



Postgres Enterprise Manager

リリース 7.15

Upgrade and Migration Guide

2020 年 08 月 25 日

目次

第 1 章	PEM インストールのアップグレード	2
1.1	Windows ホストでの PEM インストールのアップグレード	2
1.1.1	Windows ホストでの PEM エージェントのアップグレード	4
1.1.2	Windows ホストでの PEM サーバーのアップグレード	10
1.1.3	Windows ホストでの SQL プロファイラーのアップグレード	20
1.2	Linux ホスト上の RPM パッケージを使用した PEM インストールのアップグレード	22
1.2.1	RPM パッケージとともにインストールされた PEM エージェントのアップグレード	22
1.2.2	RPM パッケージとともにインストールされた PEM サーバーのアップグレード	24
1.2.3	Linux ホストでの SQL プロファイラーのアップグレード	26
1.3	隔離されたネットワークでの PEM リポジトリの作成	28
1.4	Linux ホストでのグラフィカル PEM インストールのアップグレード	29
1.4.1	グラフィカルインストーラーでインストールされた PEM サーバーのアップグレード	29
1.4.2	グラフィカルインストーラーでインストールされた PEM エージェントのアップグレード	33
1.4.3	Linux ホストでの SQL プロファイラーのアップグレード	36
第 2 章	BackingPostgres データベースのアップグレード	38
第 3 章	PostgresEnterpriseManager™ サーバーの移動	48
第 4 章	トラブルシューティング	59
4.1	PEM サーバーの再構成	59
4.2	pem.alert テーブルが復元に失敗する	61
第 5 章	結論	62
索引		63

このガイドは、PostgresEnterpriseManager (PEM) コンポーネントのアップグレードに関する詳細情報を提供します。

- **PEM インストールのアップグレード**-このセクションでは、PEM サーバー、PEM エージェント、および SQL プロファイラーをあるメジャーバージョンから別のメジャーバージョン（つまり、6.0 から 7.14）にアップグレードする方法について説明します。
- **バックアップデータベースのアップグレード**-このセクションでは、同じバージョンの PEM サーバーを維持しながら、バックアップデータベースのアップグレードに関する詳細情報を提供します。
- **PEM サーバーの移動**-このセクションでは、あるホストから別のホストへの PEM サーバーの移動に関する詳細情報を提供します。
- **トラブルシューティング**-このセクションでは、PEM のアップグレード中に発生する可能性のあるエラーのトラブルシューティングに関する詳細情報を提供します。

このドキュメントでは、***Postgres***という用語を使用して、PostgreSQL または AdvancedServer データベースを意味します。

PEM インストールのアップグレード

PEM インストールのアップグレードプロセスはプラットフォーム固有です。Windows 用の PEM グラフィカルインストーラーを使用して、Windows ホスト上の PEM エージェントまたはサーバーを更新できます。PEM7.8 リリースより前は、グラフィカルインストーラーまたは RPM を使用して、PEM エージェントまたはサーバーを Linux にインストールできました。PEM バージョン 7.8 以降、Linux 用の PEM グラフィカルインストーラーは廃止されています。Linux ホスト上の PEM エージェントまたはサーバーを下位バージョンから PEM7.9 以降のバージョンに更新するには、RPM を使用する必要があります。

PEM および SQL プロファイラーのインストーラーと RPM へのリンクは、EnterpriseDB Web サイト<<https://www.enterprisedb.com/enterprise-postgres/edb-postgres-enterprise-manager-pem/>>から入手できます。

1.1 Windows ホストでの PEM インストールのアップグレード

Windows ホスト上の PEM コンポーネントソフトウェアをアップグレードするには、次の順序で新しいバージョンの PEM コンポーネントインストーラーを起動するだけです。

1. PEM サーバーノードを除く各監視対象ノードで PEM エージェントインストーラーを呼び出します。
2. PEM サーバーインストーラーを起動します。このインストーラーは、PEM サーバーと PEM サーバーノードにある PEM エージェントの両方をアップグレードします。

インストール中に、コンポーネントインストーラーは既存のインストールを自動的に検出し、アップグレードを実行します。PEM エージェントとサーバーをアップグレードした後、SQL プロファイラーをアップグレードする必要があります。このステップはプラットフォーム固有です。

以下のセクションでは、Windows ホストでのアップグレードプロセスについて順を追って説明します。

1.1.1 Windows ホストでの PEM エージェントのアップグレード

現在 PEM エージェントによって監視されているシステムをより新しいバージョンの PEM エージェントにアップグレードするには、エージェントが監視しているシステムに新しいバージョンのプラットフォーム固有の PEM エージェントインストーラーをダウンロードして呼び出します。

ダウンロードしたインストーラーのアイコンを右クリックして Run as Administrator を選択すると、インストーラーを起動できます。PEM Agent Setup Wizard が開き、歓迎されます。



図 1 PEM エージェントインストーラーのウェルカムウィンドウ

続行するには、Next をクリックする前に License Agreement を読んで受け入れます。

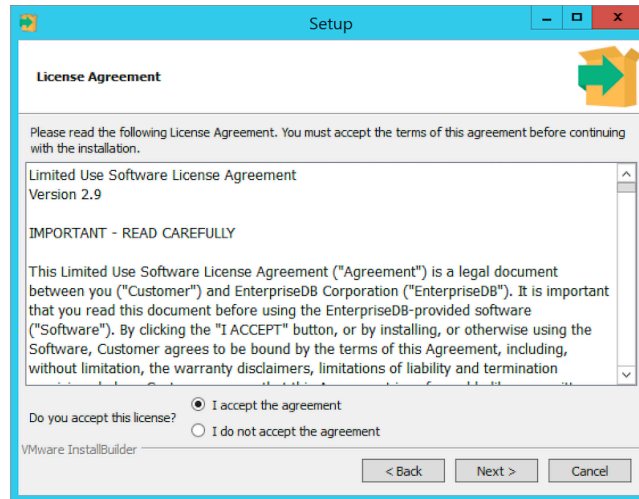


図 2 PEM ライセンス契約

セットアップウィザードは、既存のエージェントを自動的に検出し、インストールされているバージョンをアップグレードします。Next をクリックして続行します。

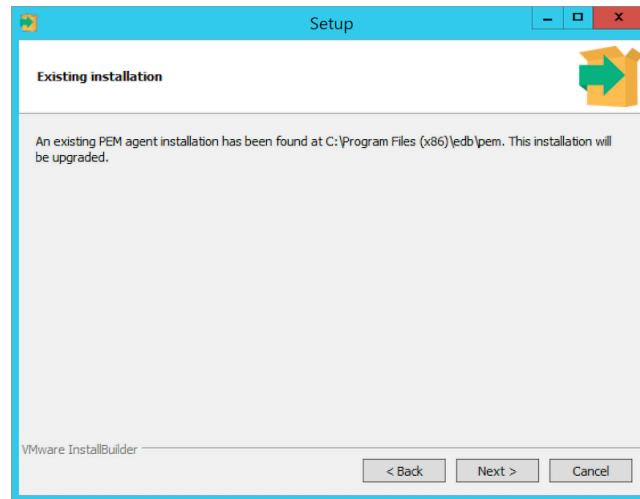


図 3 インストーラーは既存のインストールを検出します

pemAgent service account ダイアログは、PEM エージェントサービスが実行されるアカウントのパスワードを要求する場合があります。

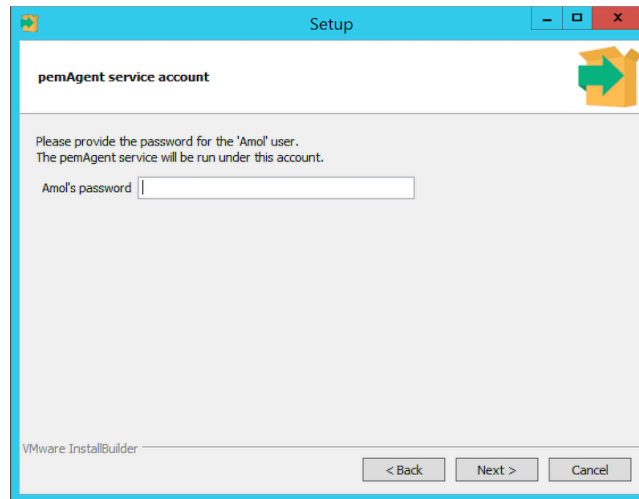


図 4 pemAgent サービスアカウントのパスワード

プロンプトが表示されたら、パスワードを入力し、Next をクリックして続行します。

Ready to Install ダイアログがインストールを開始しようとしていることを通知したら、Next をクリックして続行します。

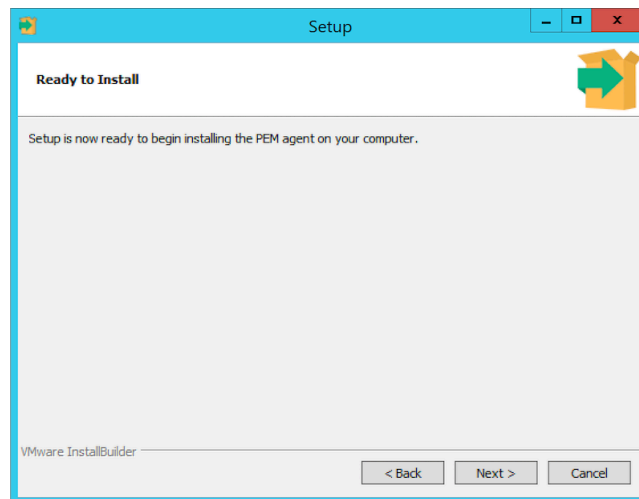


図 5 PEM エージェントがアップグレードを実行します

セットアップウィザードに進行状況バーが表示され、インストールされている各コンポーネントが通知されます。

インストールが完了すると、PEM Agent Setup Wizard が通知されます。Finish をクリックしてウィザードを終了し、ウィンドウを閉じます。

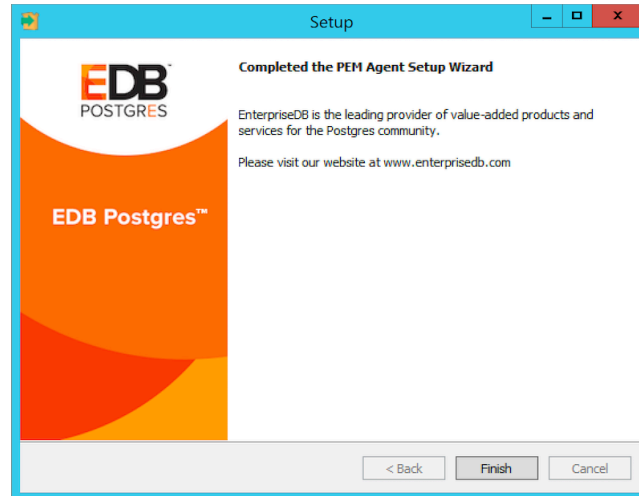


図 6 PEM エージェントセットアップウィザードは更新を終了しました

インストールが完了すると、ウィンドウがポップアップしてマシンを再起動します。yes をクリックして、マシンと httpd サービスを再起動します。

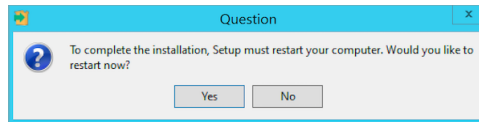


図 7 PEM エージェントウィザード-再起動ポップアップ

注釈: 以前のバージョンからバージョン 7.11 にアップグレードされた Windows 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すでに設定している場合、または設定する予定の場合は、で ‘*AllowBatchJobSteps*’ パラメーターを True に設定する必要があります。²‘ファイル。デフォルトでは、PEM エージェントはバッチ/シェルスクリプトを実行しません。

1.1.2 Windows ホストでの PEM サーバーのアップグレード

PEM サーバーインストーラーを使用すると、PEM サーバーのメジャーバージョン間で直接アップグレードできます（たとえば、最初にバージョン 6.0 にアップグレードしなくても、バージョン 5.0 からバージョン 7.5 に直接アップグレードできます）。

ダウンロードしたインストーラーのアイコンを右クリックして Run as Administrator を選択すると、インストーラーを起動できます。

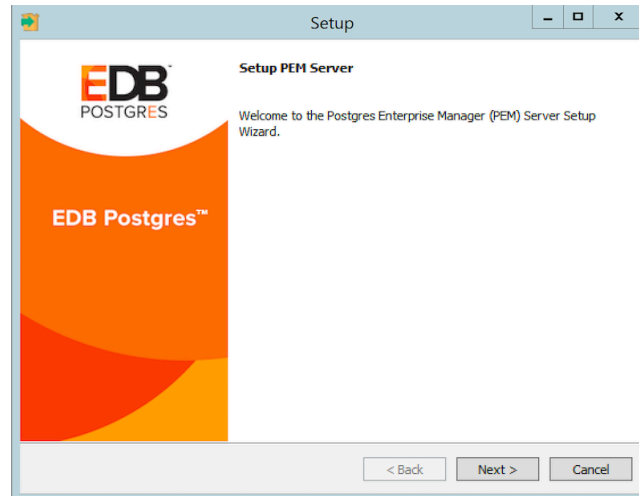


図 8 PEM サーバーセットアップウィザード

画像に示すように、PEM Server Setup Wizard はあなたを歓迎します。Next をクリックして、License Agreement に進みます。

PEM server setup wizard は License Agreement を受け入れるように促します。使用許諾契約を確認した後、I accept the agreement の横のラジオボタンをチェックし、Next をクリックして Existing installation dialog に進みます。

