。
<code>--enable-snmp</code>	<code>enable-snmp</code> パラメータを有効にして、PEM エージェントが PEM サーバーに代わって SNMP トラップを送信できるようにします。デフォルト値は <code>False</code> です。
<code>-o</code>	構成ファイルオプションをオーバーライドするかどうかを指定します。

データベースサーバーのリスタートが必要な PEM 機能（監査マネージャー、ログマネージャー、チューニングウィザードなど）を使用する前に、まず `agent.cfg` ファイルの `allow_server_restart` の値を `true` に設定する必要があります。

注釈: PEM 7.11 以降のバージョンがインストールされている PEM エージェントによって実行されるシェルスクリプトを構成する場合、`batch_script_user` パラメータのユーザを指定する必要があります。ルートユーザを使用してスクリプトを実行することを強くお勧めします。ルートユーザを使用すると、データセキュリティとオペレーティングシステムのセキュリティが低下する可能性があります。ただし、ルートユーザを使用してスクリプトを実行して `pemagent` をオリジナルの設定に復元する場合は、`batch_script_user` パラメータ値を `root` に設定する必要があります。

`PEM_SERVER_PASSWORD` 環境変数を使用して、PEM 管理ユーザーのパスワードを設定できます。`PEM_SERVER_PASSWORD` が設定されていない場合、サーバーは PEM データベースサーバーへの接続時に `PGPASSWORD` または `.pgpass` ファイルを使用します。

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。他の必須だが省略された情報の入力を求められます。登録が完了すると、サーバーはエージェントが正常に登録されたことを確認します。

2.1 PEM エージェント構成パラメーターの設定

PEM エージェント RPM インストーラは、`/usr/edb/pem/agent/etc` ディレクトリに `agent.cfg.sample` 名前付けのサンプル構成ファイルを作成します。PEM エージェントを登録すると、`pemworker` プログラムは実際のエージェント構成ファイル（`agent.cfg` 名前付け）を作成します。`agent.cfg` ファイルを変更して、次の設定パラメータを追加する必要があります。

```
heartbeat_connection = true
```

`ca-bundle.crt` ファイル（認証権限）の場所も追加する必要があります。デフォルトでは、インストーラは `agent.cfg.sample` ファイルで指定された場所に `ca-bundle.crt` ファイルを作成しファイル。サンプルファイルからデフォルトのパラメータ値をコピーするか、別の場所に保存されている `ca-bundle.crt` ファイルを使用する場合は、`ca_file` パラメータでその値を指定します。

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

次に、プラットフォーム固有のコマンドを使用して、PEM エージェントサービスをスタートします。サービスの名前付けは `pemagent` です。

RHEL または CentOS 7.x または 8.x ホストで、`systemctl` を使用してサービスをスタートします。

```
systemctl start pemagent
```

サービスは、エージェントを開始していることを確認します。エージェントが登録および起動されると、Global Overview ダッシュボードおよび PEM Web インタフェースのオブジェクトブラウザーツリーコントロールに表示されます。

`pemworker` ユーティリティを使用してサーバーを登録する方法については、`PEM Administrator's Guide` を参照してください

2.2 ルートユーザアカウントを使用して PEM エージェントを登録する

非ルートユーザアカウントを使用して PEM エージェントを登録するには、まず PEM エージェントをルートユーザとしてインストールする必要があります。インストール後、非ルートユーザ（`edb` 例）の ID を想定して、次の手順を実行します。

1. `.pem` ディレクトリと `logs` ディレクトリを作成し、ファイルに読み取り、書き込み、実行のアクセス許可を割り当てます。

```
mkdir /home/<edb>/.pem
mkdir /home/<edb>/.pem/logs
chmod 700 /home/<edb>/.pem
chmod 700 /home/<edb>/.pem/logs
```

2. PEM サーバーにエージェントを登録します。

```
./pemworker --register-agent --pem-server <172.19.11.230> --pem-user <postgres> --pem-
port <5432> --display-name <non_root> --cert-path /home/<edb> --config-dir /home/
<edb>
```

The above command creates agent certificates and an agent configuration file (`agent.cfg`) in the `/home/edb/.pem` directory. Use the following command to assign read and write permissions to these files:

```
chmod -R 600 /home/edb/.pem/agent*
```

3. `agent.cfg` ファイルのパラメーターを変更します。

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

4. `pemagent` サービスファイルの構成ファイルパスとユーザの値を更新します：`edb_notranlate_4 +`

5. エージェントプロセスを停止し、非ルートユーザ `edb_notranlate_5 +` を使用してエージェントサービスをリスタートします。

6. PEM ダッシュボードでエージェントのステータスを確認します。

第3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。

3.1 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 System	PEM Functionality Affected
Audit Log Collection	Linux/Windows	PEM は PEM データベースから空のデータを受けます。
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。

3.2 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.

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_smtp	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.
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.

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
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。	
webhook_ssl.crt	webhook の SSL クライアント証明書ファイルへの完全なパス。	
webhook_ssl.crl	webhook サーバー証明書を検証するための CRL ファイルの完全なパス。	
webhook_ssl_ca.crt	webhook の SSL ca 証明書ファイルへの完全なパス。	
allow_insecure_webhooks	true に設定されている場合、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.
AgentCrtPath	PEM エージェントの証明書ファイルへの完全なパス。	%APP-DATA%\Roaming\pem\agent.crt
AgentFlagDir	HA サポートに使用されます。別のサーバーの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

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
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.
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

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
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。	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。	
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

3.3 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。

[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レクトターを使用します。

[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メール通知設定を構成します。

Postgres Enterprise Manager Host

General Job Notifications Agent Configurations

Description: Postgres Enterprise Manager Host

Group: PEM Agents

Team:

Heartbeat interval: 0 Minutes 30 Seconds

Buttons: Cancel, Reset, Save

図 1: PEM Agent Properties dialog - General tab

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.

