



Postgres Enterprise Manager

リリース 7.16

Upgrade and Migration Guide

2020 年 10 月 10 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	3
1.1.2	Windows ホストでの PEM サーバーのアップグレード	9
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	Linux ホストでの PEM ネイティブパッケージインストールのアップグレード	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	22
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	22
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	23
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	23
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	23
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	23
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	24
1.2.3.3	SLES ホスト上の PEM サーバーのアップグレード	24
1.2.4	SQLProfilerNativePackage インストールのアップグレード	25
1.3	隔離されたネットワークでの PEM リポジトリの作成	27
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	28
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	28
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	29
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	33
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	33
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	37
1.5	Linux ホストでの PEM サーバーの設定	39
第 2 章	PEM バックエンド Postgres データベースのアップグレード	41

第 3 章	PostgresEnterpriseManager サーバーの移動	51
第 4 章	トラブルシューティング	62
4.1	PEM サーバーの再構成	62
第 5 章	結論	64
索引		65

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、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。
- **バックエンド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。
- **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 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呼び出すだけです。

1. 監視対象ノード except の PEM サーバーホストで 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、歓迎されます。

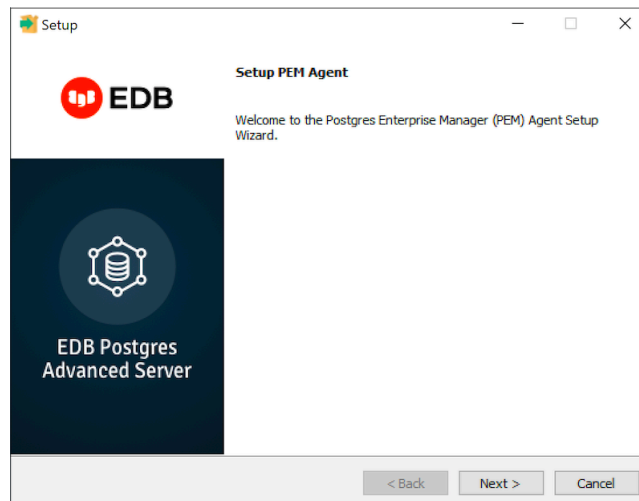


図 1 PEM エージェントインストーラー-ようこそウィンドウ

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。

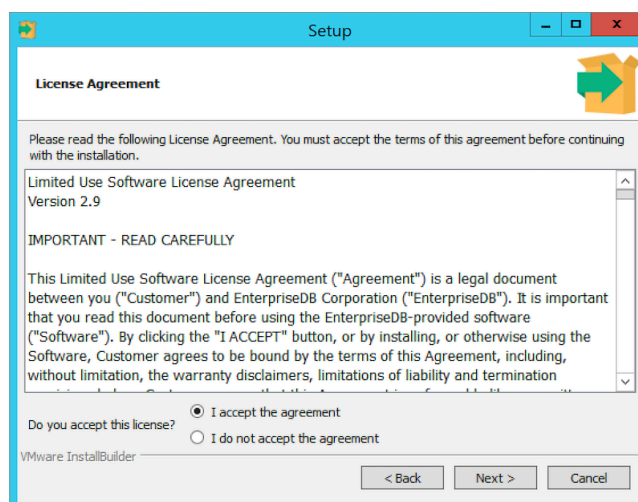


図 2 PEM エージェントインストーラー-ライセンス契約

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。

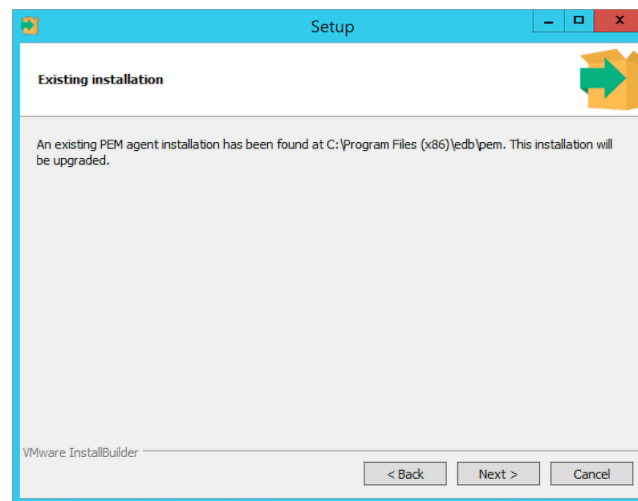


図 3 PEM エージェントインストーラー-既存のインストール場所

pemAgent service account ダイアログは、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。

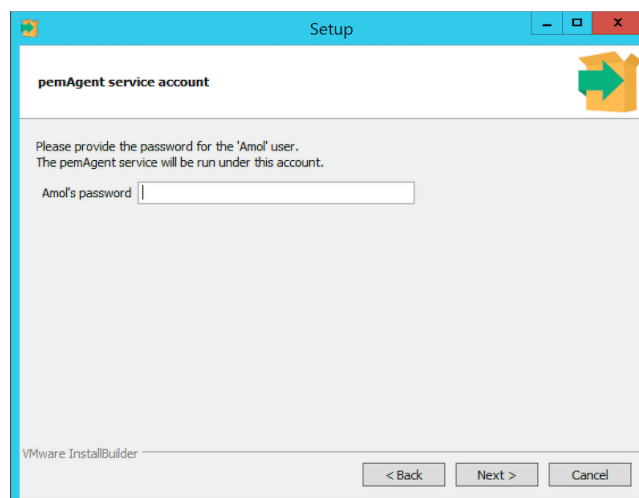


図 4 PEM エージェントインストーラー-pemAgent サービスアカウントのパスワード

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。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。

