



PEM Installation Guide on Linux

リリース **8.3.0**

EnterpriseDB

2021 年 12 月 25 日

目次

第 1 章 PEM - Hardware and Software Requirements	3
1.1 ハードウェアの前提条件	3
1.2 ソフトウェアの前提条件	3
第 2 章 PEM Architecture	5
2.1 PEM サーバー	6
2.2 PEM エージェント	8
2.3 PEM Web クライアント	10
2.4 SQL プロファイラープラグイン	10
第 3 章 Installing Postgres Enterprise Manager	11
3.1 Prerequisites for Installing the PEM Server on Linux Platforms	11
3.2 Web Server Hosting Preferences	13
3.3 ‘Installing the PEM Server on Linux Platforms’	13
3.3.1 CentOS または RHEL ホストへの PEM サーバーのインストール	13
3.3.2 Debian または Ubuntu ホストへの PEM サーバーのインストール	15
3.3.3 SLES ホストへの PEM サーバーのインストール	16
3.4 Creating a PEM Repository on an Isolated Network	17
3.5 Configuring the PEM Server on Linux Platforms	18
3.6 Configuring the PEM Server to use Kerberos Authentication	20
3.6.1 1. Kerberos、PEM サーバー、および PEM バックエンドデータベースをインストールします	20
3.6.2 2. Kerberos サーバーにプリンシパルを追加する	21
3.6.3 3. Kerberos サーバーからキーテーブルを抽出する	21
3.6.4 4. PEM バックエンドデータベースサーバーの構成	22
3.6.5 5. 初期チケットを取得してビューする	23
3.6.6 6. PEM サーバーの構成	23
3.6.7 7. ブラウザ設定	24
3.7 Configuring the PEM Server to use Windows Active Directory Domain Services for Kerberos Authentication (SSPI)	25
3.7.1 1. Active Directory、PEM サーバー、および PEM バックエンドデータベースサーバーをインストールします	25
3.7.2 2. Active Directory でユーザーを作成して、サービスプリンシパルとマップする	25
3.7.3 3. Active Directory からキーテーブルを抽出する	26
3.7.4 4. PEM バックエンドデータベースサーバーの構成	30
3.7.5 5. 初期チケットを取得してビューする	30
3.7.6 6. PEM サーバーの構成	31
3.7.7 7. ブラウザ設定	32
3.8 Configuring the PEM Server to use Two-Factor Authentication	33
3.9 Installing a PEM Agent on Linux Platforms	34

3.9.1	CentOS または RHEL ホストへの PEM エージェントのインストール	34
3.9.2	Debian または Ubuntu ホストへの PEM エージェントのインストール	36
3.9.3	SLES ホストへの PEM エージェントのインストール	37
3.10	Registering a PEM Agent	37
第 4 章	The PEM Web Interface	41
第 5 章	Uninstalling Postgres Enterprise Manager Components	45
5.1	CentOS または RHEL ホストからの PEM コンポーネントのアンインストール	45
5.2	Debian または Ubuntu ホストからの PEM コンポーネントのアンインストール	46
5.3	SLES ホストからの PEM コンポーネントのアンインストール	46
第 6 章	Reference - Linux Service Script	47

Postgres Enterprise Manager (PEM) は、PostgreSQL および Advanced Server データベースサーバーを管理、モニタリング、およびチューニングする際に、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ように設計されています。PEM は、単一のコンソールから単一サーバーまたはマルチプルサーバーを管理およびモニタするように設計されており、監視対象データベースを完全に制御できます。

このドキュメントは、Linux ホストへの Postgres Enterprise Manager のインストールをガイドするステップバイステップの手順を提供します。

このガイド全体で、**Postgres** という用語は、PostgreSQL または EDB Postgres Advanced Server または EDB Postgres Extended (旧称 2ndQPostgres) インストールのいずれかを指し、どちらか適切な場合です。

第1章 PEM - Hardware and Software Requirements

</div>

1.1 ハードウェアの前提条件

サーバーをモニタリングし、ダッシュボードをレンダリングするときに最適なパフォーマンスを得るには、少なくとも次のシステムに PEM をインストールすることをお勧めします。

- 4つの CPU コア
- 8 ギガバイトの RAM
- 100 ギガバイトのストレージ

データストレージには追加のディスクスペースが必要です。リソースの使用量は、どのプローブが定義および有効化されているか、監視対象データベースのアクティビティレベルに応じて異なることにノートしてください。サーバーリソースの監視（PEM の使用中）は、初期システム構成を拡張する必要がある場合に通知します。

1.2 ソフトウェアの前提条件

プラットフォームとバージョンのサポート

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。

**** pg_hba.conf ファイルの変更****

pg_hba.conf ファイルは、Postgres サーバーの接続を管理します。各監視対象サーバー上の pg_hba.conf ファイルで、PEM サーバー、モニタリング PEM エージェント、および PEM-HTTPD サーバーのホストからの接続が許可されていることを保証する必要があります。

pg_hba.conf ファイルの変更については、[PEM Administrator's Guide](#) を参照してください。

認証の管理に関する情報は、[Postgres core documentation](#) でも入手できます。

ファイアウォールの制限

PEM コンポーネント間の通信を許可するには、ファイアウォールを調整する必要があることにノートしてください。

サポートされているロケール

現在、PEM サーバーと Web インタフェースサポートは、**English(US)** **en_US** のロケールと言語区切り文字としてのピリオド (.) の使用をサポートしています。代替ロケール、またはピリオド以外の区切り文字を使用すると、エラーが発生する場合があります。

第2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：Postgres という用語は、PostgreSQL または EDB Postgres Advanced Server または EDB Postgres Extended を指します。

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

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

- **** PEM サーバー****-PEM サーバーは、データをモニタリングするためのデータリポジトリとして、およびエージェントとクライアントの両方が接続するサーバーとして使用されます。PEM サーバーは、PostgreSQL のインスタンスと、モニタリングデータをストレージするための関連データベース、および Web サービスを提供するサーバーで構成されます。
- **** PEM エージェント****-PEM エージェントは、タスクの実行と、エージェントホストおよび監視対象の Postgres インスタンスから PEM サーバーへの統計のレポートを担当します。単一の PEM エージェントは、1 つ以上のホストに存在する Postgres のマルチプルのインストール済みインスタンスをモニタできます。
- **** PEM Web クライアント****-PEM Web インタフェースにより、Postgres サーバーを管理およびモニタし、PEM 拡張機能を利用できます。Web インタフェースソフトウェアは PEM サーバーとともにインストールされ、サポートされている Web ブラウザーを介してアクセスされます。
- **** SQL Profiler ****- SQL Profiler は、PEM の SQL Profiler ツールで分析されるモニタリングデータとクエリ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。

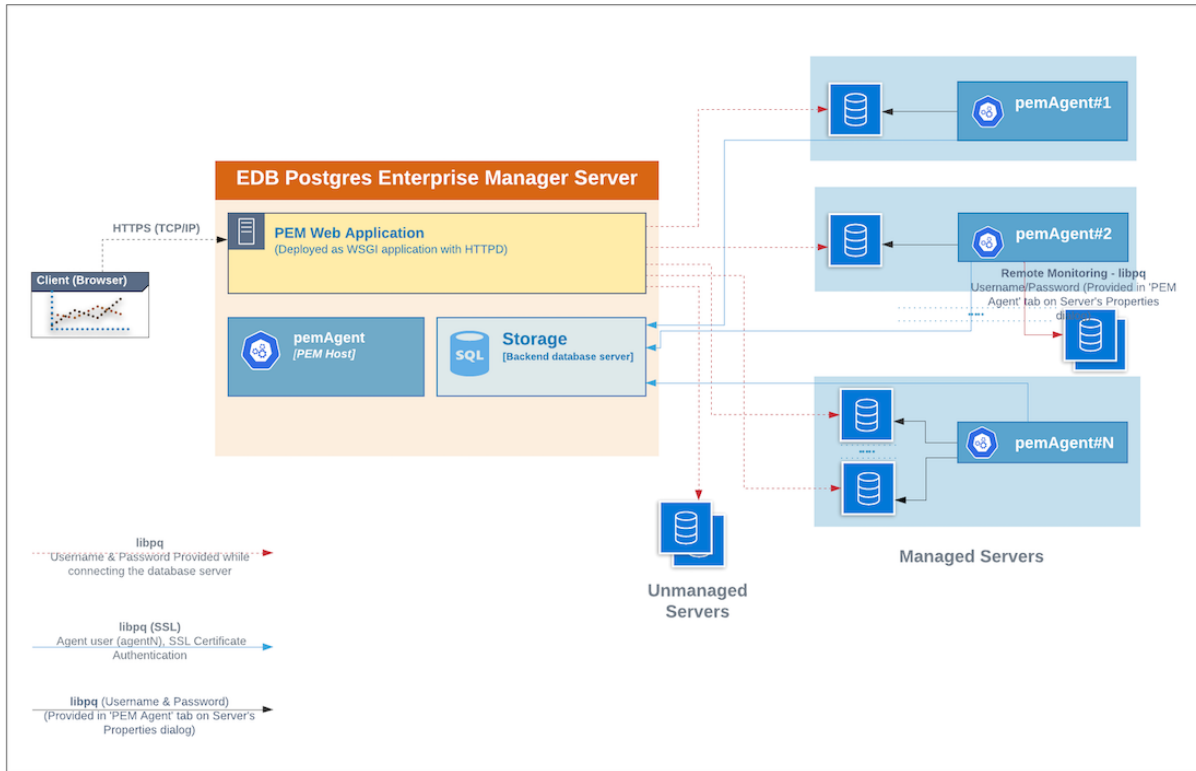


図 1: PEM Architecture

PEM アーキテクチャ

2.1 PEM サーバー

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） -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、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。