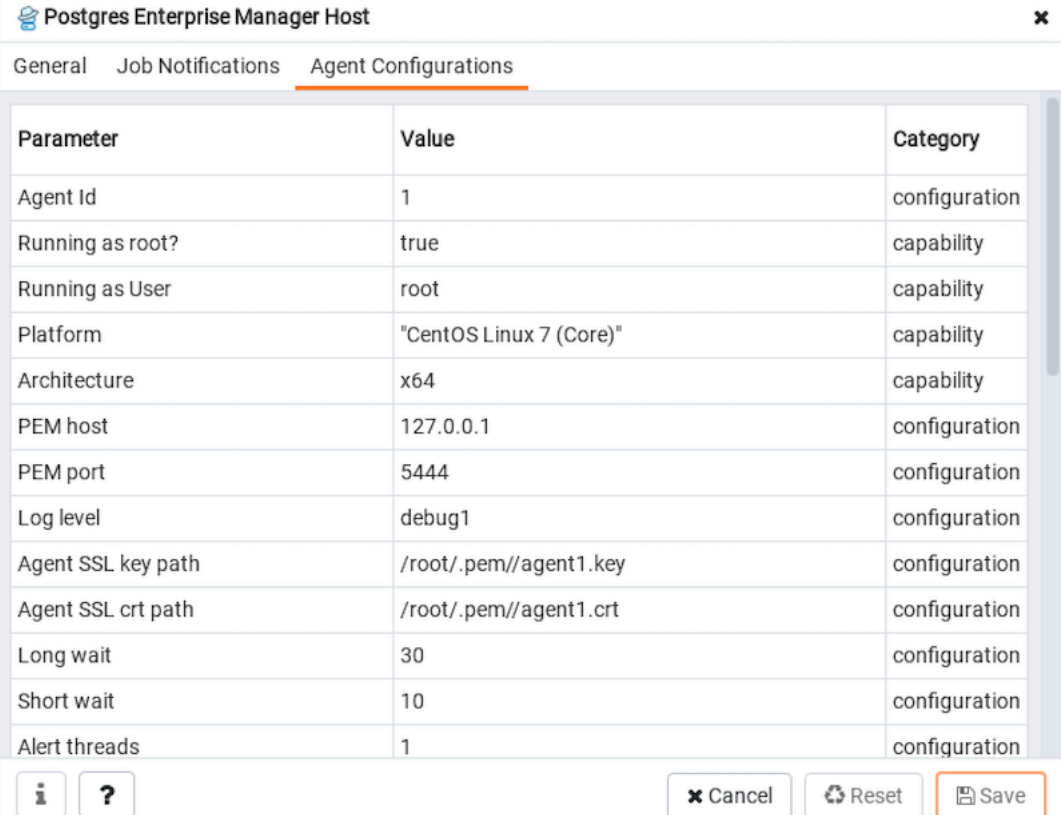
Email on a job failure? ☐ No
Select to receive a notification email only on failure of a job of this agent.

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

Buttons: Cancel, Reset, Save

図 2: PEM Agent Properties dialog - Job Notifications tab

- **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。



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

図 3: PEM Agent Properties dialog - Agent Configurations tab

[PEM エージェントプロパティ] ダイアログ-[エージェント構成] 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。

第4章 PEM agent troubleshooting

4.1 削除された PEM エージェントの復元

エージェントが pem.agent テーブルから削除されている場合、復元できません。エージェントを再登録するには、pemworker ユーティリティを使用する必要があります。

エージェントが PEM Web クライアントから削除されたが、active = f の値を持つ pem.agent テーブルにまだエントリーがある場合、次の手順を使用してエージェントを復元できます。

1. 次のコマンドを使用して、id および active フィールドの値を確認します。

```
pem=# SELECT * FROM pem.agent;
```

2. pem.agent テーブルでエージェントのステータスを true に更新します。

```
pem=# UPDATE pem.agent SET active=true WHERE id=<x>;
```

x は、手順 1 で使用したクエリーの出力に表示された識別子です。

3. PEM Web クライアントを更新します。

削除されたエージェントが再び復元されます。ただし、その特定のエージェントにバインドされたサーバーはダウンしているように見える場合があります。この問題を解決するには、サーバーの PEM エージェントプロパティを変更して、バインドされたエージェントを再度追加する必要があります。変更が成功すると、サーバーは正常に実行されていると表示されます。

4.2 コマンドラインを使用して、ダウンまたは不明なステータスの PEM エージェントを削除する

Down または Unknown ステータスの PEM エージェントを削除するために PEM Web インタフェースを使用するのは、そのようなエージェントの数が大きい場合は困難です。このようなシチュエーションでは、コマンドラインインタフェースを使用して Down または Unknown エージェントを削除することができます。

1. 次のクエリーを使用して、* N *時間以上 Down であるエージェントを削除します。

```
UPDATE pem.agent SET active=false WHERE id IN
(SELECT a.id FROM pem.agent
a JOIN pem.agent_heartbeat b ON (b.agent_id=a.id)
WHERE a.id IN
(SELECT agent_id FROM pem.agent_heartbeat WHERE (EXTRACT (HOUR FROM now()))-
EXTRACT (HOUR FROM last_heartbeat)) > <N> ));
```

2. 次のクエリーを使用して、Unknown ステータスのエージェントを削除します。

```
UPDATE pem.agent SET active=false WHERE id IN  
(SELECT id FROM pem.agent WHERE id NOT IN  
(SELECT agent_id FROM pem.agent_heartbeat));
```