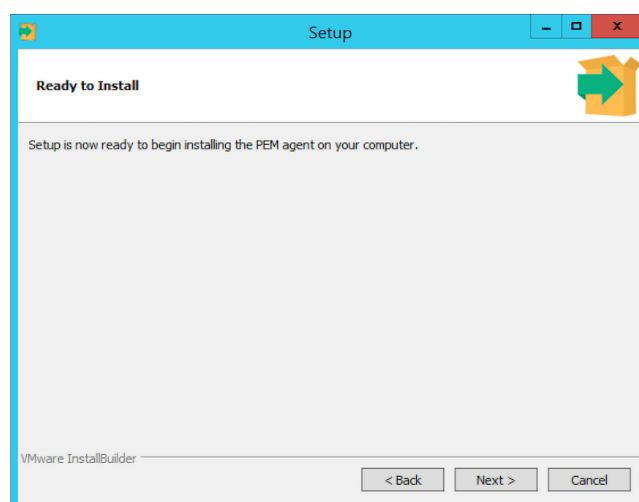


図 5 PEM エージェントインストーラー-インストール準備完了

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。

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。

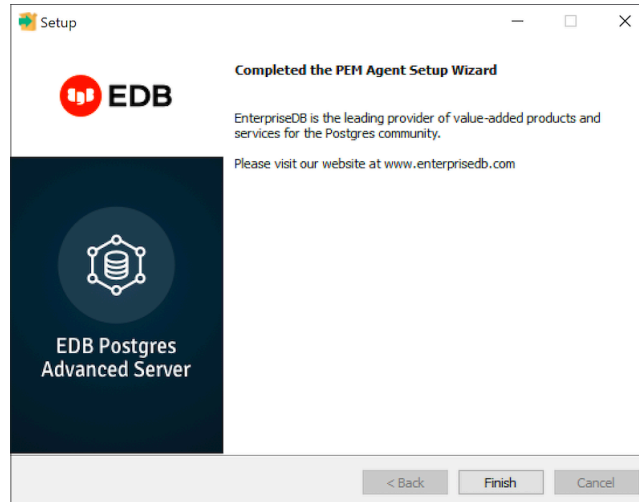


図 6 PEM エージェントインストーラー-完了

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。

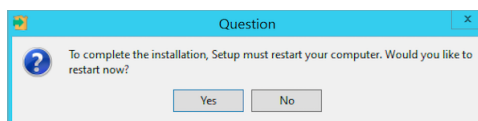


図 7 PEM エージェントインストーラー-再起動ポップアップ

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すでに構成している、または構成する予定がある場合は、AllowBatchJobSteps パラメーターを True に設定する必要があります agent.cfg ファイル内。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。