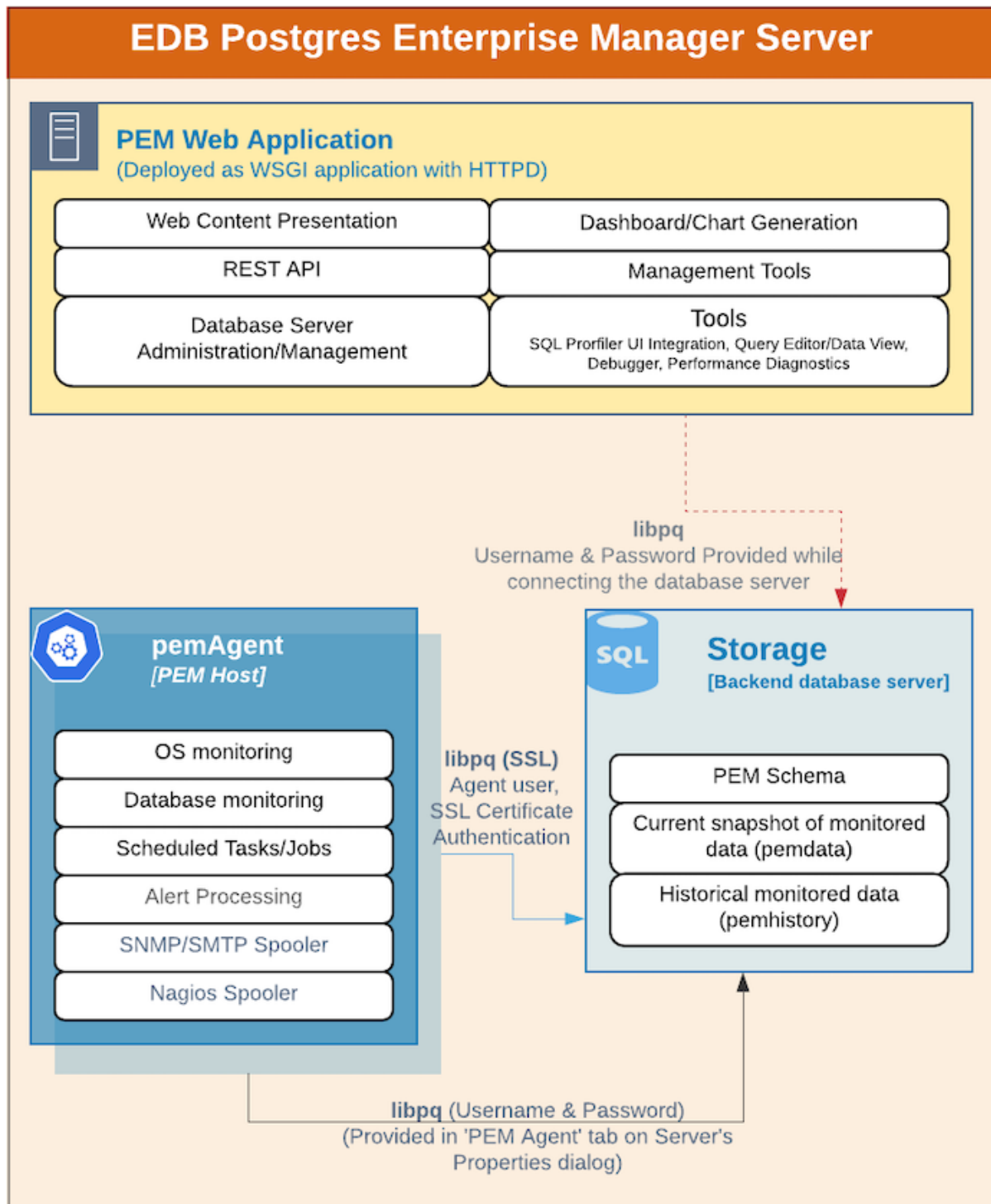


図 2: PEM Server

- **** Apache Web Server (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、..)。
- **** REST API ****-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。
- **** SQL Profiler UI の統合 ****- SQL Profiler は、簡単に分析されたセッションコンテンツのトレースを生成します。
- **クエリエディタ/データビュー**-クエリエディタを使用すると、データをクエリー、編集、およびビューします。
- **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、Postgres インスタンスのパフォーマンス分析にヘルプします。

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。

2.2 PEM エージェント

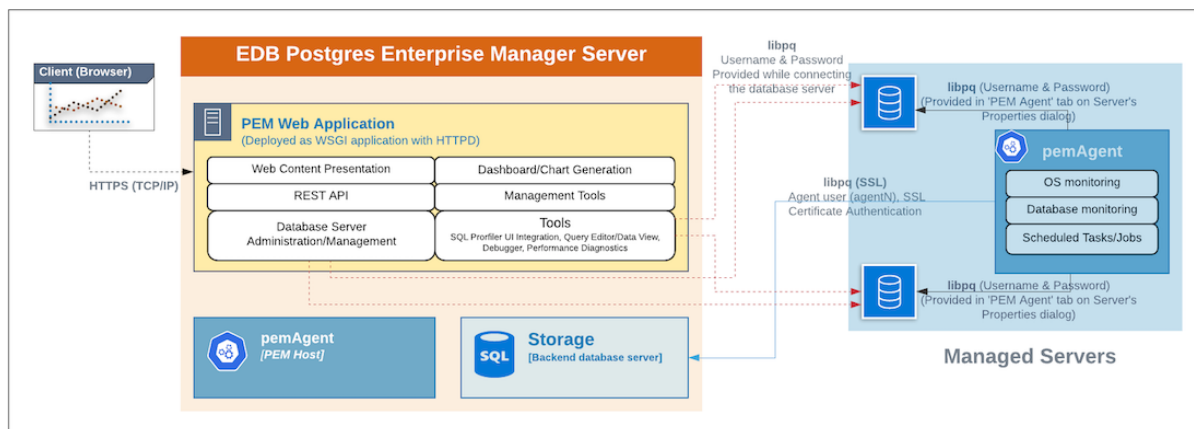


図 3: PEM Agent

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のメトリックを収集します。収集されるメトリックの例は次のとおりです。

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

PEM プローブのリストは [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。

2.3 PEM Web クライアント

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

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

2.4 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 プロファイラー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3章 Installing Postgres Enterprise Manager

Linux プラットフォーム用の `edb-pem` パッケージは、PEM サーバー、PEM エージェント、およびサポートされているブラウザで PEM Web インタフェースに接続するために必要なソフトウェアをインストールします。

PEM サーバーは、Postgres インストールとバックエンドデータベースを使用してデータを管理します。PEM を構成すると、`pem` バックエンドデータベースが作成されます。

PEM サーバーのインストールの詳細については、を参照してください。PEM サーバーの構成については、を参照してください。

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、同じホストまたはリモートホストにある複数のサーバーをモニタリングできます。リモートエージェントによって監視されるサーバー上の PEM 機能が制限される場合があることにノートしてください。リモートモニタリング機能の詳細については、[PEM Agent Privileges](#) を参照してください。

PEM エージェントのインストールと設定の詳細については、[Installing the PEM Agent on Linux](#) を参照してください。

3.1 Prerequisites for Installing the PEM Server on Linux Platforms

1. バックエンドデータベースをインストールします。

- Linux ホストに PEM サーバーをインストールする場合、`pem` データベースを保持するバックエンドデータベースクラスタを最初にインストールする必要があります。PEM サーバーのバックエンドデータベースは、Linux 用のパッケージを介してインストールできます。バックエンドデータベースは、次のバージョンのいずれかである必要があります。

EDB Postgres Advanced Server バージョン 11 以降

PostgreSQL バージョン 11 以降

Advanced Server または PostgreSQL データベースのインストールの詳細については、EDB ウェブサイトの製品文書を参照してください。

2. バックエンドデータベースで Postgres 認証を構成します。

- バックエンドデータベース上の `pg_hba.conf` ファイルは、接続にサポートされている認証方法（例、`md5`、`trust`、…）を使用するように構成できます。`pg_hba.conf` ファイルの変更については、[PostgreSQL core documentation](#) を参照してください。

3. PostgreSQL データベースを使用している場合、次のコマンドを使用して `hstore contrib module` をインストールします。

```
* yum install postgresql<x>-contrib
```

- ここで、**x** はサーババージョンです。

4. sslutils 拡張機能がインストールされていることを確認します。

- -Advanced Advanced Server バックエンドデータベースでは、**sslutils** 拡張機能がデフォルトでインストールされます。
- -PostgreSQL バックエンドデータベースを使用している場合、PostgreSQL コミュニティリポジトリにアクセスできることを保証し、次のコマンドを使用し PostgreSQL。

```
* yum install sslutils_<x>
```

- ここで、**x** はサーババージョンです。
- Debian 10 および Ubuntu 20 では、セキュリティの理由により、証明書を受け入れるための要件が増加していることにノートしてください。ユーザがマシンに PEM エージェントをインストールする場合、4096 ビット RSA キーと sha256 シグネチャアルゴリズムのサポートが追加されている **sslutils** を 1.3 にアップグレードする必要があります。ユーザが **sslutils** を 1.3 にアップグレードしない場合、PEM エージェントは PEM バックエンドデータベースサーバに接続すると、エラー **ca md too weak** がログに記録される場合があります。

5. ファイアウォールの制限を調整します。

ファイアウォールを使用している場合、PEM バックエンドデータベースのポート 8443 へのアクセスを許可する必要があります。

```
* firewall-cmd --permanent --zone=public --add-port=8443/tcp
* firewall-cmd --reload
```

6. EDB リポジトリへのアクセスを許可する資格情報を要求します。PEM サーバーをインストールするには、EDB リポジトリへのアクセスを許可する資格情報が必要です。リポジトリの資格情報を要求するには、EDB に連絡してください。以降のセクションでコマンドを使用する場合、**username** および **password** プレースホルダーを EDB が提供する資格情報に置き換えます

7. PEM は、python3、libboost、openssl、snmp++、libcurl など、ベンダリポジトリのサードパーティコンポーネントに依存しています。これらのコンポーネントが最新であることを保証するには、次のプラットフォーム固有のコマンドを使用してオペレーティングシステムを更新する必要があります。openssl に必要な最小バージョンは 1.0.2k です。

- CentOS または RHEL 7.x ホストでパッケージソフトをアップグレードするには

```
* yum upgrade
```

- CentOS または RHEL 8.x ホストでパッケージソフトをアップグレードするには

```
* dnf upgrade
```

- Debian または Ubuntu ホスト上のパッケージソフトをアップグレードするには

```
* apt-get update
```

- SLES ホストでパッケージソフトをアップグレードするには

```
* zypper update
```

3.2 Web Server Hosting Preferences

PEM サーバーのインストール中に、Apache Web サーバー（PEM-HTTPD）のホスティング設定を指定できます。

同じホストに PEM サーバーと Apache Web サーバー（PEM-HTTPD）をインストールするには

インストール手順に従います。構成スクリプトの実行中に、Web Services and Database オプションを選択して、PEM サーバーと Apache Web サーバーを同じホストにインストールします。

**** PEM サーバーと Apache Web サーバー（PEM-HTTPD）を別々のホストにインストールするには****

両方のホストでインストール手順に従います。構成スクリプトの実行中に、最初のホストで Database オプションを選択して PEM サーバーホストを構成し、次に 2 番目のホストで Web Services オプションを選択して Apache Web サーバー（PEM-HTTPD）を構成します。

PEM サーバーの構成の詳細については、[Configuring the PEM Server on Linux Platforms](#) を参照してください。

3.3 ‘Installing the PEM Server on Linux Platforms’

特定のプラットフォームに PEM サーバーをインストールする詳細な手順に従う前に、[Prerequisites for installing PEM Server](#) で説明されている前提条件の手順を実行する必要があります。

3.3.1 CentOS または RHEL ホストへの PEM サーバーのインストール

CentOS または RHEL システムでは、yum パッケージマネージャまたは dnf コマンドを使用して PEM サーバーをインストールできます。使用するインストールツールは、ホストオペレーティングシステムのバージョンに依存します。サーバーをインストールする前に、システムに以下の必須ソフトウェアが含まれていることを保証必要があります。

バージョン固有のソフトウェアをインストール

以下にリストされているバージョン固有の指示に従って、ホストシステムを準備します。

- CentOS または RHEL 7.x ホストの場合：

```
* yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.  
↪rpm
```

- CentOS または RHEL 8.x ホスト：

```
* yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.
→rpm
```

!!! Note * CentOS-Base.repo ファイル (/etc/yum.repos.d にある) で [extras] リポジトリ定義を有効にする必要がある場合があります。

**** Red Hat Network ユーザの場合****

rhel-<x>-server-extras-rpms リポジトリを有効にする必要があります。x は RHEL バージョンを指定します。

また、EPEL パッケージソフトを使用するには rhel-<x>-server-optional-rpms リポジトリを有効にする必要があります。* x *は RHEL バージョンを指定します。RHN-Classic の RHEL optional subchannel を有効にすると、リポジトリにアクセスできる make になります。証明書ベースのサブスクリプションがある場合は、EPEL パッケージソフトを使用するために rhel-<x>-server-eus-optional-rpms リポジトリも有効にする必要があります。必要なリポジトリの詳細については、Red Hat Subscription Management Guide を参照してください。

**** edb.repo ファイルをインストールして設定****

EnterpriseDB リポジトリ構成ファイルを作成するには、スーパーユーザ権限を引き受けて、次のコマンドを呼び出します。

```
yum -y install https://yum.enterprisedb.com/edb-repo-rpms/edb-repo-latest.noarch.rpm
```

