



Postgres Enterprise Manager

リリース **8.0**

Upgrade and Migration Guide

2020 年 12 月 30 日

目次

第 1 章	Upgrading a PEM Installation	3
1.1	Upgrading a PEM Installation on Windows Host	3
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	19
1.2	Upgrading a PEM Native Package Installation on a Linux Host	21
1.2.1	Linux ホスト上の PEM インストールをアップグレードするための前提条件	21
1.2.2	PEM エージェントネイティブパッケージインストールのアップグレード	23
1.2.2.1	CentOS または RHEL ホストでの PEM エージェントのアップグレード	23
1.2.2.2	Debian または Ubuntu ホストでの PEM エージェントのアップグレード	24
1.2.2.3	SLES ホスト上の PEM エージェントのアップグレード	24
1.2.3	PEM サーバーネイティブパッケージインストールのアップグレード	24
1.2.3.1	CentOS または RHEL ホスト上の PEM サーバーのアップグレード	24
1.2.3.2	Debian または Ubuntu ホストでの PEM サーバーのアップグレード	25
1.2.3.3	SLES ホスト上の PEM サーバーのアップグレード	25
1.2.4	SQLProfilerNativePackage インストールのアップグレード	26
1.3	Creating a PEM Repository on an Isolated Network	28
1.4	Upgrading a Graphical PEM Installation on a Linux Host	29
1.4.1	Linux ホスト上の PEM インストールをアップグレードするための前提条件	29
1.4.2	CentOS または RHEL ホストでの PEM エージェント（グラフィカルインストール）のアップグレード	30
1.4.3	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	34
1.4.3.1	CentOS または RHEL ホストに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	34
1.4.4	グラフィカルインストーラーを使用した SQL プロファイラーインストールのアップグレード	39
1.5	Configuring a PEM Server on a Linux Host	41
第 2 章	Upgrading the PEM Backend Postgres Database	43
第 3 章	Moving the Postgres Enterprise Manager Server	53
第 4 章	Troubleshooting	65

4.1 PEM サーバーの再構成	65
第 5 章 Conclusion	67
索引	69

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。

- **Upgrading a PEM Installation** -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。
- **Upgrading the Backend Database** -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、バックエンド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。
- **Moving a PEM Server** -このセクションは、PEM サーバーをあるホストから別のホストに移動することに関する詳細情報を提供します。
- **Troubleshooting** -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。

第 1 章

Upgrading a PEM Installation

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ました。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、ネイティブパッケージ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、EDBWeb サイト<<https://www.enterprisedb.com/downloads/edb-postgres-enterprise-manager>>_から入手できます。

1.1 Upgrading a PEM Installation on Windows Host

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呼び出すだけです。

1. PEM サーバーホストの各監視対象ノード `except` で PEM エージェントインストーラーを起動します。
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 サーバーホストに存在する PEM エージェントを `both` に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、必要に応じて SQL プロファイラーをアップグレードできます。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

PEM エージェントによって現在監視されているシステムを最新の 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 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、お迎えします。

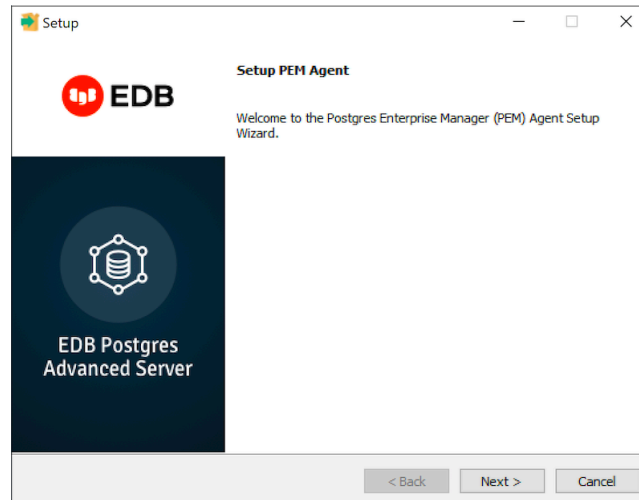


図 1 PEM エージェントインストーラー-ようこそウィンドウ

Next をクリックして続行する前に、License Agreement を読んで受け入れます。

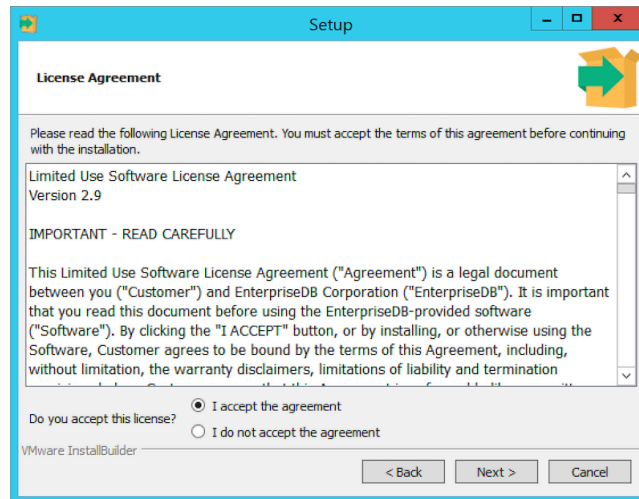


図 2 PEM エージェントインストーラー-ライセンス契約

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。

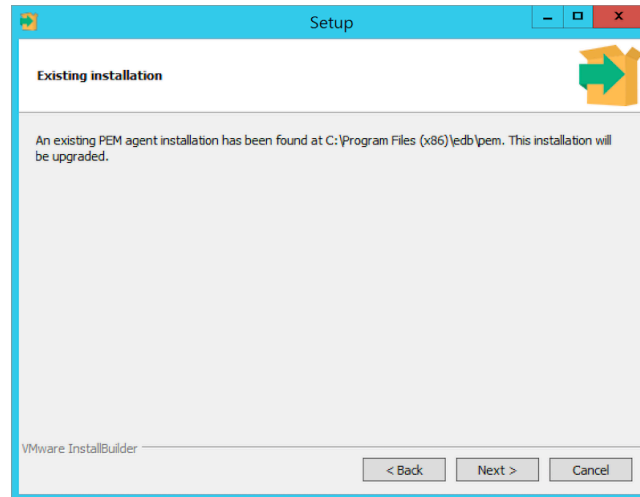


図 3 PEM エージェントインストーラー-既存のインストール場所

pemAgent service account ダイアログは、PEMAgent 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。

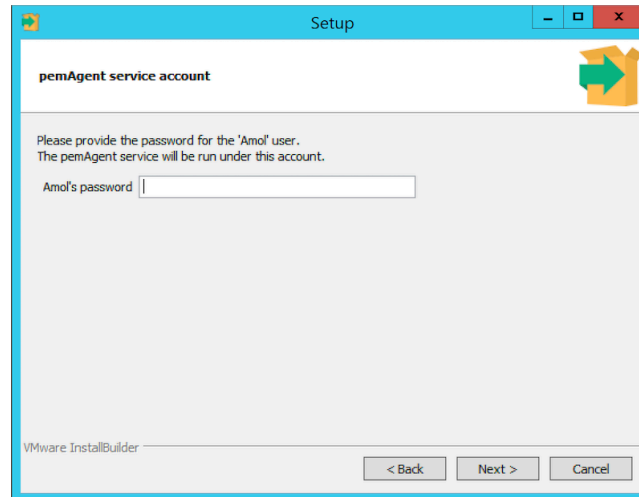


図 4 PEM エージェントインストーラー-pemAgent サービスアカウントのパスワード

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 ダイアログでインストールが開始されることが通知されたら、Next をクリックして続行します。PEM エージェントが最新バージョンにアップグレードされます。

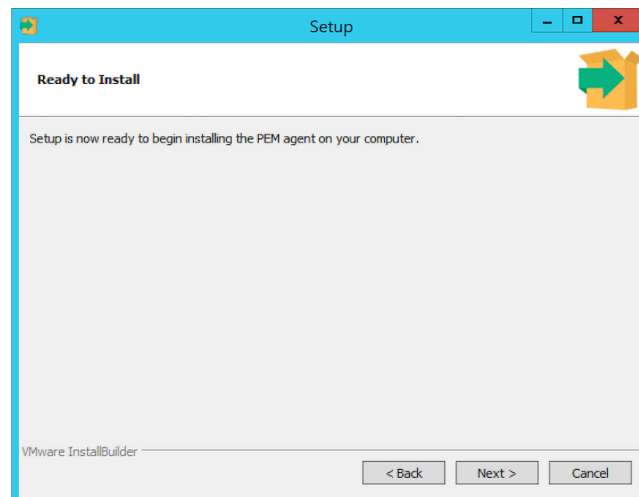


図 5 PEM エージェントインストーラー-インストール準備完了

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。

PEM Agent Setup Wizard は、インストールが完了すると通知します。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。