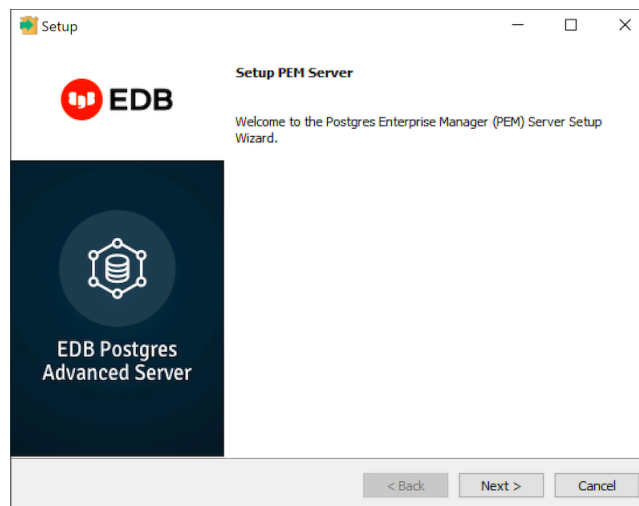


図 8 PEM サーバーインストーラー-ようこそウィンドウ

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。

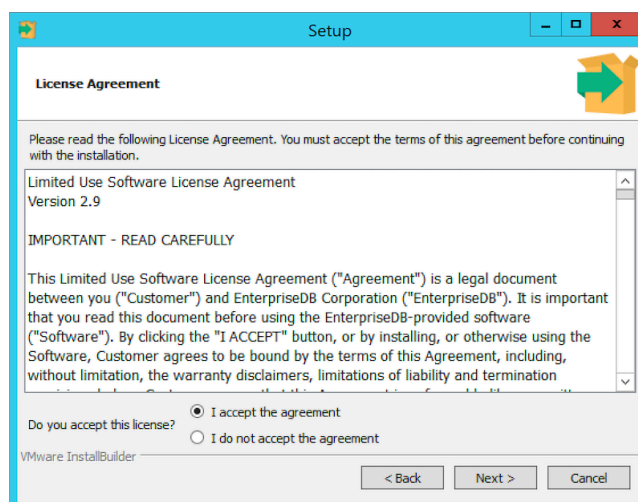


図 9 PEM サーバーインストーラー-ライセンス契約

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。

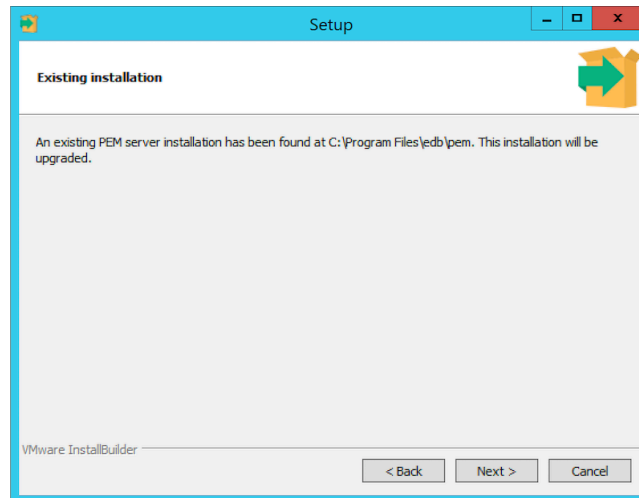


図 10 PEM サーバーインストーラー-既存の PEM サーバーの場所

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。

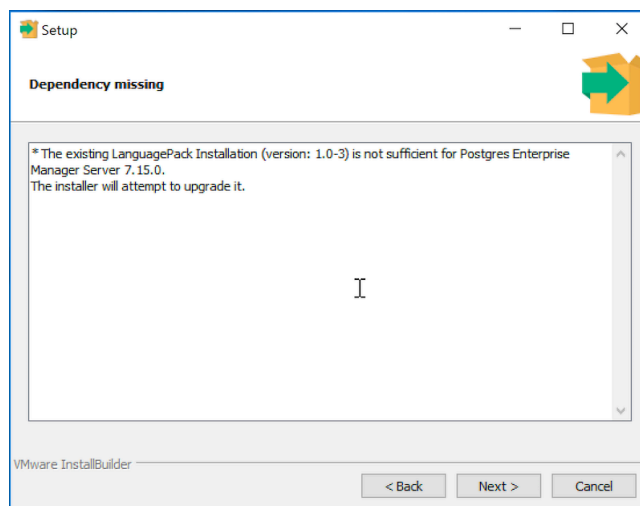


図 11 PEM サーバーインストール-不足している依存関係

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。

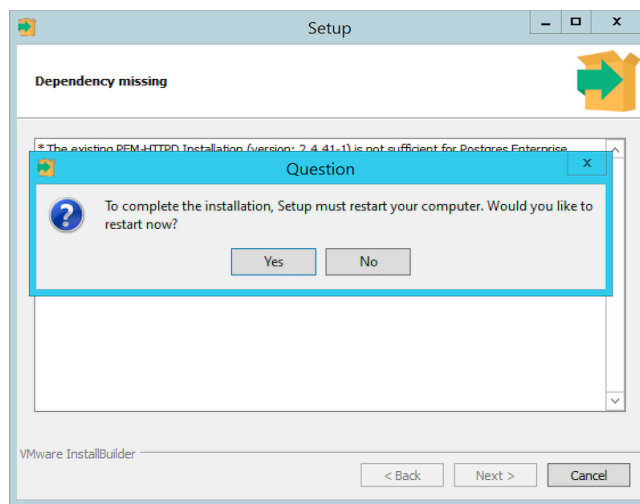


図 12 PEM サーバーインストーラー-再起動ポップアップ

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を要求します。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。