リポジトリ構成ファイルの名前付けは edb.repo です。ファイルは/etc/yum.repos.d にあります。edb.repo ファイルを作成した後、次のコマンドを使用して、baseurl 仕様の USERNAME および PASSWORD プレースホルダーを、登録済み EDB ユーザの<username>および<password>に置き換えます。

```
sed -i "s@<username>:<password>@USERNAME:PASSWORD@" /etc/yum.repos.d/edb.repo
```

**** PEM サーバーをインストール****

上記のプラットフォーム固有の前提条件を満たした後、yum または dnf を使用して PEM サーバーをインストールできます。

- CentOS または RHEL 7.x または 8.x では、yum を使用して PEM サーバーをインストールできます。

```
* yum install edb-pem
```

- CentOS または RHEL 8.x では、dnf を使用して PEM サーバーをインストールできます。

```
* dnf install edb-pem
```

!!! Note * PEM サーバーのインストール中に、オペレーティングシステムユーザ pem が作成されます。PEM サーバー Web アプリケーションは WSGI アプリケーションであり、Apache HTTPD の下で実行されます。PEM アプリケーションデータとセッションは、このユーザーのホームディレクトリに保存されます。

CentOS または RHEL 7.x ホストで PEM サーバーの新規インストールを実行している場合、インストーラは、オペレーティングシステムの要件に従って、インストールとともに edb-python3-mod_wsgi パッケージソフトもインストールします。

CentOS または RHEL 7.x ホストで PEM サーバーをアップグレード処理する場合、`mod_wsgi system` パッケージは、オペレーティングシステムの要件に従って `edb-python3-mod_wsgi` パッケージに置き換えられます。

システムで認識されないソースによって署名された RPM パッケージをインストールすると、`yum` はローカルサーバにキーをインポートするパーミッションを求める場合があります。プロンプトが表示され、パッケージソフトが信頼できるソースからのものであることに満足したら、`y` と入力し、`Return` を押して続行します。

インストール中に、`yum` は解決できない依存関係に遭遇する場合があります。存在する場合、手動で解決する必要がある依存関係のリストが提供されます。

分離されたネットワークにあるマシンに PEM サーバーをインストールする場合、最初にそのマシンに PEM リポジトリを作成する必要があります。分離されたネットワークでの PEM リポジトリの作成の詳細については、[Creating a PEM repository in an Isolated Network](#) を参照してください。

`yum` または `dnf` を使用して PEM サーバーをインストールした後、PEM サーバーを構成する必要があります。詳細については、[Configuring the PEM Server on Linux platforms](#) を参照してください。

3.3.2 Debian または Ubuntu ホストへの PEM サーバーのインストール

次の手順では、EDB apt リポジトリを使用して Debian パッケージをインストールする手順を説明します。

!!! Note * < <https://repos.enterprisedb.com/> > にアクセスし、プラットフォームと製品を選択して、インストールの手順をビューすることもできます。

1. Debian または Ubuntu ホストでルートとしてログインします。

```
* sudo su -
```

2. EDB リポジトリを構成します。

- -Debian 9 の場合：

```
* sh -c 'echo "deb https://username:password@apt.enterprisedb.com $(lsb_release_
↪cs)-edb/ $(lsb_release -cs) main" > /etc/apt/sources.list.d/edb-$(lsb_release -
↪cs).list'
```

- `username:password` は、EDB が提供する資格情報に置き換えられます。

- -Debian 10 の場合：

- a. EnterpriseDB リポジトリをセットアップします。

```
* sh -c 'echo "deb [arch=amd64] https://apt.enterprisedb.com/$(lsb_release -cs)-
↪edb/ $(lsb_release -cs) main" > /etc/apt/sources.list.d/edb-$(lsb_release -cs).list'
```

- b. 次のコマンドで、`<username>` および `<password>` プレースホルダーを EnterpriseDB 資格情報に置き換えます。

```
*      sh -c 'echo "machine apt.enterprisedb.com login <username> password <password>  
↪" > /etc/apt/auth.conf.d/edb.conf'
```

3. 安全な APT リポジトリのサポートをシステムに追加します。

```
*      apt-get install apt-transport-https
```

4. EBD 署名キーを追加します。

```
*      wget -q -O -https://username:password@apt.enterprisedb.com/edb-deb.gpg.key | apt-  
↪key add -`
```

5. リポジトリメタデータを更新します。

```
*      apt-get update
```

6. 次のコマンドを使用して、PEM サーバーの Debian パッケージをインストールします。

```
*      apt-get install edb-pem
```

- !!! Note

- オペレーティングシステムユーザ `pem` は、PEM サーバーのインストール中に作成されます。PEM サーバー Web アプリケーションは WSGI アプリケーションであり、Apache HTTPD の下で実行されます。PEM アプリケーションデータとセッションは、このユーザーのホームディレクトリに保存されます。

インストールが完了したら、PEM サーバーを構成する必要があります。詳細については、[Configuring the PEM Server on Linux Platforms](#) を参照してください。

3.3.3 SLES ホストへの PEM サーバーのインストール

次のコマンドを使用して、EDB リポジトリ構成ファイルを SLES ホストに追加します。

```
zypper addrepo https://zypp.enterprisedb.com/suse/edb-sles.repo
```

このコマンドは、`/etc/zypp/repos.d` ディレクトリに `edb.repo` 名前付けのリポジトリ構成ファイルを作成します。登録済みの EDB ユーザのユーザー名とパスワードを追加して、リポジトリ構成ファイルを変更します。

PEM をインストールする前に、前提条件パッケージソフトをインストールする必要があります。次のコマンドを呼び出して、`* sp_no *` を使用しているサービスパック（SP4 など）に置き換えます。

```
SUSEConnect -p sle-module-legacy/12/x86_64
```

```
SUSEConnect -p sle-sdk/12/x86_64
```

(次のページに続く)

(前のページからの続き)

```
zypper addrepo https://download.opensuse.org/repositories/Apache:Modules/SLE_12_<sp_
→no>/Apache:Modules.repo

zypper addrepo http://download.opensuse.org/repositories/Cloud:/OpenStack:/Newton:/
→cisco-apic:/2.3.1/SLE_12_<sp_no>/ pem_opensuse_boost
```

次に、リポジトリをリフレッシュし、PEM サーバーをインストールします。

```
zypper refresh

zypper install edb-pem
```

!!! Note * オペレーティングシステムユーザ `pem` は、PEM サーバーのインストール中に作成されます。PEM サーバー Web アプリケーションは WSGI アプリケーションであり、Apache HTTPD の下で実行されます。PEM アプリケーションデータとセッションは、このユーザーのホームディレクトリに保存されます。

`zypper` を使用して PEM サーバーをインストールした後、PEM サーバーを構成する必要があります。詳細については、[Configuring the PEM Server on Linux Platforms](#) を参照してください。

3.4 Creating a PEM Repository on an Isolated Network

PEM をアップグレードするサーバーが EnterpriseDB リポジトリに直接アクセスできない場合、PEM RPM パッケージソフトのホストとして機能するローカルリポジトリを作成できます。これは、必要な手順のノートです。個々のネットワークのプロセスを変更する必要がある場合があります。ローカルリポジトリを作成して使用するには、以下を行う必要があります。

1. インターネットにアクセスできるシステムで次のコマンドを使用して、PEM の依存関係をダウンロードします。

```
* yum install yum-plugin-downloadonly
* mkdir /<pem_dir>
* yum install --downloadonly --downloadaddir=/<pem_dir>/ edb-pem
* mkdir /<epel_dir>
* yum install --downloadonly --downloadaddir=/<epel_dir>/ epel-release*
```

- `<pem_dir>` および `<epel_dir>` は、RPM をダウンロードするために作成するローカルディレクトリです。

2. ディレクトリ `/<pem_dir>` および `/<epel_dir>` を分離されたネットワーク内のマシンにコピーします。

3. リポジトリを作成します。

```
* createrepo /<pem_dir>
* createrepo /<epel_dir>
```

4. 次を指定する接続情報を使用して、`/etc/yum.repos.d/pem.repo` というリポジトリ構成ファイルを作成します。

```
* [pemrepo]
* name=PEM Repository
* baseurl=file:///pem7/
* enabled=1
* gpgcheck=0
```

5. 次を指定する接続情報を使用して、`/etc/yum.repos.d/epel.repo` というリポジトリ構成ファイルを作成します。

```
* [epelrepo]
* name=epel Repository
* baseurl=file:///pem7/
* enabled=1
* gpgcheck=0
```

6. ローカルリポジトリの場所と接続情報を指定したら、`yum` コマンドを使用して PEM サーバーをインストールまたはアップグレードします。

- PEM サーバーをインストールするには：

```
* yum install edb-pem
```

- PEM サーバーをアップグレードするには：

```
* yum upgrade edb-pem
```

ローカル yum リポジトリの作成の詳細については、次の Web サイトをご覧ください：< <https://wiki.centos.org/HowTos/CreateLocalRepos> >

3.5 Configuring the PEM Server on Linux Platforms

PEM サーバーを構成する前に、保証拡張モジュールと `hstore contrib` モジュールがバックエンドデータベースにインストールされていることを確認してください。

- Advanced Server バックエンドデータベースの場合、`sslutils` 拡張モジュールと `hstore contrib` モジュールは、デフォルトで Advanced Server とともにインストールされます。
- PostgreSQL バックエンドデータベースの場合、PostgreSQL コミュニティリポジトリにアクセスできることを保証してから、コマンドで `sslutils` 拡張機能と `hstore contrib` モジュールをインストールし PostgreSQL。

```
* yum install sslutils_<x> postgresql<X>-contrib
```

- ここで、`x` はサーババージョンです。

PEM サーバーパッケージには、Linux プラットフォームインストールの構成プロセスを自動化するのにヘルプスクリプト (`configure-pem-server.sh`) が含まれています。スクリプトは `/usr/edb/pem/bin` ディレクトリにインストールされます。スクリプトを呼び出すには、次のコマンドを使用します。

```
/usr/edb/pem/bin/configure-pem-server.sh
```

!!! Note * -SSL 証明書を使用している場合、すべての SSL 証明書がバックエンドデータベースサーバのデータディレクトリ内にあることを確認します。証明書がデータディレクトリ内がない場合、PEM サーバーの構成中にデータディレクトリを調べると、PEM サーバーの構成スクリプトが失敗する場合があります。-Debian 9 では、PEM サーバーの設定中に apache2 WSGI モジュールが無効になり、edb-python3-mod-wsgi モジュールがインストールされます

スクリプトを呼び出すときに、構成プロパティを指定するコマンドラインオプションを含めることができます。スクリプトは、コマンドラインで省略した値のプロンプトを求めます。受け入れられるオプションは次のとおりです。

Option	Description
-acp	PEM エージェント証明書パスを定義します。デフォルトは`/root/.pem`です。
-ci	エージェントがサーバーに接続し、サーバーの`pg_hba.conf`ファイルに追加される CIDR 形式のネットワークアドレスレンジ。例、192.168.1.0/24。デフォルトは`0.0.0.0/0`です。
-dbi	データベースサーバをインストールするためのディレクトリ。例、Advanced Server の場合は`/usr/edb/as12`、PostgreSQL の場合は`/usr/pgsql-12`です。
-ds	PEM データベースサーバのユニットファイル名前。Advanced Server の場合、デフォルトのファイル名前は`edb-as-12`です。PostgreSQL の場合は`postgresql-12`です。
-ho	PEM データベースサーバのホストアドレス。
-p	PEM データベースサーバのポート番号。
-ps	pemagent のサービス名前。デフォルト値は`pemagent`です。
-sp	PEM データベースサーバのスーパーユーザパスワード。この値は必須です。
-su	PEM データベースサーバのスーパーユーザ名前。
-t	インストールの種類：構成が Web サービスとバックエンドデータベースの場合は 1、Web サービスを構成する場合は 2、バックバックエンドデータベースを構成する場合は 3 を指定します。3 を指定する場合、データベースはローカルホストに存在する必要があることにノートしてください。
-un	PEM サーバーの登録を解除します。
	NA

