



Postgres Enterprise Manager

リリース 7.13

Agent User Guide

2020 年 03 月 01 日

目次

第 1 章	Postgres Enterprise Manager-概要	2
1.1	PEM アーキテクチャ	2
第 2 章	PEM エージェントのインストール	4
2.1	Windows ホストへのエージェントのインストール	5
2.2	コマンドラインからグラフィカルインストーラーを呼び出す	15
2.2.1	無人モードでのグラフィカルインストーラーの呼び出し	15
2.3	RHEL または CentOS ホストへのエージェントのインストール	17
2.4	SLES ホストへのエージェントのインストール	20
2.5	Debian または Ubuntu ホストへのエージェントのインストール	21
第 3 章	エージェントの登録	22
3.1	PEM エージェント構成パラメーターの設定	26
3.2	非ルートユーザーアカウントを使用して PEM エージェントを登録する	27
第 4 章	PEM エージェントの管理	29
4.1	エージェント権限	29
4.2	エージェント構成	32
4.3	エージェントのプロパティ	39
第 5 章	PEM エージェントのトラブルシューティング	41
5.1	削除された PEM エージェントの復元	41
5.2	PEM サーバーの再構成	43

5.3	コマンドラインを使用して、ダウンまたは不明なステータスの PEM エージェントを削除する	44
第 6 章	PEM エージェントのアンインストール	45
第 7 章	結論	46
索引		47

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

PEM でサポートされているプラットフォームとバージョンについては、次の EnterpriseDB Web サイトにアクセスしてください。

<https://www.enterprisedb.com/services-support/edb-supported-products-and-platforms#pem>

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

- **Postgres Enterprise Manager-概要**-このセクションでは、PEM アーキテクチャの概要を説明し、PEM エージェントをインストールするためのハードウェアおよびソフトウェアの前提条件に関する情報も提供します。
- **PEM エージェントの登録**-このセクションでは、PEM エージェントの登録に関する情報を提供します。
- **PEM エージェントの管理**-このセクションでは、PEM エージェントの構成と管理に関する情報を提供します。
- **PEM エージェントのトラブルシューティング**-このセクションでは、PEM エージェントのトラブルシューティングに関する情報を提供します。
- **PEM エージェントのアンインストール**-このセクションでは、PEM エージェントのアンインストールに関する情報を提供します。

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

Postgres Enterprise Manager-概要

1.1 PEM アーキテクチャ

Postgres Enterprise Manager (PEM) は、PEM の管理および分析機能を提供するコンポーネントで構成されています。

- **PEM サーバー**: PEM サーバーは、データを監視するためのデータリポジトリとして、およびエージェントとクライアントの両方が接続するサーバーとして使用されます。 PEM サーバーは、PostgreSQL のインスタンス、監視データの保存用の関連データベース、および Web サービスを提供するサーバーで構成されています。
- **PEM Web インターフェイス**: PEM Web インターフェイスを使用すると、Postgres サーバーを管理および監視し、PEM 拡張機能を利用できます。 Web インターフェイスソフトウェアは PEM サーバーインストーラーでインストールされ、選択した Web ブラウザーからアクセスします。
- **PEM エージェント**: PEM エージェントは、タスクの実行と、エージェントホストおよび監視対象の Postgres インスタンスから PEM サーバーへの統計のレポートを担当します。単一の PEM エージェントは、1 つ以上のホストに存在する Postgres の複数のインストール済みインスタンスを監視できます。
- **SQL Profiler プラグイン**: Postgres サーバーへのこのプラグインは、SQL Profiler ツールで使用

される監視データを生成するために使用されます。SQL Profiler プラグインのインストールはオプションですが、プラグインはプロファイルする Postgres の各インスタンスにインストールする必要があります。SQL Profiler は、PostgreSQL サーバーまたは Advanced Server (PEM サーバーで管理されているものだけでなく) の EnterpriseDB ディストリビューションのサポートされているバージョンで使用できます。

PEM エージェントインストーラーは、PEM ワーカー (`pemworker.exe`) と PEM エージェント (`pemagent.exe`) の 2 つの実行可能ファイルを作成します。各 PEM ワーカーには、PEM ワーカーの起動または停止に使用できる対応する PEM エージェントがあります。PEM エージェントは、予期せず終了した場合にも PEM ワーカーを再起動します。PEM ワーカーのログファイルには、PEM ワーカーのアクティビティ (プローブアクティビティ、ハートビート応答など) に関連する情報が含まれ、`/var/log/pem/worker.log` に保存されます。

次のアーキテクチャ図は、一般的な PEM インストールに関係するさまざまなサーバーとワークステーションの関係を示しています。

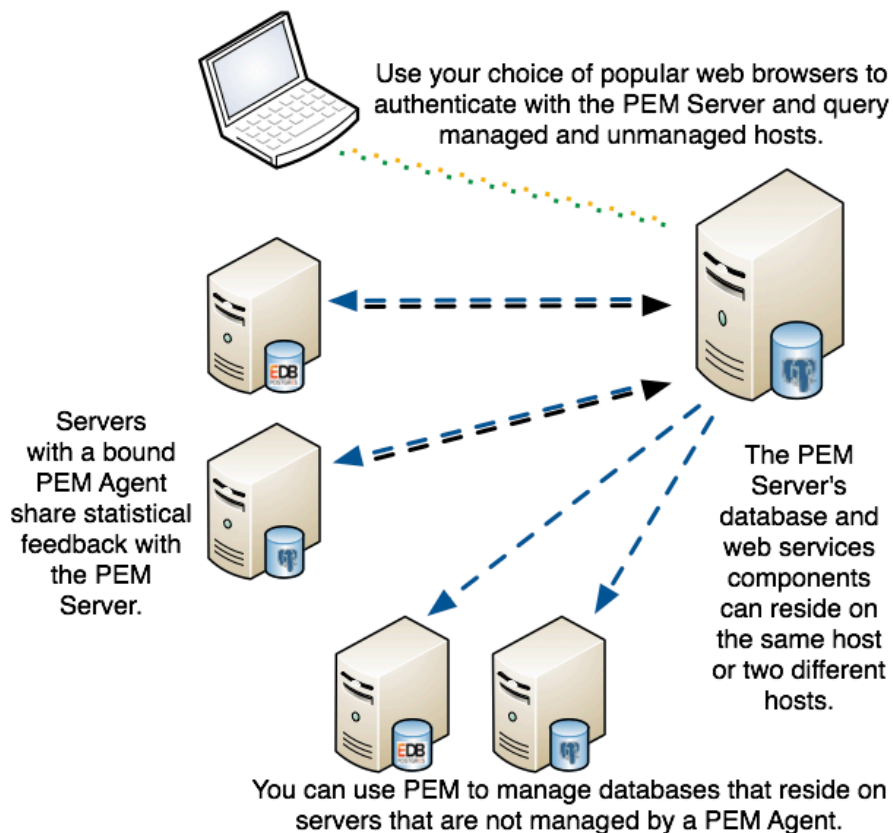


図 1 典型的な PEM インストール。

PEM エージェントのインストール

グラフィカルインストーラーを使用して、Windows ホストに Postgres Enterprise Manager エージェントをインストールできます。このグラフィカルインストーラーは、コマンドラインから呼び出すこともできます。

Linux ホストに Postgres Enterprise Manager エージェントをインストールするには、RPM パッケージを使用する必要があります。

インストーラーは、EnterpriseDB Web サイトから入手できます。

<http://www.enterprisedb.com/download-postgres-enterprise-manager>

2.1 Windows ホストへのエージェントのインストール

Windows システムでは、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 を選択することにより、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。