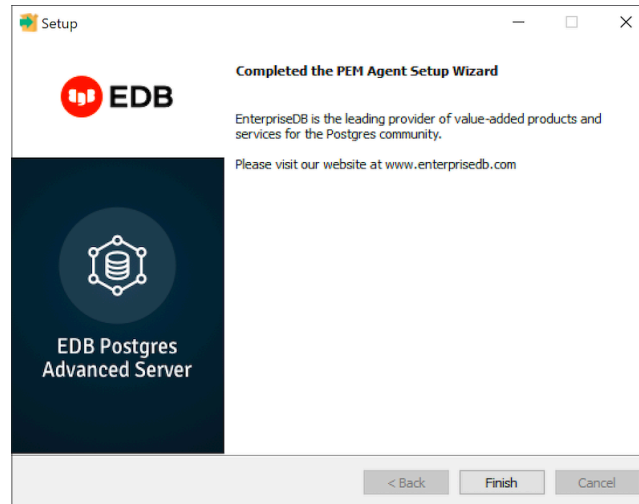


図 6 PEM エージェントインストーラー-完了

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 PEM エージェントを再起動します。

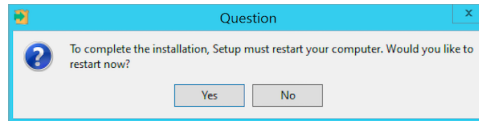


図 7 PEM エージェントインストーラー-再起動ポップアップ

注釈: 7.11 より前の PEM バージョンを持つ Windows エージェントをアップグレードする場合、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既に構成している、または構成する予定がある場合は、`agent.cfg` で `AllowBatchJobSteps` パラメーターを `True` に設定する必要があります」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。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 7.16 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。

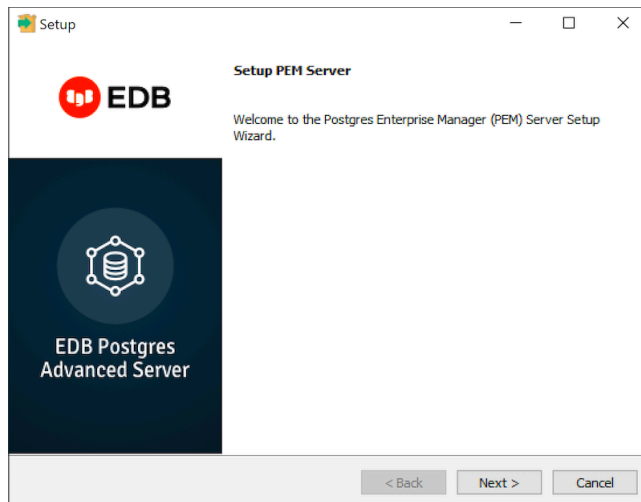


図 8 PEM サーバーインストーラー-ようこそウィンドウ

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。

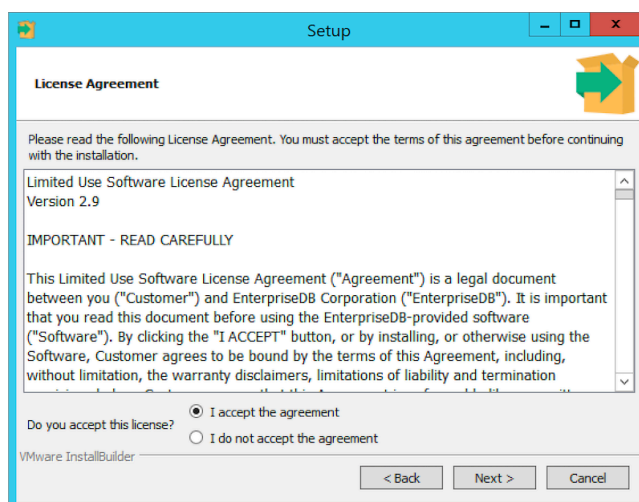


図 9 PEM サーバーインストーラー-ライセンス契約

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。

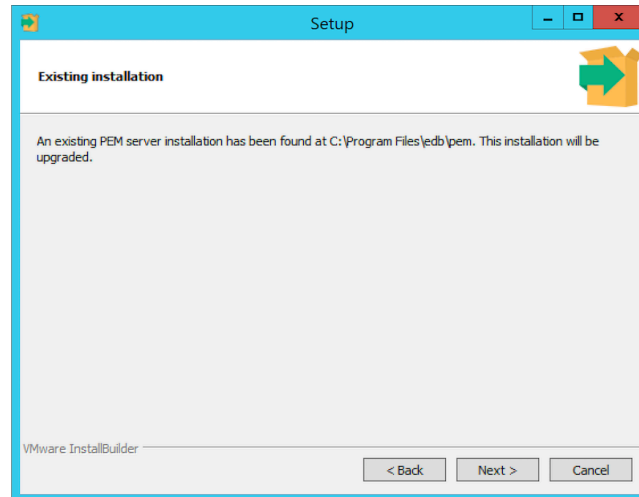


図 10 PEM サーバーインストーラー-既存の PEM サーバーの場所

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見つからない場合、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。

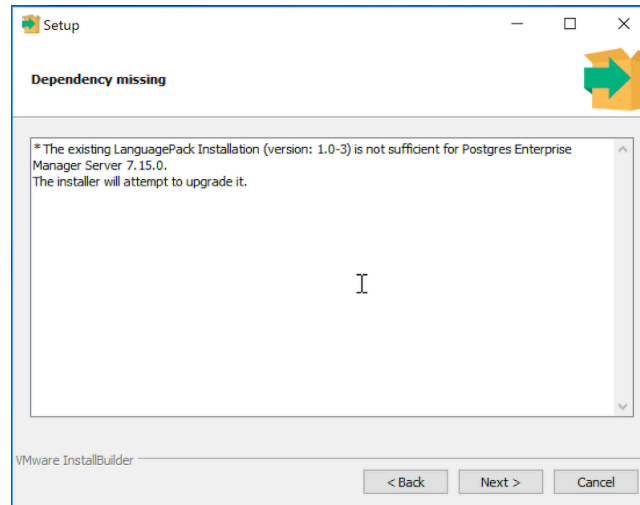


図 11 PEM サーバーインストール-不足している依存関係

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。

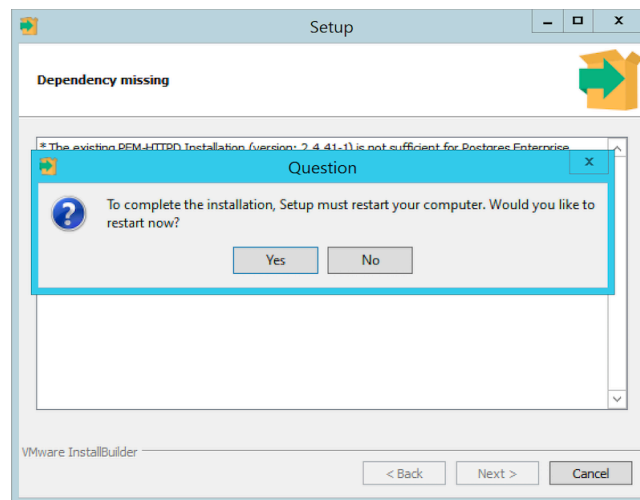


図 12 PEM サーバーインストーラー-再起動ポップアップ

No をクリックして、アップグレードプロセスを続行します。

次に、ウィザードは Database Server Installation Details ダイアログを開き、PEM バックエンド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お願いします。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。