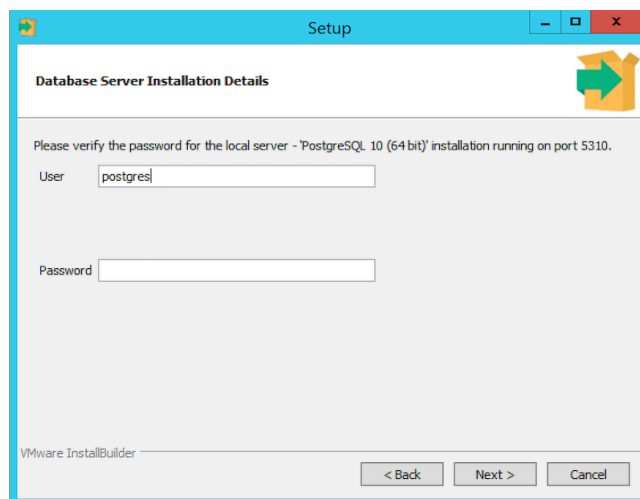


図 13 PEM サーバーインストーラー-データベースサーバーのインストールの詳細

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。

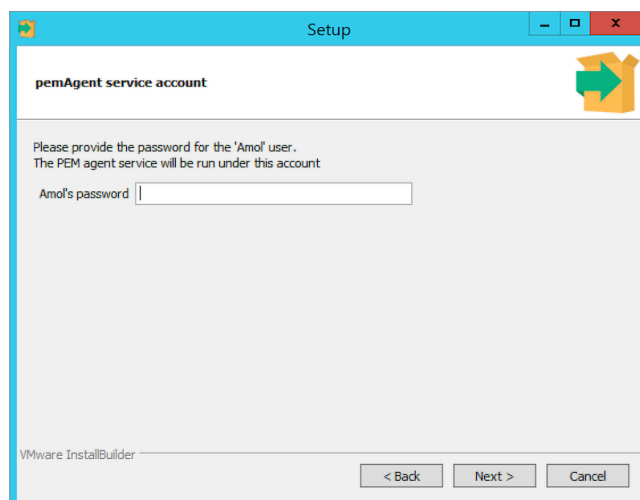


図 14 PEM サーバーインストーラー-pemAgent サービスアカウントのパスワード

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。

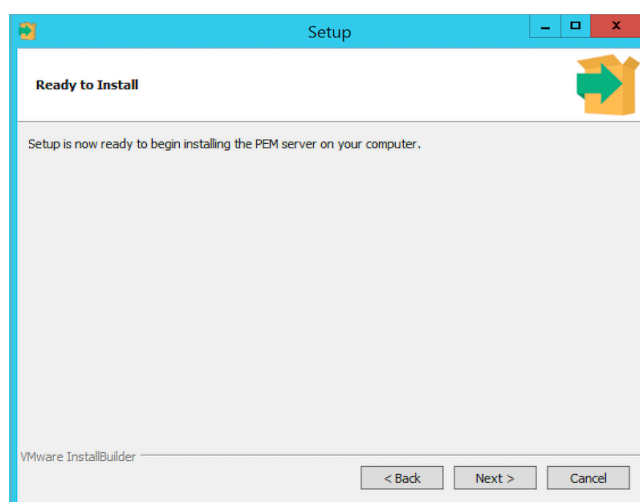


図 15 PEM サーバーインストーラー-インストール準備完了

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。

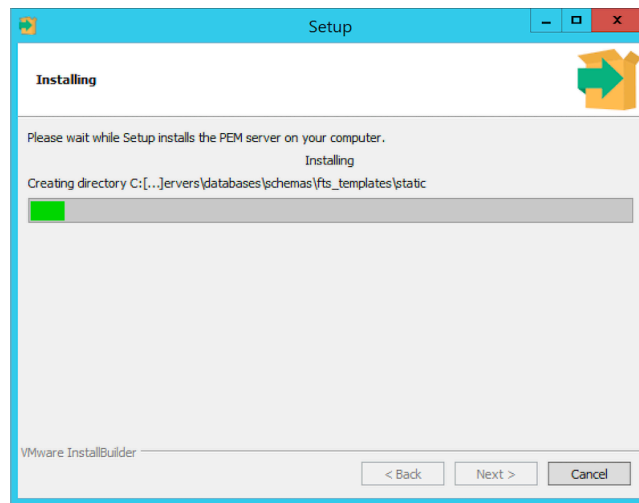


図 16 進行中の PEM サーバーのインストール

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。

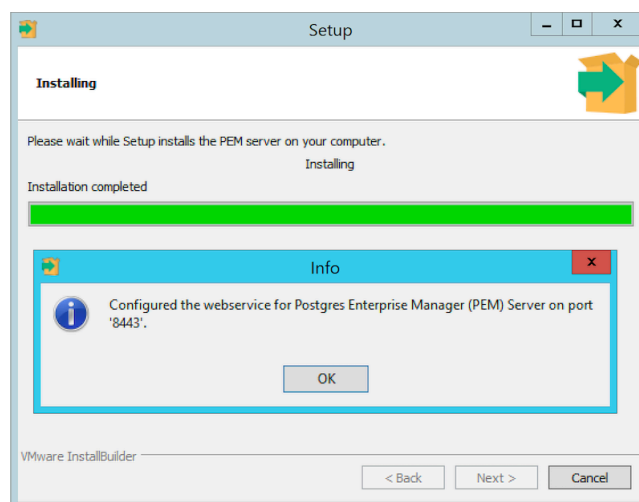


図 17 PEM サーバーインストーラー-設定された PEMWeb サービス

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。

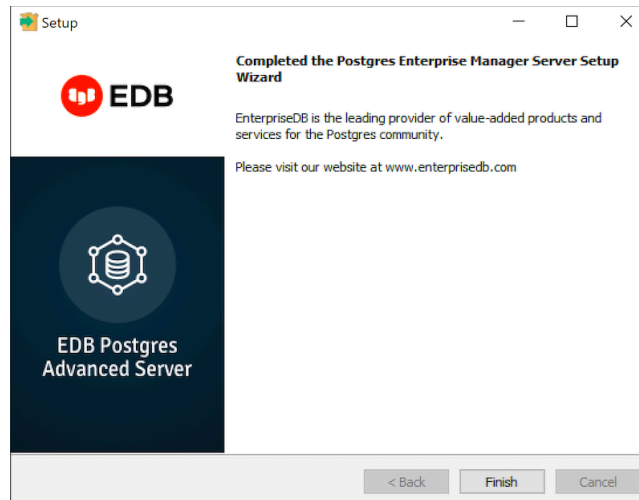


図 18 PEM サーバーインストーラー-アップグレードの完了

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、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>
```

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。

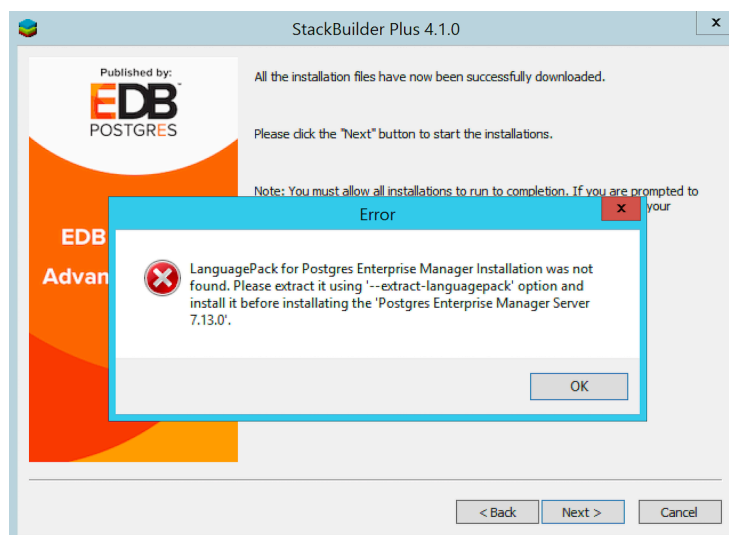


図 19 PEM サーバーのアップグレードエラー

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、:ref:`バックエンド Postgres データベースのアップグレード<upgrading\_backend\_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
```