コマンドラインで構成プロパティを指定しない場合、スクリプトによって値の入力が求められます。スクリプトを呼び出すとき、次から選択します。

1. Web Services and Database- ウェブサーバとデータベースの両方が PEM サーバーと同じホストにある場合、このオプションを選択します。
2. Web Services- ウェブサーバが PEM サーバーとは異なるホストにある場合、このオプションを選択します。
3. Database-PEM サーバーで使用する PEM バックエンドデータベースを構成するには、このオプションを選択します。指定したデータベースはローカルホストに存在する必要があることにノートしてください。

!!! Note * ウェブサーバ（PEM-HTTPD）とバックエンドデータベース（PEM サーバー）が別々のホストに

存在する場合、最初にデータベースサーバを構成し（オプション 3）、次に Web サービスを構成します（オプション 2）。Web サービスの前にバックエンドデータベースが設定されていない場合、スクリプトは終了します。

構成オプションを選択すると、スクリプトは続行して構成プロパティのプロンプトを求めます。スクリプトが完了すると、PEM サーバーに必要なオブジェクトが作成されるか、必要な構成手順が実行されます。1 スクリプト関連のヘルプをビューするには、次のコマンドを使用します。

```
/usr/edb/pem/bin/configure-pem-server.sh -help
```

PEM サーバーを構成したら、ブラウザで PEM Web インタフェースにアクセスできます。以下に移動します：

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

3.6 Configuring the PEM Server to use Kerberos Authentication

!!! Tip “New Feature” * Kerberos サポートは PEM 8.1.0 以降で利用可能です。

Kerberos サーバーは、ip addresses ではなくホスト名で動作します。Kerberos Authentication を使用して PEM サーバーでシングルサインオンを使用するには、DNS（REALM）を使用してホスト名で次のマシンを構成する必要があります。

Kerberos サーバー-PEM サーバー（PEM Web サーバーおよび PEM バックエンドデータベースサーバー）-クライアントマシン

- 例、Kerberos サーバーの REALM は edbpem.org であり、Kerberos サーバーのホスト名は Krb5server.edbpem.org、PEM サーバーのホスト名は pem.edbpem.org、クライアントのホスト名は pg12.edbpem.org に設定できます。
- レルムの名前として DNS ドメイン名前を使用することは慣例です。

手順に従って、PEM サーバーの Kerberos 認証を構成します。

3.6.1 1. Kerberos、PEM サーバー、および PEM バックエンドデータベースをインストールします

- 認証サーバーとして機能するマシンに Kerberos をインストールします。
- PEM サーバーを別のマシンにインストールします。詳細については、[Installing the PEM Server](#) を参照してください。
- PEM サーバーと同じまたは異なるマシンに PEM バックエンドデータベース（Postgres/ EDB Postgres Advanced Server）をインストールします。詳細については、[EDB Docs Website](#) のインストールガイドを参照してください。

3.6.2 2. Kerberos サーバーにプリンシパルを追加する

- Apache Web サーバー (HTTPD / Apache2) および PEM バックエンドデータベースサーバー (PostgreSQL / EDB Postgres Advanced Server) の下にデプロイされた PEM Web アプリケーションのプリンシパルを追加します。

```
$ sudo kadmin.local -q "addprinc -randkey HTTP/<HOSTNAME_OF_PEM_SERVER>"
$ sudo kadmin.local -q "addprinc -randkey postgres/<HOSTNAME_OF_PEM_SERVER>"
```

ここで、HOSTNAME_OF_PEM_SERVER には Kerberos Server の REALM が含まれている必要があります。例、pembdb.edbpem.org を PEM サーバーのホスト名として指定し、edbpem.org を REALM として指定できます。

!!! Note * -PEM Web アプリケーションと PEM バックエンドデータベースサーバーが異なるマシン上にある場合、両方のホスト名は異なります。

3.6.3 3. Kerberos サーバーからキーテーブルを抽出する

a. PEM Web アプリケーションおよび PEM バックエンドデータベースサーバーのキーテーブルを Kerberos から抽出します。

```
sudo kadmin.local "ktadd -k <NAME_OF_PEM_WEB_FILE>.keytab HTTP/<HOSTNAME_OF_PEM_
→SERVER>"
sudo kadmin.local "ktadd -k <NAME_OF_PEM_DB_FILE>.keytab postgres/<HOSTNAME_OF_PEM_
→SERVER>"
```

b. キーテーブルを Kerberos サーバーから PEM サーバーにコピーします。

```
scp <NAME_OF_PEM_WEB_FILE>.keytab <OS_USERNAME_ON_PEM_SERVER>@<HOSTNAME_OF_PEM_SERVER>
→:/tmp
scp <NAME_OF_PEM_DB_FILE>.keytab <OS_USERNAME_ON_PEM_SERVER>@<HOSTNAME_OF_PEM_SERVER>
→:/tmp
```

c. PEM サーバーで、キーテーブルを必要な場所に移動し、次のように所有権を変更します。

```
mv /tmp/<NAME_OF_PEM_WEB_FILE>.keytab <PEM_INSTALLATION_DIRECTORY>/share
chown pem <PEM_INSTALLATION_DIRECTORY>/share/<NAME_OF_PEM_WEB_FILE>.keytab
```

```
mv /tmp/<NAME_OF_PEM_DB_FILE>.keytab <DATA_DIRECTORY_OF_POSTGRES>/
chown enterprisedb <DATA_DIRECTORY_OF_POSTGRES>/<NAME_OF_PEM_DB_FILE>.keytab
```

- どこで、
- NAME_OF_PEM_WEB_FILE は、PEM Web アプリケーションのキーテーブルに指定された名前です。
- NAME_OF_PEM_DB_FILE は、PEM バックエンドデータベースサーバーのキーテーブルに指定された名前です。
- OS_USERNAME_ON_PEM_SERVER は、PEM サーバー上のオペレーティングシステムユーザの名前です。

- DATA_DIRECTORY_OF_POSTGRES は、インストールされた postgres (PostgreSQL/ EDB Postgres Advanced Server) のデータディレクトリのパスです。

3.6.4 4. PEM バックエンドデータベースサーバーの構成

- postgresql.conf ファイルにキーテーブルの場所を追加します

```
*   krb_server_keyfile='FILE: /<DATA_DIRECTORY_OF_POSTGRES>/<NAME_OF_PEM_DB_FILE>.  
→keytab'
```

- どこで、
- -NAME_OF_PEM_DB_FILE は、PEM バックエンドデータベースサーバーのキーテーブルに指定された名前です。
- -DATA_DIRECTORY_OF_POSTGRES は、インストールされた postgres (PostgreSQL/ EDB Postgres Advanced Server) のデータディレクトリのパスです。
- krb5.conf ファイルを編集します

```
*   $ sudo vim /etc/krb5.conf  
*   [libdefaults]  
*   default_realm = EDBPEM.ORG  
*   Forwardable = True  
*   [domain_realm]  
*   .edbpem.org = EDBPEM.ORG  
*   edbpem.org = EDBPEM.ORG  
*   [realms]  
*   EDBPEM.ORG = {  
*       kdc = krb5server.edbpem.org  
*       admin_server = krb5server.edbpem.org  
*   }
```

- データベースサーバを再起動して、変更を反映します

```
*   systemctl restart <POSTGRES_SERVICE_NAME>
```

- どこで、
- -POSTGRES_SERVICE_NAME は、Postgres (PostgreSQL/ EDB Postgres Advanced Server) データベースのサービス名前です。例: CentOS または RHEL プラットフォーム上の PostgreSQL 13 データベースの postgresql-13。

3.6.5 5. 初期チケットを取得してビューする

- **kinit** ユーティリティは、Kerberos チケットを取得してキャッシュします。このユーティリティは通常、ユーザが入力したパスワードを使用してキーディストリビューションセンター（KDC）から資格情報を復号化し、チケット許可チケットを取得するために使用されます。チケット許可チケットは、ユーザーの資格情報キャッシュに保存されます。
- **klist** ユーティリティを使用して、チケットの詳細をビューできます。

!!! Note * **kinit** と **klist** を使用するには、PEM サーバーとクライアントマシンに **Kerberos Client** をインストールする必要があります。

```
$ kinit <USERNAME@REALM>
$ klist
```

Kerberos チケットとともにプリンシパルを表示します。

!!! Note * ここで指定する **USERNAME@REALM** は、**pem_admin** ロールと **pem** データベースに対する **CONNECT** 権限を持つデータベースユーザでなければなりません。

3.6.6 6. PEM サーバーの構成

- PEM サーバーで PEM 構成スクリプトを実行して、Kerberos 認証を使用します

```
* $ sudo PEM_APP_HOST=<HOSTNAME_OF_PEM_SERVER> PEM_KRB_KTNAME=<PEM_INSTALLATION_
→DIRECTORY/share/<NAME_OF_PEM_WEB_FILE>.keytab <PEM_INSTALLATION_DIRECTORY>/bin/
→configure-pem-server.sh
```

- **.install-config** ファイルで **HOST** を構成します

```
* $ sudo vim <PEM_INSTALLATION_DIRECTORY>/share/.install-config
* HOST=`<HOSTNAME_OF_PEM_SERVER>`
```

- PEM サーバーが Kerberos 認証を使用する場合、
 - すべての監視対象サーバーは、デフォルトで同じ認証を使用します。デフォルトを上書きするには、**ALLOW_DATABASE_CONNECTION_WITHOUT_KERBEROS** パラメータを追加し、**config_local.py** ファイルで **True** に設定します。
 - 認証されたユーザプリンシパルにレルム（**USERNAME @ REALM**）が追加され、デフォルトでデータベースユーザ名前として渡されます。デフォルトを上書きするには、**PEM_USER_KRB_INCLUDE_REALM** パラメータを追加し、**config_local.py** ファイルで **False** に設定します。
- Apache サーバーを再起動します

```
* sudo systemctl restart <SERVICE_NAME>
```

- **gss** 認証メソッドを使用してデータベースサーバをリロードするには、**pg_hba.conf** の上部のエントリを編集します。

```
* host pem +pem_user <ip_of_pem_server>/32 gss
* host postgres +pem_user <ip_of_pem_server>/32 gss
```

```
* systemctl reload <POSTGRES_SERVICE_NAME>
```

- どこで、
- POSTGRES_SERVICE_NAME は、Postgres (PostgreSQL/ EDB Postgres Advanced Server) データベースのサービス名前です。例：CentOS または RHEL プラットフォーム上の PostgreSQL 13 データベースの postgresql-13。

!!! Note * PostgreSQL または EDB Postgres Advanced Server 12 以降を使用している場合、接続タイプを hostgssenc に指定して、gss 暗号化接続のみを許可できます。