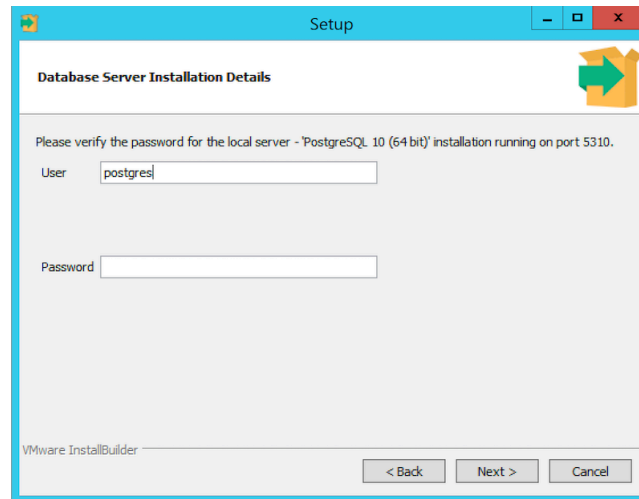


図 13 PEM サーバーインストーラー-データベースサーバーのインストールの詳細

pemAgent service account ダイアログでは、PEM エージェントサービスを実行するアカウントのパスワードを求められる場合があります。

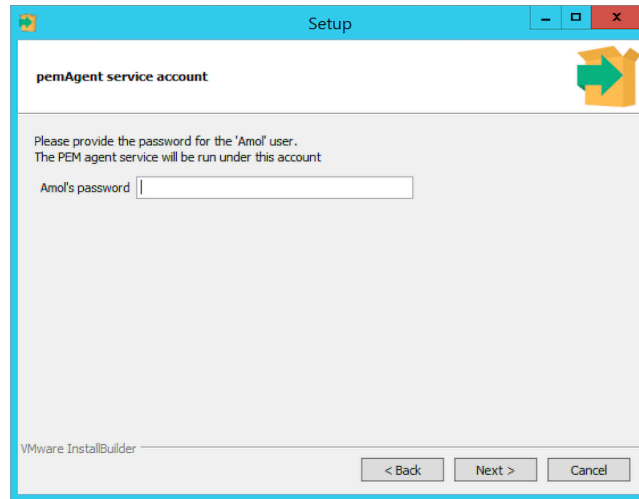


図 14 PEM サーバーインストーラー-pemAgent サービスアカウントのパスワード

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。

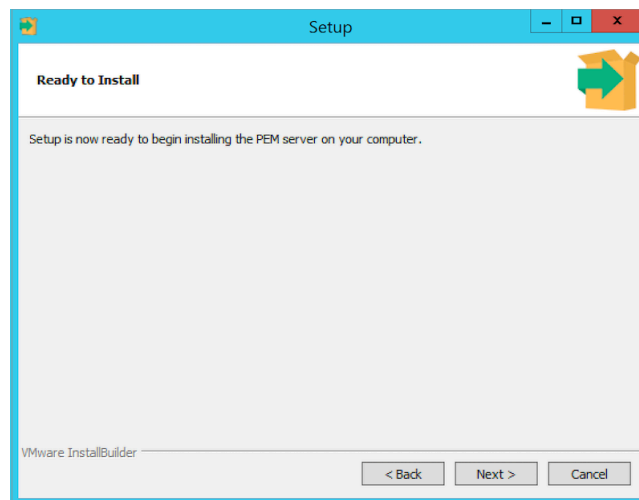


図 15 PEM サーバーインストーラー-インストール準備完了

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。

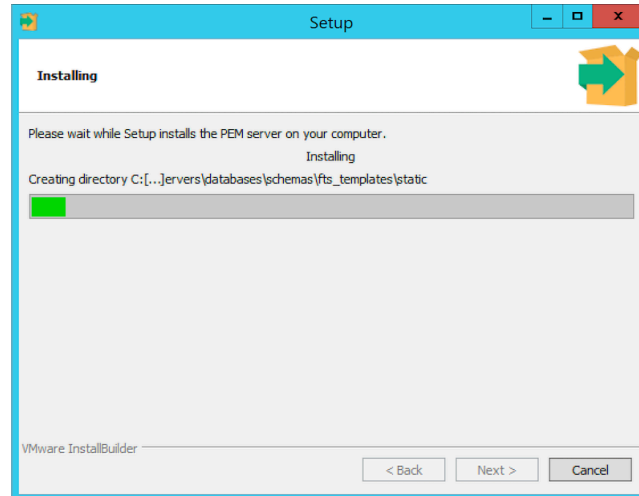


図 16 進行中の PEM サーバーのインストール

PEM サーバー（および PEM サーバーと同じホストに存在するエージェント）をアップグレードし、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ッ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。

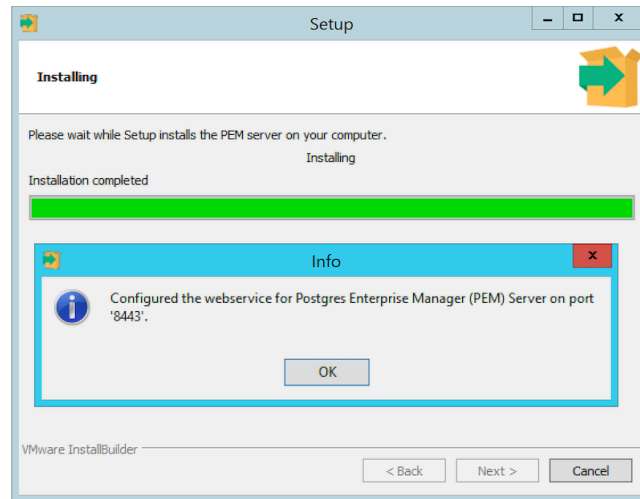


図 17 PEM サーバーインストーラー-設定された PEMWeb サービス

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。

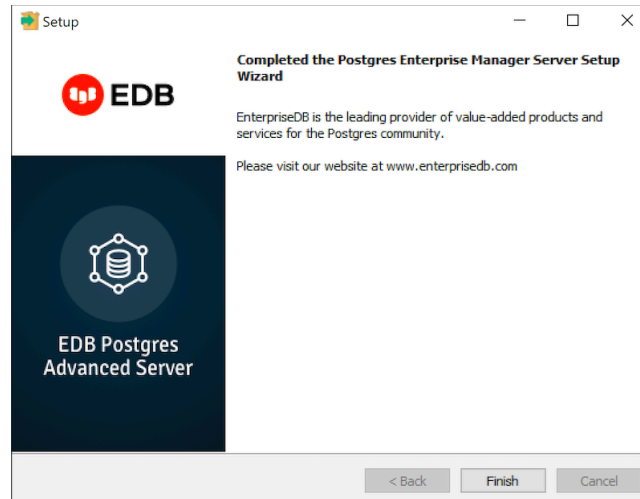


図 18 PEM サーバーインストーラー-アップグレードの完了

ウィンドウがポップアップしてマシンの再起動を求められたら、Yes をクリックしてマシンを再起動し、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（各ホストで 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>
```

<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。

PEM サーバーをアップグレードした後、バックエンド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することができます。バックエンドデータベースのアップグレードについては、[バックエンド Postgres データベースのアップグレード](#) を参照してください。

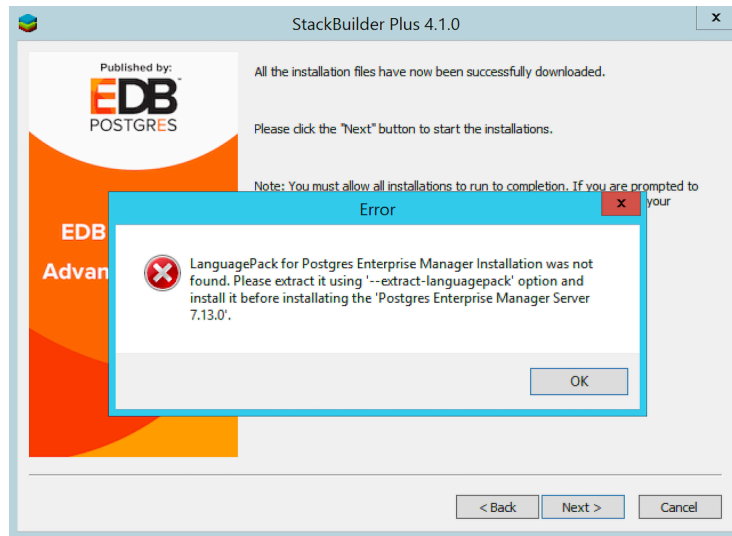


図 19 PEM サーバーのアップグレードエラー

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たオプションを使用してサービスを Stop 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 プロファイラーの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
```

x は AdvancedServer のバージョンで、-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、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
```

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 -U postgres
```

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
```

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、バックエンドデータベースを更新する準備ができました。

1.2 Upgrading a PEM Native Package Installation on a Linux Host

Linux ホストで 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、新しいバージョンの PEM コンポーネントネイティブパッケージを次の順序でインストールするだけです。

1. PEM サーバーホストの各監視対象ノード `except` で PEM エージェントネイティブパッケージインストーラーを起動します。
2. PEM サーバーのネイティブパッケージインストーラーを起動します。PEM サーバーと PEM サーバーホストに存在する PEM エージェントを `both` にアップグレードします。

インストール中、コンポーネントのインストール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、必要に応じて SQL プロファイラーをアップグレードできます。このステップはプラットフォーム固有です。

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している場合、または構成する予定の場合、`batch_script_user` パラメーターのユーザーを `agent.cfg` ファイルで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。

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。

1.2.1 Linux ホスト上の PEM インストールをアップグレードするための前提条件

PEM は、`python3`、`libboost`、`openssl`、`snmp++`、`libcurl` など、ベンダー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確認するには、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。バージョン 10 より古いバージョンの PostgreSQL または AdvancedServer を使用している場合、アップグレードの前に、PEM サーバーがインストールされているシステムにバージョン 10 以上の `libs` パッケージをインストールする必要があります。次のプラットフォーム固有のコマンドを使用して、ホストに `libs` バージョン 10 以降をインストールします。

Prerequisites to Upgrade a PEM Installation on a CentOS or RHEL Host

CentOS または RHEL7.x ホストでパッケージをアップグレードするには

```
yum upgrade
```

CentOS または RHEL8.x ホストでパッケージをアップグレードするには

```
dnf upgrade
```

AdvancedServer ライブラリをアップグレードするには:

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

PostgreSQL ライブラリをアップグレードするには:

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

<X> は、libs パッケージをインストールする PostgreSQL または AdvancedServer バージョンです。

Prerequisites to Upgrade a PEM Installation on a Debian or Ubuntu Host

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