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、`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 -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
```

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 Linux ホストでの PEM ネイティブパッケージインストールのアップグレード

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: If you have already configured or are planning to configure any shell/batch script run by a Linux agent that is upgraded from any lower version to 7.11 or later version, the user for the `batch_script_user` parameter must be specified in the `agent.cfg` file. It is strongly recommended that a non-root user is used to run the scripts. Using the root user may result in compromising the data security and operating system security. However, if you want to restore the `pemagent` to its original settings using a root user to run the scripts, then the `batch_script_user` parameter value must be set to `root`.

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

If you are using a version of PostgreSQL or Advanced Server older than 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 platform-specific commands to install the `libs` version 10 or above on your host:

CentOS または RHEL ホストで PEM インストールをアップグレードするための前提条件

AdvancedServer の場合:

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

PostgreSQL の場合:

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

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

Debian または Ubuntu ホストで PEM インストールをアップグレードするための前提条件

AdvancedServer の場合:

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

PostgreSQL の場合:

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

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

SLES 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

AdvancedServer の場合:

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

PostgreSQL の場合:

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

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

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 RHEL6.x、7.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、[:ref:分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。

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 RHEL6.x、7.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、[:ref:Linux プラットフォームでの PEM サーバーの設定<configuring_the_pem_server_on_linux_platforms>](#)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 Debian または Ubuntu のバージョンで PEM サーバーのインストール済み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、[:ref:Linux プラットフォームでの PEM サーバーの設定<configuring_the_pem_server_on_linux_platforms>](#)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、[:ref:Linux プラットフォームでの PEM サーバーの設定<configuring_the_pem_server_on_linux_platforms>](#)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 インストールの下での `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
```

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 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。

```

[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、バックエンドデータベースを更新する準備ができました。

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*
```

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。

2. 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。
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 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

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 より古いバージョンの Postgres 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。

AdvancedServer の場合:

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

PostgreSQL の場合:

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

ここで、<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
```

CentOS または RHEL6.x ホストの場合:

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

2. 次のコマンドのいずれかを実行して、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
```

ここで、`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かもしれません。

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 を有効にすることで、リポジトリにアクセスできます。証明書ベースのサブスクリプションを持っている場合、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か、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設定します:

- 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%202004-09-2019.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。

- 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置き換えられるようにします登録済みの EDB 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
```

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、“[: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。

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 RHEL6.x、7.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 RHEL6.x、7.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** エージェント設定ファイル（`agent.cfg`）を以前の場所から **RPM** インストーラー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
```

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
```

CentOS または **RHEL6.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` や `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
```

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 (agent2 や 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
```

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、または構成する予定の場合は、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 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 **PEM-HTTPD** 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.x0r8.x ホストの場合:

```
systemctl stop httpd
```

CentOS または RHEL6.x ホストの場合:

```
/etc/init.d/httpd stop
```

3. 次のコマンドのいずれかを実行して、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
```

ここで、`<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かもしれません。

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** を有効にすることで、リポジトリにアクセスできます。証明書ベースのサブスクリプションを持っている場合、**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か、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。

- 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%2004-09-2019.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。

- 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置き換えられるようにします登録済みの EDB 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
```

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、“[: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。

5. `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 RHEL6.x、7.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 RHEL6.x、7.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. 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
```

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
```

(次のページに続く)

(前のページからの続き)

```
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. 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
```

14. 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
```

15. 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
```

CentOS または RHEL6.x ホストの場合:

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

16. 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。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。

1.4.4 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:

```
/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-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、バックエンドデータベースを更新する準備ができました。

1.5 Linux ホストでの PEM サーバーの設定

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 <code>/root/.pem</code> です。
-ci	サーバーの <code>pg_hba.conf</code> 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、 <code>192.168.1.0/24</code> 。デフォルトは <code>0.0.0.0/0</code> 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 <code>/usr/edb/as10</code> 、PostgreSQL の場合は <code>/usr/pgsql-10</code> 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 <code>edb-as-10</code> です。PostgreSQL の場合、 <code>postgresql-10</code> 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 <code>pemagent</code> 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
```

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*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

PEM バックエンド Postgres 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、[: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 backend server to a more recent version. `pg_upgrade` facilitates migration between any version of PostgreSQL (version 9.5 or later), and any subsequent release of PostgreSQL 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 backend database to a PostgreSQL 10 backend database, but not to an Advanced Server 10 backend database. If you wish to migrate to a different type of backend 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>

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設定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。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>

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

Linux で

- 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。
- PostgreSQL バックエンドデータベース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>
```

ここで、`<X>` はサーバーのバージョンです。

Please note that Debian 10 and Ubuntu 20 has increased the requirements to accept the certificates due to security reason. If a user wants to install the PEM Agent on any of the machines, they must upgrade `sslutils` to 1.3 where 4096 bit RSA key and sha256 signature algorithm support has added. If the user does not upgrade `sslutils` to 1.3, then PEM Agent may fail to connect to the PEM backend database server, and it might log the error `ca md too weak`.

Windows で

`sslutils` must be compiled on the new backend database with the same compiler that was used to compile `sslutils` on the original backend database. If you are moving to a Postgres database that was installed using a PostgreSQL one-click installer (from EDB) 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、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ー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、必要な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 OPENSSL_PATH=C:\OpenSSL-Win32

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

SET PG_PATH="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--unpackaged--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\  
  
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。

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.

4. Use the `pg_upgrade` utility to perform an in-place transfer of existing data between the old backend database and the new backend 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つ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\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）。

Once invoked, pg_upgrade will perform consistency checks before moving the data to the new backend 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>

5. Copy the following certificate files from the data directory of the old backend database to the data directory of the new backend 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、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。