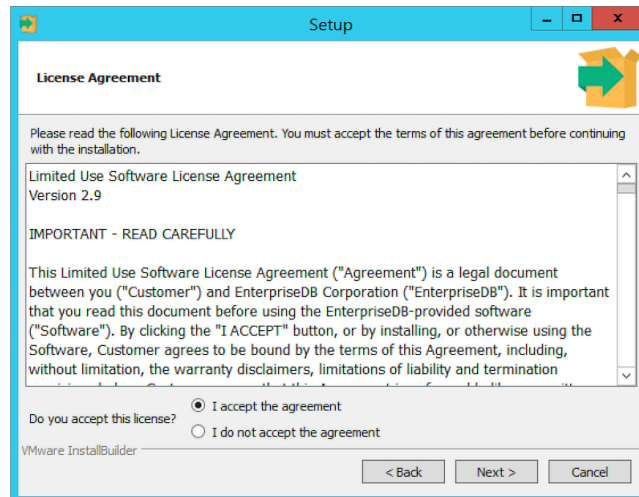


図 9 続行するにはライセンス契約に同意します

ウィザードは、既存の PEM サーバーのインストールについて PEM サーバーホストを確認します。ウィザードがインストールを見つけると、アップグレードを実行します。Next をクリックして続行します。

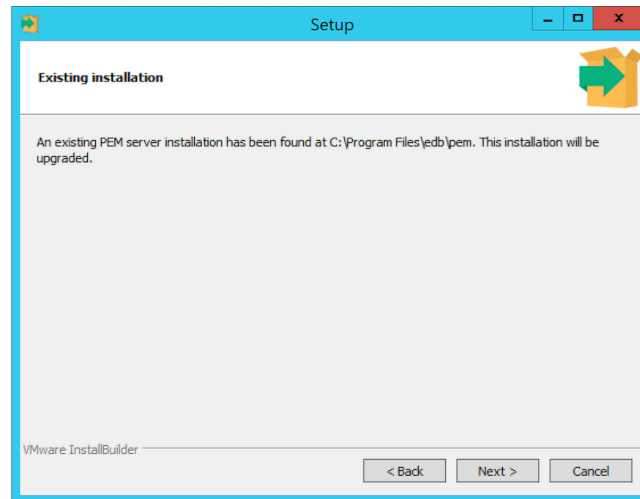


図 10 PEM サーバーインストーラーは、既存の PEM サーバーのインストールを検出します

PEM サーバーをアップグレードする前に、ウィザードは新しい PEM サーバーの要件が存在することを確認します。サポートするコンポーネントが欠落している場合、または新しい PEM インストールをサポートしないバージョンの場合、PEM インストールウィザードは依存関係をアップグレードする必要があることを通知し、必要なインストーラーを起動します。

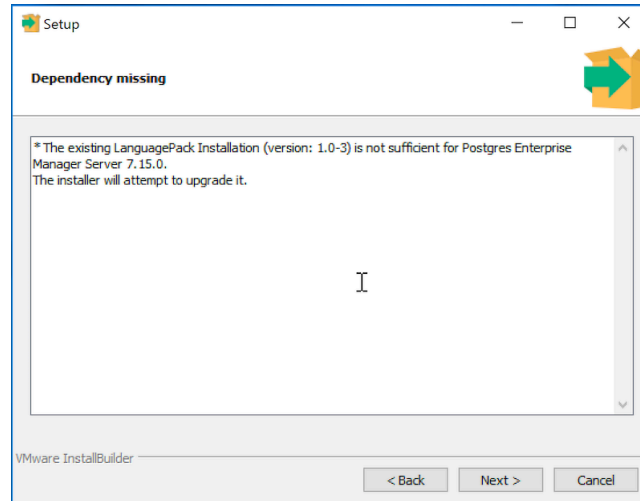


図 11 インストールには依存関係のアップグレードが必要です

インストールウィザードが依存関係のアップグレードを完了すると、マシンを再起動するかどうかを尋ねるウィンドウがポップアップ表示されます。

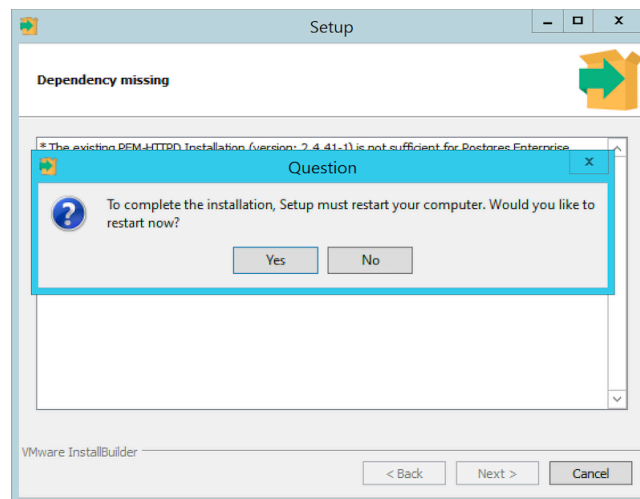


図 12 PEM サーバーアップグレード-マシンの再起動ポップアップ

アップグレードプロセスを続行するには、No をクリックします。

次に、ウィザードは Database Server Installation Details ダイアログを開き、PEM バックアップデータベースのデータベーススーパーユーザーの接続資格情報の入力を求めます。提供:

- User フィールドのデータベーススーパーユーザーの名前。
- Password フィールドのデータベーススーパーユーザーに関連付けられたパスワード。

Next をクリックして続行します。

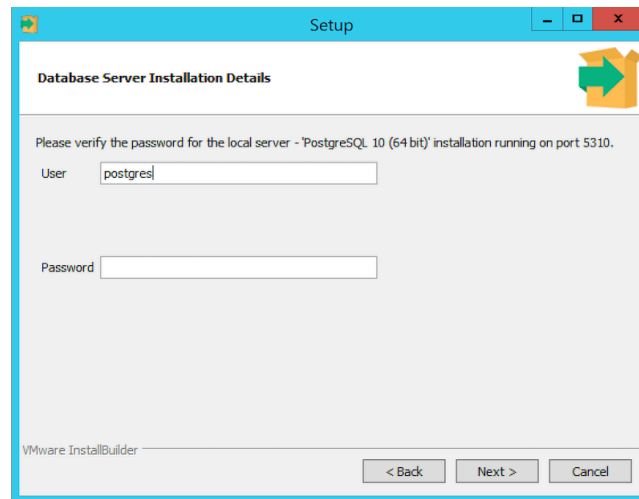


図 13 バックアップデータベースの接続情報を提供

pemAgent service account ダイアログは、PEM エージェントサービスが実行されるアカウントのパスワードを要求する場合があります。

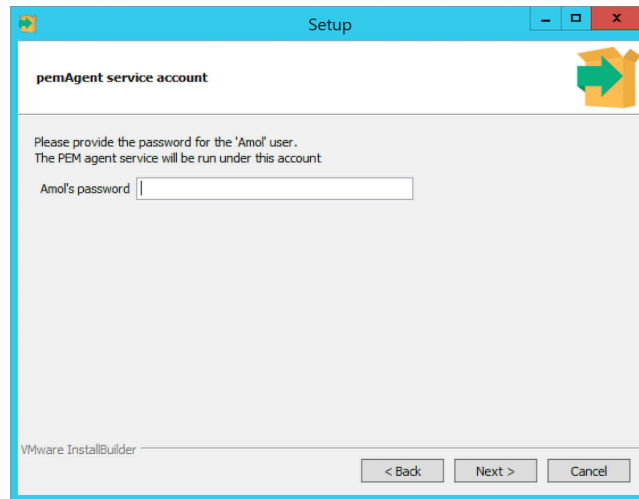


図 14 pemAgent サービスアカウントのパスワード

プロンプトが表示されたら、パスワードを入力し、Next をクリックして続行します。

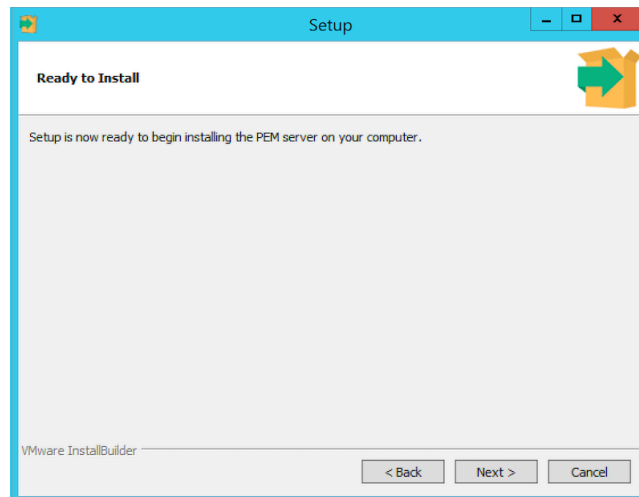


図 15 PEM セットアップウィザードは PEM サーバーをインストールする準備ができています

Ready to Install ダイアログは、セットアップウィザードがインストールを実行する準備ができたことを通知します。Next をクリックしてインストールを開始します。

インストール中、進行状況バーに更新の進行状況が表示されます。

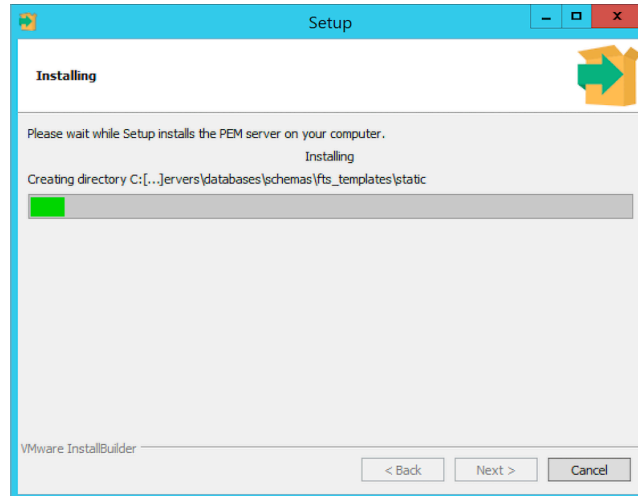


図 16 進行中の PEM サーバーのインストール

PEM サーバー（および PEM サーバーと同じホスト上にあるエージェント）を更新し、Web サービスを構成すると、PEM セットアップウィザードはサービスがリスンしているポートを通知します。PEM クライアントを使用して PEM サーバーに接続するときに、このポート番号を使用します。

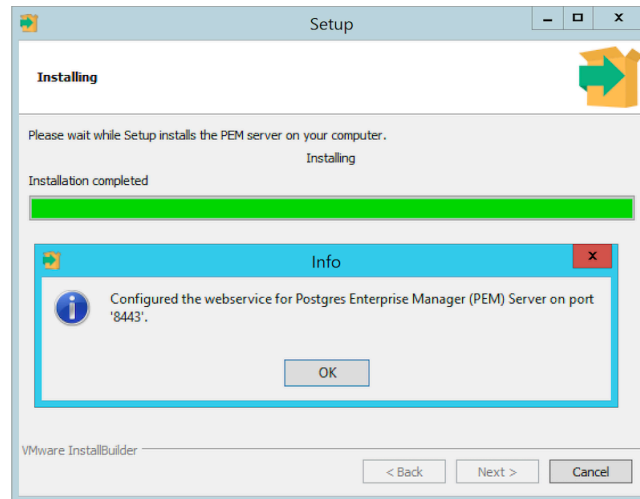


図 17 セットアップウィザードは *PEMWeb* サービスを設定します

OK をクリックして、Info ポップアップを閉じます。PEM サーバーセットアップウィザードは、インストールが完了したことを通知します。

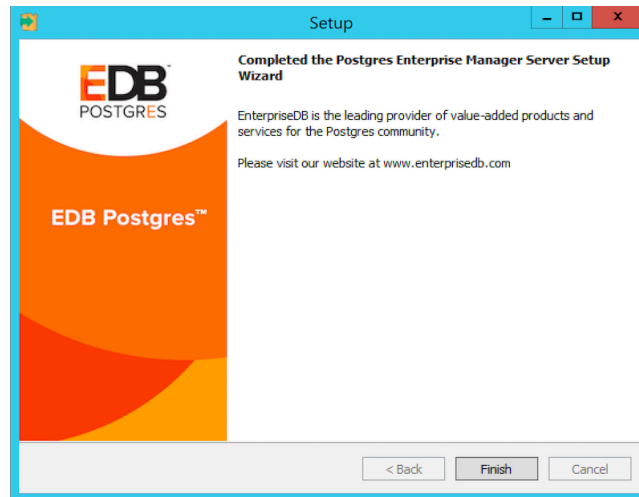


図 18 PEM サーバーのアップグレードが完了しました

ウィンドウがポップアップしてマシンの再起動を要求する場合は、yes をクリックしてマシンを再起動し、それにより httpd サービスを再起動します。

PEM バックエンドデータベースサーバーと PEM-HTTPD を異なるホストにインストールした場合は、PEM サーバーインストーラーを 2 回（各ホストに 1 回）実行する必要があります。言語パックインストーラーを抽出し、PEM インストーラーを起動する前に PEM-HTTPD のホストにインストールします。インストーラーを呼び出して言語パックを抽出する場合、次のキーワードを含めます。

```
--extract-languagepack <path>
```