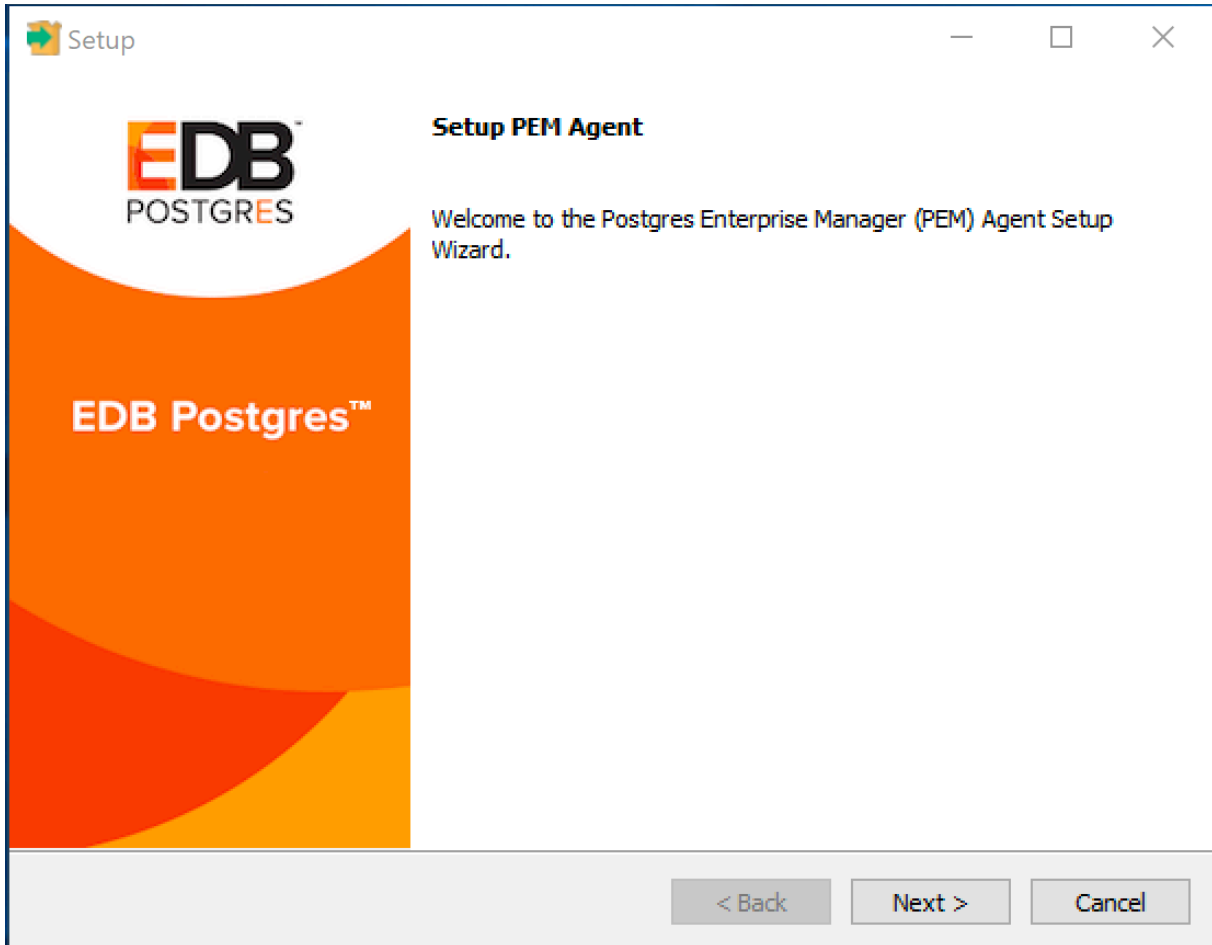


図 1 PEM Agent Installation - Welcome window

Next をクリックして、License Agreement に進みます。

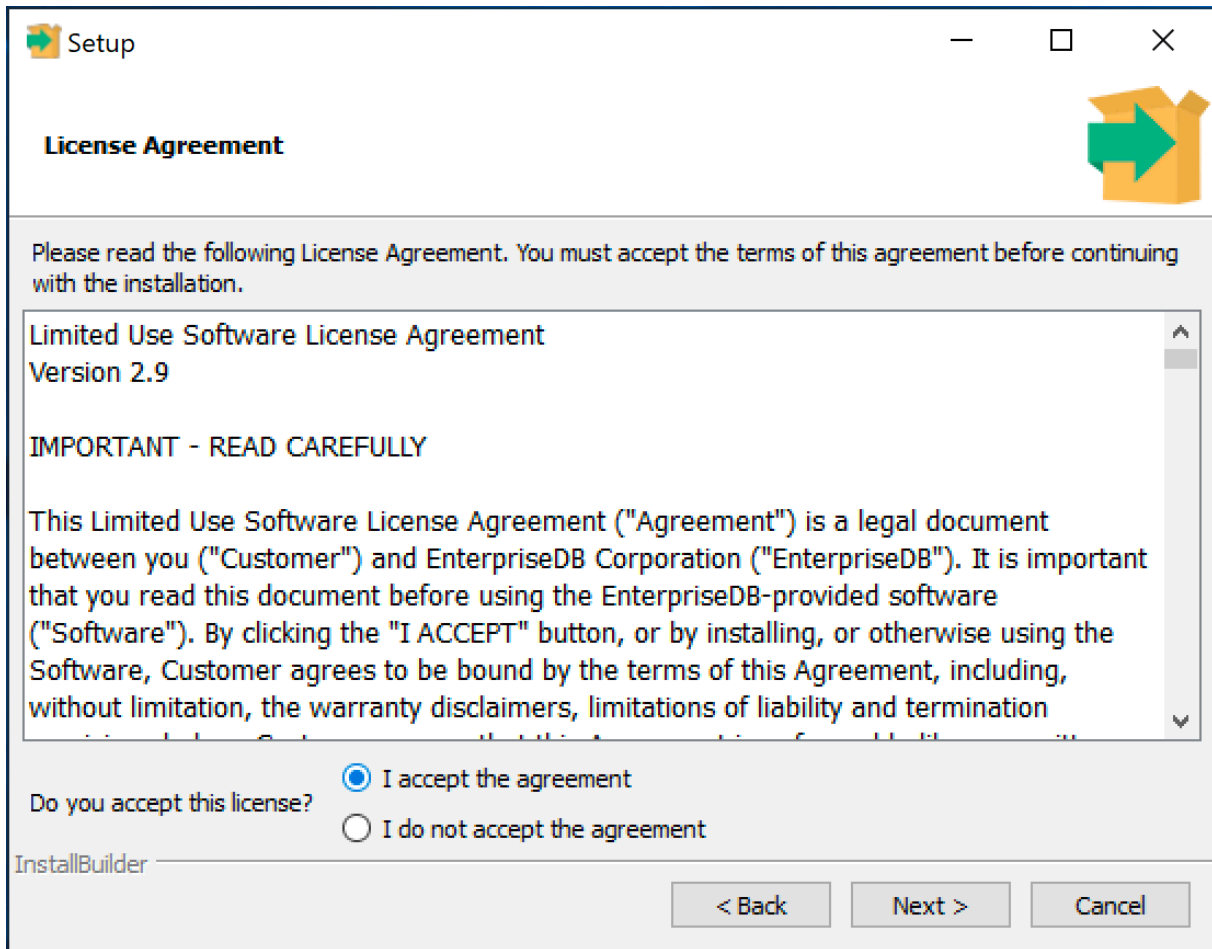


図 2 PEM Agent Installation - License Agreement

適切なラジオボタンを強調表示して契約に同意する前に、ライセンス契約を注意深く確認してください。Next をクリックして、Installation Directory ダイアログに進みます。