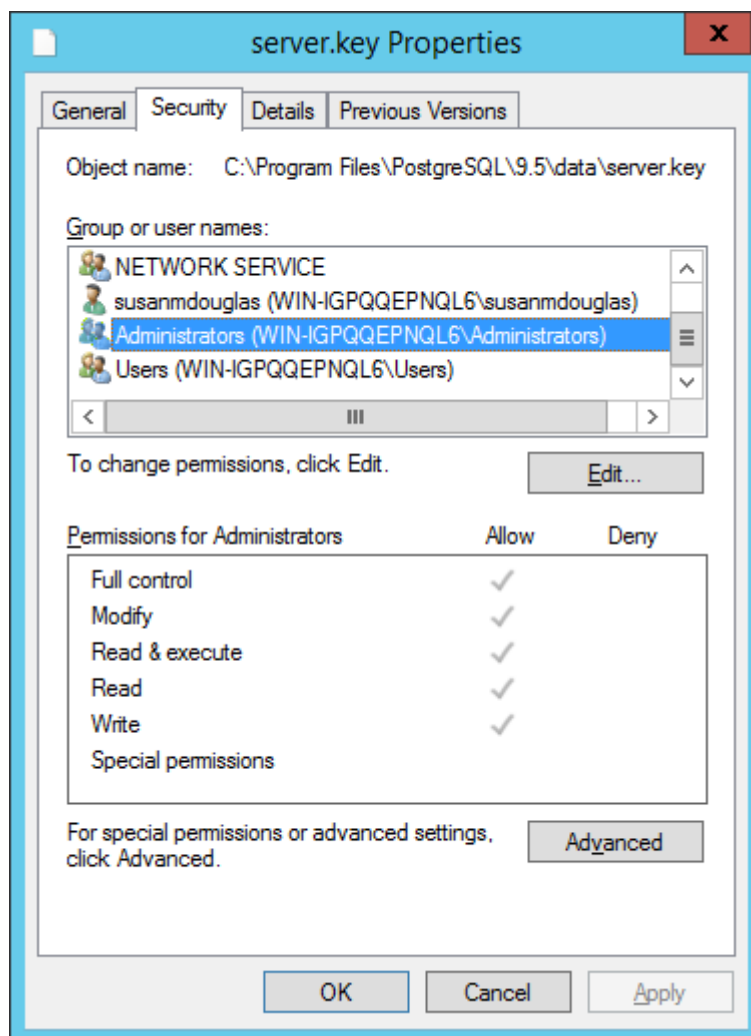


図 1 /セキュリティ/タブ

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. 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場合の 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。

- 元のバックエンド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。

7. 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 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/<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.

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 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` はバックエンド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.

第 3

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りますことに注意してください。

The backend database of the target server may be a different version or type than the backend database of the source. If the new PEM server does *not* reside on the same type of backend 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 EDB 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 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、ターゲットホストの 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。

When the psql client opens, connect to the pem backend database as the database superuser. After connecting to the pem database on the target host, use the following commands to drop the schemas:

```
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。

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 サーバーインストーラーによってターゲットホストに 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 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。

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

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。

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、(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の) `bin` ディレクトリに移動します。 `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、データベースのスーパーユーザー 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 RHEL6.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。

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

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> であり、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。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.

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ーのバックエンドデータベースを提供する Postgres インストール下の 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。

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。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。

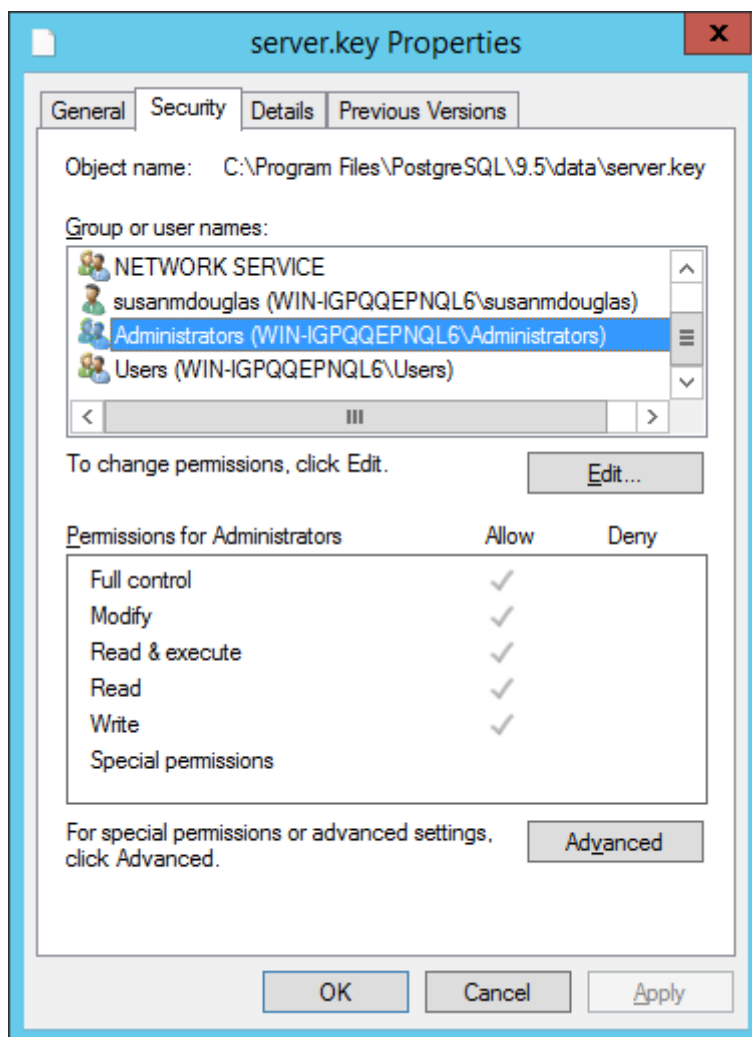


図 1 [権限] タブ

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。

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。

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
```

ここで、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。

10. ターゲットホストの 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.