```
apt-get update
```

AdvancedServer ライブラリをアップグレードするには:

```
apt-get install edb-as<X>-server-libs
```

PostgreSQL ライブラリをアップグレードするには:

```
apt-get install postgresql<X>-libs
```

<X> は、libs パッケージをインストールする PostgreSQL または AdvancedServer バージョンです。

Prerequisites to Upgrade a PEM Installation on a SLES Host

SLES ホストでパッケージをアップグレードするには

```
zypper update
```

AdvancedServer ライブラリをアップグレードするには:

```
zypper install edb-as<x>-server-libs
```

PostgreSQL ライブラリをアップグレードするには:

```
zypper install postgresql<x>-libs
```

<X> は、libs パッケージをインストールする PostgreSQL または AdvancedServer バージョンです。

1.2.2 PEM エージェントネイティブパッケージインストールのアップグレード

ネイティブパッケージを使用して、ネイティブパッケージを使用して最初にインストールされた既存の PEM 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。

1.2.2.1 CentOS または RHEL ホストでの PEM エージェントのアップグレード

CentOS または RHEL7.x または 8.x の場合:

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
```

CentOS または RHEL8.x の場合、次のコマンドも使用できます。

```
dnf 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。

1.2.2.2 Debian または Ubuntu ホストでの PEM エージェントのアップグレード

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。

```
apt-get upgrade edb-pem-agent
```

1.2.2.3 SLES ホスト上の PEM エージェントのアップグレード

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。

```
zypper update edb-pem-agent
```

1.2.3 PEM サーバーネイティブパッケージインストールのアップグレード

最初にネイティブパッケージを使用して PEM サーバーをインストールした場合、ネイティブ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。アップグレードするコマンドはプラットフォーム固有です。以下のプラットフォームのセクションを参照してください。

分離されたネットワーク内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、そのマシン上に 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、[分離されたネットワークでの PEM リポジトリの作成](#) を参照してください。

1.2.3.1 CentOS または RHEL ホスト上の PEM サーバーのアップグレード

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。

yum パッケージマネージャーを使用して、CentOS または RHEL7.x または 8.x に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できます。

```
yum upgrade edb-pem
```

CentOS または RHEL8.x で dnf コマンドを使用することもできます。

```
dnf upgrade edb-pem
```

注釈: CentOS または RHEL7.x ホストに PEM サーバーを新規インストールする場合、パッケージはオペレーティングシステムの要件であるため、インストーラーは edb-python3-mod_wsgi パッケージをインストールとともにインストールします。CentOS または RHEL7.x ホスト上の PEM サーバーを*アップグレード*している場合、mod_wsgi パッケージは、オペレーティングシステムの要件を満たすために edb-python3-mod_wsgi パッケージに置き換えられます。

yum または dnf を使用して PEM サーバーをアップグレードした後、PEM サーバーを構成する必要があります。詳細については、[Linux プラットフォームでの PEM サーバーの設定](#) を参照してください。

1.2.3.2 Debian または Ubuntu ホストでの PEM サーバーのアップグレード

apt-get パッケージマネージャーを使用して、サポートされているバージョンの Debian または Ubuntu に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できます。

```
apt-get upgrade edb-pem
```

apt-get で PEM サーバーをアップグレードした後、PEM サーバーを構成する必要があります。詳細については、[Linux プラットフォームでの PEM サーバーの設定](#) を参照してください。

1.2.3.3 SLES ホスト上の PEM サーバーのアップグレード

zypper パッケージマネージャーを使用して、サポートされているバージョンの SLES ホストに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できます。

```
zypper update edb-pem
```

zypper を使用して PEM サーバーをアップグレードした後、PEM サーバーを構成する必要があります。詳細については、[Linux プラットフォームでの PEM サーバーの設定](#) を参照してください。

1.2.4 SQLProfilerNativePackage インストールのアップグレード

Linux ホストにある 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、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 upgrade postgresql<x>-sqlprofiler
```

ここで、`x` は PostgreSQL のバージョンです。

AdvancedServer の場合:

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

`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 プロファイラの最新バージョンでアップグレードします。

PostgreSQL 用の SQL プロファイラをアップグレードする次の例を参照してください。

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:

```
[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

```
/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、バックエンドデータベースを更新する準備ができました。

1.3 Creating a PEM Repository on an Isolated Network

PEM をアップグレードするサーバーが EDB 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> 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。

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. 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
```

5. 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 Upgrading a Graphical PEM Installation on a Linux Host

Linux ホストで 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、新しいバージョンの PEM コンポーネントネイティブパッケージを次の順序でインストールするだけです。

- PEM サーバーホストの各監視対象ノード `except` で PEM エージェントネイティブパッケージのインストールを呼び出します。
- PEM サーバーのネイティブパッケージインストールを呼び出します。PEM サーバーと PEM サーバーホストにある PEM エージェントを `both` にアップグレードします。

インストール中、コンポーネントのインストール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、必要に応じて SQL プロファイラーをアップグレードできます。このステップはプラットフォーム固有です。

次のセクションでは、Linux ホスト上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 Linux ホスト上の PEM インストールをアップグレードするための前提条件

バージョン 10 より古いリリースの Postgres または AdvancedServer のバージョンを使用している場合、アップグレードの前に、PEM サーバーがインストールされているシステムにバージョン 10 以上の `libs` パッケージをインストールする必要があります。

AdvancedServer の場合:

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

PostgreSQL の場合:

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

<X> 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。

1.4.2 CentOS または RHEL ホストでの PEM エージェント（グラフィカルインストール）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 エージェント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。

CentOS または RHEL7.x0r8.x ホストの場合:

```
systemctl stop pemagent
```

2. 次のいずれかのコマンドを実行して、サポートする epel-release パッケージをホストにインストールします。

```
yum install epel-release
```

```
yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-<X>.  
↪noarch.rpm
```

X は OS バージョンです。

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場合があります。

RedHatNetwork ユーザーの場合、

- また、EPEL パッケージを使用するには rhel-<x>-server-optional-rpms リポジトリを有効にする必要があります。x はホスト上の RHEL の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ようになります。証明書ベースのサブスクリプションがある場合、EPEL パッケージを使用するには rhel-<x>-server-eus-optional-rpms リポジトリを有効にする必要があります。必要なリポジトリについては Red Hat Subscription Management Guide を参照してください。
 - また、rhel-<x>-server-extras-rpms リポジトリを有効にする必要があります。x はホスト上の RHEL のバージョンを指定します。
-

3. edb.repo ファイルをインストールして構成します。

- a. EDB 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。資格情報を要求するには、次をご覧ください。

EDB リポジトリアクセス手順<https://info.enterprisedb.com/rs/069-ALB-339/images/Repository%20Access%20004-09-2019.pdf>。`\_`。

- b. リポジトリ構成ファイルを作成します。スーパーユーザー特権を想定し、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は edb.repo です。ファイルは /etc/yum.repos.d にあります。

- c. 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登録済み EDB ユーザーの名前とパスワードに置き換えられるようにします、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
```

分離されたネットワークにあるマシンに PEM エージェントをインストールする場合は、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。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、「[隔離されたネットワークでの PEM リポジトリの作成](#)」を参照してください。

4. yum makecache コマンドは、現在有効な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 エージェントが含まれていることを確認します。

CentOS または RHEL7.x または 8.x の場合:

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

CentOS または RHEL8.x の場合:

```
dnf makecache  
  
dnf list edb-pem-agent
```

5. 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。

CentOS または RHEL7.x または 8.x の場合:

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

CentOS または RHEL8.x の場合:

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

6. インストール後、PEM エージェント構成ファイル (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
```

注釈: パッケージ管理機能と複製レプリケーション機能は、PEM7.16 バージョンから非推奨になりました。PEM エージェントサービスの再起動後にワーカログファイルに警告メッセージが表示されないように、ファイルが存在する場合は、allow_package_management および allow_streaming_replication をコピーするときに、これらのパラメーターを agent.cfg ファイルから削除してください。

7. 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 パラメーターの値を設定します。

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

8. 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。

CentOS または RHEL7.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
```

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
```

9. 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
```

10. 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
```

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

11. 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
```

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、または構成する予定の場合、agent.cfg ファイルを変更し、batch_script_user パラメーターに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。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 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

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。

1.4.3.1 CentOS または RHEL ホストに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

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にある場合、Web サーバーでコマンドを実行します。

CentOS または RHEL7.xOr8.x ホストの場合:

```
systemctl stop PEMHTTPD
systemctl stop pemagent
```

3. 次のいずれかのコマンドを実行して、サポートする epel-release パッケージをホストにインストールします。

```
yum install epel-release
```

```
yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-<X>.  
↪noarch.rpm
```