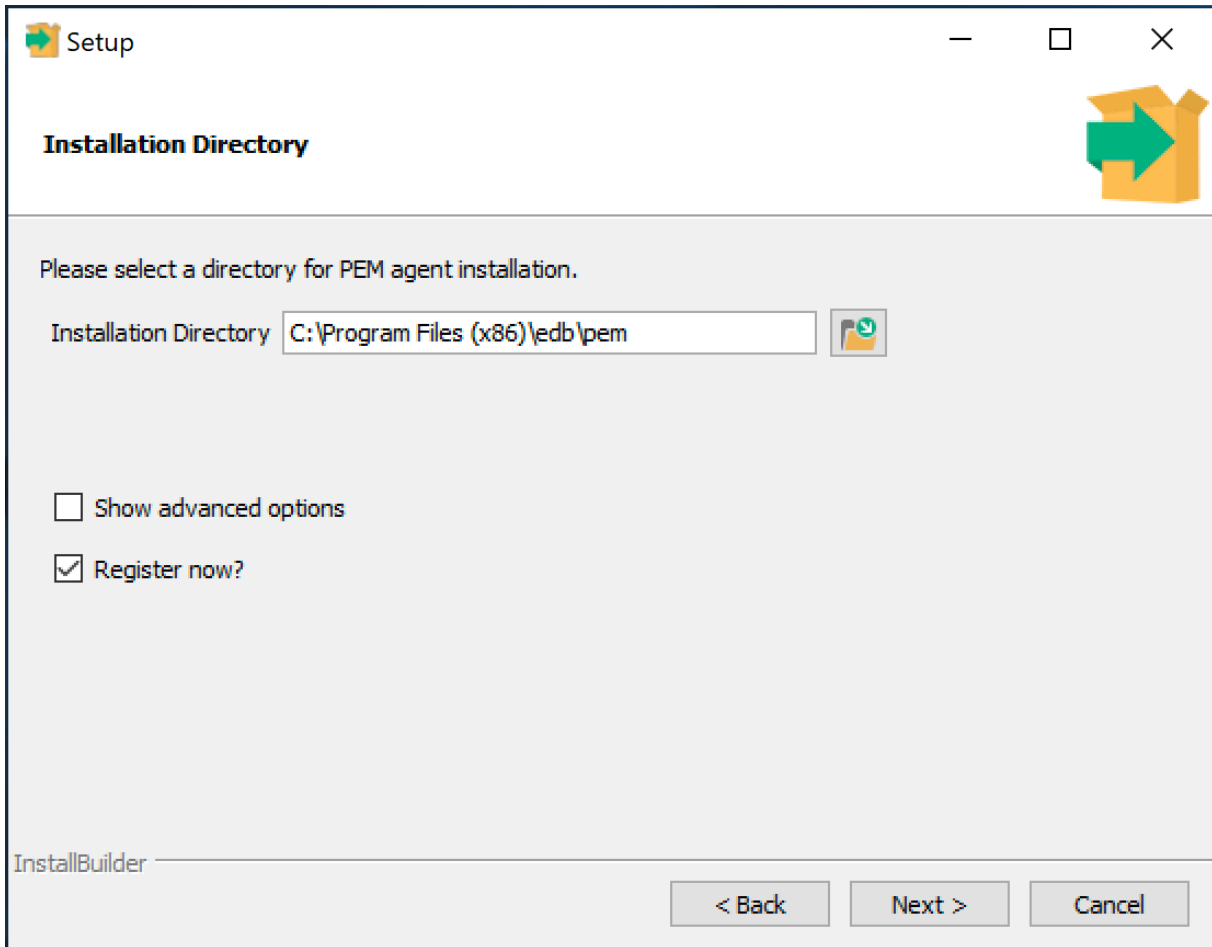


図 3 PEM Agent Installation - Installation Directory

デフォルトでは、**PEM** エージェントは `C:\Program Files (x86)\edb\pem` ディレクトリにインストールされます。デフォルトのインストールディレクトリをそのまま使用するか、[インストールディレクトリ] フィールドの内容を変更して、**PEM** エージェントの代替インストールディレクトリを指定できます。

デフォルトでは、**PEM** エージェントインストーラーは管理者の `%APPDATA%\edb\pem` ディレクトリに証明書を配置します。Show advanced options ボックスをチェックして、証明書ファイルの代替パスを指定できるダイアログを **PEM** エージェントインストーラーに含めることを示します。

Register now? の隣のボックスをチェックして、新しくインストールされた **PEM** エージェントを **PEM** サーバーに登録するようインストーラーに指示します。Next をクリックして、PEM Server Installation Details ダイアログに進みます。

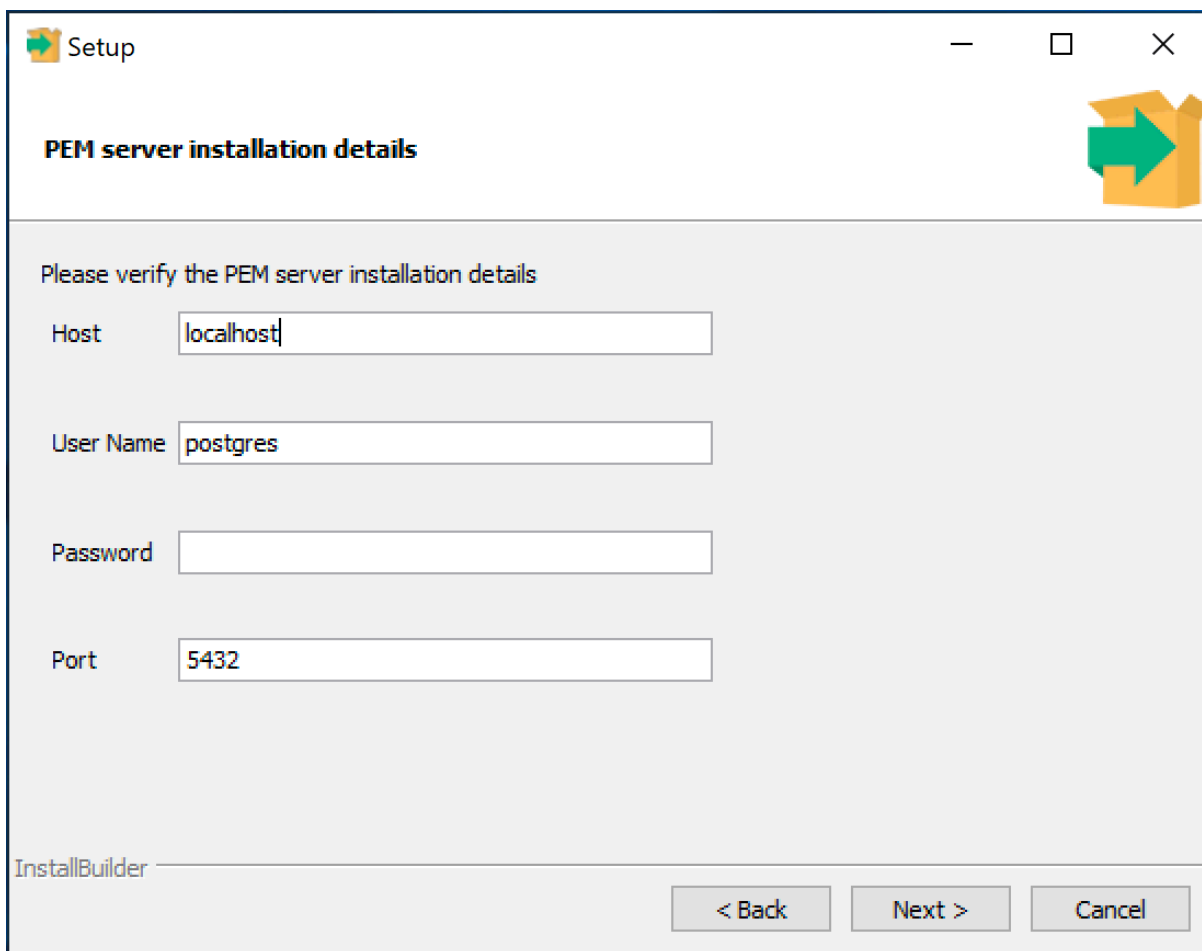


図 4 PEM Agent Installation - PEM server installation details

PEM サーバーのインストールの詳細ダイアログで、PEM サーバーの接続の詳細を入力します。

- PEM データベースサーバーが存在するシステムの名前または IP アドレスを Host フィールドに指定します。注意: PEM-HTTPD Web サーバーと PEM データベースが異なるシステムでホストされている場合、PEM データベースのホストを指定する必要があります。
- User Name フィールドにデータベースのスーパーユーザーの名前を指定します。
- Password フィールドで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指定します。
- PostgreSQL が監視しているポートを Port フィールドに指定します。