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
```

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.

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
 - :ref:`Windows<`pem_agent_restart_on_windows`>`の場合

PEM エージェントが Linux 上にある場合

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。

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

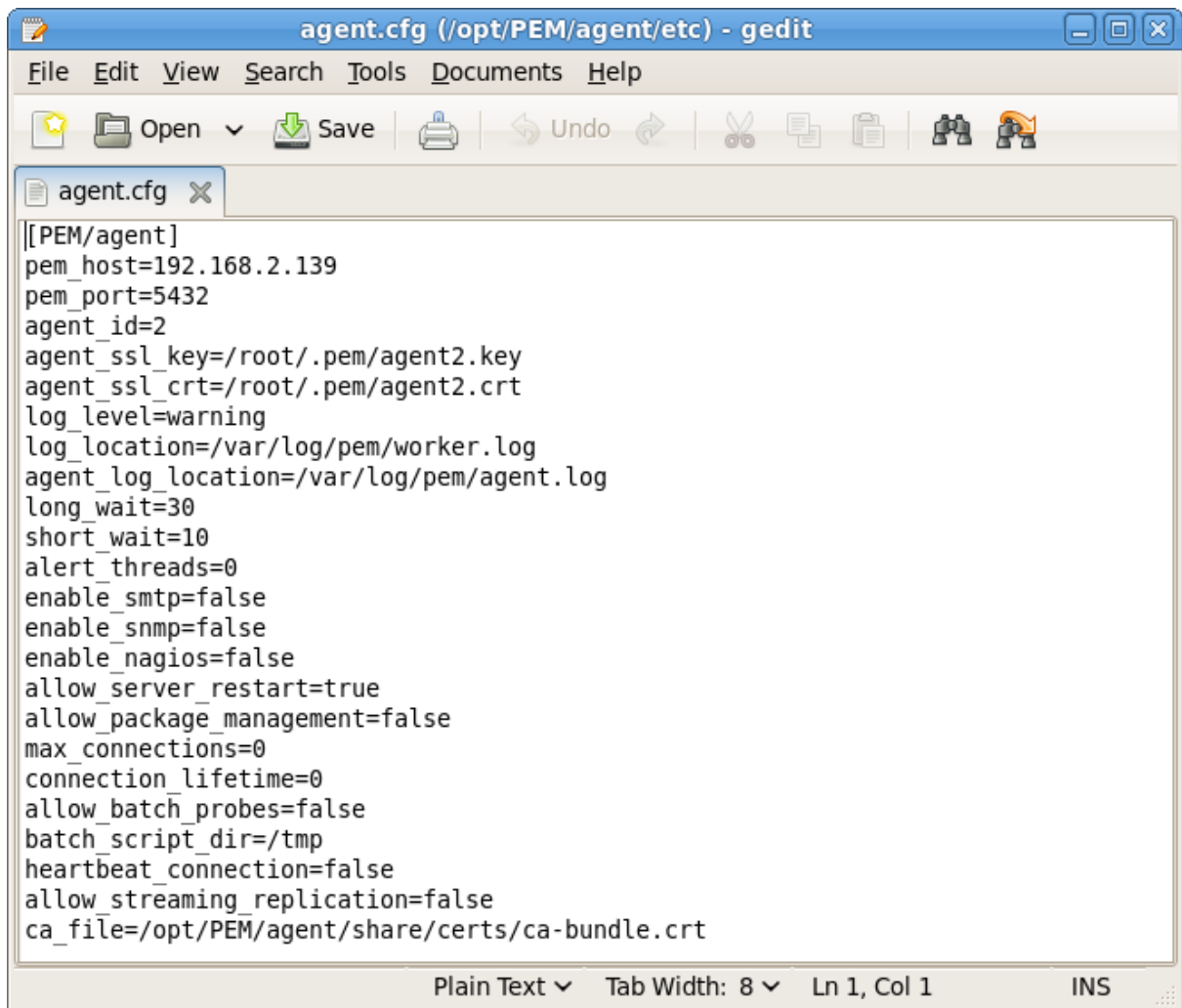


図 2 agent.cfg ファイル

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

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 のエントリを確認および編集し、ホストを正しく識別することを確認できますおよび PEM サーバーが使用するポート。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

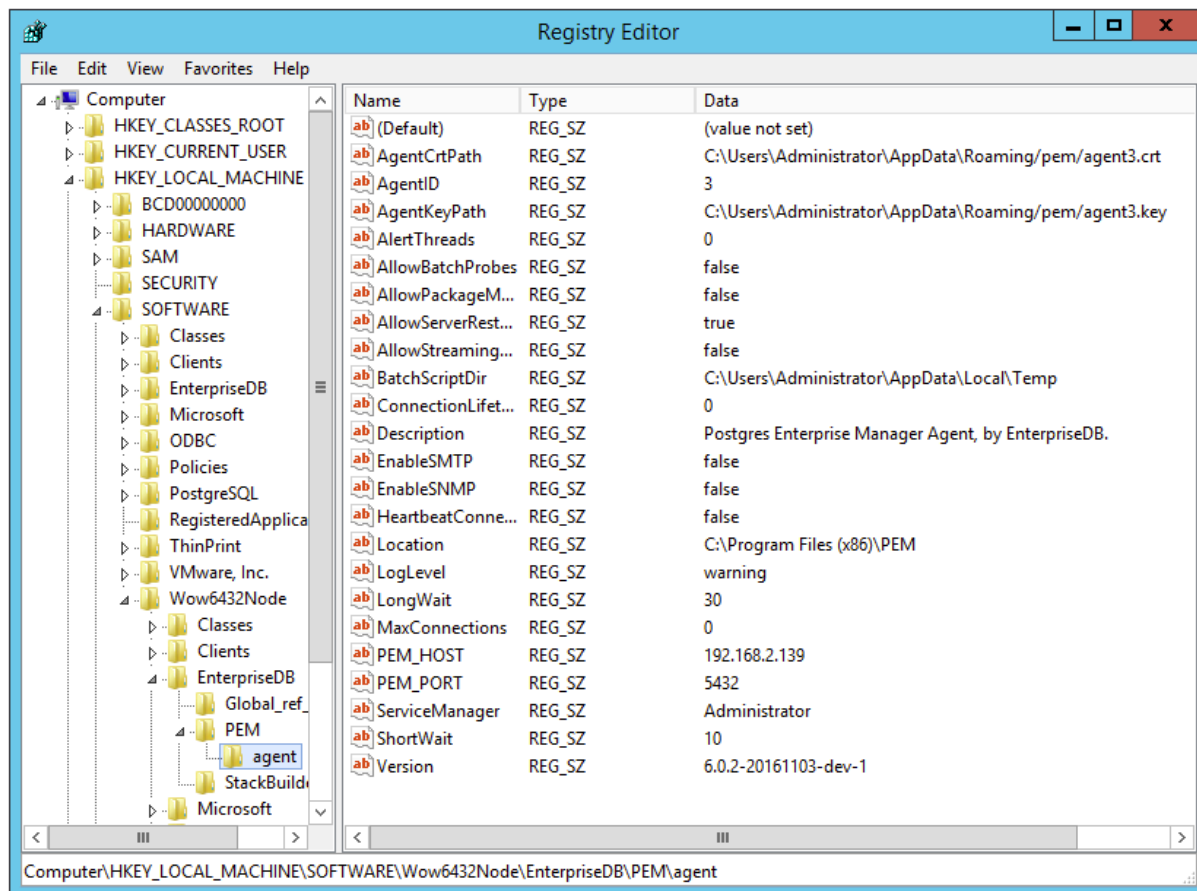


図 3 Windows レジストリエディター

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。

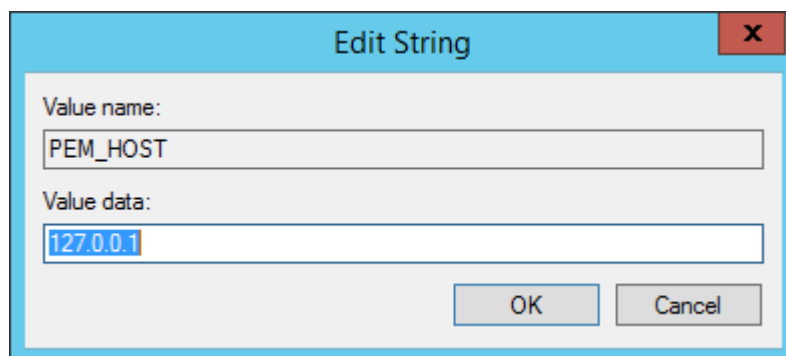


図 4 Windows レジストリエディター

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 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。