3.6.7 7. ブラウザ設定

Spnego/Kerberos を使用するように PEM Web クライアントにアクセスするように、クライアントマシンのブラウザを構成します。

- Mozilla Firefox の場合
 - -about:config ページをローディングして、低レベルの Firefox 設定ページを開きます。-[検索] テキストボックスに network.negotiate-auth.trusted-uris と入力します-network.negotiate-auth.trusted-uris 設定をダブルクリックし、Kerberos HTTP SPNEGO で保護されているウェブサーバのホスト名またはドメインを入力します。マルチプルのドメインとホスト名をコンマで区切ります。-[検索] テキストボックスに次のように入力します。network.negotiate-auth.delegation-uris-network.negotiate-auth.delegation-uris 設定をダブルクリックし、Kerberos HTTP SPNEGO で保護されているウェブサーバのホスト名またはドメインを入力します。マルチプルのドメインとホスト名をコンマで区切ります。-[OK] をクリックします。
- Linux または MacOS 上の Google Chrome の場合：
 - ---auth-server-whitelist パラメータを google-chrome コマンドに追加します。例、Linux プロンプトから Chrome を実行するには、次のように google-chrome コマンドを実行します。

```
* google-chrome --auth-server-whitelist = "hostname/domain"
```

- PEM サーバーの構成後、ブラウザで PEM Web インタフェースにアクセスできます。以下に移動します：

```
* https://<ip_address_of_PEM_server>:8443/pem
```

!!! Note * Postgres クラスターへの接続中に次のエラーが発生した場合：

- psql -h hostname template1 psql: GSSAPI continuation error: Unspecified GSS failure. Minor code may provide more information GSSAPI continuation error: Key version is not available

- 解決策は、`ktutil` を使用してキータブに追加の暗号化タイプを追加するか、AD のすべての暗号システムで Postgres キータブを再作成することです。

3.7 Configuring the PEM Server to use Windows Active Directory Domain Services for Kerberos Authentication (SSPI)

Windows Active Directory ドメインサービスは、`ip addresses` ではなくホスト名で動作します。`Active Directory Domain Services` を使用して PEM サーバーでシングルサインオンを使用するには、DNS を使用してホスト名で次のマシンを構成する必要があります。

Windows Server (ドメインコントローラー) -PEM サーバー (PEM Web サーバーおよび PEM バックエンドデータベースサーバー)-クライアントマシン-例、Windows Active Directory の REALM は `edbpem.internal` であり、Windows Server ホスト名は `Krb5server.edbpem.internal`、PEM サーバーホスト名は `pem.edbpem.internal` として設定できますクライアントのホスト名は `pg12.edbpem.internal` です。

手順に従って、PEM サーバーの Kerberos 認証を構成します。

3.7.1 1. Active Directory、PEM サーバー、および PEM バックエンドデータベースサーバーをインストールします

- 認証サーバーとして機能する Windows サーバー (ドメインコントローラー) に Active Directory をインストールします。また、Kerberos 認証を使用するように Active Directory ドメインサービスを構成し、それが稼働している必要があります。
- PEM サーバーを別の Linux マシンにインストールします。詳細については、[Installing the PEM Server](#) を参照してください。
- PEM サーバーと同じまたは異なる Linux マシンに PEM バックエンドデータベース (Postgres/ EDB Postgres Advanced Server) をインストールします。詳細については、[EDB Docs Website](#) のインストールガイドを参照してください。

3.7.2 2. Active Directory でユーザーを作成して、サービスプリンシパルとマップする

Windows Server の Active ディレクトリにユーザを作成して、PEM Web アプリケーションの HTTP サービスプリンシパルにマップします。

- a。Active Directory Users and Computers を開く-> <DOMAIN_NAME>-> Users->右クリック-> New-> User

![Active Directory Users and Computers](../images/active_directory_users_and_computers.png)

- b。ユーザの詳細を入力してください

```
![Active Directory Users and Computers Create User]( ../images/active_directory_
→users_and_computers_create_user.png)
```

- c。パスワードを入力し、User must change password at next logon の選択を解除し、User cannot change password および make を選択します。

```
![Active Directory Users and Computers set password]( ../images/active_directory_
→users_and_computers_set_password.png)
```

- d。ユーザの詳細を確認する

```
![Active Directory Users and Computers User Details]( ../images/active_directory_
→users_and_computers_user_details.png)
```

- e。PEMServer Web Properties で、ユーザーを Domain Users グループのメンバとして追加します。

```
![PEM Server Web Properties]( ../images/pem_server_web_properties_member_of.png)
```

- PEM バックエンドデータベースの Postgres サービスプリンシパルとマップするために、Windows Server の Active ディレクトリにユーザ（pemserverdb など）を作成します。
 - 上記のように、a から e までの手順を繰り返します。

3.7.3 3. Active Directory からキーテーブルを抽出する

- サービスプリンシパルのキーテーブルを抽出し、上記の手順で作成した各ドメインユーザーにマップします。

a。Windows PowerShell ユーザとして Windows PowerShell を開き、ユーザー pemserverweb を使用して HTTP サービスプリンシパルマッピングのキーテーブルを作成し、ユーザ pemserverdb を使用して Postgres サービスプリンシパルのキーテーブルを作成しユーザ。

```
ktpass /out pemserver.keytab /princ HTTP/pem.edbpem.internal@EDBPEM.INTERNAL /mapuser_
→pemserverweb /crypto AES256-SHA1 +rndpass /target EDBPEM.INTERNAL -ptype KRB5_NT_
→PRINCIPAL
ktpass /out pemdb.keytab /princ postgres/pem.edbpem.internal@EDBPEM.INTERNAL /mapuser_
→pemserverdb /crypto AES256-SHA1 +rndpass /target EDBPEM.INTERNAL -ptype KRB5_NT_
→PRINCIPAL
```

どこで、

- pemserver.keytab は、PEM Web アプリケーションのキーテーブルの名前です。
- pemdb.keytab は、PEM バックエンドデータベースサーバーのキーテーブルの名前です。
- pem.edbpem.internal@EDBPEM.INTERNAL は PEM サーバーのホスト名です（ここで@EDBPEM.INTERNAL は@REALM を意味します）
- pemserverweb は PEM Web アプリケーションのユーザです

- pemserverdb は、PEM バックエンドデータベースサーバーのユーザです。
- EDBPEM.INTERNAL はターゲットのドメインです

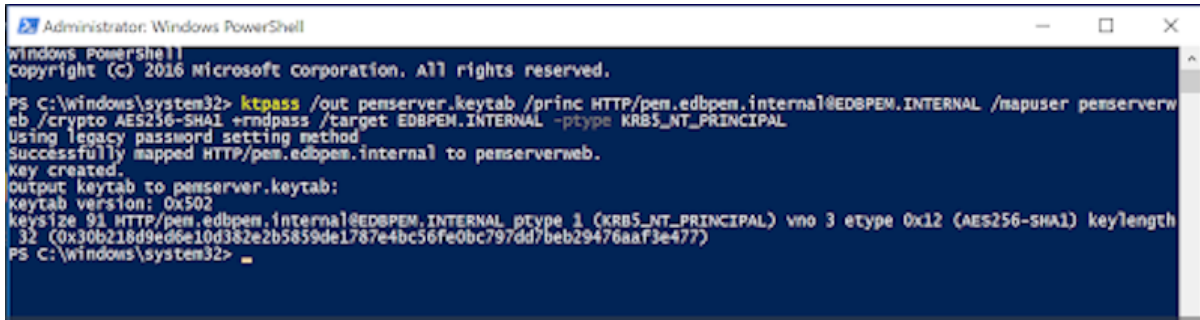


図 1: Windows PowerShell ktpass

!!! Note * コマンドライン引数+rndpass は、ドメインユーザー pemserverweb のパスワードをランダムパスワードにリセットします。

b. Accounts タブでユーザーアカウントの Kerberos サポートを追加します

!!! Note * [アカウント] タブで、User logon name は HTTP/pem.edbpem.internal@EDBPEM.INTERNAL を示し、pemserverweb ユーザに対して Delegation タブが有効になっています。

c. 作成されたユーザーの Delegation タブで Trust this user for delegation to any service (Kerberos only) オプションを選択します。

d. 両方のキーテーブルを PEM サーバーホスト（または、異なるホストにインストールされている場合は、それぞれ PEM Web サーバーおよび PEM バックエンドデータベースサーバーホスト）にコピーします。

e. PEM サーバーで、キーテーブルを必要な場所に移動し、次のように所有権を変更します。

```
mv /tmp/pemserver.keytab <PEM_INSTALLATION_DIRECTORY>/share
chown pem <PEM_INSTALLATION_DIRECTORY>/share/pemserver.keytab
```

```
mv /tmp/pemdb.keytab <DATA_DIRECTORY_OF_POSTGRES>/
chown enterprisedb <DATA_DIRECTORY_OF_POSTGRES>/pemdb.keytab
```

- どこで、
- OS_USERNAME_ON_PEM_SERVER は、PEM サーバー上のオペレーティングシステムユーザの名前です。
- DATA_DIRECTORY_OF_POSTGRES は、インストールされた postgres (PostgreSQL/ EDB Postgres Advanced Server) のデータディレクトリのパスです。

PEMServer Web Properties

Organization	Member Of	Dial-in	Environment	Sessions
Remote control	Remote Desktop Services Profile			COM+
General	Address	Account	Profile	Telephones
		Delegation		

User logon name:

HTTP/pem.edbpem.intenal @EDBPEM.INTERNAL

User logon name (pre-Windows 2000):

EDBPEM\ pemserverweb

Logon Hours... Log On To...

☐ Unlock account

Account options:

- ☐ Use only Kerberos DES encryption types for this account
- ☒ This account supports Kerberos AES 128 bit encryption.
- ☒ This account supports Kerberos AES 256 bit encryption.
- ☐ Do not require Kerberos preauthentication