Next をクリックして、pemAgent service account ダイアログに進みます。インストーラーはサーバーに接続して、詳細が正しいことを確認しようとします。

注釈: PEM サーバーは、PEM エージェントインストーラーからの接続を許可する必要があります。接続エラーが発生した場合は、PEM サーバーのインストールの詳細ダイアログで指定された接続プロパティが正しいことを確認し、pg_hba.conf ファイル (PEM サーバー上) がエラーメッセージに記載されているサーバーへの接続を許可することを確認します。

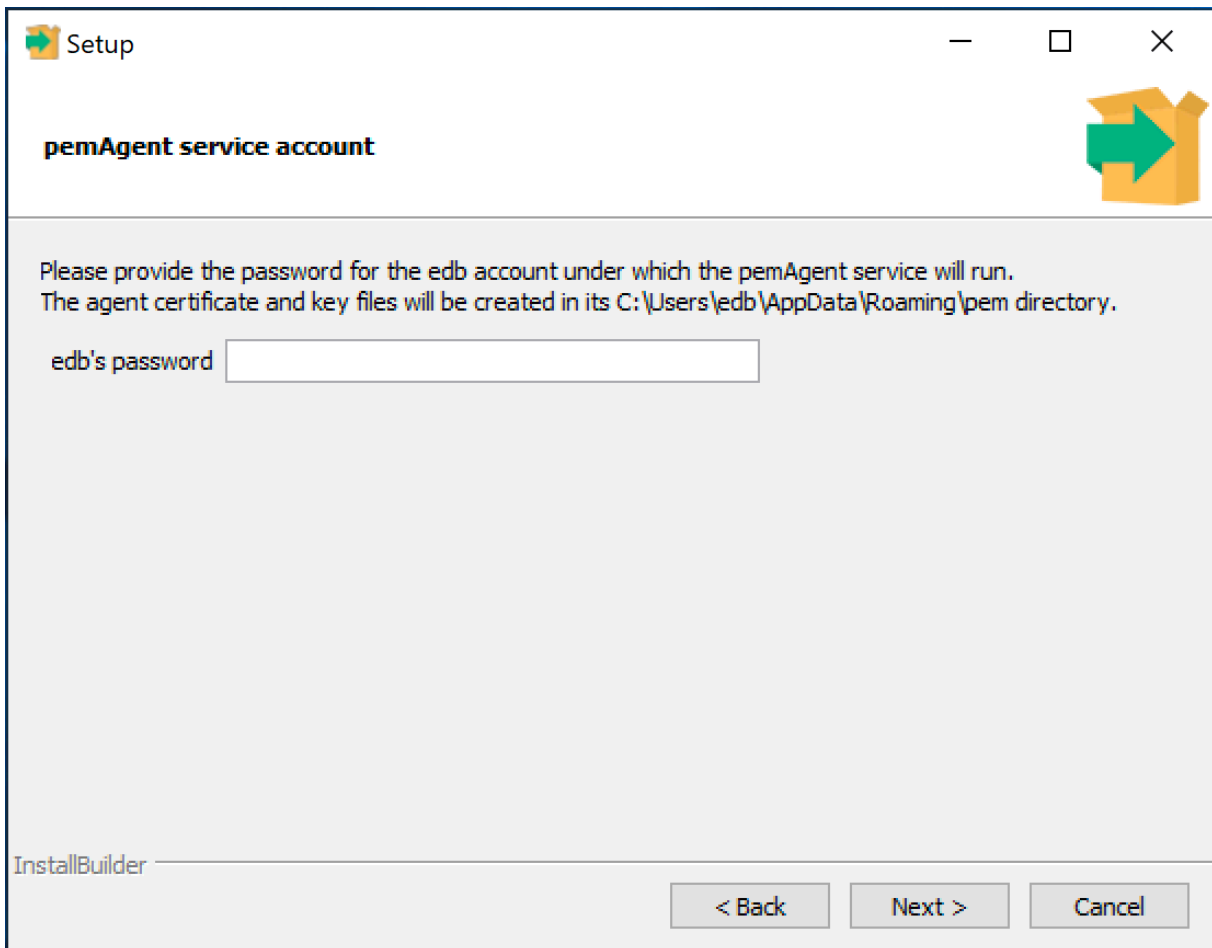


図 5 PEM Agent Installation - pemAgent Service Account password

pemAgent サービスを実行する edb アカウントのパスワードを入力します。エージェント証明書とキーファイルは C:/Users/edb/AppData/Roaming/pem ディレクトリに作成されます。Next をクリックして、Agent Details ダイアログに進みます。