<X> は OS バージョンです。

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場合があります。

RedHatNetwork ユーザーの場合、

- また、EPEL パッケージを使用するには rhel-<x>-server-optional-rpms リポジトリを有効にする必要があります。x はホスト上の RHEL の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ようになります。証明書ベースのサブスクリプションがある場合、EPEL パッケージを使用するには rhel-<x>-server-eus-optional-rpms リポジトリを有効にする必要があります。必要なリポジトリについては Red Hat Subscription Management Guide を参照してください。

- また、`rhel-<x>-server-extras-rpms` リポジトリを有効にする必要があります。x はホスト上の RHEL のバージョンを指定します。

4. edb.repo ファイルをインストールして構成します。

- EDB 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。資格情報を要求するには、次をご覧ください。

EDB リポジトリアクセス手順<https://info.enterprisedb.com/rs/069-ALB-339/images/Repository%20Access%202004-09-2019.pdf>。

- リポジトリ構成ファイルを作成します。スーパーユーザー特権を想定し、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、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登録済み EDB ユーザーの名前とパスワードに置き換えられるようにします、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
```

分離されたネットワークにあるマシンに PEM エージェントをインストールする場合は、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。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、「[隔離されたネットワークでの PEM リポジトリの作成](#)」を参照してください。

- `yum makecache` コマンドは、現在有効な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。

CentOS または RHEL7.x または 8.x の場合:

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

CentOS または RHEL8.x の場合:

```
dnf makecache  
  
dnf list edb-pem
```

6. PEM サーバー 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 サーバーのインストール情報を確認できます。

CentOS または RHEL7.x または 8.x の場合:

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

CentOS または RHEL8.x の場合:

```
dnf install edb-pem
```

7. 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
```

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` パラメーターの値を設定します。

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

9. `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/
```

10. `passwd` ファイルのホームディレクトリを、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
```

11. 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
```

12. 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
```

13. 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
```

14. RPM またはインストーラーを使用して `sslutils` をインストールします（Windows の場合、インストーラーはファイルを上書きできないため、インストールの前にデータベースサーバーを停止する必要があります。そうしないと、システムの再起動が必要になります）

データベースサーバーを再起動します（拡張機能の共有オブジェクトがアップグレードされたため、データベースサーバーを再起動する必要があります）

参照セクション [sslutils のインストール](#)

`/"ALTEREXTENSION/"` ステートメントを使用してバージョンをアップグレードします

```
ALTER EXTENSION sslutils UPDATE [ TO new_version ]
```

15. PEM サーバーおよび 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、バックエンド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
```

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。

CentOS または RHEL7.x0r8.x ホストの場合:

```
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す。
18. 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。

```
/opt/edb/pem/httpd/uninstall-pemhttpd
```

1.4.4 グラフィカルインストーラーを使用した SQL プロファイラーインストールのアップグレード

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 プロファイラーの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 `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
```

`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
```

`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。

- 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。

- 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、バックエンドデータベースを更新する準備ができました。

1.5 Configuring a PEM Server on a Linux Host

PEM サーバーをアップグレードした後、次のコマンドを使用して PEM サーバーを構成できます。

```
/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/as11、PostgreSQL の場合は/usr/pgsql-11 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-11 です。PostgreSQL の場合、postgresql-11 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:設定が Web サービスとバックエンドデータベース用の場合は 1、Web サービスを設定する場合は 2、バックエンドデータベースを設定する場合は 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 サーバーを登録解除します。
-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
```

PEM サーバー構成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*PEMLinux 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>

第 2 章

Upgrading the PEM Backend Postgres Database

PEM コンポーネントと PEM バックエンドデータベースの両方を更新する場合は、バックエンドデータベースを更新する前に、PEM コンポーネントの更新（サーバーとエージェント）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、[PEM インストールのアップグレード](#) を参照してください。

注釈: PEM8.0 以降では、PostgreSQL または EPAS バージョン 11 以降は、バックエンドデータベースサーバーとしてのみサポートされています。したがって、バックエンドデータベースサーバーがバージョン 11 よりも低い場合、まずバックエンドデータベースサーバーをアップグレードしてから、PEM コンポーネントをアップグレードする必要があります。

このセクションで説明する更新プロセスでは、pg_upgrade ユーティリティを使用して、バックエンドサーバーのあるバージョンからより新しいバージョンに移行します。pg_upgrade は、Postgres の任意のバージョン（バージョン 9.5 以降）と、同じプラットフォームでサポートされる Postgres の後続のリリースとの間の移行を容易にします。

pg_upgrade は、同じタイプのサーバー間のデータ転送をサポートします。たとえば、pg_upgrade を使用して、PostgreSQL9.6 バックエンドデータベースから PostgreSQL11 バックエンドデータベースにデータを移動できますが、AdvancedServer11 バックエンドデータベースには移動できません。別のタイプのバックエンドデータベースに（つまり、PostgreSQL サーバーから AdvancedServer に）移行する場合は、[PostgresEnterpriseManager サーバー](#) の移動を参照してください。

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>

1. 更新されたインストーラーをダウンロードして呼び出します。PostgreSQL および AdvancedServer のインストーラーは、EDBWeb 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 バックエンドデータベースのホストとして使用できます。ポート 5432 でリッスンする PostgreSQL バックエンド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構成する必要があることに注意してください。

2. 新しいサーバーで SSL ユーティリティを構成します。新しいバックエンドデータベースは、現在のバックエンドデータベースが実行し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。EDB 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>

AdvancedServer を新しいバックエンドデータベースとして使用する場合、`sslutils` 拡張子を手動で追加する必要はありません。`sslutils` を設定するプロセスはプラットフォーム固有です。

On Linux

- AdvancedServer 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。
- PEM バックエンドデータベースとして PostgreSQL を使用している場合は、PostgreSQL コミュニティリポジトリにアクセスできることを確認し、次のコマンドを使用します。

```
yum install sslutils_<X>
```

<X> はサーバーのバージョンです。

- PEM バックエンドデータベースとして PostgreSQL の EDB ワンクリックインストーラーを使用している場合

```
yum install gcc openssl-devel

Set the value of PATH so it can locate the pg_config program
export PATH=$PATH:/opt/postgres_inst_dir/<X>/bin/

Move into the sslutils folder, and enter
make USE_PGXS=1
make USE_PGXS=1 install

Use psql to create the sslutils extension
CREATE EXTENSION sslutils
```

Debian10 および Ubuntu20 では、セキュリティ上の理由により、証明書を受け入れるための要件が増加していることに注意してください。ユーザーがいずれかのマシンに 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 署名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。sslutils を 1.3 にアップグレードしない場合、PEM エージェントが PEM バックエンドデータベースサーバーへの接続に失敗し、エラー `ca md too weak` が記録される場合があります。

On Windows

sslutils は、元のバックエンドデータベースで sslutils をコンパイルするために使用したのと同じコンパイラを使用して、新しいバックエンドデータベースでコンパイルする必要があります。PostgreSQL ワンクリックインストーラー（EDB から）または AdvancedServer インストーラーを使用してインストールされた Postgres データベースに移動する場合は、VisualStudio を使用して sslutils をビルドします。PostgreSQL9.5 以降にアップグレードする場合は、VisualStudio2010 を使用してください。

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、次の EDBWeb サイトで:

<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、使用している 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 インストーラーを起動する前に、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

VisualStudio コマンドラインを開き、sslutils ディレクトリに移動します。次のコマンドを使用して sslutils をビルドします。

```
SET USE_PGXS=1

SET GETTEXT_PATH=\ <path_to_gettext>

SET OPENSSL_PATH=\ <path_to_openssl>

SET PG_PATH=\ <path_to_pg_installation_dir>
```

(次のページに続く)

(前のページからの続き)

```
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、次の一連のコマンドは、`PostgreSQL11` サーバーに `OpenSSL` サポートを構築します。

```
SET USE_PGXS=1

SET OPENSSELPATH=C:\OpenSSL-Win32

SET GETTEXTPATH="C:\Program Files\PostgreSQL\11"

SET PGPATH="C:\Program Files\PostgreSQL\11"

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--1.3.sql

sslutils--unpackaged--1.3.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\"

COPY sslutils.control "%PGPATH%\share\extension\"
```

3. 古いバックエンドデータベースと新しいバックエンド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。

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 メニューに移動します。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サービスを停止します。

4. pg_upgrade ユーティリティを使用して、古いバックエンドデータベースと新しいバックエンドデータベースの間で既存のデータのインプレース転送を実行します。サーバーが md5 認証を強制するように構成されている場合、データベーススーパーユーザーの接続プロパティ（およびパスワード）を指定する .pgpass ファイルにエントリを追加するか、pg_hba.conf ファイルを変更して信頼接続を許可する必要があります pg_upgrade を呼び出す前。 .pgpass ファイルにエントリを作成する方法の詳細については、次の場所にある PostgreSQL コードドキュメントを参照してください。