Account expires

☒ Never

☐ End of: Wednesday, July 14, 2021

OK Cancel Apply Help

図 2: PEM Server Web Properties

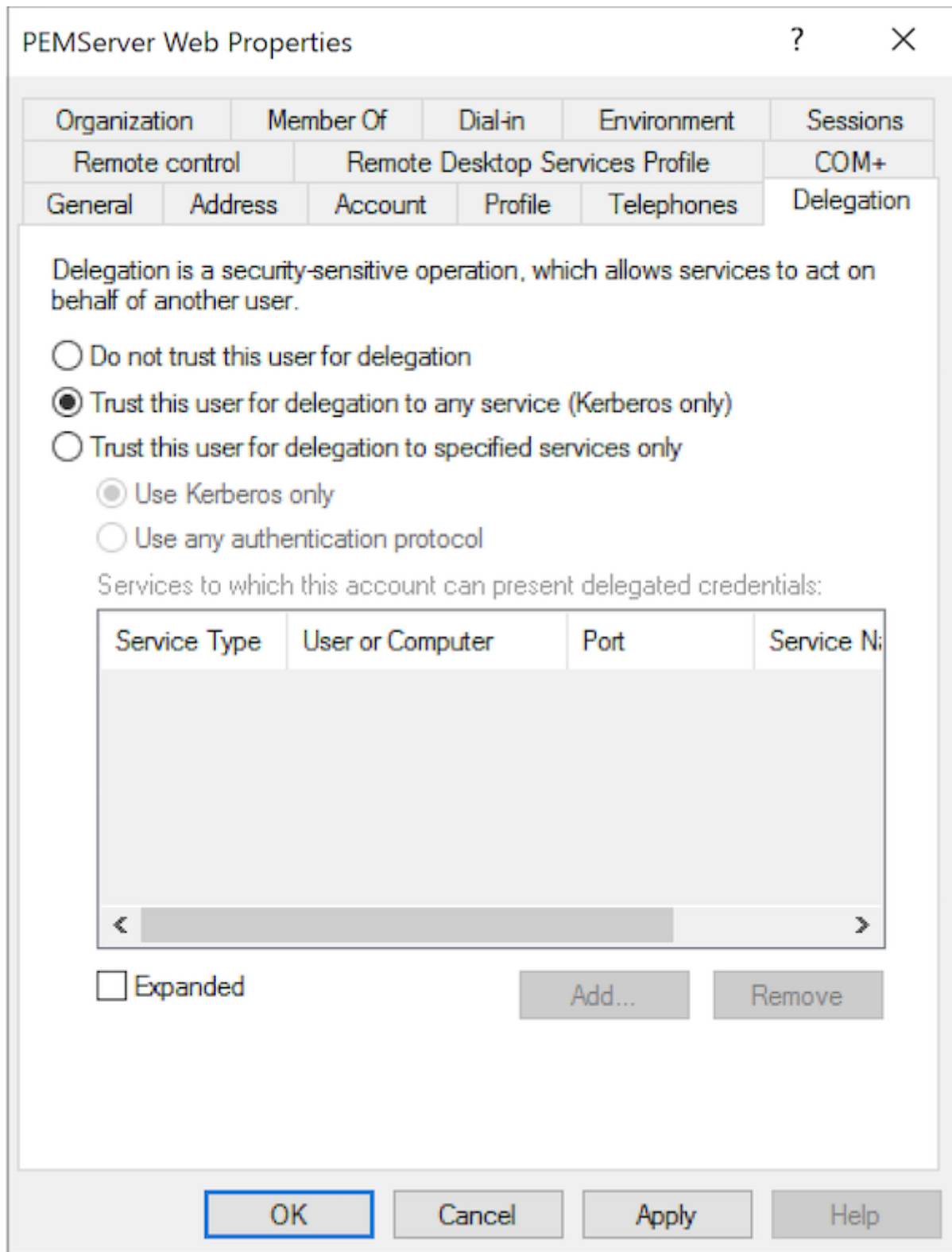


図 3: PEM Server Web Properties Delegation

3.7.4 4. PEM バックエンドデータベースサーバーの構成

- postgresql.conf ファイルにキーテーブルの場所を追加します

```
* krb_server_keyfile='FILE:/<DATA_DIRECTORY_OF_POSTGRES>/pemdb.keytab'
```

- どこで、
- -DATA_DIRECTORY_OF_POSTGRES は、インストールされた postgres (PostgreSQL/EDB Postgres Advanced Server) のデータディレクトリのパスです。
- krb5.conf ファイルを編集します。

```
* $ sudo vim /etc/krb5.conf
* [libdefaults]
* default_realm = EDBPEM.INTERNAL
* Forwardable = True
* [domain_realm]
* .edbpem.org = EDBPEM.INTERNAL
* edbpem.org = EDBPEM.INTERNAL
* [realms]
* EDBPEM.INTERNAL = {
*     kdc = krb5server.edbpem.internal
*     admin_server = krb5server.edbpem.internal
* }
```

- データベースサーバを再起動して、変更を反映します

```
* systemctl restart <POSTGRES_SERVICE_NAME>
```

- どこで、
- -POSTGRES_SERVICE_NAME は、Postgres (PostgreSQL/EDB Postgres Advanced Server) データベースのサービス名前です。例：CentOS または RHEL プラットフォーム上の PostgreSQL 13 データベースの postgresql-13。

3.7.5 5. 初期チケットを取得してビューする

- kinit ユーティリティは、Kerberos チケットを取得してキャッシュします。このユーティリティは通常、ユーザが入力したパスワードを使用してキーディストリビューションセンター (KDC) から資格情報を復号化し、チケット許可チケットを取得するために使用されます。チケット許可チケットは、ユーザーの資格情報キャッシュに保存されます。
- klist ユーティリティを使用して、チケットの詳細をビューできます。

!!! Note * kinit と klist を使用するには、PEM サーバーとクライアントマシンに Kerberos Client をインストールする必要があります。

```
$ kinit <USERNAME@REALM>
$ klist
```

Kerberos チケットとともにプリンシパルが表示されます。

!!! Note * ここで指定する USERNAME@REALM は、pem_admin ロールと pem データベースに対する CONNECT 権限を持つデータベースユーザでなければなりません。

3.7.6 6. PEM サーバーの構成

- PEM サーバーで PEM 構成スクリプトを実行して、Kerberos 認証を使用します

```
* $ sudo PEM_APP_HOST=pem.edbpem.internal PEM_KRB_KTNAME=<PEM_INSTALLATION_
→DIRECTORY/share/pemserver.keytab <PEM_INSTALLATION_DIRECTORY>/bin/configure-pem-
→server.sh
```

- .install-config ファイルで HOST を構成します

```
* $ sudo vim <PEM_INSTALLATION_DIRECTORY>/share/.install-config
* HOST=`pem.edbpem.internal`
```

- PEM サーバーが Kerberos 認証を使用する場合、
 - すべての監視対象サーバーは、デフォルトで同じ認証を使用します。デフォルトを上書きするには、ALLOW_DATABASE_CONNECTION_WITHOUT_KERBEROS パラメータを追加し、config_local.py ファイルで True に設定します。
 - 認証されたユーザプリンシパルにレルム (USERNAME @ REALM) が追加され、デフォルトでデータベースユーザ名前として渡されます。デフォルトを上書きするには、PEM_USER_KRB_INCLUDE_REALM パラメータを追加し、config_local.py ファイルで False に設定します。
- Apache サーバーを再起動します

```
* sudo systemctl restart <SERVICE_NAME>
```

- gss 認証メソッドを使用してデータベースサーバをリロードするには、pg_hba.conf の上部のエントリを編集します。

```
* host pem +pem_user <ip_of_pem_server>/32 gss
* host postgres +pem_user <ip_of_pem_server>/32 gss
```

```
* systemctl reload <POSTGRES_SERVICE_NAME>
```

- どこで、
- POSTGRES_SERVICE_NAME は、Postgres (PostgreSQL/ EDB Postgres Advanced Server) データベースのサービス名前です。例: CentOS または RHEL プラットフォーム上の PostgreSQL 13 データベースの postgresql-13。

!!! Note * Windows は gss 暗号化接続をサポートしていないため、接続タイプを `hostgssenc` として指定することはできません。

3.7.7 7. ブラウザ設定

Spnego/Kerberos を使用するように PEM Web クライアントにアクセスするように、クライアントマシンのブラウザを構成します。

- Mozilla Firefox の場合
 - `-about:config` ページをローディングして、低レベルの Firefox 設定ページを開きます。-[検索] テキストボックスに `network.negotiate-auth.trusted-uris` と入力します-`network.negotiate-auth.trusted-uris` 設定をダブルクリックし、Kerberos HTTP SPNEGO で保護されているウェブサーバのホスト名またはドメインを入力します。マルチプルのドメインとホスト名をコンマで区切ります。-[検索] テキストボックスに次のように入力します。`network.negotiate-auth.delegation-uris-network.negotiate-auth.delegation-uris` 設定をダブルクリックし、Kerberos HTTP SPNEGO で保護されているウェブサーバのホスト名またはドメインを入力します。マルチプルのドメインとホスト名をコンマで区切ります。-[OK] をクリックします。
- Linux または MacOS 上の Google Chrome の場合：
 - `---auth-server-whitelist` パラメータを `google-chrome` コマンドに追加します。例、Linux プロンプトから Chrome を実行するには、次のように `google-chrome` コマンドを実行します。

```
* google-chrome --auth-server-whitelist = "hostname/domain"
```

- PEM サーバーの構成後、ブラウザーで PEM Web インタフェースにアクセスできます。以下に移動します：

```
* https://<ip_address_of_PEM_server>:8443/pem
```

!!! Note * Postgres クラスターへの接続中に次のエラーが発生した場合：

- `psql -h hostname template1` `psql: GSSAPI continuation error: Unspecified GSS failure. Minor code may provide more information GSSAPI continuation error: Key version is not available`
- 解決策は、`ktutil` を使用してキータブに追加の暗号化タイプを追加するか、AD のすべての暗号システムで Postgres キータブを再作成することです。

3.8 Configuring the PEM Server to use Two-Factor Authentication

!!! Tip “New Feature” * Two-Factor Authentication (2FA) サポートは、PEM 8.2.0 以降で使用できます。

- PEM は 2FA の 2 つの方法をサポートしています。
 - メール認証-オーセンティケーターアプリ（例：Google オーセンティケーター）

2FA を有効にするには、これらの設定を `config.py` ファイルから `config_local.py` ファイルにコピーし、次のパラメーターを変更します。

Parameter	Description
MFA_ENABLED	これを true に設定して、2 要素認証を有効にします。デフォルト値は false です。
MFA_FORCE_REGISTRATION	これを true に設定して、ログイン時に 2 要素認証方法を強制的に登録するようにユーザーに依頼します。デフォルト値は false です。
MFA_SUPPORTED_METHODS	2FA 認証メソッドを使用するには `email` に設定し（電子メールでワンタイムコードをメール）、TOTP ベースのアプリケーション認証メソッドを使用するには `authenticator` に設定します。
MFA_EMAIL_SUBJECT	これをメール認証のメールのサブジェクトに設定します。デフォルト値は ` <app_name> - Verification Code` です。</app_name>
	NA

メールサーバーの構成

メール認証メソッドを使用するには、メールサーバー設定を構成する必要があります。

PEM サーバーは、PEM 構成に保存されている SMTP 構成または Flask-Mail を使用して、メールを送信できます。

PEM 構成から内部 SMTP 構成を使用してメール検証コードを送信するには、パラメータ `MAIL_USE_PEM_INTERNAL` を `True` に設定します。`False` に設定した場合、次のメール構成を使用して、ユーザが指定したメールアドレスでコードを送信します。

- `MAIL_SERVER = 'ローカルホスト'`
- `MAIL_PORT = 25`
- `MAIL_USE_TLS = False`
- `MAIL_USE_SSL = False`
- `MAIL_USERNAME = なし`
- `MAIL_PASSWORD = なし`
- `MAIL_DEFAULT_SENDER = なし`

これらの構成の詳細については、[Flask-Mail documentation](#) を参照してください。

3.9 Installing a PEM Agent on Linux Platforms

PEM エージェントは、1 つ以上のホスト上の 1 つ以上のサーバーをモニタできます。PEM エージェントの管理に関する包括的な情報については、[PEM Agent User Guide](#) を参照してください。

3.9.1 CentOS または RHEL ホストへの PEM エージェントのインストール

CentOS または RHEL システムでは、yum パッケージマネージャまたは dnf コマンドを使用して PEM エージェントをインストールできます。使用するインストールツールは、ホストオペレーティングシステムのバージョンに依存します。エージェントをインストールする前に、システムに以下の必須ソフトウェアが含まれていることを保証必要があります。

プラットフォーム固有のソフトウェアをインストール

- CentOS または RHEL 7.x ホストの場合：

```
* yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.
→rpm
```

- CentOS または RHEL 8.x ホスト：

```
* yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.
→rpm
```

!!! Note * **CentOS-Base.repo** ファイル (/etc/yum.repos.d にある) で [extras] リポジトリ定義を有効にする必要がある場合があります。

** Red Hat Network ユーザの場合**

rhel-<x>-server-extras-rpms リポジトリを有効にする必要があります。x は RHEL バージョンを指定します。

また、EPEL パッケージソフトを使用するには rhel-<x>-server-optional-rpms リポジトリを有効にする必要があります。* x *は RHEL バージョンを指定します。RHN-Classic の RHEL optional subchannel を有効にすると、リポジトリにアクセスできる make になります。証明書ベースのサブスクリプションがある場合は、EPEL パッケージソフトを使用するために rhel-<x>-server-eus-optional-rpms リポジトリも有効にする必要があります。必要なリポジトリの詳細については、Red Hat Subscription Management Guide を参照してください。

** edb.repo ファイルをインストールして設定**

EnterpriseDB リポジトリ構成ファイルを作成するには、スーパーユーザ権限を引き受けて、次のコマンドを呼び出します。

```
yum -y install https://yum.enterprisedb.com/edb-repo-rpms/edb-repo-latest.noarch.rpm
```