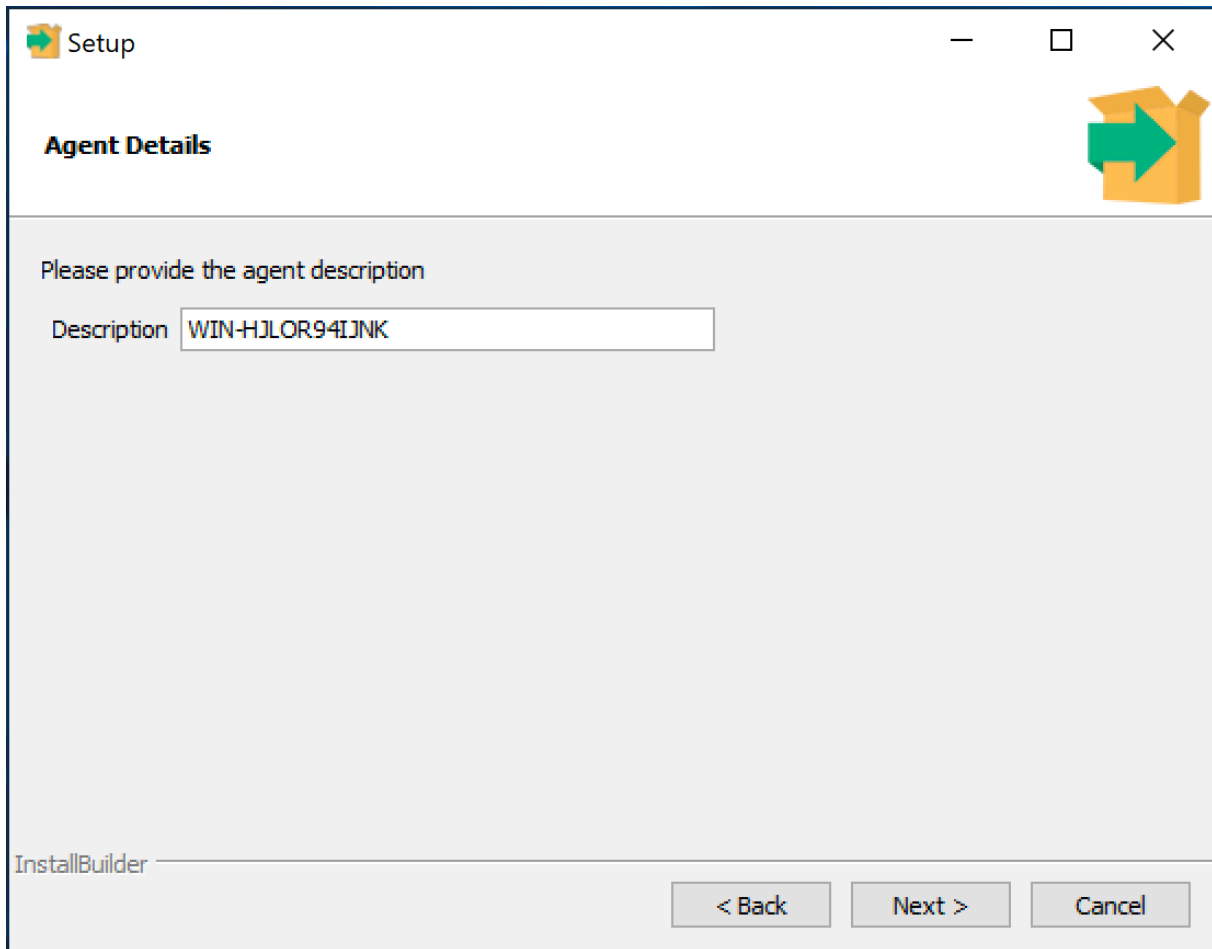


図 6 PEM Agent Installation - Agent Details

PEM Web インターフェイスのブラウザパネルに表示されるツリーコントロールには、PEM エージェントを識別するために Description フィールドに入力された値が表示されます。エージェントがインストールされているマシンのホスト名など、エージェントの説明的な名前、またはホストの機能を反映する名前を指定します。わかりやすい名前を指定するか、PEM エージェントホストによって提供されるデフォルトを受け入れ、Next をクリックして続行します。

Show advanced options チェックボックスをチェックした場合、Advanced options ダイアログが開きます:

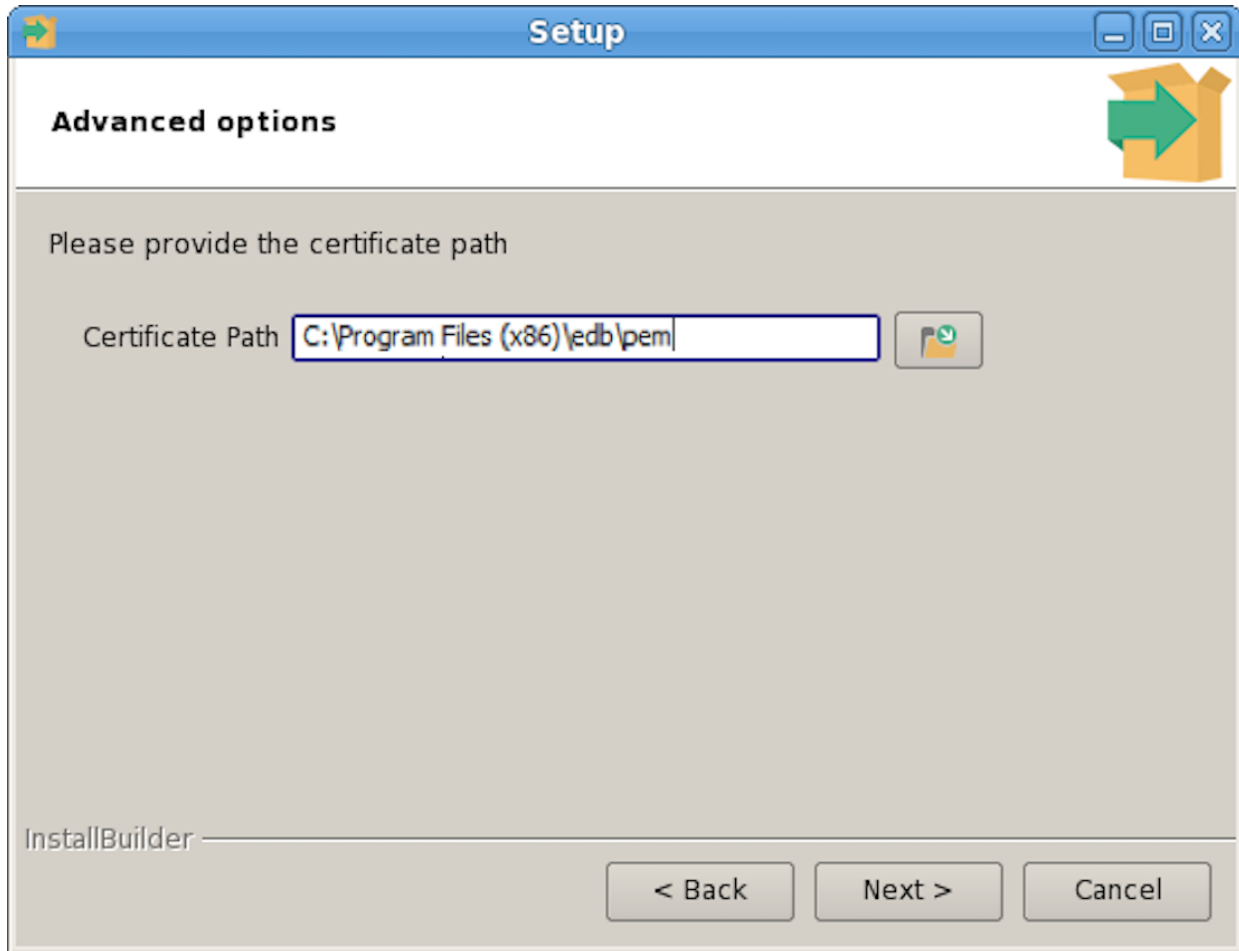


図7 PEM Agent Installation - Advanced Options - Certificate Path

デフォルトでは、PEM エージェントインストーラーは証明書を C:\Program Files (x86)\edb\pem ディレクトリに配置します。証明書の代替パスを指定するか、デフォルトを受け入れて Next をクリックします。これで、ウィザードで PEM エージェントをインストールする準備ができました。Back をクリックしてインストールディレクトリを修正するか、Next をクリックして続行します。