ここで、<path> は言語パックインストーラーを抽出するための既存のパスを指定します。

注釈: デフォルトでは、EDBLanguagepack は C:/edb/languagepack/v1 にインストールされます。

StackBuilderPlus を介して PEM サーバーをアップグレードする場合、以下に示すエラーが発生する可能性があります。エラーを表示した後、PEM はインストールが完了したことを通知します。インストールは完了していないため、ダウンロードした場所からインストーラーファイルを呼び出してインストールする必要があることに注意してください。

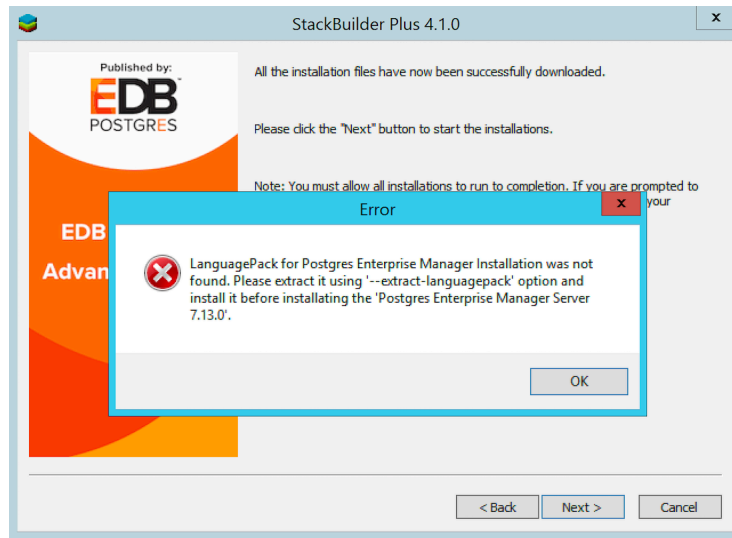


図 19 PEM サーバーのアップグレードエラー

PEM サーバーをアップグレードした後、バックアップデータベースをより新しいバージョンにアップグレードできます。バックアップデータベースのアップグレードについては、:ref:`バックアップ Postgres データベースのアップグレード<upgrading\_backing\_database>`を参照してください。

1.1.3 Windows ホストでの SQL プロファイラーのアップグレード

Windows ホストで SQL プロファイラーを使用している場合、Windows は実行またはメモリにロードされたファイルをロックします。ロックされたファイルを解放するには、アップグレードを実行する前に Postgres サーバーを停止する必要があります。

Windows では、Services ダイアログを使用してサービスを制御できます。Services ダイアログを開くには、Control Panel から System and Security menu までナビゲートします。Administrative Tools を選択し、Services アイコンをダブルクリックします。Services ダイアログが開いたら、リスト内のサービス名を強調表示し、ダイアログに表示されるオプションを使用して Stop サービスを選択します。

Postgres サーバーを停止した後:

1. `uninstall-sql-profiler.sql` スクリプトを呼び出して、各ノードの既存の SQL プロファイラークエリセットを削除します。デフォルトでは、Windows ホストでは、スクリプトは AdvancedServer または PostgreSQL インストールの下の `share/contrib` ディレクトリにあります。

次のサーバー固有のコマンドを使用できます。PostgreSQL の場合:

```
psql -f C:\Program Files\PostgreSQL\<x>\share\contrib\uninstall-sql-profiler.sql -  
-d postgres -U postgres
```

ここで、`x` は PostgreSQL のバージョンであり、`-d` はメンテナンスデータベースの名前を指定します。

AdvancedServer の場合:

```
psql -f C:\Program Files\edb\as<x>\share\contrib\uninstall-sql-profiler.sql -d  
-edb -U enterprisedb
```

ここで、`x` は AdvancedServer のバージョンであり、`-d` はメンテナンスデータベースの名前を指定します。

2. プロファイリングする各ノードで新しい SQLProfiler インストーラーを呼び出します。インストーラーを Administrator として実行します。

PostgreSQL の場合:

```
sqlprofiler-pg-<x>-<y>-windows-x64.exe
```

ここで、`x` は PostgreSQL のバージョンであり、`y` は SQL プロファイラーのバージョンです。
例: `sqlprofiler-pg-12-7.14.0-1-windows-x64.exe`。

AdvancedServer の場合:

```
sqlprofiler-edb-as<x>-<y>-windows-x64.exe
```

ここで、`x` は AdvancedServer のバージョン、`y` は SQL プロファイラーのバージョンです。
例: `sqlprofiler-edb-as12-7.14.0-1-windows-x64.exe`。

SQLProfiler インストーラーは、既存の SQL Profiler インストールを検出し、最新バージョンの SQLProfiler でアップグレードします。

3. メンテナンスデータベースで `sql-profiler.sql` スクリプトファイルを実行します。

PostgreSQL の場合:

```
psql -f C:\Program Files\PostgreSQL\<x>\share\contrib\sql-profiler.sql -d postgres  
→postgres -U postgres
```

ここで、`x` は PostgreSQL のバージョンであり、`-d` はメンテナンスデータベースの名前を指定します。

AdvancedServer の場合:

```
psql -f C:\Program Files\edb\as<x>\share\contrib\sql-profiler.sql -d edb -U enterprisedb  
→enterprisedb
```

ここで、`x` は AdvancedServer のバージョンであり、`-d` はメンテナンスデータベースの名前を指定します。

4. 次に、Postgres サーバーを再起動して、PEM クライアントからノードのプロファイリングを再開します。

PEM コンポーネントを更新したら、バックアップデータベースを更新できます。

1.2 Linux ホスト上の RPM パッケージを使用した PEM インストールのアップグレード

以下のセクションでは、Linux ホストでのアップグレードプロセスを順を追って説明します。

前提条件:

If you are using a version of Postgres or Advanced Server prior to version 10, before the upgrade you must install the `libs` package for version 10 or above on the system where the PEM server is installed. Use the following commands to install the `libs` version 10 or above:

- AdvancedServer の場合:

```
yum install edb-as<version>-server-libs
```

- Postgres の場合:

```
yum install postgresql<version>-libs
```

ここで、`version` は、`libs` パッケージをインストールする Postgres または AdvancedServer のバージョンです。

1.2.1 RPM パッケージとともにインストールされた PEM エージェントのアップグレード

RPM パッケージを使用して、パッケージとともに最初にインストールされた既存のエージェントをアップグレードできます。アップグレードプロセスでは、PEM エージェントの構成ファイルは更新されません。新しいエージェントをインストールした後、既存のエージェントの構成ファイルを新しいインストール場所に手動でコピーする必要があります。

RPM パッケージを使用して PEM エージェントをアップグレードするには、次のコマンドを使用します。

```
yum upgrade edb-pem-agent
```

PEM サーバーに登録されている CentOS6.x 上の PEM エージェントの下位バージョンが 7.10 にアップグレードされると、次の警告が表示される場合があります。

```
warning:%postun(edb-pem-agent-7.9.0-3.rhel6.x86_64) scriptlet failed, exit
status 1
```

```
Non-fatal POSTUN scriptlet failure in rpm package edb-pem-agent
```

上記の警告にもかかわらず、アップグレードは正常に完了し、アップグレード後にエージェントが自動的に再起動されます。

注釈: 以前のバージョンから 7.11 以降のバージョンにアップグレードされた Linux 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既に構成または構成する予定がある場合、`agent.cfg` で `batch_script_user` パラメーターのユーザーを指定する必要があります。root 以外のユーザー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。root ユーザーを使用すると、データのセキュリティとオペレーティングシステムのセキュリティが低下する可能性があります。ただし、root ユーザーを使用してスクリプトを実行して pemagent を元の設定に復元する場合は、`batch_script_user` パラメーター値を root に設定する必要があります。

1.2.2 RPM パッケージとともにインストールされた PEM サーバーのアップグレード

RPM パッケージで PEM サーバーをインストールした場合、RPM パッケージを使用して PEM サーバーをアップグレードできます。

分離されたネットワークのマシンにインストールされている PEM サーバーをアップグレードする場合は、PEM サーバーをアップグレードする前に、そのマシンに PEM リポジトリを作成する必要があります。隔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、[:ref:隔離されたネットワークでの PEM リポジトリの作成<creating_pem_repository_in_isolated_network>](#)を参照してください。

RPM パッケージを使用して既存の RPM インストールをアップグレードするには、以下を行う必要があります。

1. パッケージマネージャーを使用して、インストールされている PEM サーバーのバージョンをアップグレードします。

```
yum upgrade edb-pem
```

2. アップグレードされたサーバーを構成します。

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

構成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入力を求めます。受け入れられるオプションは次のとおりです。

オプション	説明
-acp	PEM エージェントの証明書パスを定義します。デフォルトは/root/.pem です。
-ci	サーバーの pg_hba.conf ファイルに追加される、エージェントがサーバーに接続する CIDR 形式のネットワークアドレス範囲。たとえば、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ーのインストールディレクトリ。たとえば、AdvancedServer の場合は/usr/edb/as10、PostgreSQL の場合は/usr/pgsql-10 です。
-ds	PEM データベースサーバーのユニットファイル名。AdvancedServer の場合、デフォルトのファイル名は edb-as-10 です。PostgreSQL の場合、postgresql-10 です。
-ho	PEM データベースサーバーのホストアドレス。
-p	PEM データベースサーバーのポート番号。
-ps	pemagent のサービス名。デフォルト値は pemagent です。
-sp	PEM データベースサーバーのスーパーユーザーパスワード。この値は必須です。
-su	PEM データベースサーバーのスーパーユーザー名。
-t	インストールタイプ:構成が Web サービスとバックアップデータベース用の場合は 1、Web サービスを構成する場合は 2、バックアップ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注意してください。
-un	PEM サーバーをアンインストールします。
-h	ヘルプを表示します。

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。スクリプト関連のヘルプを表示するには、次のコマンドを使用します。

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

RHEL6.x 上の 7.10PEM サーバーを PEM バージョン 7.11 にアップグレードする場合、アップグレード中に次のエラーテキストが表示される場合があります。

```
/var/tmp/rpm-tmp.<xxxxxxx>:line 9:Requires::command not found
```

上記のエラーにもかかわらず、アップグレードは正常に完了します。

PEM サーバー設定ファイルを実行した後、バージョン固有のサービス制御コマンドを使用して、httpd サービスを再起動します。

RPM パッケージを使用して PEM サーバーまたは PEM エージェントをインストールまたは構成する方法の詳細については、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>

1.2.3 Linux ホストでの SQL プロファイラーのアップグレード

Linux ホストにある SQL プロファイラーのインストールをアップグレードするには:

1. `uninstall-sql-profiler.sql` スクリプトを呼び出して、各ノードの既存の SQL プロファイラー クエリセットを削除します。デフォルトでは、Linux ホストでは、スクリプトは AdvancedServer または PostgreSQL インストールの下の `share/contrib` ディレクトリにあります。

次のサーバー固有のコマンドを使用できます。

PostgreSQL の場合:

```
/usr/pgsql-<x>/bin/psql -f /usr/pgsql-<x>/share/contrib/uninstall-sql-profiler.  
→sql -d postgres -U postgres
```

ここで、`x` は PostgreSQL のバージョンであり、`-d` はメンテナンスデータベースの名前を指定します。

AdvancedServer の場合:

```
/usr/edb/as<x>/bin/psql -f /usr/edb/as<x>/share/contrib/uninstall-sql-profiler.  
→sql -d edb -U enterprisedb
```

ここで、`x` は AdvancedServer のバージョンであり、`-d` はメンテナンスデータベースの名前を指定します。

2. プロファイリングする各ノードで新しい SQLProfiler インストーラーを呼び出します。

PostgreSQL の場合:

```
yum install postgresql<x>-sqlprofiler
```

ここで、`x` は PostgreSQL のバージョンです。

AdvancedServer の場合:

```
yum install edb-as<x>-server-sqlprofiler
```

ここで、`x` は AdvancedServer のバージョンです。

インストーラーは既存の SQL Profiler インストールを検出し、SQL プロファイラーの最新バージョンでアップグレードします。

PostgreSQL 用の SQLProfiler のアップグレードの次の例を参照してください。

```
[root@localhost Downloads]# yum install postgresql12-sqlprofiler
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirrors.piconets.webwerks.in
 * epel: hkg.mirror.rackspace.com
 * extras: mirrors.piconets.webwerks.in
 * updates: mirrors.piconets.webwerks.in
Resolving Dependencies
--> Running transaction check
--> Package postgresql12-sqlprofiler.x86_64 0:7.12.0-1.rhel7 will be updated
--> Package postgresql12-sqlprofiler.x86_64 0:7.14.0-1.rhel7 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                               Arch              Version            Repository          Size
=====
Updating:
postgresql12-sqlprofiler              x86_64            7.14.0-1.rhel7     edb                 88 k
=====
Transaction Summary
=====
Upgrade 1 Package

Total download size: 88 k
Is this ok [y/d/N]: y
Downloading packages:
No Presto metadata available for edb
postgresql12-sqlprofiler-7.14.0-1.rhel7.x86_64.rpm      | 88 kB  00:00:04
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Updating : postgresql12-sqlprofiler-7.14.0-1.rhel7.x86_64      1/2
  Cleanup  : postgresql12-sqlprofiler-7.12.0-1.rhel7.x86_64      2/2
  Verifying : postgresql12-sqlprofiler-7.14.0-1.rhel7.x86_64      1/2
  Verifying : postgresql12-sqlprofiler-7.12.0-1.rhel7.x86_64      2/2
Updated:
  postgresql12-sqlprofiler.x86_64 0:7.14.0-1.rhel7
Complete!
```