リポジトリ構成ファイルの名前付けは edb.repo です。ファイルは/etc/yum.repos.d にあります。edb.repo ファイルを作成したら、次のコマンドを使用して、baseurl 仕様の USERNAME および PASSWORD プレースホルダーを <username> および < に置き換えます。登録済み EDB ユーザのパスワード>

```
sed -i "s@<username>:<password>@USERNAME:PASSWORD@" /etc/yum.repos.d/edb.repo
```

**** PEM エージェントのインストール****

- `yum install` コマンドを使用して、CentOS または RHEL 7.x または 8.x に PEM エージェントをインストールします。

```
* yum install edb-pem-agent
```

- また、`dnf` コマンドを使用して、CentOS または RHEL 8.x に PEM エージェントをインストールできます。

```
* dnf install edb-pem-agent
```

インストールが完了すると、インストールされたパッケージソフトと依存関係のリストを確認できます。

```

root@localhost:/etc/yum.repos.d
File Edit View Search Terminal Help
Is this ok [y/N]: y
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : edb-as12-server-libs-12.2.3-1.rhel7.x86_64      1/8
  Installing : libcurl-pem-7.61.1-2.rhel7.x86_64             2/8
  Installing : boost-atomic-1.53.0-27.el7.x86_64             3/8
  Installing : boost-program-options-1.53.0-27.el7.x86_64    4/8
  Installing : snmp++-3.3.8-1.rhel7.x86_64                   5/8
  Installing : boost-regex-1.53.0-27.el7.x86_64              6/8
  Installing : boost-chrono-1.53.0-27.el7.x86_64             7/8
  Installing : edb-pem-agent-7.12.0-2.rhel7.x86_64           8/8
  Verifying   : boost-chrono-1.53.0-27.el7.x86_64            1/8
  Verifying   : boost-regex-1.53.0-27.el7.x86_64            2/8
  Verifying   : snmp++-3.3.8-1.rhel7.x86_64                  3/8
  Verifying   : boost-program-options-1.53.0-27.el7.x86_64  4/8
  Verifying   : boost-atomic-1.53.0-27.el7.x86_64           5/8
  Verifying   : edb-pem-agent-7.12.0-2.rhel7.x86_64         6/8
  Verifying   : libcurl-pem-7.61.1-2.rhel7.x86_64           7/8
  Verifying   : edb-as12-server-libs-12.2.3-1.rhel7.x86_64  8/8

Installed:
  edb-pem-agent.x86_64 0:7.12.0-2.rhel7

Dependency Installed:
  boost-atomic.x86_64 0:1.53.0-27.el7          boost-chrono.x86_64 0:1.53.0-27.el7
  boost-program-options.x86_64 0:1.53.0-27.el7  boost-regex.x86_64 0:1.53.0-27.el7
  edb-as12-server-libs.x86_64 0:12.2.3-1.rhel7  libcurl-pem.x86_64 0:7.61.1-2.rhel7
  snmp++.x86_64 0:3.3.8-1.rhel7

Complete!
[root@localhost yum.repos.d]#
[root@localhost yum.repos.d]#

```

システムで認識されないソースによって署名された RPM パッケージをインストールすると、`yum` はローカルサーバにキーをインポートするパーミッションを求める場合があります。プロンプトが表示され、パッケージソフトが信頼できるソースからのものであることに満足したら、`y` と入力し、`Return` を押して続行します。

インストール中に、`yum` は解決できない依存関係に遭遇する場合があります。存在する場合、手動で解決する必要がある依存関係のリストが提供されます。

`yum` または `dnf` を使用して PEM エージェントをインストールした後、PEM エージェントを登録する必要があります。詳細については、[Registering a PEM Agent](#) を参照してください。

3.9.2 Debian または Ubuntu ホストへの PEM エージェントのインストール

Debian または Ubuntu ホストに PEM をインストールするには、EDB リポジトリへのアクセスを許可する資格情報が必要です。リポジトリ [contact EDB](#) の資格情報を要求するには。

次の手順では、EDB apt リポジトリを使用して Debian パッケージをインストールします。コマンドを使用する場合、ユーザー名とパスワードを EDB が提供する資格情報に置き換えます。

1. ルートとしてログインします。

```
* sudo su -
```

2. EDB リポジトリを構成します。

- -Debian 9 の場合：

```
* sh -c 'echo "deb https://username:password@apt.enterprisedb.com $(lsb_release_
↪- cs)-edb/ $(lsb_release -cs) main" > /etc/apt/sources.list.d/edb-$(lsb_release -
↪cs).list'
```

- username:password は、EDB が提供する資格情報に置き換えられます。

- -Debian 10 の場合：

- a。 EnterpriseDB リポジトリをセットアップします。

```
* sh -c 'echo "deb [arch=amd64] https://apt.enterprisedb.com/$(lsb_release -cs)-
↪edb/ $(lsb_release -cs) main" > /etc/apt/sources.list.d/edb-$(lsb_release -cs).list'
```

- b。 次のコマンドで、`username` および `password` プレースホルダーを EnterpriseDB 資格情報に置き換えます。

```
* sh -c 'echo "machine apt.enterprisedb.com login <username> password <password>
↪" > /etc/apt/auth.conf.d/edb.conf'
```

3. 安全な APT リポジトリのサポートをシステムに追加します。

```
* apt-get install apt-transport-https
```

4. EDB 署名キーを追加します。

```
* wget -q -O -https://username:password@apt.enterprisedb.com/edb-deb.gpg.key | apt-
↪key add -
```

5. リポジトリメタデータを更新します。

```
* apt-get update
```

6. 次のコマンドを使用して、PEM エージェントの Debian パッケージをインストールします。

```
* apt-get install edb-pem-agent
```

apt-get を使用して PEM エージェントをインストールした後、PEM エージェントを登録する必要があります。詳細については、[Registering a PEM Agent](#) を参照してください。

3.9.3 SLES ホストへの PEM エージェントのインストール

次のコマンドを使用して、EDB リポジトリ構成ファイルを SLES ホストに追加します。

```
zypper addrepo https://zypp.enterprisedb.com/suse/edb-sles.repo
```

このコマンドは、/etc/zypp/repos.d ディレクトリに edb.repo 名前付けのリポジトリ構成ファイルを作成します。登録済みの EDB ユーザのユーザー名とパスワードを追加して、リポジトリ構成ファイルを変更します。

PEM をインストールする前に、前提条件パッケージソフトをインストールする必要があります。* sp_no * を使用しているサービスパック（SP2 または SP3）に置き換える次のコマンドを使用します。

```
SUSEConnect -p sle-module-legacy/12/x86_64
```

```
SUSEConnect -p sle-sdk/12/x86_64
```

```
zypper addrepo https://download.opensuse.org/repositories/Apache:Modules/SLE_12_<sp_
↪no>/Apache:Modules.repo
```

```
zypper addrepo http://download.opensuse.org/repositories/Cloud:/OpenStack:/Newton:/
↪cisco-apic:/2.3.1/SLE_12_<sp_no>/ pem_opensuse_boost
```

次に、リポジトリをリフレッシュして PEM エージェントを追加できます。

```
zypper refresh
```

```
zypper install edb-pem-agent
```

PEM エージェントをインストールしたら、エージェントを登録する必要があります。詳細については、[Registering a PEM Agent](#) を参照してください。

3.10 Registering a PEM 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。パッケージを使用して PEM エージェントをインストールする場合、pemworker ユーティリティを使用してエージェントを登録できます。

PEM エージェントパッケージは、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（スーパーユーザ権限を持つ）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 の場合、 <code>~/pem</code> Windows の場合、 <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 Yes です。
<code>--group</code>	エージェントが表示されるグループの名前。
<code>--team</code>	監視対象データベースサーバ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。
<code>--allow_server_restart</code>	<code>allow-server_restart</code> パラメータを有効にして、PEM が監視対象サーバをリスタートできるようにします。デフォルト値は True です。
<code>--allow_batch_probes</code>	<code>allow-batch-probes</code> パラメータを有効にして、PEM がこのエージェントでバッチプローブを実行できるようにします。デフォルト値は False です。
<code>--batch-script-user</code>	バッチ/シェルスクリプトの実行に使用するオペレーティングシステムユーザを指定します。デフォルト値は none です。このパラメータを空白のままにするか、指定され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 False です。
<code>--enable-smtp</code>	マルチプルの PEM エージェント（7.13 以下）で true に設定すると、重複する電子メールがさらに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、重複する電子メールが少なくなる場合があります。

次のページに続く

表 3 – 前のページからの続き

Option	Description
<code>--enable-snmp</code>	マルチプルの PEM エージェント（7.13 以下）で true に設定すると、より多くの重複トラップを送信する場合があります。一方、PEM エージェント（7.14 以降）の場合、重複するトラップが少なくなる可能性があります。
<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。
	NA

データベースサーバ `restart` が `pemagent` に必要な 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）を使用する場合は、`agent.cfg` ファイルで `allow_server_restart` の値を `true` に設定する必要があります。

!!! Note * PEM バージョン 7.11 以降がインストールされている PEM エージェントによって実行されるシェル/バッチスクリプトを構成する場合、`batch_script_user` parameter 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、スクリプトを実行する `root` ユーザを使用して `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 データベースで変更が行われる前に、接続エージェントは PEM データベースサーバで認証されます。`pemworker` ユーティリティを呼び出す場合、PEM サーバー管理ユーザロール（`postgres`）に関連付けられたパスワードを提供する必要があります。管理パスワードを指定するには 3 つの方法があります。できること：

- `PEM_MONITORED_SERVER_PASSWORD` 環境変数を設定します。
- コマンドラインで `PGPASSWORD` キーワードを使用してパスワードを指定します。
- `.pgpass` ファイルにエントリーを作成します。

パスワードを提供しないと、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。

** PEM エージェント構成パラメーターの設定**

PEM エージェントネイティブパッケージは、`/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、次の設定パラメータ値を `true` に変更します。

```
heartbeat_connection = true
```

デフォルトでは、`heartbeat_connection` の値は `false` ですが、`--enable-heartbeat-connection` オプションを使用して、`pemworker` ユーティリティで `pemagent` の登録中に値をオーバーライドできます。

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。

CentOS または RHEL 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、PEM Web インタフェースの Global Overview および Object browser 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。

第4章 The PEM Web Interface

PEM サーバーとエージェントをインストールした後、PostgreSQL または Advanced Server インスタンスのモニタリングと管理をスタートするように PEM を構成できます。PEM サーバーインストーラは、PEM Web インタフェースをインストールします。インタフェースを使用して、監視対象サーバーに存在するオブジェクトに関する情報を確認したり、PEM サーバーによって収集された統計情報を確認したりできます。