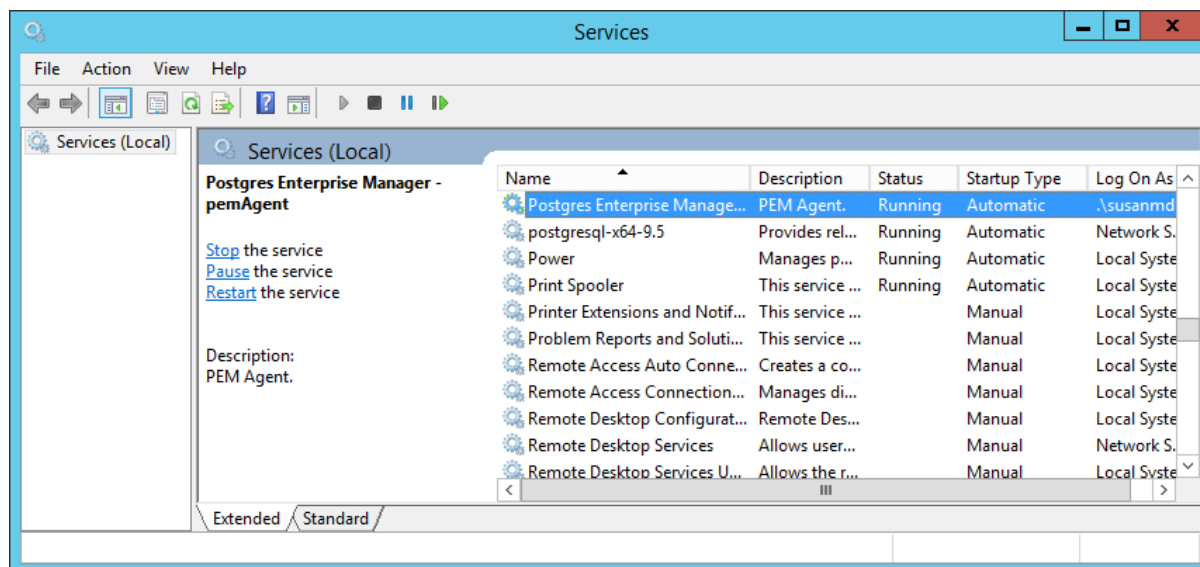


図 5 PEM エージェントのサービスの再起動

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。

トラブルシューティング

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
```

結論

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 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

Linux ホストでの PEM サーバーの設定, [39](#)
Linux ホストでの PEM ネイティブパッケージインストールの
アップグレード, [21](#)
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, [28](#)

PEM インストールのアップグレード, [2](#)
PEM バックエンド Postgres データベースのアップグレード,
[41](#)
PostgresEnterpriseManager サーバーの移動, [51](#)

reconfiguring an agent, [62](#)

Upgrading a PEM Server Installed with a
Graphical Installer, [33](#)
Upgrading PEM Agent installed with graphical
installer, [28](#)

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, [62](#)

隔離されたネットワークでの PEM リポジトリの作成, [27](#)

結論, [64](#)