図 20 SQL プロファイラーのアップグレード

3. 次に、メンテナンスデータベースで `sql-profiler.sql` スクリプトファイルを実行します。

PostgreSQL の場合:

```
/usr/pgsql-<x>/bin/psql -f /usr/pgsql-<x>/share/contrib/sql-profiler.sql -d_
↪postgres -U postgres
```

ここで、`x` は PostgreSQL のバージョンであり、`-d` はメンテナンスデータベースの名前を指定します。

AdvancedServer の場合:

```
/usr/edb/as<x>/bin/psql -f /usr/edb/as<x>/share/contrib/sql-profiler.sql -d edb -
↪U enterprisedb
```

ここで、`x` は AdvancedServer のバージョンであり、`-d` はメンテナンスデータベースの名前を指定します。

4. PostgreSQL/AdvancedServer を再起動して、PEM クライアントからノードのプロファイリングを再開します。

PEM コンポーネントを更新したら、バックアップデータベースを更新できます。

1.3 隔離されたネットワークでの PEM リポジトリの作成

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、PEMRPM パッケージ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、必要な手順の概要であることに注意してください。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*
```

ここで、<pem_dir> と <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。

1. ディレクトリ /<pem_dir> および /<epel_dir> を隔離されたネットワーク内のマシンにコピーします。
2. リポジトリを作成します。

```
createrepo /<pem_dir>

createrepo /<epel_dir>
```

1. 以下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
```

1. 以下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
```

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できます。

- 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、<https://wiki.centos.org/HowTos/CreateLocalRepos> にアクセスしてください。

1.4 Linux ホストでのグラフィカル PEM インストールのアップグレード

次のセクションでは、Linux ホストのグラフィカルインストーラーを介して実行された PEM インストールのアップグレードプロセスについて説明します。

1.4.1 グラフィカルインストーラーでインストールされた PEM サーバーのアップグレード

前提条件:

If you are using a version of Postgres or Advanced Server released prior to version 10, then before the upgrade you need to install the `libs` package for version 10 or above on the system where the PEM server is installed. Use the following commands to install `libs` version 10 or above:

- AdvancedServer の場合:

```
yum install edb-as<version>-server-libs
```

- Postgres の場合:

```
yum install postgresql<version>-libs
```

ここで、`version` は、`libs` パッケージをインストールする Postgres または AdvancedServer のバージョンです。

グラフィカルインストーラーによってインストールされたときの PEM サーバーのデフォルトのインストール場所は `/opt/edb/pem` です。次の例では、サーバーのインストール場所を `<PEM_installation_path>` に置き換えます。

1. PEM からログアウトします。

2. PEM サーバーで PEMHTTPD サービスを停止します。PEM サーバーと Web サーバーが 2 つの異なるシステム上にある場合、Web サーバーでコマンドを実行します。

```
systemctl stop PEMHTTPD
```

3. 次のコマンドのいずれかを実行して、epel-release パッケージをインストールします。

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

注釈: 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かもしれません。

4. EnterpriseDB リポジトリへのアクセスを許可する資格情報も必要です。資格情報のリクエストについては、次をご覧ください。

<https://info.enterprisedb.com/rs/069-ALB-339/./images/Repository%20Access%2004-09-2019.pdf>

5. リポジトリの資格情報を受け取った後、次のことができます。

1. リポジトリ構成ファイルを作成します。
2. ユーザー名とパスワードを指定して、ファイルを変更します。

リポジトリ設定ファイルの作成

リポジトリ構成ファイルを作成するには、スーパーユーザー特権を引き受けて、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
```

リポジトリ設定ファイルの名前は edb.repo です。ファイルは /etc/yum.repos.d にあります。

ファイルの変更、ユーザー名とパスワードの入力

edb.repo ファイルを作成した後、有効なパラメーターの値が 1 であり、baseurl 仕様の username および password プレースホルダーが置き換えられていることを確認します登録済みの EnterpriseDB ユーザーの名前とパスワードを使用して、次のコマンドを実行します。

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

6. yum makecache コマンドを使用して、現在有効になっているリポジトリのメタデータをダウンロードします。コマンドが完了したら、利用可能なパッケージをチェックして、リストに最新の PEM サーバーが含まれていることを確認します。

```
yum makecache

yum list edb-pem
```

7. PEM サーバーに PEM サーバー RPM をインストールします。PEM サーバーと Web サーバーが 2 つの異なるシステムにある場合、PEM サーバーと Web サーバーでコマンドを実行します。インストールが完了したら、yum info コマンドを使用してインストールの詳細を確認します。

```
yum install edb-pem

yum info edb-pem
```

8. インストール後、agent.cfg ファイルを現在の場所（グラフィカルインストーラーに必要な場所）から RPM パッケージに必要な場所にコピーします。

```
cp /<PEM_installation_path>/agent/etc/agent.cfg /usr/edb/pem/agent/etc/agent.cfg
```

9. 次の場所にあるエージェント構成ファイルを開きます。

```
/usr/edb/pem/agent/etc/agent.cfg
```

次に、ca_file パラメータの値を設定します:

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

10. pem.db ファイル（およびその他の必要なファイル）を RPM インストール場所にコピーし、ファイルの所有権を変更します。PEM サーバーと Web サーバーが 2 つの異なるシステム上にある場合、Web サーバーで以下のコマンドを実行します。

```
cp -r /<PEM_installation_path>/server/share/pemhome/.pem/* /var/lib/pemhome/.pem/

chown -R pem:pem /var/lib/pemhome/.pem/
```

11. グラフィカルインストーラーによって識別された場所から RPM の場所に passwd ファイルのホームディレクトリを変更します。PEM サーバーと Web サーバーが 2 つの異なるシステムにある場合、PEM サーバーと Web サーバーでコマンドを実行します。

```
usermod -m -d /var/lib/pemhome pem

cat /etc/passwd | grep pem
```

12. PEM サービスファイルとエージェント証明書のバックアップを作成します。

```
cp /usr/lib/systemd/system/pemagent.service /usr/lib/systemd/system/pemagent.service_
↪bkp

mv /root/.pem/agent1.key /root/.pem/agent1.key.bkp

mv /root/.pem/agent1.crt /root/.pem/agent1.crt.bkp
```

13. PEM サーバーおよび Web サーバマシンからグラフィカルアンインストーラーを使用して PEM サーバーをアンインストールします。

```
/<PEM_installation_path> /server/uninstall-pemserver
```

14. PEM サーバーと Web サーバーで PEMRPM 構成スクリプトを実行します。プロンプトが表示されたら、バックアップデータベースの詳細を指定します。エラーを生成せずにスクリプトを実行する必要があります。

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

15. サービスファイルのバックアップとエージェント証明書を PEM サーバーの元の場所に復元します。

```
cp /usr/lib/systemd/system/pemagent.service_bkp /usr/lib/systemd/system/pemagent.
↪service

mv /root/.pem/agent1.crt.bkp /root/.pem/agent1.crt

mv /root/.pem/agent1.key.bkp /root/.pem/agent1.key
```

16. PEM サーバーと Web サーバーで pemagent サービスを有効にします。Web サーバーで pemagent および httpd サービスを開始します。

```
systemctl enable pemagent

systemctl start pemagent

systemctl start httpd
```

17. PEMWeb インターフェイスを起動します。サーバーとエージェントをチェックして、PEM バージョン、サーバーステータス、およびスキーマバージョンを確認します。この時点で、すべてが稼働しているはずで、使用されなくなった PEMHTTD サービスを Web サーバーからアンインストールできるようになりました。

1.4.2 グラフィカルインストーラーでインストールされた PEM エージェントのアップグレード

グラフィカルインストーラーによってインストールされたときの PEM サーバーのデフォルトのインストール場所は `/opt/edb/pem` です。次の例では、サーバーのインストール場所を `<PEM_installation_path>` に置き換えます。

1. バージョン固有のコマンドを使用して、`pemagent` サービスを停止します。

- RHEL または CentOS7.x0r8.x ホストの場合:

```
systemctl stop pemagent
```

- RHEL または CentOS6.x ホストの場合:

```
/etc/init.d/pemagent stop (RHEL -6)
```

2. サポートする `epel` パッケージをインストールします。

```
yum install epel-release
```

3. EDB リポジトリ構成ファイルをインストールします。

```
yum install http://yum.enterprisedb.com/edbrepos/edb-repo-latest.noarch.rpm
```

4. リポジトリ構成ファイルのインストールが完了したら、`edb` リポジトリ定義を変更し、リポジトリが使用可能になっていることを確認し、EnterpriseDB 資格情報を提供します。ファイルの場所は次のとおりです。

```
/etc/yum.repos.d/edb.repo
```

5. The `yum makecache` command downloads the metadata for the currently enabled repositories; when the command completes, check the available packages to confirm that the list includes the latest PEM agent:

```
yum makecache  
  
yum list edb-pem-agent
```

6. PEM エージェント RPM をインストールします。インストールが完了したら、`yum info` コマンドを使用して PEM エージェントのインストール情報を確認できます。

```
yum install edb-pem-agent  
  
yum info edb-pem-agent
```

7. インストール後、PEM エージェント設定ファイル（agent.cfg）を以前の場所から RPM インストーラーが必要とする場所にコピーします。

```
cp /PEM_installation_path/agent/etc/agent.cfg /usr/edb/pem/agent/etc/agent.cfg
```

8. 次の場所にあるエージェント構成ファイルを開きます。

```
/usr/edb/pem/agent/etc/agent.cfg
```

次に、ca_file パラメータの値を設定します:

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

9. サービスファイルとエージェント証明書のバックアップを作成します。

- RHEL または CentOS7.x または 8.x では、次のコマンドを使用してサービスファイルをバックアップします。

```
cp /usr/lib/systemd/system/pemagent.service /usr/lib/systemd/system/pemagent.  
↪service_bkp
```

- RHEL または CentOS6.x では、次のコマンドを使用してサービスファイルをバックアップします。

```
cp /etc/init.d/pemagent_bkp /etc/init.d/pemagent
```

次に、エージェント証明書をコピーします。次のコマンドでは、<agent_id> はエージェント識別子（agent2 または agent3 など）を指定する必要があります。

```
mv /root/.pem/<agent_id>.key /root/.pem/<agent_id>.key.bkp  
  
mv /root/.pem/<agent_id>.crt /root/.pem/<agent_id>.crt.bkp
```

10. bitrockuninstaller を使用して PEM エージェントをアンインストールします。

```
/PEM_installation_path/agent/uninstall-pemagent
```

11. バージョン固有のコマンドを使用して、サービスファイルのバックアップとエージェント証明書を元の場所に復元します。例:

- RHEL または CentOS7.x または 8.x ホストの場合:

```
cp /usr/lib/systemd/system/pemagent.service_bkp /usr/lib/systemd/system/  
↪pemagent.service
```

- RHEL または CentOS6.x ホストの場合:

```
cp /etc/init.d/pemagent /etc/init.d/pemagent_bkp
```

次に、エージェント証明書ファイルを移動します。次のコマンドでは、<agent_id> はエージェント識別子を指定する必要があります (agent2 または agent3) :

```
mv /root/.pem/<agent_id>.key.bkp /root/.pem/<agent_id>.key  
mv /root/.pem/<agent_id>.cert.bkp /root/.pem/<agent_id>.cert
```

12. pemagent サービスを有効にして、pemagent と httpd を開始します。

- RHEL または CentOS7.x または 8.x ホストでは、次のコマンドを使用します。

```
systemctl enable pemagent  
  
systemctl start pemagent
```

- RHEL または CentOS6.x ホストの場合:

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

この時点で、PEM エージェントが稼働しているはずです。PEMWeb インターフェイスを使用して、エージェントのバージョンとステータスを確認できます。

注釈: バージョン 7.11 より前のバージョンからアップグレードされた Linux 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既に構成している場合、または構成しようとしている場合は、agent.cfg ファイルを変更し、batch_script_user のユーザーを指定する必要があります “パラメータ。root 以外のユーザー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。スクリプトの実行時に root ユーザーを使用すると、データのセキュリティとオペレーティングシステムのセキュリティが低下する可能性があります。ただし、PEM エージェントの以前の動作を復元し、root ユーザーを使用してスクリプトを実行する場合は、batch_script_user パラメーターの値を root に設定する必要があります。

1.4.3 Linux ホストでの SQL プロファイラーのアップグレード

注釈: SQL プロファイラーのグラフィカルインストーラーは、11.x より前の PostgreSQL および AdvancedServer バージョン (10.x、9.6、9.5、..) で使用できます。

Linux ホスト上の SQLProfiler インストールをアップグレードするには:

1. `uninstall-sql-profiler.sql` スクリプトを呼び出して、各ノードの既存の SQL プロファイラークエリセットを削除します。デフォルトでは、グラフィカルインストーラーでインストールされた Linux ホストで PostgreSQL を使用している場合、スクリプトは PostgreSQL インストールの下の `share/postgresql/contrib` ディレクトリにあります。AdvancedServer の場合、スクリプトは AdvancedServer インストールの `share/contrib` ディレクトリにあります。次のサーバー固有のコマンドを使用できます。

PostgreSQL の場合:

```
/opt/PostgreSQL/<x>/bin/psql -f /opt/PostgreSQL/<x>/share/postgresql/contrib/  
↪uninstall-sql-profiler.sql -d postgres -U postgres
```

ここで、`x` は PostgreSQL のバージョンであり、`-d` はメンテナンスデータベースの名前を指定します。

AdvancedServer の場合:

```
/opt/edb/as<x>/bin/psql -f /opt/edb/as<x>/share/contrib/uninstall-sql-  
↪profiler.sql -d edb -U enterprisedb
```