PEM をインストールして構成したら、ブラウザを使用して PEM Web インタフェースにアクセスできます。ブラウザを開き、次の場所に移動します。

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

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

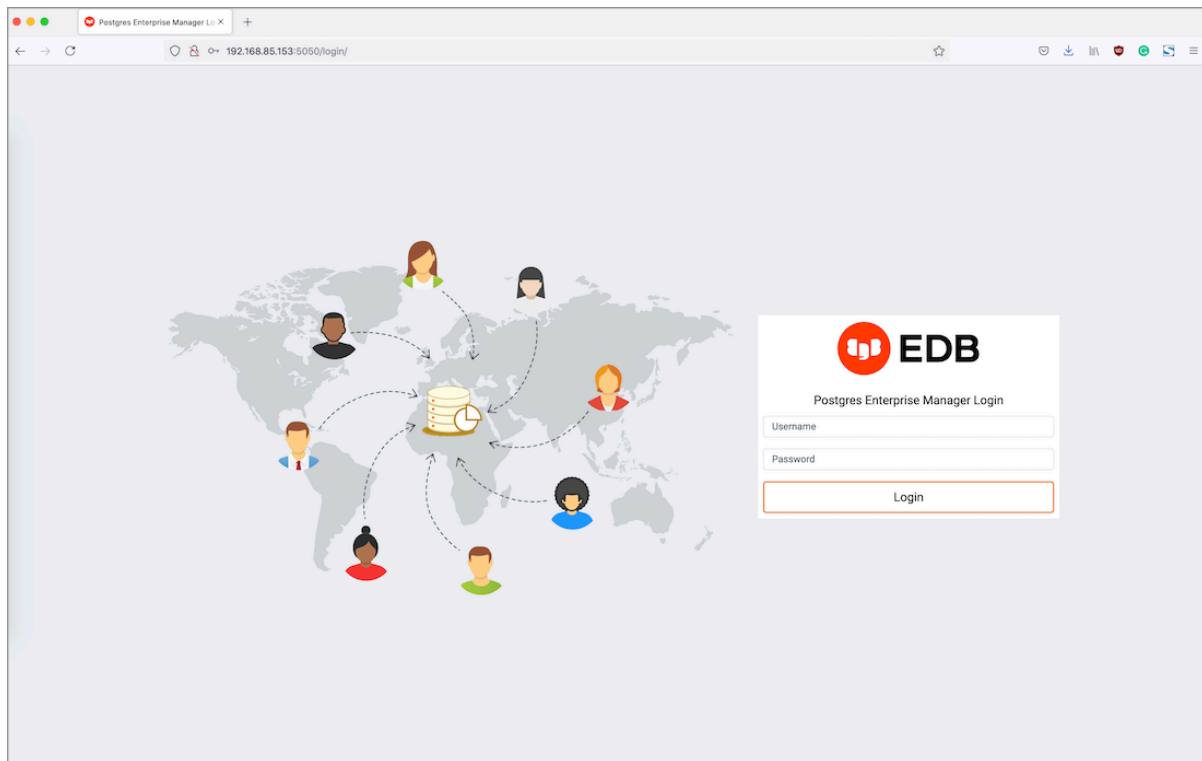


図 1: The PEM Web Login page

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

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

Login ボタンをクリックして、PEM サーバーに接続します。

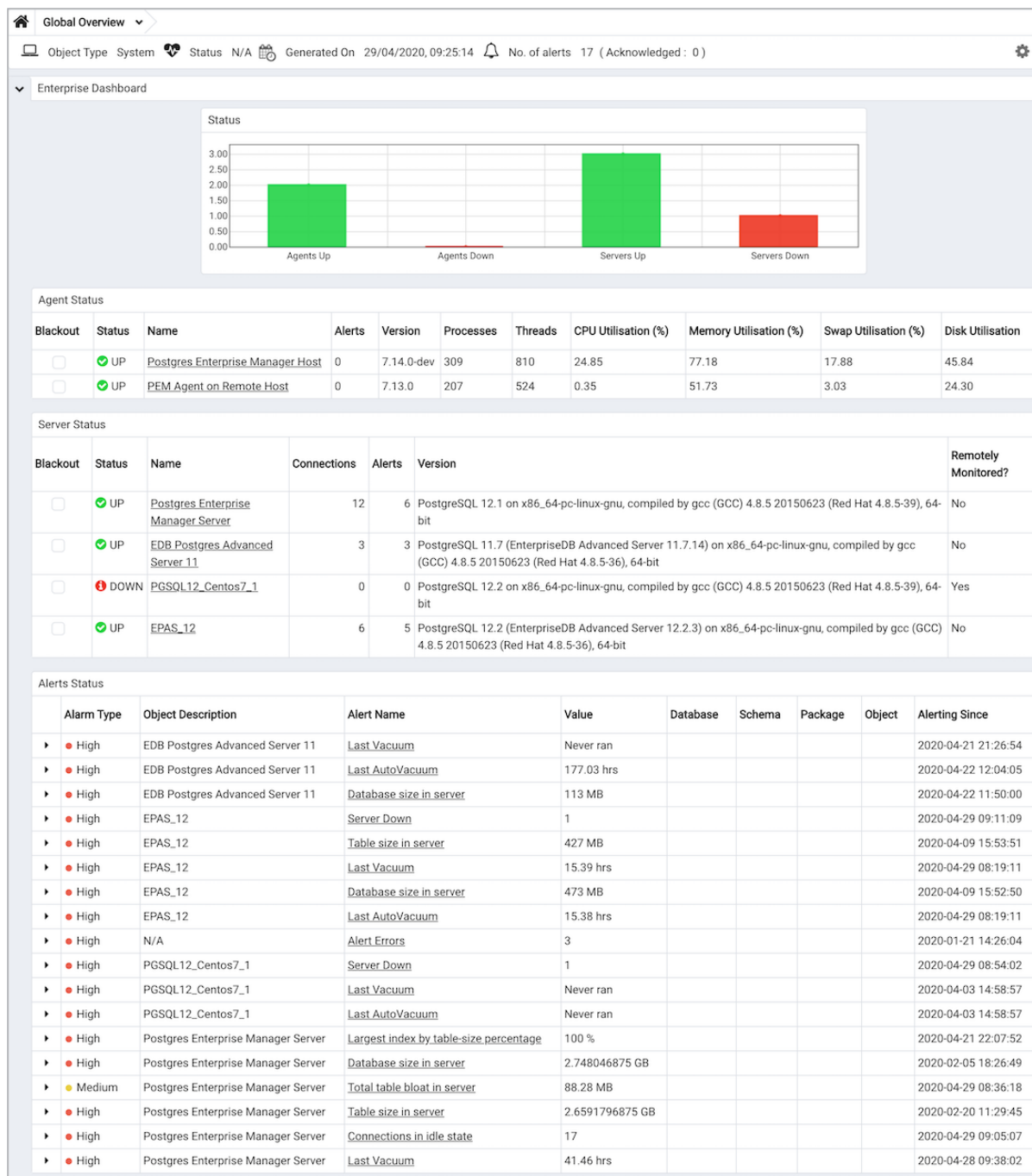


図 2: The Global Overview Dashboard

PEM Web インタフェースを使用してデータベースサーバを管理またはモニタする前に、サーバーを PEM サーバーに登録する必要があります。サーバーを登録するとき、サーバーへの接続を記述し、接続の認証情報を提供し、管理設定を指定します（オプションでエージェントをバインドします）。

サーバーは管理されている場合と管理されていない場合があります。

- **managed** サーバーは PEM エージェントにバインドされています。PEM エージェントは、バインド先のサーバーをモニタし、タスクを実行したり、PEM ダッシュボードに表示する統計をレポートします。管理対象サーバーは、カスタムアラートなどの拡張 PEM 機能にアクセスできます。サーバー

を登録するときに、必要に応じて PEM によって管理対象サーバーを再起動することもできます。

- **unmanaged** サーバーは PEM エージェントにバインドされていません。非管理対象サーバーでデータベースオブジェクトを作成できますが、拡張 PEM 機能（カスタムアラートなど）は非管理対象サーバーではサポートされていません。

また、登録するサーバーの `pg_hba.conf` ファイルが PEM Web インタフェースのホストからの接続を許可していることを保証する必要があります。

PEM Web インタフェースに関するオンラインヘルプ情報にアクセスするには、メニューバーから **Help** を選択します。オンラインヘルプのコンテンツは、[EDB website](#) から入手できます。

第5章 Uninstalling Postgres Enterprise Manager Components

PEM サーバーまたはエージェントをアンインストールするプロセスはプラットフォーム固有です。PEM サーバーのパッケージの名前は `edb-pem-server` で、PEM エージェントのパッケージの名前は `edb-pem-agent` です。

ホストから PEM サーバーパッケージをアンインストールする場合、同じホストにインストールされている PEM エージェントパッケージはアンインストールされません。ただし、PEM Agent パッケージをアンインストールすると、同じホストにインストールされている PEM サーバーパッケージもアンインストールされます。

!!! Note * PEM エージェントをアンインストールする前に、最初にエージェントの登録を解除する必要があります。 `pemworker` コマンドラインユーティリティを使用して、エージェントの登録を解除できます。その後、アンインストール手順に進むことができます。

5.1 CentOS または RHEL ホストからの PEM コンポーネントのアンインストール

さまざまな `rpm`、`yum remove`、または `yum erase` コマンドを使用して、インストールされたパッケージソフトを CentOS または RHEL 7.x または 8.x ホストから削除できます。また、`dnf remove` コマンドを使用して、インストールしたパッケージを CentOS または RHEL 8.x ホストから削除します。パッケージを削除しても、PEM データディレクトリは破損しません。

- インストールされたパッケージを削除するために `rpm` コマンドを呼び出すときに、`-e` オプションを含めます。コマンド構文は次のとおりです。

```
* rpm -e package_name
```

- `yum remove` コマンドを使用して、`yum` によってインストールされた PEM サーバーまたはエージェントパッケージを削除できます。パッケージを削除するには、ターミナルウィンドウを開き、スーパーユーザ権限を引き受けて、次のコマンドを入力します。

```
* yum remove package_name
```

- `yum erase` コマンドを使用して、`edb-pem` および `edb-pem-docs` 依存関係とともに `pem` サーバーまたはエージェントパッケージを削除できます。パッケージを削除するには、ターミナルウィンドウを開き、スーパーユーザ権限を引き受けて、次のコマンドを入力します。

```
* yum erase package_name
```

`package_name` は、削除するパッケージの名前です。

- `dnf remove` コマンドを使用して、CentOS または RHEL 8.x ホストの `edb-pem` および `edb-pem-docs` 依存関係とともに `pem` サーバーまたはエージェントを削除できます。パッケージを削除するには、ターミナルウィンドウを開き、スーパーユーザ権限を引き受けて、次のコマンドを入力します。

```
* dnf remove package_name
```

5.2 Debian または Ubuntu ホストからの PEM コンポーネントのアンインストール

`apt-get remove` または `apt-get purge` コマンドを使用して、Debian または Ubuntu ホストから PEM サーバーまたはエージェントパッケージをアンインストールできます。

- 構成ファイルとデータディレクトリに影響を与えずに Debian または Ubuntu ホストから PEM サーバーまたはエージェントをアンインストール、次のコマンドを呼び出します。

```
* apt-get remove package_name
```

- PEM サーバーまたはエージェントを構成ファイルとデータディレクトリとともにアンインストール、次のコマンドを呼び出します。

```
* apt-get purge package_name
```

`package_name` は、削除するパッケージの名前です。

5.3 SLES ホストからの PEM コンポーネントのアンインストール

SLES ホストから PEM サーバーまたはエージェントをアンインストール、次のコマンドを呼び出します。

```
zypper remove package_name
```

`package_name` は、削除するパッケージの名前です。

第6章 Reference - Linux Service Script

- PEM サーバーは、サービススクリプトを使用して、構成管理、証明書管理、またはその他の管理タスクを実行するときに、必要に応じてサーバーをスタート、停止、またはリスタートできます。
- PEM サーバーが存在する Postgres サーバーには、サービススクリプトが含まれている必要があります。EDB によって生成された Windows の Postgres インストーラーは、サービススクリプトを作成します。ネイティブパッケージソフトのような別のソースから Postgres サーバーを使用している場合は、サービススクリプトを提供する必要があります。
 - ****CentOS または RHEL 7.x または 8.x ****では、サービススクリプトは`/usr/lib/systemd/system`ディレクトリにあります。
- サービススクリプトはプラットフォーム固有です。

Postgres サービスのカスタマイズについては、次をご覧ください。

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