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

アップグレードプロセス中に、pg_upgrade は一連のログファイルを書き込みます。クラスターの所有者は、書き込み権限があるディレクトリから pg_upgrade を呼び出す必要があります。アップグレードが正常に完了した場合、pg_upgrade はアップグレードの完了時にログファイルを削除します。アップグレードログファイルを削除しないように pg_upgrade に指示するには、pg_upgrade を呼び出すときに --retain キーワードを含めます。

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、古いバックエンド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、新しいバックエンド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、古いバックエンドデータベースの `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、古いバックエンドデータベースの `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\11\bin\pg_upgrade.exe"  
  
-d "C:\Program Files\PostgreSQL\9.6\data"  
  
-D "C:\Program Files\PostgreSQL\11\data"  
  
-b "C:\Program Files\PostgreSQL\9.6\bin"  
  
-B "C:\Program Files\PostgreSQL\11\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 PostgreSQL11 に移行するように `pg_upgrade` に指示します（バックエンド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）。

呼び出されると、`pg_upgrade` はデータを新しいバックエンドデータベースに移動する前に一貫性チェックを実行します。アップグレードが完了すると、`pg_upgrade` はアップグレードが完了したことを通知します。

`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>

5. 次の証明書ファイルを古いバックエンドデータベースの data ディレクトリから新しいバックエンドデータベースの data ディレクトリにコピーします。

```
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。

Permissions and Ownership on Linux

ファイル名	所有者	許可
ca_certificate.crt	ポストGRES	-rw-----
ca_key.key	ポストGRES	-rw-----
root.crt	ポストGRES	-rw-----
root.crl	ポストGRES	-rw-----
server.key	ポストGRES	-rw-----
server.crt	ポストGRES	-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。

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。

Permissions and Ownership on Windows

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 サーバーとバックエンド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。「インストーラーのコンテキストメニューから選択した」Run as Administrator オプションを使用して PEM サーバーと Postgres インストーラーを起動した場合、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。

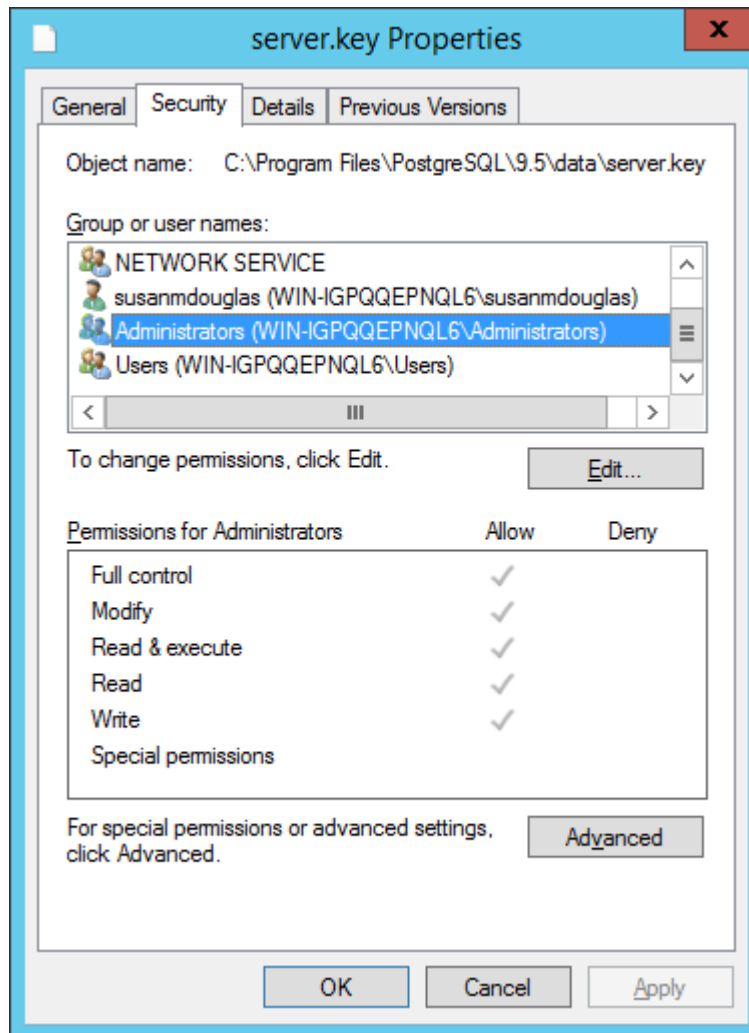


図1 [セキュリティ]タブ

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。

6. postgresql.conf ファイルには、サーバーの動作を指定するパラメーター設定が含まれています。

古いサーバーの `postgresql.conf` ファイルで指定された構成に一致するように、新しいサーバーの `postgresql.conf` ファイルを変更する必要があります。

デフォルトでは、`postgresql.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/<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/<X>/data`
- Linux に RPM をインストールした Postgres バージョン 10 以上の場合、`/usr/pgsql/<X>/data`
- Windows の Postgres バージョンの場合、`C:/Program Files/PostgreSQL/<X>/data`

ここで、`<X>` はシステム上の Postgres のバージョンです。

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。

- 元のバックエンドデータベースによって監視されるポート（通常は 5432）でリッスンする `port` パラメーター。
- `ssl` パラメータを `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は、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、新しいバックエンドデータベースが古いバックエンド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ます。

7. `pg_hba.conf` ファイルには、サーバーがホストベースの認証を強制する方法を指定するパラメーター設定が含まれています。PEM サーバーをインストールすると、インストーラーは `pg_hba.conf` ファイルを変更し、ファイルの先頭にエントリを追加します。

```
# Adding entries for PEM Agens 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 Agens 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/<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/<X>/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/<X>/data
- Linux 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<X>/data
- Windows の Postgres バージョンの場合、 C:/Program Files/PostgreSQL/<X>/data

ここで、<X> はシステム上の Postgres のバージョンです。

選択したエディターを使用して、古いサーバーの pg_hba.conf ファイルから新しいサーバーの pg_hba.conf ファイルにエントリをコピーします。

8. 新しいバックエンドデータベースのサービスを再起動します。

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>
```

service_name は、バックエンドデータベースサーバーの名前です。

Windows を使用している場合、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 メニューに移動します。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、ダイアログで提供されるオプションを使用してサービスを開始します。

第 3 章

Moving the Postgres Enterprise Manager Server

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）のバックエンドデータベースは、ソース PEM サーバーのバックエンド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可能性があります。PostgreSQL ホスト上にある PEM サーバーは、AdvancedServer ホストに移行できます。その逆も可能です。

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。

1. ターゲットホストを準備する

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。

ターゲットサーバーのバックエンドデータベースは、ソースのバックエンドデータベースとは異なるバージョンまたはタイプである場合があります。新しい PEM サーバーが元のサーバーと同じタイプのバックエンドデータベースに存在しない場合は、新しいサーバーホストに同じバージョンの `sslutils` 拡張機能がインストールされていることを確認する必要があります。PEM インストーラーと共に配布される `sslutils` のバージョンは、EDBWeb サイトから無料でダウンロードできます。

<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 Guides を参照してください。

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

2. 新しい PEM サーバーから既存のスキーマを削除する

移行プロセスにより、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 スキーマが再作成されます。移動の準備として、psql クライアントを使用して、ターゲットホスト上の pem データベースからこれらのスキーマを削除します。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 バックエンドデータベースに接続します。ターゲットホストの pem データベースに接続した後、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場合、CASCADE キーワードを含めて、すべての依存オブジェクトを削除するようサーバーに指示する必要があります。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 を表示して、各スキーマが削除されることを確認します。

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上の 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 サーバーにあ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 データベースと 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>;
```

<X> はエージェント番号を指定します。agent1 は、PEM サーバーインストーラーによってターゲットホスト上に作成されます。

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

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

<X> はエージェント番号を指定します。

4. ソース PEM サーバーのバックアップスクリプトを生成する

pg_dump ユーティリティを使用して、ターゲットホスト上で pem データベースを再作成するために必要なコマンドを含むスクリプトを生成できます。デフォルトでは、pg_dump は Postgres インストールの bin ディレクトリにインストールされます。pg_dump を呼び出すには、コマンドラインを開き、bin ディレクトリに移動して、次のように入力します。

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

どこで:

- <user_name> は、PEM バックエンド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 バックエンド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。スクリプトには backup.sql という名前が付けられ、tmp ディレクトリに作成されます。pg_dump は、ユーザー postgres の資格情報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。

5. バックアップをターゲットホストに移動する

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。

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 バックエンド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 を開始し、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、バックエンドデータベースの接続権限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。

この例では、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。

7. ターゲットホストでデータベースサーバーを停止します

CentOS または RHEL7.x または 8.x 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。

```
systemctl stop <service_name>
```

ここで、<service_name> はバックエンドデータベースサーバーの名前を指定します。PostgreSQL バックエンドデータベースの場合、サービス名は postgresql-<x> です。AdvancedServer バックエンドデータベースの場合、サービス名は edb-as-<X> です。<X> はバージョン番号を示します。