ここで、`x` は AdvancedServer のバージョンであり、`-d` はメンテナンスデータベースの名前を指定します。

2. 次に、プロファイリングする各ノードで新しい SQLProfiler インストーラーを呼び出します。管理者としてインストーラーを実行します。

PostgreSQL ホストの場合:

```
sqlprofiler-pg-<x>-<y>-linux.exe
```

ここで、`x` は PostgreSQL のバージョン、`y` は SQL プロファイラーのバージョンです。
例: `sqlprofiler-pg-12-7.14.0-1-linux.exe`。

AdvancedServer の場合:

```
sqlprofiler-edb-as<x>-<y>-linux.exe
```

ここで、`x` は AdvancedServer のバージョン、`y` は SQL プロファイラーのバージョンです。
例: `sqlprofiler-edb-as12-7.14.0-1-linux.exe`。

SQLProfiler インストーラーは、既存の SQL Profiler インストールを検出し、最新バージョンの SQLProfiler でアップグレードします。

3. メンテナンスデータベースで `sql-profiler.sql` スクリプトファイルを実行します。

PostgreSQL の場合:

```
/opt/PostgreSQL/<x>/bin/psql -f /opt/PostgreSQL/<x>/share/postgresql/contrib/sql-  
→profiler.sql -d postgres -U postgres
```

ここで、`x` は PostgreSQL のバージョンであり、`-d` はメンテナンスデータベースの名前を指定します。

AdvancedServer の場合:

```
/opt/edb/as<x>/bin/psql -f /opt/edb/as<x>/share/contrib/sql-profiler.sql -d edb -  
→U enterprisedb
```

ここで、`x` は AdvancedServer のバージョンであり、`-d` はメンテナンスデータベースの名前を指定します。

4. PostgreSQL/AdvancedServer を再起動して、PEM クライアントからノードのプロファイリングを再開します。

PEM コンポーネントを更新したら、バックアップデータベースを更新できます。

BackingPostgres データベースのアップグレード

PEM コンポーネントと PEM バッキングデータベースの両方を更新する場合は、バッキングデータベースを更新する前に PEM コンポーネントの更新（サーバー、エージェント、およびクライアント）を実行する必要があります。PEM コンポーネントソフトウェアの更新の詳細については、:ref:PEM インストールのアップグレード<upgrading_pem_installation>を参照してください。

The update process described in this section uses the `pg_upgrade` utility to migrate from one version of the backing server to a more recent version. `pg_upgrade` facilitates migration between any version of Postgres (version 9.3 or later), and any subsequent release of Postgres that is supported on the same platform.

`pg_upgrade` supports a transfer of data between servers of the same type. For example, you can use `pg_upgrade` to move data from a PostgreSQL 9.6 backing database to a PostgreSQL 10 backing database, but not to an Advanced Server 10 backing database. If you wish to migrate to a different type of backing database (i.e from a PostgreSQL server to Advanced Server), see *Moving the Postgres Enterprise Manager™ Server*.

`pg_upgrade` の使用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/static/pgupgrade.html>

ステップ 1-更新されたバッキングデータベースインストーラーのダウンロードと呼び出し

PostgreSQL および AdvancedServer のインストーラーは、EnterpriseDBWeb サイトから入手できます。

<https://www.enterprisedb.com/software-downloads-postgres>

アップグレードするサーバーバージョンのインストーラーをダウンロードしたら、PEM サーバーのホストでインストーラーを起動します。インストールウィザードの画面上の指示に従って、Postgres サーバーを構成およびインストールします。

オプションで、カスタム構築の PostgreSQL サーバーを PEM バックアップデータベースのホストとして使用できます。ポート 5432 をリッスンする PostgreSQL バックアップデータベースからアップグレードする場合、新しいサーバーは別のポートでリッスンするように設定する必要があることに注意してください。

ステップ 2-新しいサーバーで SSL ユーティリティを構成する

新しいバックアップデータベースは、現在のバックアップデータベースが実行されているのと同じバージョンの `sslutils` を実行している必要があります。EnterpriseDB インストーラーで使用される `SSLUtils` パッケージは、次の場所からダウンロードできます。

<https://www.enterprisedb.com/downloads/modified-gpl-source-code>

You are *not* required to manually add the `sslutils` extension when using the Advanced Server as the new backing database. The process of configuring `sslutils` is platform-specific.

Linux で

Linux を使用している場合、アーカイブされた `SSLUtils` ファイルのバージョンを次からダウンロードできます。

<https://www.enterprisedb.com/downloads/modified-gpl-source-code>

ダウンロードが完了したら、`sslutils` フォルダを抽出し、アップグレード先の Postgres バージョンの Postgres インストールディレクトリに移動します。

コマンドラインを開き、スーパーユーザー権限を取得し、`PATH` 環境変数の値を設定して、`make` が `pg_config` プログラムを見つけられるようにします。

```
export PATH=$PATH:/usr/pgsql/<X>/bin/
```

どこで:

`X` は、移行先の Postgres のバージョンを指定します。

次に、`yum` を使用して `sslutil` の依存関係をインストールします。

```
yum install openssl-devel
```

Navigate into the `sslutils` folder, and build the `sslutils` package by entering:

```
make USE_PGXS=1  
  
make USE_PGXS=1 install
```

Windows で

`sslutils` must be compiled on the new backing database with the same compiler that was used to compile `sslutils` on the original backing database. If you are moving to a Postgres database that was installed using a PostgreSQL one-click installer (from EnterpriseDB) or an Advanced Server installer, use Visual Studio to build `sslutils`. If you are upgrading to PostgreSQL 9.5 or later, use Visual Studio 2010.

Windows で Postgres の特定のバージョンを構築する詳細については、そのバージョンのコアドキュメントを参照してください。コアドキュメントは、PostgreSQL プロジェクトの Web サイトで入手できます。

<http://www.postgresql.org/docs/>

または、EnterpriseDB Web サイト:

<https://www.enterprisedb.com/edb-docs>

プロセスの具体的な詳細はプラットフォームとコンパイラによって異なりますが、各プラットフォームの基本的な手順は同じです。次の例は、32 ビット Windows システムでの PostgreSQL の OpenSSL サポートのコンパイルを示しています。

OpenSSL 拡張機能をコンパイルする前に、ご使用の Windows バージョンの OpenSSL を見つけてインストールする必要があります。OpenSSL インストーラーを呼び出す前に、前提条件の再配布可能ファイル (vcredist_x86.exe など) をダウンロードしてインストールする必要がある場合があります。

OpenSSL をインストールした後、次から入手できる sslutils ユーティリティパッケージをダウンロードして解凍します。

<https://www.enterprisedb.com/downloads/modified-gpl-source-code>

解凍した sslutils フォルダを Postgres インストールディレクトリ (つまり C:/ProgramFiles/PostgreSQL/<x.x>) にコピーします

Open the Visual Studio command line, and navigate into the sslutils directory. Use the following commands to build sslutils:

```
SET USE_PGXS=1

SET GETTEXT_PATH=\ <path_to_gettext>

SET OPENSSL_PATH=\ <path_to_openssl>

SET PG_PATH=\ <path_to_pg_installation_dir>

SET ARCH=x86

msbuild sslutils.proj /p:Configuration=Release
```

どこで:

- path_to_gettext は GETTEXT ライブラリとヘッダーファイルの場所を指定します。
- path_to_openssl は openssl ライブラリとヘッダーファイルの場所を指定します。
- path_to_pg_installation_dir は Postgres インストールの場所を指定します。

たとえば、次のコマンドセットは、OpenSSL サポートを PostgreSQL10 サーバーに組み込みます。

```
SET USE_PGXS=1

SET OPENSSELPATH=C:\OpenSSL-Win32

SET GETTEXT_PATH="C:\Program Files\PostgreSQL\10"

SET PGPATH="C:\Program Files\PostgreSQL\10"

SET ARCH=x86

msbuild sslutils.proj /p:Configuration=Release
```

ビルドが完了すると、sslutils ディレクトリに次のファイルが含まれます。

- sslutils--1.1.sql
- sslutils--unpacked--1.1.sql
- sslutils--pemagent.sql.in
- sslutils.dll

コンパイル済みの sslutils ファイルをインストールに適したディレクトリにコピーします。例えば:

```
COPY sslutils*.sql "%PGPATH%\share\extension\"

COPY sslutils.dll "%PGPATH%\lib\"
```

ステップ 3-サービスの停止

古いバックアップデータベースと新しいバックアップデータベースの両方のサービスを停止します。

-RHEL または CentOS6.x で、コマンドラインを開き、スーパーユーザーの ID を引き継ぎます。次のコマンドを入力します。

```
/etc/init.d/<service_name> stop
```

-RHEL または CentOS7.x または 8.x で、コマンドラインを開き、スーパーユーザーの ID を引き継ぎます。次のコマンドを入力します。

```
systemctl <service_name> stop
```

ここで、service_name は Postgres サービスの名前を指定します。

On Windows, you can use the Services dialog to control the service. To open the Services dialog, navigate through the Control Panel to the System and Security menu. Select Administrative Tools, and then double-click the Services icon. When the Services dialog opens, highlight the service name in the list, and use the option provided on the dialog to Stop the service.

ステップ 4-pg_upgrade を使用してサーバーを更新

You can use the `pg_upgrade` utility to perform an in-place transfer of existing data between the old backing database and the new backing database. If your server is configured to enforce `md5` authentication, you may need to add an entry to the `.pgpass` file that specifies the connection properties (and password) for the database superuser, or modify the `pg_hba.conf` file to allow trust connections before invoking `pg_upgrade`. For more information about creating an entry in the `.pgpass` file, please see the PostgreSQL core documentation, available at:

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

During the upgrade process, `pg_upgrade` will write a series of log files. The cluster owner must invoke `pg_upgrade` from a directory in which they have write privileges. If the upgrade completes successfully, `pg_upgrade` will remove the log files when the upgrade completes. To instruct `pg_upgrade` to not delete the upgrade log files, include the `--retain` keyword when invoking `pg_upgrade`.

`pg_upgrade` を呼び出すには、クラスター所有者の ID を想定し、クラスター所有者が書き込み権限を持っているディレクトリに移動して、コマンドを実行します。

```
<path_to_pg_upgrade> pg_upgrade
-d <old_data_dir_path>
-D <new_data_dir_path>
-b <old_bin_dir_path> -B <new_bin_dir_path>
-p <old_port> -P <new_port>
-u <user_name>
```

どこで:

- `path_to_pg_upgrade` は `pg_upgrade` ユーティリティの場所を指定します。デフォルトでは、`pg_upgrade` は Postgres ディレクトリの下に `bin` ディレクトリにインストールされます。
- `old_data_dir_path` は、古いバックアップデータベースのデータディレクトリへの完全なパスを指定します。
- `new_data_dir_path` は、新しいバックアップデータベースのデータディレクトリへの完全なパスを指定します。
- `old_bin_dir_path` は、古いバックアップデータベースの `bin` ディレクトリへの完全なパスを指定します。
- `new_bin_dir_path` は、古いバックアップデータベースの `bin` ディレクトリへの完全なパスを指定します。
- `old_port` は古いサーバーがリッスンしているポートを指定します。
- `new_port` は、新しいサーバーがリッスンするポートを指定します。

- `user_name` はクラスター所有者の名前を指定します。

たとえば、次のコマンド:

```
C:\>"C:\Program Files\PostgreSQL\10\bin\pg_upgrade.exe"  
  
-d "C:\Program Files\PostgreSQL\9.6\data"  
  
-D "C:\Program Files\PostgreSQL\10\data"  
  
-b "C:\Program Files\PostgreSQL\9.6\bin"  
  
-B "C:\Program Files\PostgreSQL\10\bin"  
  
-p 5432 -P 5433  
  
-u postgres
```

Windows システムで PEM データベースを PostgreSQL9.6 から PostgreSQL10 に移行するように `pg_upgrade` に指示します (バックアップデータベースがデフォルトの場所にインストールされている場合)。

Once invoked, `pg_upgrade` will perform consistency checks before moving the data to the new backing database. When the upgrade is finished, `pg_upgrade` will notify you that the upgrade is complete.