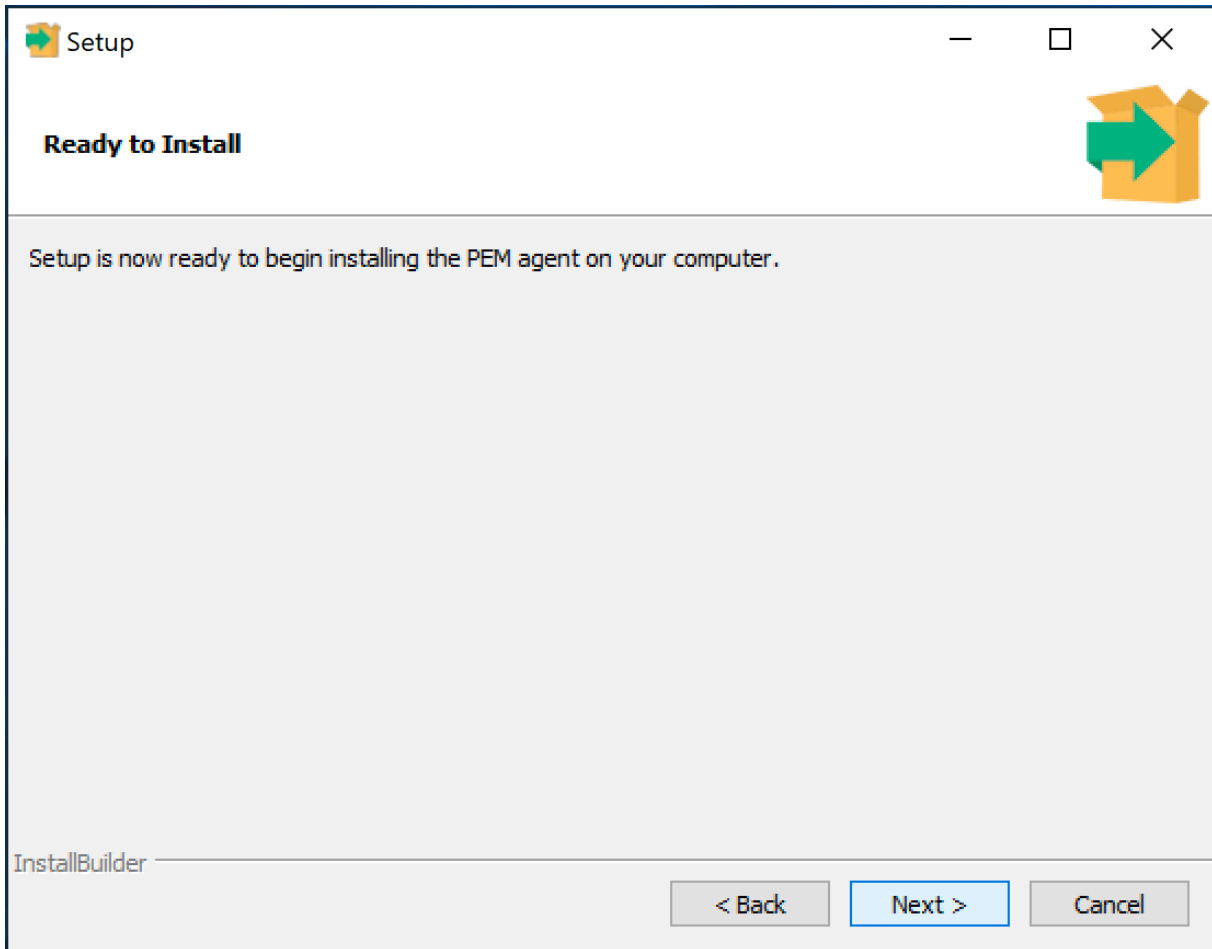


図 8 PEM Agent Installation - Ready to Install

Ready to Install ダイアログで Next をクリックして、インストーラーにファイルをシステムにコピーし、PEM サーバーにエージェントを登録するよう指示します。