Windows を使用している場合、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 メニューに移動します。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、リストで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サービスを停止します。

8. 証明書ファイルをターゲットホストにコピーする

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エンドデータベースを提供する Postgres インストールの data ディレクトリにファイルをコピーします。

Linux では、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すでに存在します。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したら、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。

Permissions and Ownership on Linux

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

Linux では、証明書ファイルは postgres が所有する必要があります。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 サーバーとバックエンド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。「インストーラーのコンテキストメニューから選択した」Run as Administrator オプションを使用して PEM サーバーと Postgres インストーラーを起動した場合、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。

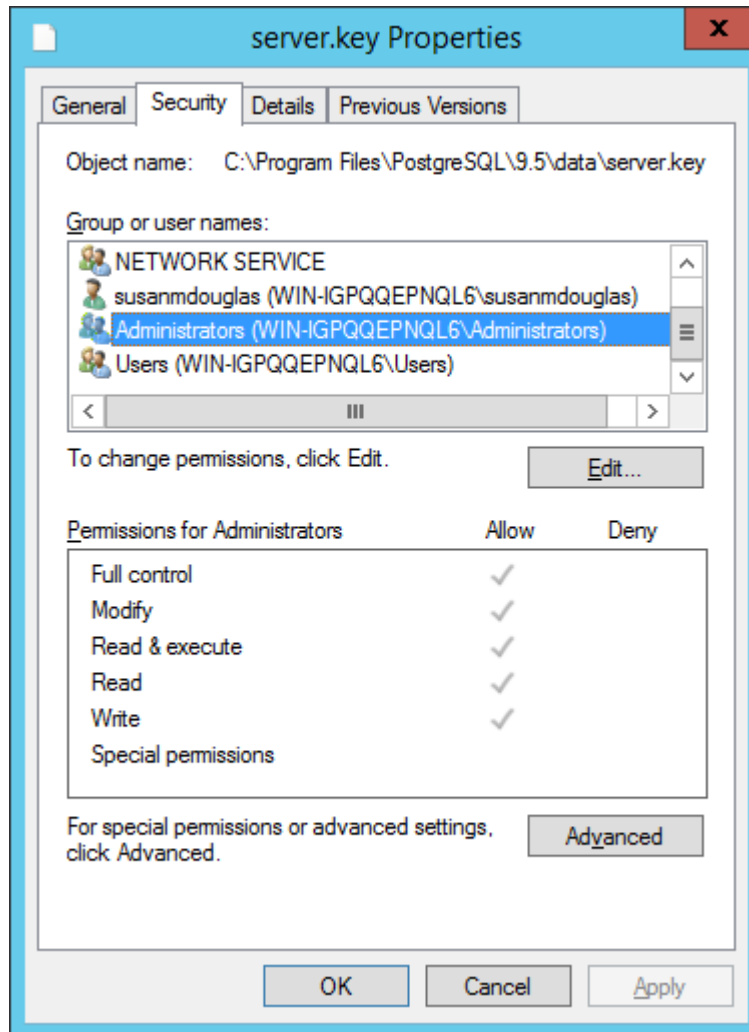


図1 [権限] タブ

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。

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する

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。このステップはプラットフォーム固有です。

On Linux

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上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 ファイルの権限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 ファイルの権限と所有権を変更します。

```
chmod 644 agent1.crt

chown root agent1.crt
```

On Windows

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 インストーラーを起動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。「インストーラーのコンテキストメニューから選択した」Run as Administrator オプションを使用して PEM サーバーと Postgres インストーラーを起動した場合、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。

10. ターゲットホスト上の pg_hba.conf ファイルを更新します

ターゲットホスト上の pg_hba.conf ファイルを変更して、各 PEM エージェントからの接続を許可します。デフォルトでは、pg_hba.conf ファイルは Postgres インストールの下のデータディレクトリにあります。