`pg_upgrade` オプションの使用またはアップグレードプロセスのトラブルシューティングの詳細については、以下を参照してください。

<http://www.postgresql.org/docs/current/static/pgupgrade.html>

ステップ 5-古いデータベースから新しいデータベースに証明書ファイルをコピー

Copy the following certificate files from the `data` directory of the old backing database to the `data` directory of the new backing database:

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

ターゲットサーバーに配置したら、ファイルには以下で説明する (プラットフォーム固有の) 権限が必要です。

Linux での許可と所有権

ファイル名	所有者	許可
ca_certificate.crt	postgres	-rw-----
ca_key.key	postgres	-rw-----
root.crt	postgres	-rw-----
root.crl	postgres	-rw-----
server.key	postgres	-rw-----
server.crt	postgres	-rw-r--r--

Linux では、証明書ファイルは postgres によって所有されている必要があります。コマンドラインで次のコマンドを使用して、ファイルの所有権を変更できます。

```
chown postgres <file_name>
```

ここで、file_name は証明書ファイルの名前を指定します。

The server.crt file may only be modified by the owner of the file, but may be read by any user. You can use the following command to set the file permissions for the server.crt file:

```
chmod 644 server.crt
```

他の証明書ファイルは、ファイルの所有者のみが変更または読み取ることができます。次のコマンドを使用して、ファイルのアクセス許可を設定できます。

```
chmod 600 <file_name>
```

ここで、file_name はファイルの名前を指定します。

Windows での権限と所有権

Windows では、ソースホストから移動した証明書ファイルは、ターゲットホストで PEM サーバーとバックギングデータベースのインストールを実行したサービスアカウントが所有している必要があります。Run as Administrator オプション（インストーラーのコンテキストメニューから選択）を使用して PEM サーバーと Postgres インストーラーを呼び出した場合、証明書ファイルの所有者は Administrators になります。

Windows でファイルのアクセス許可を確認および変更するには、ファイル名を右クリックし、Properties を選択します。

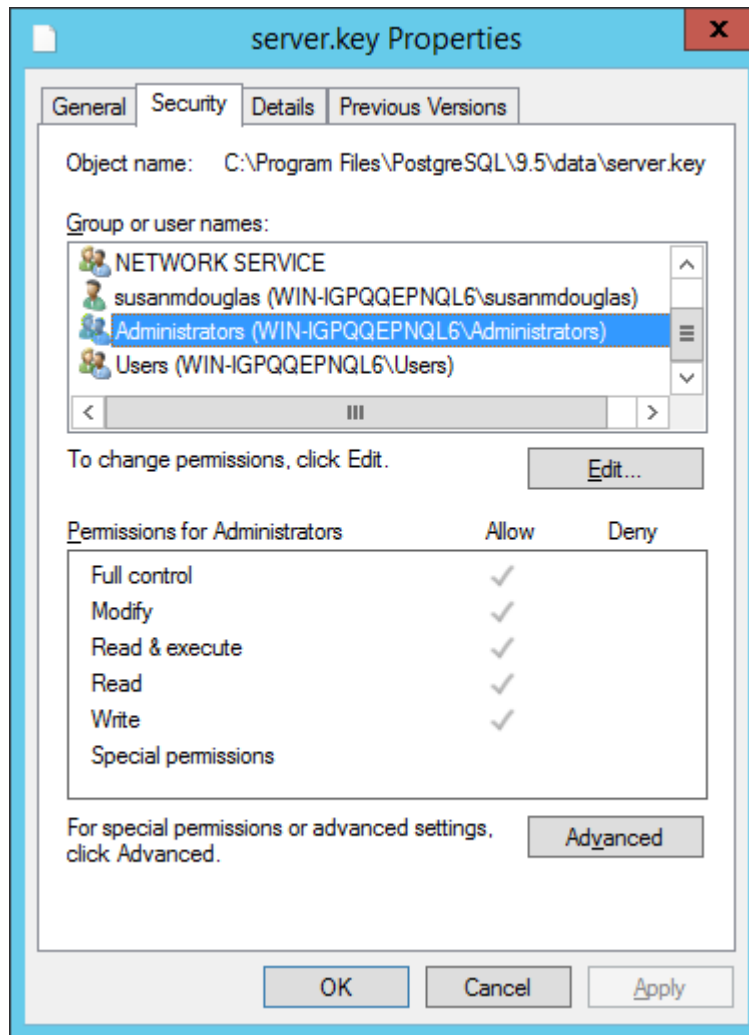


図1 [セキュリティ] タブ。

Security タブに移動し、Group or user name を強調表示して、割り当てられた権限を表示します。Edit または Advanced を選択して、選択したユーザーに関連付けられた権限を変更できるダイアログにアクセスします。

ステップ 6-新しいサーバー構成ファイルの更新

The `postgresql.conf` file contains parameter settings that specify server behavior. You will need to modify the `postgresql.conf` file on the new server to match the configuration specified in the `postgresql.conf` file of the old server.

デフォルトでは、`postgresql.conf` ファイルがあります:

- Linux の Postgres バージョンが 10 未満の場合、`/opt/PostgreSQL/<version.x>/data`
- Linux のグラフィカルインストーラーでインストールされた Postgres バージョン 10 以降の場合、`/opt/PostgreSQL/<version>/data`

- Linux に RPM をインストールした Postgres バージョン 10 以降の場合、`/usr/pgsql/<version>/data`
- Windows の Postgres バージョンの場合、`C:/Program Files/PostgreSQL/<version.x>/data`

ここで、`version` はシステム上の Postgres のメジャーバージョンです。

選択したエディターを使用して、新しいサーバーの `postgresql.conf` ファイルを更新します。以下のパラメーターを変更します。

- 元のバックアップデータベースによって監視されるポートでリッスンする `port` パラメーター（通常は 5432）。
- `ssl` パラメーターは `on` に設定する必要があります。

また、次のパラメーターが有効になっていることを確認する必要があります。パラメーターがコメント化されている場合、各 `postgresql.conf` ファイルエントリの前からポンド記号を削除します。

- `ssl_cert_file = 'server.crt' # (change requires restart)`
- `ssl_key_file = 'server.key' # (change requires restart)`
- `ssl_ca_file = 'root.crt' # (change requires restart)`
- `ssl_crl_file = 'root.crl'`

インストールには、新しいバックアップデータベースが古いバックアップデータベースと同等の方法で動作することを保証するために変更が必要な他のパラメータ設定がある場合があります。 `postgresql.conf` ファイルを注意深く確認して、新しいサーバーの構成が古いサーバーの構成と一致することを確認してください。

ステップ 7-新しいサーバー認証ファイルの更新

The `pg_hba.conf` file contains parameter settings that specify how the server will enforce host-based authentication. When you install the PEM server, the installer modifies the `pg_hba.conf` file, adding entries to the top of the file:

```
# Adding entries for PEM agents and admins to connect to PEM server

hostssl pem +pem_user 192.168.2.0/24 md5

hostssl pem +pem_agent 192.168.2.0/24 cert

# Adding entries (localhost) for PEM agents and admins to connect to PEM
server

hostssl pem +pem_user 127.0.0.1/32 md5

hostssl postgres +pem_user 127.0.0.1/32 md5

hostssl pem +pem_user 127.0.0.1/32 md5

hostssl pem +pem_agent 127.0.0.1/32 cert
```

デフォルトでは、`pg_hba.conf` ファイルは次の場所にあります。

- Linux の Postgres バージョンが 10 未満の場合、`/opt/PostgreSQL/<version>.x/data`
- Linux のグラフィカルインストーラーでインストールされた Postgres バージョン 10 以降の場合、`/Opt/PostgreSQL/<version>/data`
- Linux に RPM をインストールした Postgres バージョン 10 以降の場合、`/var/lib/pgsql/<version>/data`
- Linux 上に RPM をインストールした AdvancedServer バージョン 10 以降の場合、`/var/lib/edb/as<version>/data`
- Windows の Postgres バージョンの場合、`C:/Program Files/PostgreSQL/version.x/data`

ここで、`version` はシステム上の Postgres のメジャーバージョンであり、`*x*`はマイナーバージョンです。

Using your editor of choice, copy the entries from the `pg_hba.conf` file of the old server to the `pg_hba.conf` file for the new server.

ステップ 8-新しい Postgres サーバーを再起動します

新しいバックアップデータベースのサービスを開始します。

-RHEL または CentOS6.x で、コマンドラインを開き、スーパーユーザーの ID を引き継ぎます。次のコマンドを入力します。

```
/etc/init.d/<service_name> start
```

-RHEL または CentOS7.x または 8.x で、コマンドラインを開き、スーパーユーザーの ID を引き継ぎます。次のコマンドを入力します。

```
systemctl stop <service_name>
```

ここで、`service_name` はバックアップデータベースサーバーの名前です。

If you are using Windows, you can use the Services dialog to control the service. To open the Services dialog, navigate through the Control Panel to the System and Security menu. Select Administrative Tools, and then double-click the Services icon. When the Services dialog opens, highlight the service name in the list, and use the option provided on the dialog to start the service.

PostgresEnterpriseManager™ サーバーの移動

このセクションの手順では、**PEM** サーバーをあるホストマシンから新しいホストマシンに移動する方法について説明します。新しいホスト（ターゲット）上の **PEM** サーバーは、元のホスト（ソース）と同じバージョンの **PEM** サーバーインストーラーでインストールする必要があります。同じインストーラーバージョンを使用しない場合、スキーマ不一致エラーが発生する可能性があることに注意してください。

ターゲットサーバー（PostgreSQL または AdvancedServer）のバックアップデータベースは、ソース **PEM** サーバーのバックアップデータベースと同じタイプとバージョン、または異なるタイプとバージョンである場合があります。PostgreSQL ホストにある **PEM** サーバーは、AdvancedServer ホストに移行できます。逆も同様です。

サーバーの移行を開始する前に、ソースホスト、ターゲットホスト、および **PEM** エージェントのホスト間のファイアウォールがサービス間の接続を許可することを確認する必要があります。

ステップ 1-ターゲットホストの準備

ターゲットホストで **PEM** サーバーのインストーラーを起動します。ソース **PEM** サーバーのインストール時に使用したのと同じバージョンの **PEM** サーバーインストーラーを使用する必要があることに注意してください。

The backing database of the target server may be a different version or type than the backing database of the source. If the new PEM server does *not* reside on the same type of backing database as the original server, you must ensure that the same version of the `sslutils` extension is installed on the new server host. The version of `sslutils` that is distributed with the PEM installers is freely available for download from the EnterpriseDB website at:

<https://www.enterprisedb.com/downloads/modified-gpl-source-code>

PEM サーバーまたは `sslutils` 拡張機能のインストールについては、次の場所にある **PEM Installation Guide** を参照してください。

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

ステップ 2-新しい PEM サーバーから既存のスキーマを削除

移行プロセスは、ターゲット PEM サーバー上のソース PEM サーバーから pem、pemdata、および pemhistory スキーマを再作成します。移動の準備として、psql クライアントを使用して、これらのスキーマをターゲットホスト上の pem データベースから削除します。コマンドラインで psql クライアントを開くか、Postgres Enterprise Manager メニューから SQL Shell (psql) を選択します。

psql クライアントが開いたら、データベーススーパーユーザーとして pem バックアップデータベースに接続します。ターゲットホスト上の pem データベースに接続したら、次のコマンドを使用してスキーマを削除します。

```
DROP SCHEMA pem CASCADE;  
  
DROP SCHEMA pemdata CASCADE;  
  
DROP SCHEMA pemhistory CASCADE;
```

スキーマを削除するとき、すべての依存オブジェクトを削除するようにサーバーに指示する CASCADE キーワードを含める必要があります。コマンドを実行すると、psql クライアントは依存オブジェクトのリストを表示します。クライアントは、DROP SCHEMA を表示することにより、各スキーマが削除されることを確認します。

ステップ 3-新しい PEM サーバーで PEM エージェントを準備する

PEM サーバーを移動する前に、ソース PEM サーバーによって監視されているエージェントの数を特定し、ターゲットサーバーでその数のエージェント（1 つ未満）の ID を作成する必要があります。PEM サーバーによって監視されている PEM エージェントの総数を検出するには、psql クライアントを使用してソースホストの pem データベースに接続し、pem.agent テーブルを照会します。

```
SELECT id FROM pem.agent WHERE active = true;
```

元の PEM サーバーに存在するエージェントの数を 1 つ少なくして手動で作成する必要があります。PEM サーバーインストーラーは、ターゲットホスト上に既に 1 つのエージェントを作成しています。たとえば、ソースサーバーに 3 つのエージェントが含まれる場合、2 つの追加エージェントを手動で作成する必要があります。ターゲットサーバー上の pem データベースで psql セッションを開き、必要なエージェントを作成します。次のコマンドを使用します。

```
CREATE USER agent <x>;
```

ここで、<x> はエージェント番号を指定します。PEM サーバーのインストーラーによってターゲットホスト上に agent1 が作成されることを忘れないでください。

次に、GRANT コマンドを使用して、ターゲット PEM サーバーの pem_agent 権限にある各エージェントを割り当てます。

```
GRANT pem_agent TO agent <x>;
```

ここで、<x> はエージェント番号を指定します。

ステップ 4-ソース PEM サーバーのバックアップスクリプトを生成

You can use the `pg_dump` utility to generate a script that contains the commands required to recreate the pem database on the target host. By default, `pg_dump` is installed in the `bin` directory under your Postgres installation. To invoke `pg_dump`, open a command line, navigate to the `bin` directory, and enter:

```
pg_dump -U <user_name> <db_name> > <file_name>
```

どこで:

- <user_name> は、PEM バックアップデータベースのデータベーススーパーユーザーの名前を指定します。
- <db_name> は PEM バックアップデータベースの名前を指定します。
- <file_name> は `pg_dump` によって生成されたスクリプトの名前を指定します。

プロンプトが表示されたら、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。

表示されるコマンドは、`pg_dump` に指示して、(実行時に) `pem` データベースを再作成するスクリプトを生成します。スクリプトの名前は `backup.sql` で、`tmp` ディレクトリに作成されます。`pg_dump` は、ユーザーの資格情報 `postgres` を使用してサーバーに接続しています。

`pg_dump` ユーティリティを呼び出しても、現在のデータベースユーザーは中断されません。

ステップ 5-バックアップをターゲットホストに移動

`pg_dump` ユーティリティによって生成されたスクリプトを PEM サーバーのターゲットホストに移動します。

ステップ 6-ターゲットホストでバックアップを復元する

ターゲットホストでコマンドラインを開き、`bin` ディレクトリ (Postgres バックアップデータベースのインストールディレクトリの下) に移動します。`psql` を起動し、`pg_dump` ユーティリティによって生成されたスクリプトを実行します:

```
psql -U <user_name> -d pem -f <file_name>
```

どこで:

- <user_name> はデータベースのスーパーユーザーの名前を指定します。指定されたユーザーは、バックアップデータベースに対する接続特権を持っている必要があります。
- <file_name> は、`pg_dump` によって生成されたバックアップスクリプトへの完全なパスを指定します。

プロンプトが表示されたら、データベースのスーパーユーザーに関連付けられたパスワードを入力します。

示されている例は、psql クライアントを使用して backup.sql という名前のスクリプトを呼び出し、pem データベースを再作成します。スクリプトは、データベーススーパーユーザー postgres に関連付けられた特権を使用して呼び出されます。

ステップ 7-ターゲットホストでデータベースサーバーを停止する

Linux で PEM サーバーを停止するには、次のコマンドを使用します。

```
/etc/init.d/<service_name> stop
```

ここで、<service_name> はバックアップデータベースサーバーの名前を指定します。PostgreSQL バックアップデータベースの場合、サービス名は postgresql-<x> であり、AdvancedServer バックアップデータベースの場合、サービス名は edb-as-<x> です。ここで、x はバージョン番号を指定します。

If you are using Windows, you can use the Services dialog to control the service. To open the Services dialog, navigate through the Control Panel to the System and Security menu. Select Administrative Tools, and then double-click the Services icon. When the Services dialog opens, highlight the service name in the list, and use the option provided on the dialog to Stop the service.