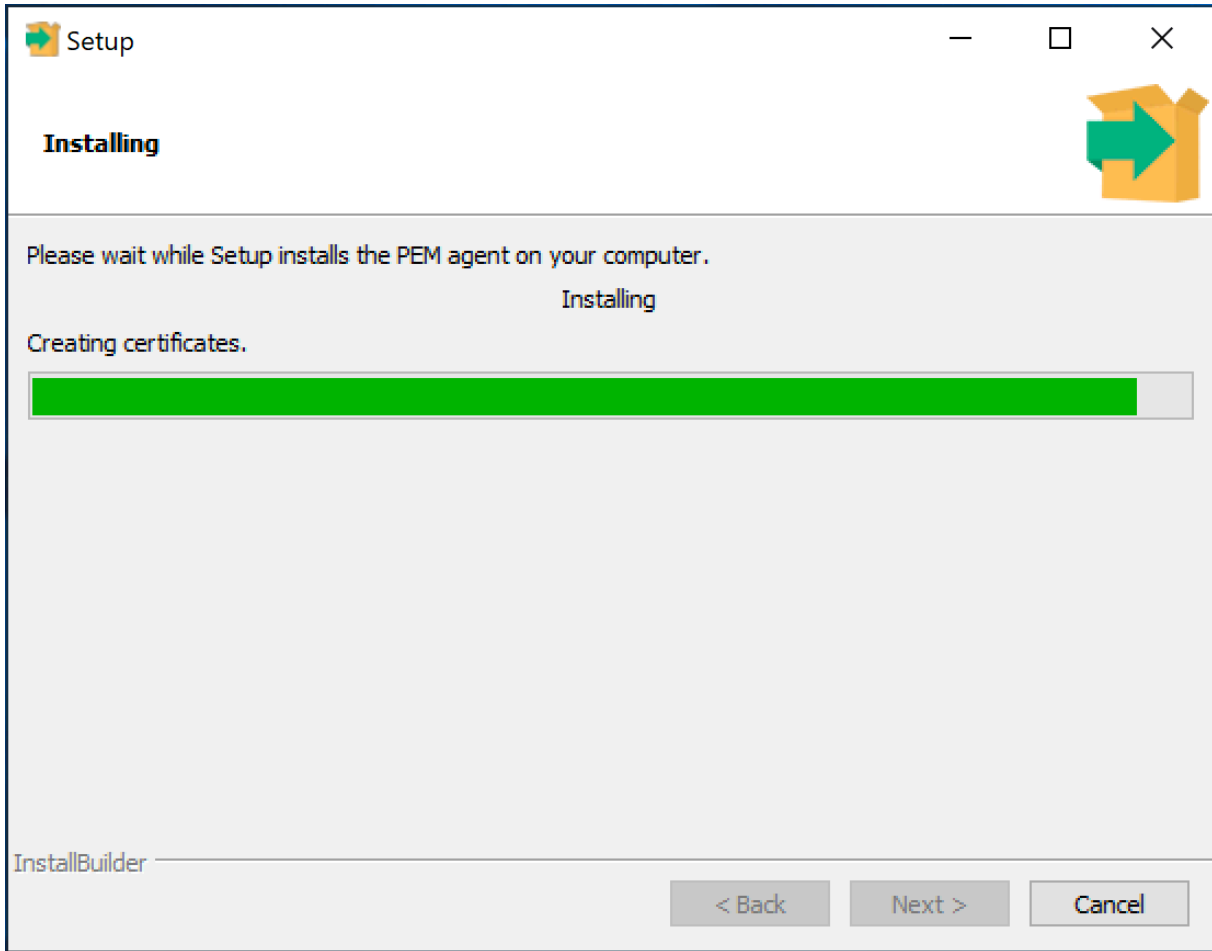


図 9 PEM Agent Installation - Installing Progress

PEM エージェントインストーラーには、PEM エージェントのインストールの進行状況を示す進行状況バーが表示されます。

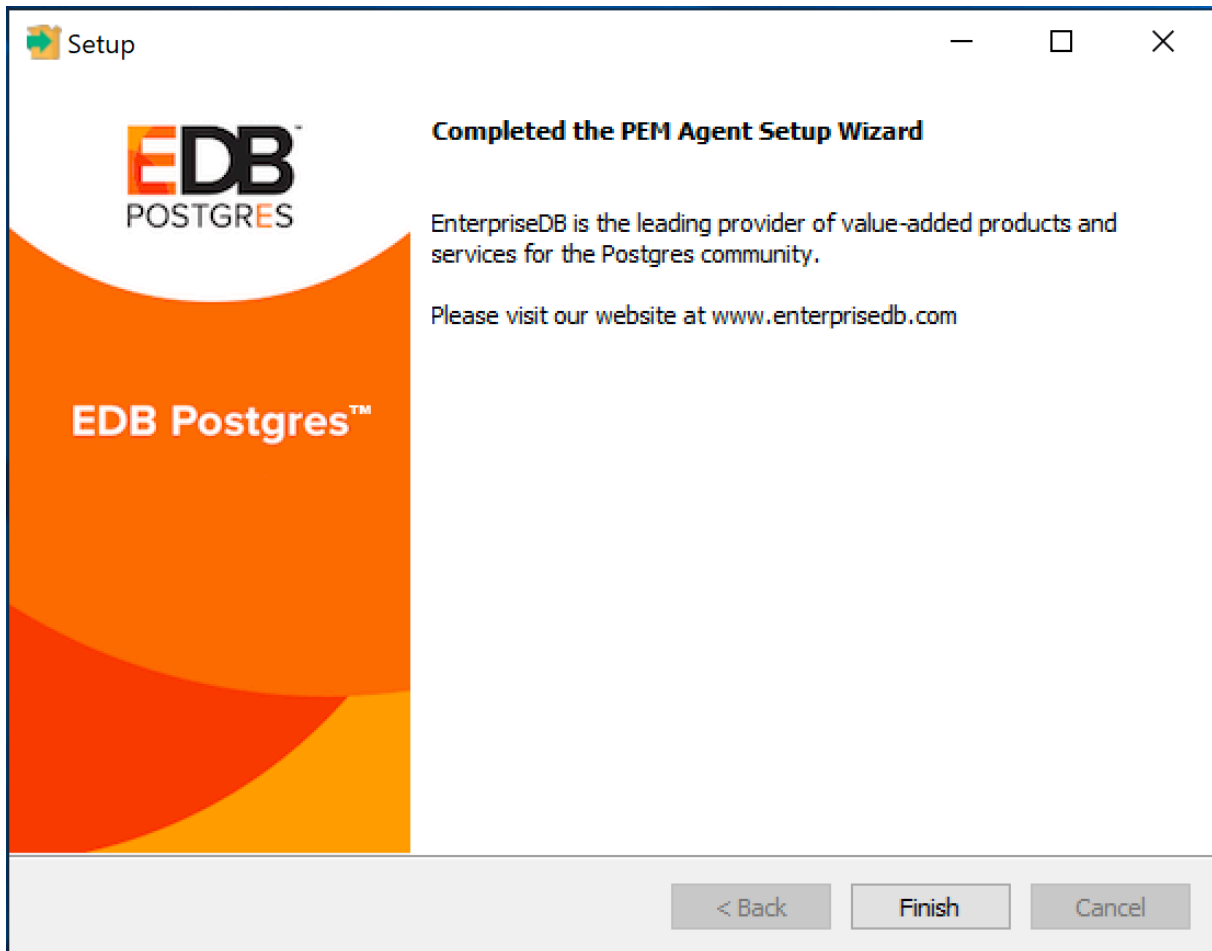


図 10 PEM Agent Installation - Finish

インストールが完了すると、PEM エージェントが実行され、オペレーティングシステムとホストデータが PEM サーバーに報告されます。PEM エージェントのホストで Postgres インスタンスの監視を開始するには、PEM のエンタープライズディレクトリに追加し、エージェントにバインドする必要があります。

2.2 コマンドラインからグラフィカルインストーラーを呼び出す

追加のユーザー入力なしでインストールを実行するためにインストーラーを呼び出すときに、`--mode unattended` オプションを含めることができます。

コマンドラインオプションの完全なリファレンスガイドについては、インストーラーを起動するときに `--help` オプションを含めてください。

2.2.1 無人モードでのグラフィカルインストーラーの呼び出し

インストーラーを呼び出すときにコマンドラインでインストール設定を指定することにより、PEM エージェントの無人インストールを実行できます。PEM サーバーをインストールするシステムは、インターネットにアクセスできる必要があることに注意してください。

無人モードで PEM エージェントインストーラーを呼び出す前に、以下を行う必要があります。

- PEM サーバーをインストールします。PEM サーバーの `pg_hba.conf` ファイルは、PEM エージェントのホストからの接続を許可する必要があります。
- 監視対象の Postgres データベースで SSL が有効になっており、接続を受け入れていることを確認してください。

PEM エージェントをインストールするには管理者権限が必要です。次のコマンドを使用して、無人モードで PEM エージェントインストーラーを呼び出します。

```
pem-agent-7<.x.x>-windows-x64.exe --mode unattended
--pgghost <pem_server_host_address> --pgport <pem_server_port>
--pguser postgres --pgpassword <pguser_password>
--agent_description <agent_name>
```

ここで、`x.x` は PEM エージェントのバージョンを指定します。`pem_server_host_address` は PEM サーバーのホストの IP アドレスを指定します。`pem_server_port` は、バックアップ PEM データベースが使用するポートを指定します。デフォルトでは、データベースはポート 5432 を使用します。`pguser_password` は PEM 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指定します。`agent_name` は PEM エージェントの説明的な名前を指定します。

注釈: PEM 7.11 以降のバージョンがインストールされている 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構成する場合、`agent.cfg` ファイルで `AllowBatchJobSteps` パ

ラメーターを True に設定する必要があります。pemagent は、デフォルトではバッチ/シェルスクリプトを実行しません。

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

Linux システムでは、yum パッケージマネージャーを使用して PEM エージェントをインストールできます。パッケージマネージャーを使用してホストに PEM エージェントをインストールする前に、以下を行う必要があることに注意してください。

- ホストに epel-release パッケージをインストールします:

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

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。

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

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

リポジトリの資格情報を受け取った後、次のことができます。

1. リポジトリ構成ファイルを作成します。
2. ユーザー名とパスワードを指定して、ファイルを変更します。
3. edb-pem-agent をインストールします。

リポジトリ設定ファイルの作成

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

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

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

ファイルの変更、ユーザー名とパスワードの入力

edb.repo ファイルを作成した後、エディターの選択を使用して、有効なパラメーターの値が 1 であることを確認し、edb.repo の username および password プレースホルダーを置き換えます 4

登録済みの EnterpriseDB ユーザーの名前とパスワードを含む仕様。

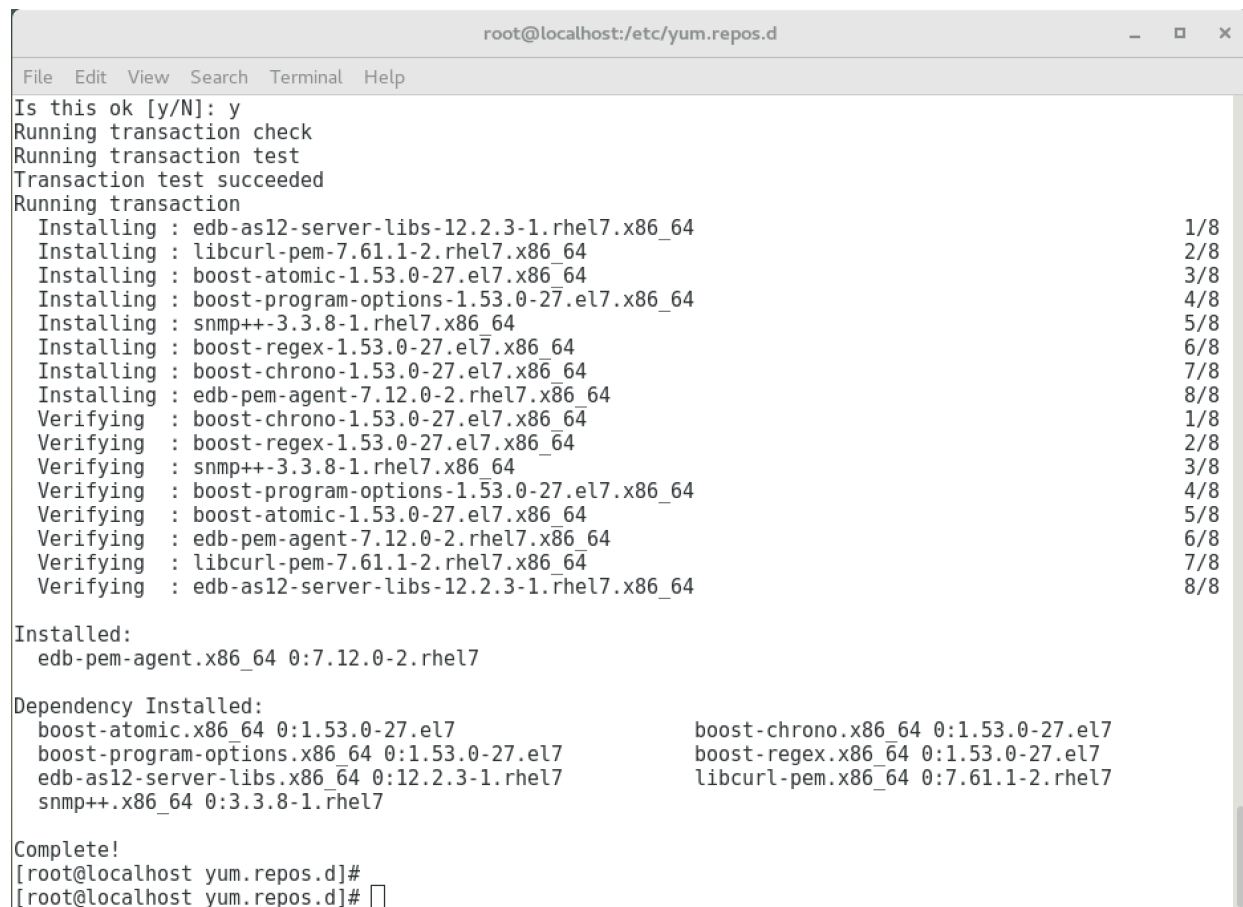
```
[edb]
name=EnterpriseDB RPMs $releasever - $basearch
baseurl=https://<username>:<password>@yum.enterprisedb.com/edb/redhat/rhel-
→$releasever-$basearch
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY
```