11. ターゲットホストでサーバーを起動します

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
```

service_name は、バックエンドデータベースサーバーの名前です。

Windows を使用している場合、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 メニューに移動します。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サービスを開始します。

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 エージェントを備えた各 Windows ホスト）またはエージェント構成ファイル（PEM エージェントを備えた各 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：
 - Linux<pem_agent_restart_on_linux>の場合
 - *Windows* の場合

If the PEM Agent Resides on Linux

選択したエディターを使用して 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。

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

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。

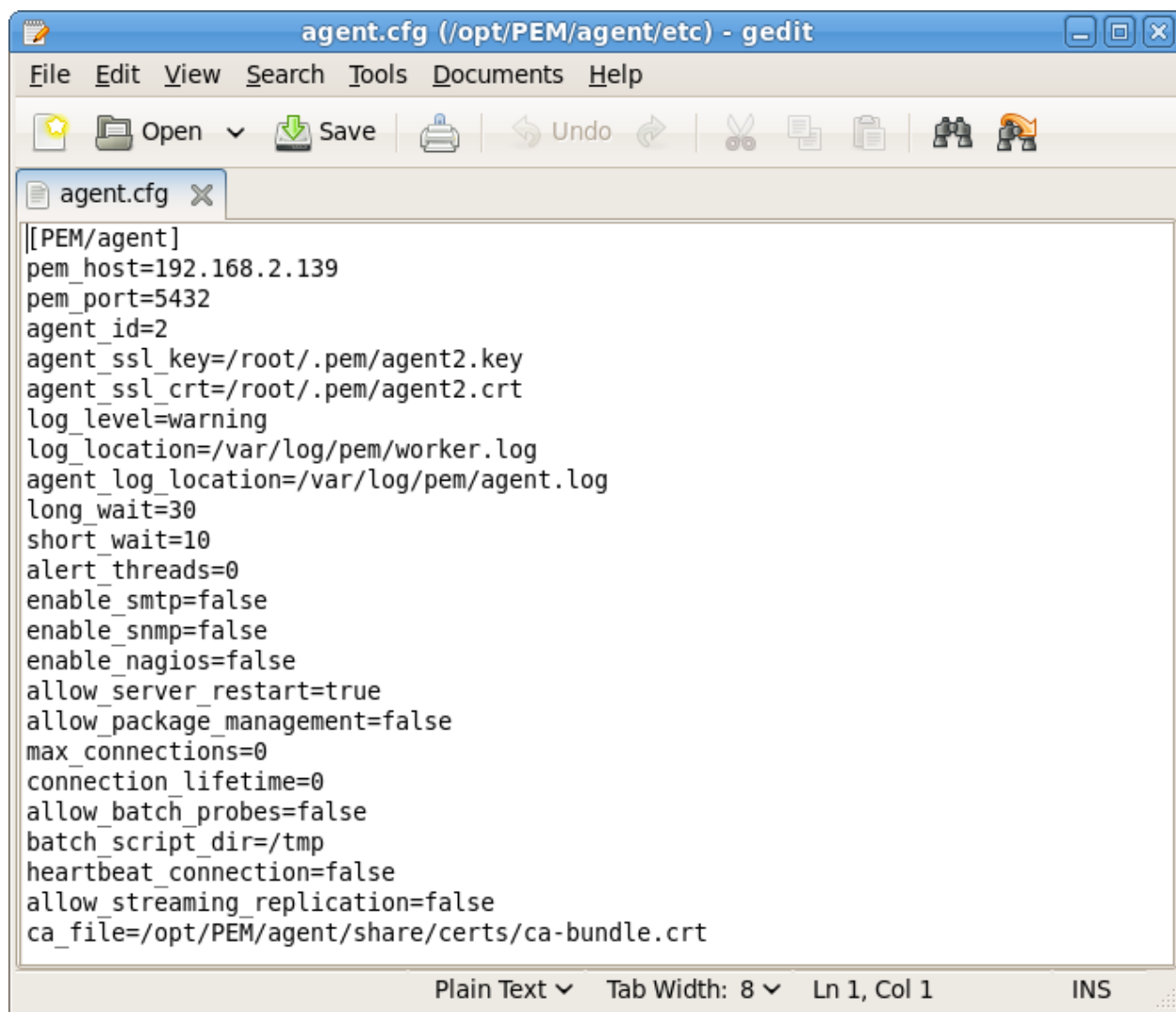


図2 agent.cfg ファイル

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

If the PEM Agent Resides on Windows

監視対象ノードの Windows レジストリを変更する前に、PEM エージェントのホスト上のファイアウォールが PEM サーバーへの接続を許可することを確認してください。PEM エージェントホストが PEM サーバーホストに接続できることを確認したら、Windows の Registry Editor を使用して PEM_HOST および PEM_PORT エントリを確認および編集し、PEM が使用するホストとポートを正しく識別できることを確認できますサーバー。Registry Editor を開くには、Windows の Run ダイアログまたは Windows のスタートメニューの検索ボックスに regedit と入力します。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

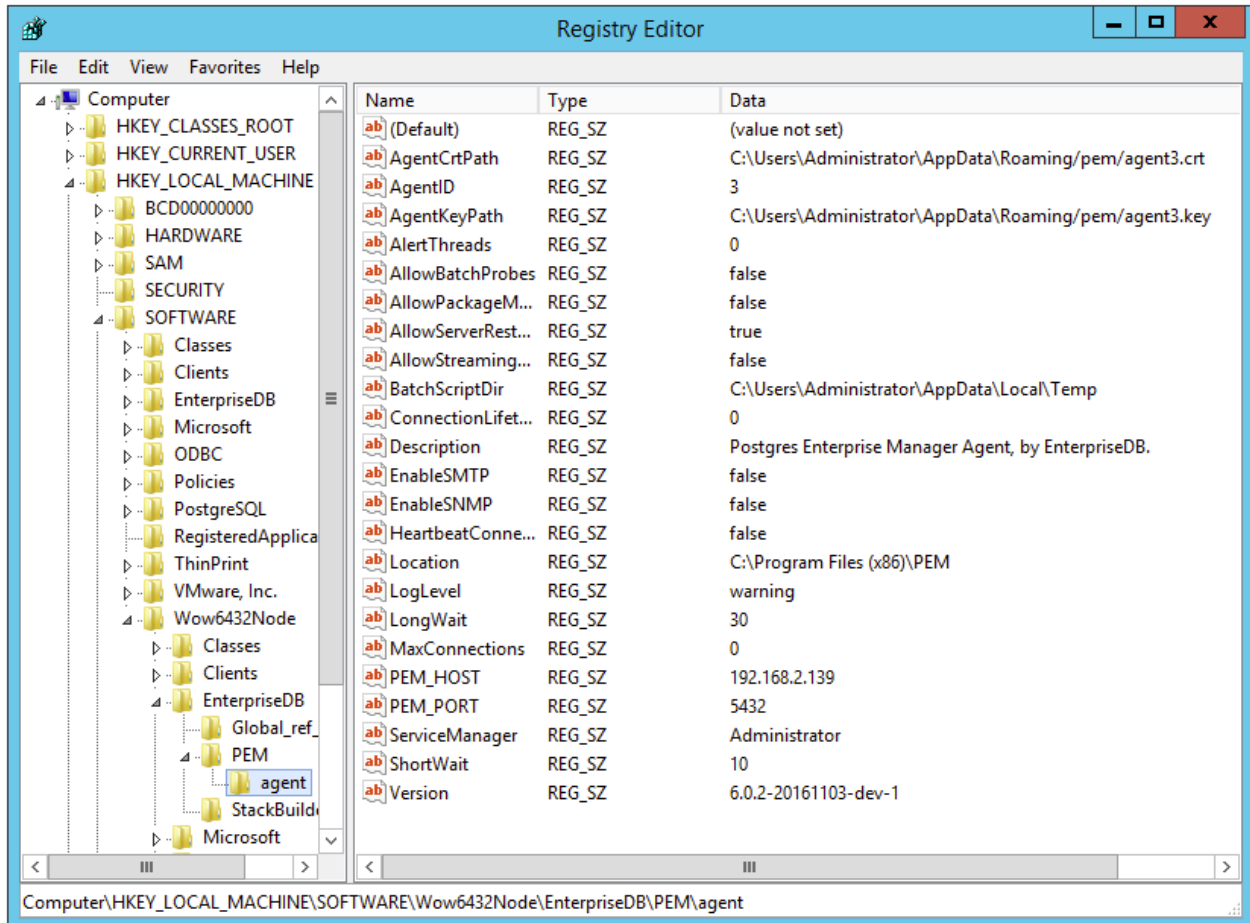


図3 Windows レジストリエディター

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。

Edit String ダイアログを使用して、エントリの値を変更します。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 の 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。

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。

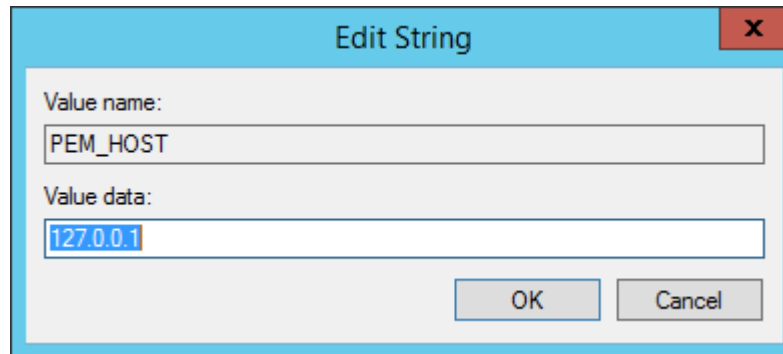


図 4 Windows レジストリエディター

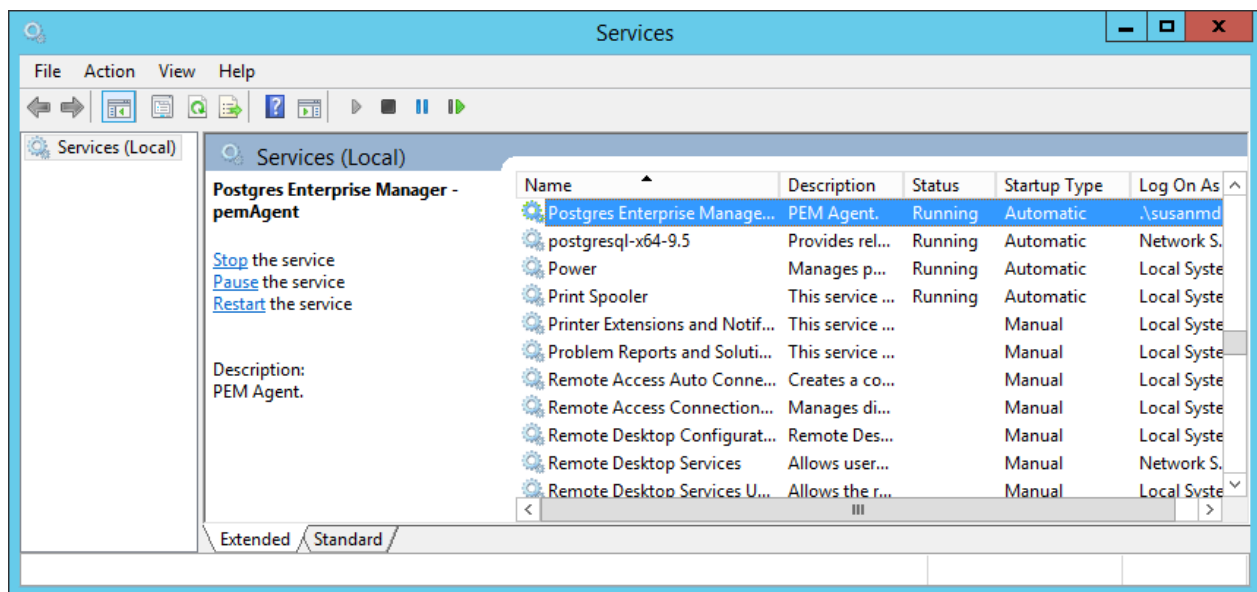


図 5 PEM エージェントのサービスの再起動

第 4 章

Troubleshooting

4.1 PEM サーバーの再構成

状況によっては、PEM サーバーをアンインストールし、再度インストールしてから、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使用します。

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
```


第 5 章

Conclusion

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

- EDB および PostgresEnterpriseManager は、EnterpriseDBCorporation の登録商標です。EDB および EDB-Postgres は、EnterpriseDBCorporation の商標です。Oracle は、Oracle、Inc の登録商標です。その他の商標は、それぞれの所有者の商標です。
- EDB は、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危険な入力を禁止することはできません。
- EDB は、将来、認証された信頼できるユーザーからのフリーフォーム SQL、WMI またはその他の潜在的に危険な入力を受け入れる製品に機能を追加する権利を留保しますが、そのような機能はすべて、可能な限り最小のリスクを提供するように設計およびテストされます可能であれば、スーパーユーザーまたは同等の特権が必要です。
- EDB は、すべての潜在的な脅威を予測できることを保証しません。したがって、すべての潜在的な脆弱性が対処または検討されたことをプロセスが完全に保証することはできません。

索引

Conclusion, [67](#)
Configuring a PEM Server on a Linux Host, [41](#)
Creating a PEM Repository on an Isolated
Network, [28](#)

Moving the Postgres Enterprise Manager Server,
[53](#)

reconfiguring an agent, [65](#)

Troubleshooting, [65](#)

Upgrading a Graphical PEM Installation on a
Linux Host, [29](#)
Upgrading a PEM Installation, [3](#)
Upgrading a PEM Installation on Windows Host, [3](#)
Upgrading a PEM Native Package Installation on
a Linux Host, [21](#)
Upgrading a PEM Server Installed with a
Graphical Installer, [34](#)
Upgrading PEM Agent installed with graphical
installer, [30](#)
Upgrading the PEM Backend Postgres Database, [43](#)