ステップ 8-証明書ファイルをターゲットホストにコピー

ターゲットホストのインストール時に作成される証明書ファイルを、ソースホストの証明書ファイルに置き換える必要があります。次のファイルをソース PEM サーバーからターゲット PEM サーバーにコピーします。

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

ターゲットクラスタのバックアップデータベースを提供する Postgres インストールの下の data ディレクトリにファイルをコピーします。Linux では、デフォルトで、ファイルは次の場所にあります。

```
/var/lib/pgsql/<X>/data/
```

Windows では、ファイルは次の場所にあります。

```
C:\Program Files\PostgreSQL\<X>\data
```

どこで:

x はシステム上の PostgreSQL のバージョンを指定します。

ファイルはターゲットクラスタに既に存在します。コピーを実行する前に既存のファイルを削除するか、既存のファイルをソースサーバーのファイルで上書きします。ターゲットサーバーに配置されたら、ファイルには、以降のセクションで説明する（プラットフォーム固有の）権限が必要です。

Linux での許可と所有権

ファイル名	所有者	許可
ca_certificate.crt	postgres	-rw-----
ca_key.key	postgres	-rw-----
root.crt	postgres	-rw-----
root.crl	postgres	-rw-----
server.key	postgres	-rw-----
server.crt	postgres	-rw-r--r--

Linux では、証明書ファイルは postgres が所有している必要があります。コマンドラインで次のコマンドを使用して、ファイルの所有権を変更できます。

```
chown postgres <file_name>
```

ここで、file_name は証明書ファイルの名前を指定します。

server.crt ファイルは、ファイルの所有者のみが変更できますが、すべてのユーザーが読み取ることができます。次のコマンドを使用して、server.crt ファイルのファイル許可を設定できます。

```
chmod 644 server.crt
```

他の証明書ファイルは、ファイルの所有者のみが変更または読み取ることができます。次のコマンドを使用して、ファイルのアクセス許可を設定できます。

```
chmod 600 <file_name>
```

ここで、file_name はファイルの名前を指定します。

Windows での権限と所有権

Windows では、ソースホストから移動した証明書ファイルは、ターゲットホストで PEM サーバーとバックギングデータベースのインストールを実行したサービスアカウントが所有している必要があります。Run as Administrator オプション（インストーラーのコンテキストメニューから選択）を使用して PEM サーバーと Postgres インストーラーを呼び出した場合、証明書ファイルの所有者は Administrators になります。

Windows でファイルのアクセス許可を確認および変更するには、ファイル名を右クリックし、Properties を選択します。

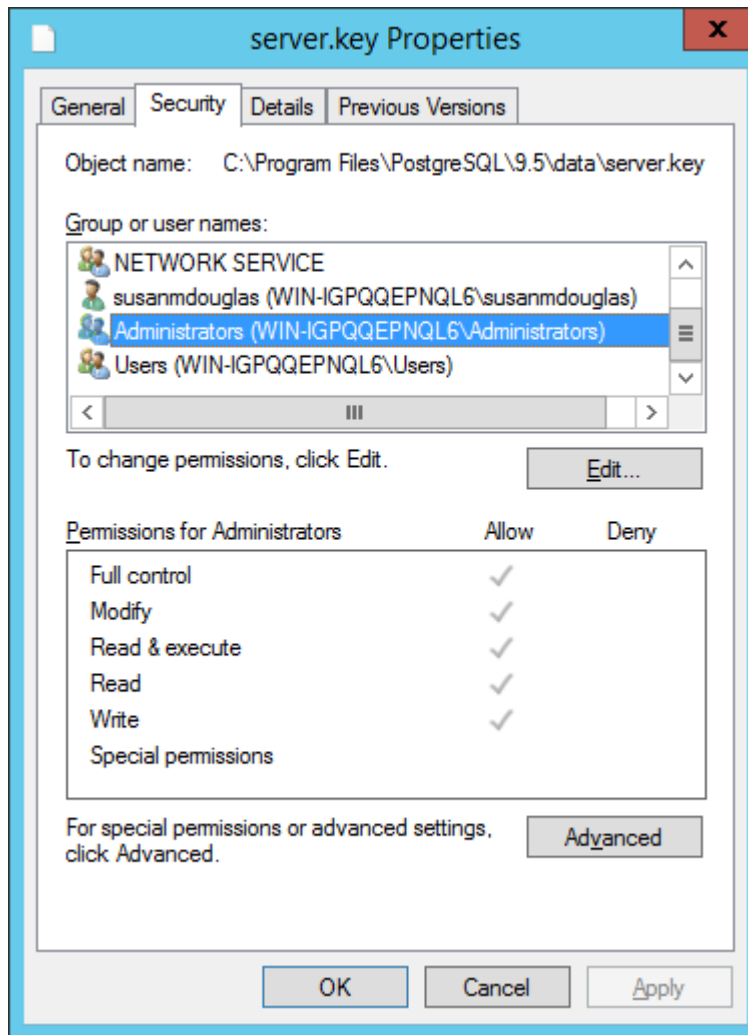


図1 [権限] タブ

Security タブに移動し、Group or user name を強調表示して、割り当てられた権限を表示します。Edit または Advanced を選択して、選択したユーザーに関連付けられた権限を変更できるダイアログにアクセスします。

ステップ 9-PEM エージェント証明書ファイルを PEM サーバーホストに移動

ソース PEM サーバーの PEM エージェントが使用する証明書ファイルをターゲットホストに移動する必要があります。この手順はプラットフォーム固有です。

Linux で

agent1.key および agent1.crt ファイルをソースホストからターゲットホストにコピーします。デフォルトでは、Linux では、ファイルは /root/.pem にインストールされます。ファイルをターゲットホストの同じディレクトリにコピーします。

ファイルのファイル所有権とアクセス許可は、次のように設定する必要があります。

ファイル名	所有者	許可
agent1.key	根	-rw-----
agent1.crt	根	-rw-r--r--

必要に応じて、/root/.pem に移動し、次のコマンドを使用して agent1.key ファイルのアクセス権と所有権を変更します。

```
chmod 600 agent1.key  
  
chown root agent1.key
```

次のコマンドを使用して、agent1.crt ファイルのアクセス権と所有権を変更します。

```
chmod 644 agent1.crt  
  
chown root agent1.crt
```

Windows で

agent1.key および agent1.crt ファイルをソースホストからターゲットホストにコピーします。Windows では、ファイルは次の場所にあります。

```
C:\Users\<user_name>\AppData\Roaming\pem
```

user_nameは、PEM インストーラーを呼び出したユーザーの名前です。

ターゲットマシン上の証明書ファイルに関連付けられている所有権と許可は、ソースマシン上の証明書ファイルの所有権と許可と一致する必要があります。Run as Administrator オプション（インストーラーのコンテキストメニューから選択）を使用して PEM サーバーと Postgres インストーラーを呼び出した場合、エージェント証明書ファイルの所有者は Administrators になります。

Windows でファイルのアクセス許可を確認および変更するには、ファイル名を右クリックし、Properties を選択します。Security タブに移動し、Group or user name を強調表示して、割り当てられた権限を表示します。Edit または Advanced を選択して、選択したユーザーに関連付けられた権限を変更できるダイアログにアクセスします。

ステップ 10-ターゲットホスト上の “pg_hba.conf” ファイルを更新します

Modify the pg_hba.conf file on the target host to allow connections from each PEM agent. By default, the pg_hba.conf file is located in the data directory under your Postgres installation.

ステップ 11-ターゲットホストでサーバーを起動

pg_hba.conf ファイルを変更した後、変更を有効にするにはサーバーを再起動する必要があります。

Linux でデータベースサーバーを再起動するには、次のコマンドを使用します。

```
/etc/init.d/<service_name> start
```

ここで、`service_name` はバックアップデータベースサーバーの名前です。

If you are using Windows, you can use the Services dialog to control the service. To open the Services dialog, navigate through the Control Panel to the System and Security menu. Select Administrative Tools, and then double-click the Services icon. When the Services dialog opens, highlight the service name in the list, and use the option provided on the dialog to Start the service.

ステップ 12-監視対象エージェントを新しい PEM サーバーホストに接続する

既存の PEM エージェントに新しい PEM サーバーホストに接続するよう指示するには、以下を行う必要があります。

- PEM エージェントホストが新しい PEM サーバーホストに接続できることを確認します。
- 新しい PEM サーバーの IP アドレスとポートを指定して、レジストリ（PEM エージェントを使用する各 Windows ホスト）またはエージェント構成ファイル（PEM エージェントを使用する各 Linux ホスト）を変更します。
- PEM エージェントのサービスを再起動します。

これらの手順はプラットフォーム固有です。

PEM エージェントが Linux に存在する場合

エディターの選択を使用して、`agent.cfg` ファイルを変更し、`pem_host` および `pem_port` パラメーターで PEM サーバーの新しい IP アドレスとポート番号を指定します。

デフォルトでは、`agent.cfg` ファイルは次の場所にあります。

```
/opt/PEM/agent/etc/agent.cfg
```

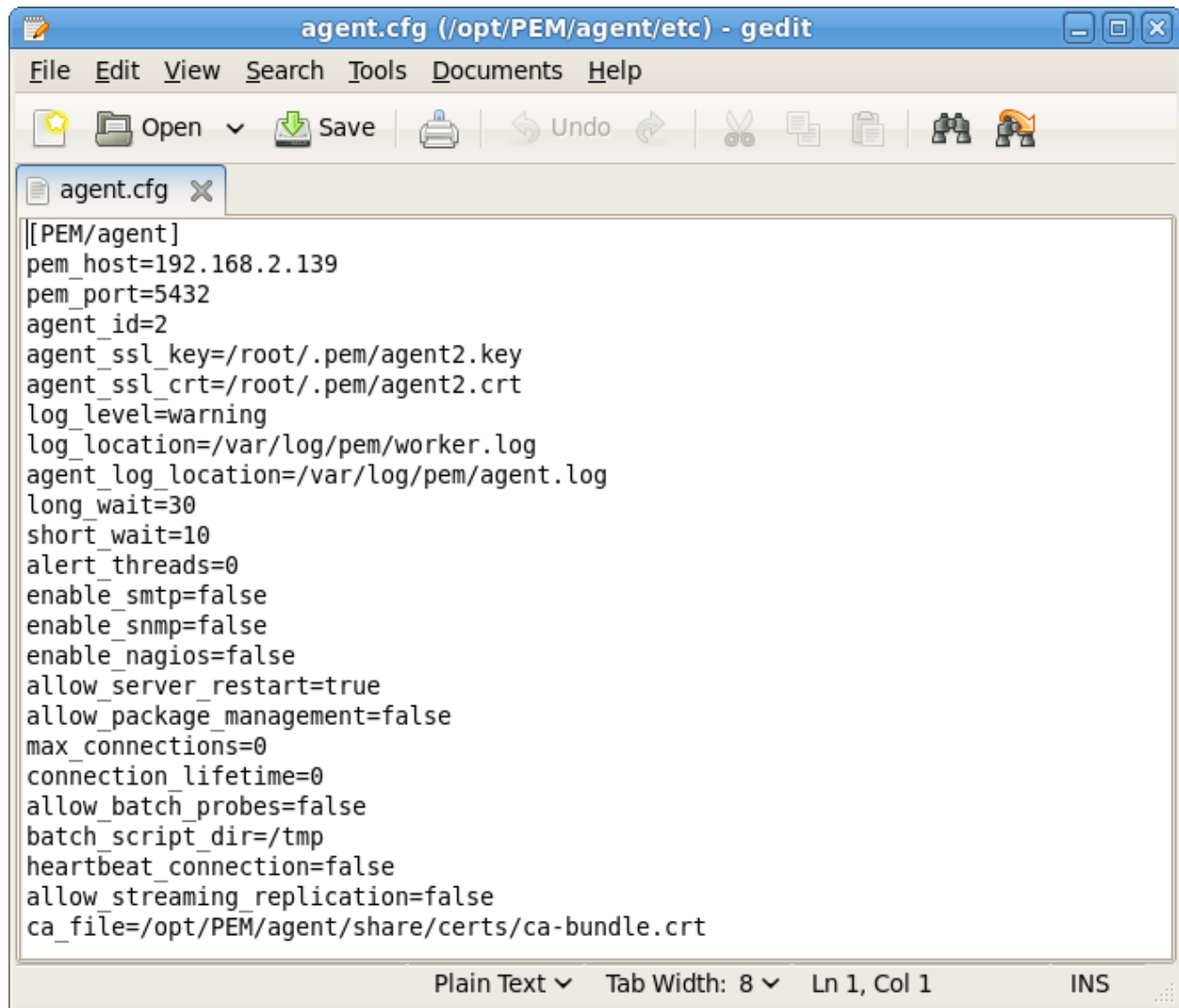


図2 agent.cfg ファイル

agent.cfg ファイルを変更した後、PEM エージェントサービスを再起動する必要があります。Linux コマンドラインで pemagent サービススクリプトを使用して、サービスを再起動できます。

```
/etc/init.d/pemagent restart
```

PEM エージェントが Windows に存在する場合

Before modifying the Windows registry on the monitored node, confirm that the firewall on the host of the PEM agent will allow connections to the PEM server. After confirming that the PEM agent host can connect to the PEM server host, you can use the Windows Registry Editor to review and edit the PEM_HOST and PEM_PORT entries to ensure that they correctly identify the host and port used by the PEM server. To open the Registry Editor, enter regedit in the Windows Run dialog or in the Windows start menu search box.

レジストリツリーコントロールをナビゲートして、レジストリエントリを表示または変更します。64 ビット

Windows では、PEM エージェントのレジストリエントリは次の場所にあります。

```
HKEY_LOCAL_MACHINE SOFTWARE wow6432Mode EnterpriseDB PEM agent
```

32 ビット Windows では、PEM エージェントのレジストリエントリは次の場所にあります。

```
HKEY_LOCAL_MACHINE SOFTWARE EnterpriseDB PEM agent
```

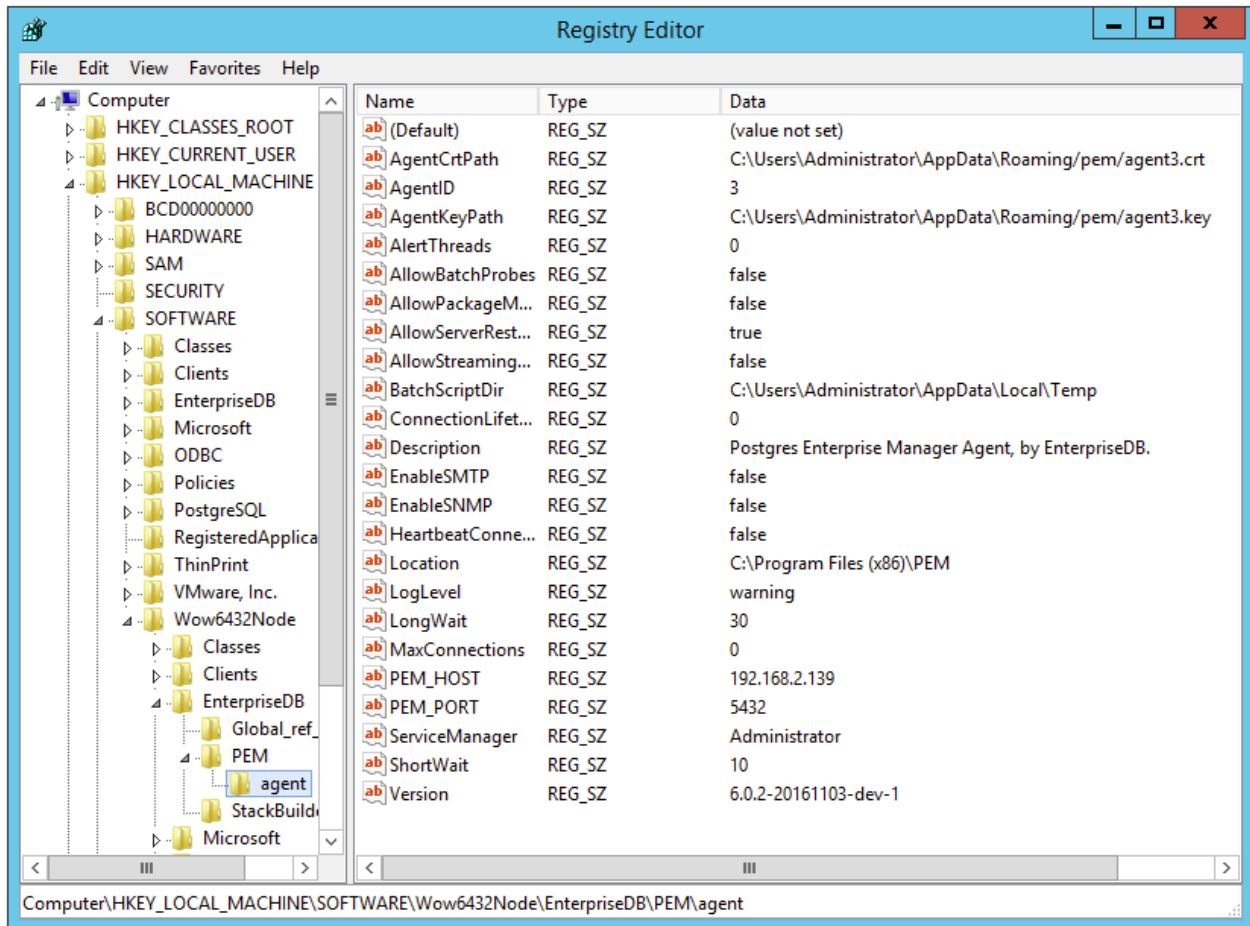


図 3 Windows レジストリエディター

PEM_HOST および PEM_PORT エントリは、ターゲットホスト上の新しい PEM サーバーのアドレスとポート番号を指定する必要があります。レジストリエントリを変更するには、エントリ Name を右クリックし、コンテキストメニューから Modify を選択して Edit String ダイアログを開きます。

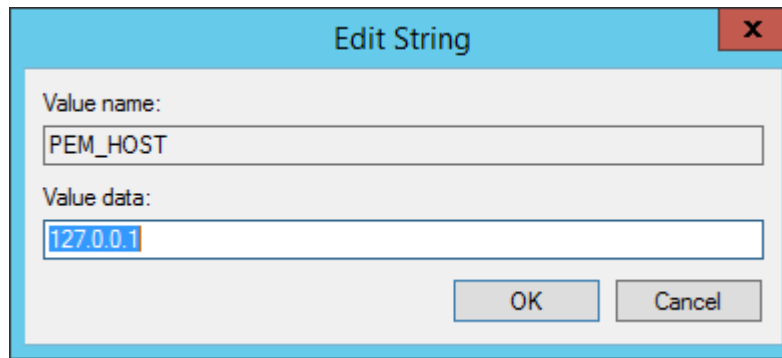


図 4 Windows レジストリエディター

エントリの値を変更するには、Edit String ダイアログを使用します。終了したら、OK をクリックして変更を保存するか、Cancel をクリックして保存せずに終了します。

レジストリを変更した後、PEM エージェントのサービスを再起動する必要があります。Services ダイアログ (Windows Control Panel からアクセス) を使用して Postgres Enterprise Manager - pemAgent サービスを再起動できます。

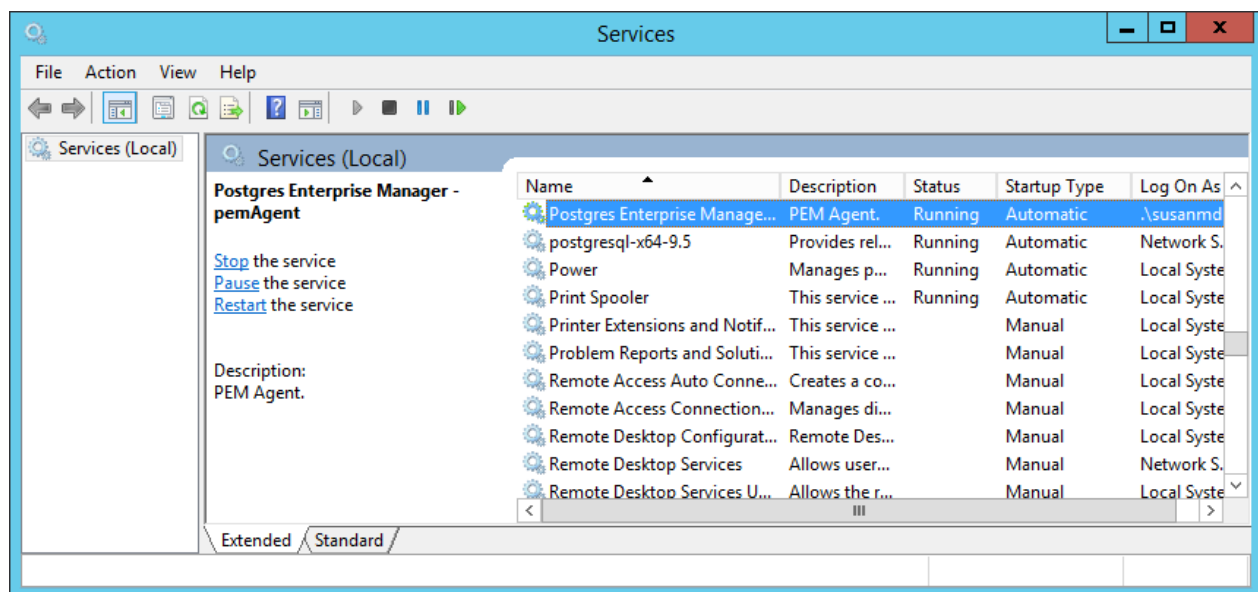


図 5 PEM エージェントのサービスの再起動

サーバーを移動した後、インストールされている PEM クライアントの接続プロパティを変更して、PEM サーバー、エージェント、監視対象サーバーの新しいホストに接続します。

第 4

トラブルシューティング

4.1 PEM サーバーの再構成

特定の状況では、PEM サーバーをアンインストールし、再度インストールし、PEM サーバーを再構成する必要があります。次のコマンドを指定された順序で 사용합니다。

1. 次のコマンドを使用して、PEM サーバー構成を削除し、アンインストールします。

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

2. 次のコマンドを使用して、PEM パッケージを削除します。

```
yum erase edb-pem-server
```

3. 次のコマンドを使用して、pem データベースを削除します。

```
DROP DATABASE pem
```

4. 証明書を /root/.pem/ から別の場所に移動します:

```
mv /root/.pem/* <new_location>
```

5. agent.cfg ファイルを /usr/edb/pem/agent/etc/agent.cfg から別の場所に移動します:

```
mv /usr/edb/pem/agent/etc/agent.cfg <new_location>
```

6. 次に、次のコマンドを使用して PEM サーバーを再構成します。

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

4.2 pem.alert テーブルが復元に失敗する

When restoring the pem backing database from backup, you may encounter an error during the restoration of the pem.alert table. This is caused by a missing table pre-requisite for the table - the pg_restore utility may restore the pem.alert pre-requisites *after* it attempts to restore pem.alert.

この場合、pg_restore からの出力には、alert テーブルを参照するエラーメッセージが含まれます。

```
pg_restore: [archiver (db)] could not execute query: ERROR: insert or update on table
↳ "alert_history" violates foreign key constraint "alert_history_alert_id_fkey"
DETAIL: Key (alert_id)=(3) is not present in table "alert".
Command was: ALTER TABLE ONLY alert_history
ADD CONSTRAINT alert_history_alert_id_fkey FOREIGN KEY (alert_id) REFERENCES alert(id)
↳ ON...
pg_restore: creating FK CONSTRAINT alert_status_alert_id_fkey
pg_restore: [archiver (db)] Error from TOC entry 3265; 2606 18355 FK CONSTRAINT alert_
↳ status_alert_id_fkey postgres
pg_restore: [archiver (db)] could not execute query: ERROR: insert or update on table
↳ "alert_status" violates foreign key constraint "alert_status_alert_id_fkey"
DETAIL: Key (alert_id)=(1) is not present in table "alert".
Command was: ALTER TABLE ONLY alert_status
ADD CONSTRAINT alert_status_alert_id_fkey FOREIGN KEY (alert_id) REFERENCES alert(id)
↳ ON U...
```

If you encounter this problem, restore the pem database before restoring the pem.alert table. Restoring the pem database will install the pre-requisites for pem.alert, and the restoration of the table should complete as expected.

EDBPostgresEnterpriseManager アップグレードおよび移行ガイド 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

- EnterpriseDB および PostgresEnterpriseManager は、EnterpriseDBCorporation の登録商標です。EDB および EDBPostgres 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、存在する場合、EDB 製品のログオン 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、および入力検証の検証を行います。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その他の危険な入力を防ぐことはできません。
- 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。

BackingPostgres データベースのアップグレード, [38](#)

Linux ホストでのグラフィカル PEM インストールのアップグレード, [29](#)

Linux ホスト上の RPM パッケージを使用した PEM インストールのアップグレード, [22](#)

PEM インストールのアップグレード, [2](#)

PostgresEnterpriseManager™ サーバーの移動, [48](#)

reconfiguring an agent, [59](#)

Upgrading a PEM Server Installed with a Graphical Installer, [29](#)

Upgrading PEM agent installed with graphical installer, [33](#)

Windows ホストでの PEM インストールのアップグレード, [2](#)

トラブルシューティング, [59](#)

隔離されたネットワークでの PEM リポジトリの作成, [28](#)

結論, [62](#)