PEM エージェントのインストール

設定ファイルへの変更を保存した後、`yum install` コマンドを使用して `edb-pem-agent` をインストールできます。たとえば、次のコマンドは `edb-pem-agent` をインストールします。

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

インストールが完了すると、`yum` はインストールされたパッケージと依存関係のリストを表示します。



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

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

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

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

図 11 Using an RPM package to install the PEM agent

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

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

2.4 SLES ホストへのエージェントのインストール

Advanced Server のインストールおよび SLES ホストへのサポートコンポーネントの詳細については、次の Web サイトで入手可能な EDB Postgres Advanced Server インストールガイドを参照してください。

<https://www.enterprisedb.com/resources/product-documentation>

SLES パッケージは以下から入手できます。

<https://zypp.enterprisedb.com>

PEM エージェントをインストールする前に、前提条件パッケージをインストールする必要があります。

次のコマンドを所定の順序で使用して、エージェントをインストールします。

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

SUSEConnect -p sle-sdk/12/x86_64

zypper addrepo https://download.opensuse.org/repositories/Apache:Modules/<SLE_
↪version_service_pack>/Apache:Modules.repo

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

zypper refresh

zypper install edb-pem-agent
```

ここで、SLE_version_service_pack は、SLE_12_SP2 や SLE_12_SP3 など、使用している SLES のバージョンとサービスパックです。

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

Debian または Ubuntu ホストに PEM エージェントをインストールするには、EnterpriseDB リポジトリへのアクセスを許可する資格情報が必要です。リポジトリの資格情報を要求するには、**“EnterpriseDB <<https://www.enterprisedb.com/repository-access-request>>”**。

次の手順では、EnterpriseDB apt リポジトリを使用して Debian パッケージをインストールする手順を説明します。コマンドを使用する場合、* username および password *を EnterpriseDB が提供する資格情報に置き換えます。

1. <https://apt.enterprisedb.com/>にアクセスし、root としてログインします:

```
sudo su -
```

2. EnterpriseDB リポジトリを構成します。

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

3. 安全な APT リポジトリのサポートをシステムに追加します。

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

4. EBD 署名キーを追加します。

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

5. リポジトリのメタデータを更新します。

```
apt-get update
```

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

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

エージェントの登録

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

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

```
pemworker --register-agent
```

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

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

次のページに続く

表 1 – 前のページからの続き

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

Audit Manager、Log Manager、Tuning Wizard など、`pemagent` でデータベースサーバーの再起動が必要な **PEM** 機能を使用する場合は、`allow_server_restart` の値を `true` として設定する必要があります `agent.cfg` ファイル。

注釈: **PEM** 7.11 以降のバージョンがインストールされている **PEM** エージェントによって実行されるシェル/バッチスクリプトを構成する場合、`batch_script_user` パラメーターのユーザーを指定す

る必要があります。スクリプトの実行には、非 root ユーザーを使用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。

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

パスワードの入力に失敗すると、パスワード認証エラーが発生します。他の必須だが省略された情報の入力を求められます。登録が完了すると、サーバーはエージェントが正常に登録されたことを確認します。

3.1 PEM エージェント構成パラメーターの設定

PEM エージェント RPM インストーラーは、`/usr/edb/pem/agent/etc` ディレクトリに `agent.cfg.sample` という名前のサンプル構成ファイルを作成します。PEM エージェントを登録すると、`pemworker` プログラムは実際のエージェント構成ファイル（`agent.cfg` という名前）を作成します。次の構成パラメーターを追加して、`agent.cfg` ファイルを変更する必要があります。

```
heartbeat_connection = true
```

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

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

次に、プラットフォーム固有のコマンドを使用して、PEM エージェントサービスを開始します。サービスの名前は `pemagent` です。たとえば、CentOS または RHEL 6.x システムでは、次のコマンドを使用します。

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

CentOS または RHEL 7.x ホストで、`systemctl` を使用してサービスを開始します。

```
systemctl start pemagent
```

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

`pemworker` ユーティリティを使用してサーバーを登録する方法については、次の Web サイトで入手可能な* PEM 入門ガイド*を参照してください。

<https://www.enterprisedb.com/resources/product-documentation>

3.2 非ルートユーザーアカウントを使用して PEM エージェントを登録する

非 root ユーザーを使用して PEM エージェントを登録するには、最初に PEM エージェントを root ユーザーとしてインストールする必要があります。インストール後、非 root ユーザー（edb など）の ID を想定して、以下の手順を実行します。

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

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

2. 次のように pemworker ユーティリティを使用して、エージェントを PEM サーバーに登録します。

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

上記のコマンドは、/home/edb/.pem ディレクトリにエージェント証明書とエージェント設定ファイル（agent.cfg）を作成します。次のコマンドを使用して、これらのファイルに読み取りおよび書き込み権限を割り当てます。

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

3. agent.cfg ファイルのパラメーターを次のように変更します。

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

4. pemagent サービスファイルのパスとユーザーの値を更新します。

- RHEL または CentOS 6 を使用している場合、agent.cfg ファイルの正しいパスを反映するように pemagent サービスファイルを更新し、ユーザー su を su edb に変更します。
- RHEL または CentOS 7 を使用している場合は、次のようにパラメーターを更新します。

```
User=edb  
ExecStart=/usr/edb/pem/agent/bin/pemagent -c /home/edb/.pem/agent.cfg
```

5. 前に開始されたエージェントプロセスを強制終了し、次のように非ルートユーザーを使用してエージェントサービスを再起動します。
 - RHEL または CentOS 6 を使用している場合、`sudo /etc/init.d/pemagent start/stop/restart`
 - RHEL または CentOS 7 を使用している場合、`sudo systemctl start/stop/restart pemagent`
6. PEM ダッシュボードでエージェントのステータスを確認します。

第 4

PEM エージェントの管理

以下のセクションでは、PEM エージェントの動作と管理に関する情報を提供します。

4.1 エージェント権限

デフォルトでは、PEM エージェントはオペレーティングシステムホストの `root` 特権とデータベースサーバーのスーパーユーザー特権でインストールされます。これらの権限により、PEM エージェントは、システムの使用状況について監視対象ホストおよびデータベースサーバーで制限のないプローブを呼び出し、情報を取得して PEM サーバーに返すことができます。

PEM エージェントの権限が低下すると、PEM 機能が低下することに注意してください。完全な機能を実現するには、PEM エージェントを `root` として実行する必要があります。PEM エージェントがデータベースサーバーのサービスアカウントで実行されている場合、PEM プローブはレポートの生成に使用される統計情報に完全にアクセスできず、機能はそのアカウントの機能に制限されます。PEM エージェントが別のより低い特権のアカウントで実行されている場合、機能はさらに制限されます。

PEM エージェントのオペレーティングシステム権限を制限すると、PEM プローブの一部が情報を返さず、次の機能が影響を受ける可能性があります。

プローブまたはアクション	オペレーティングシステム	影響を受ける PEM 機能
データとログファイルの分析	Linux / Windows	Postgres Expert は完全な情報にアクセスできません。
セッション情報	Linux	プロセスごとの統計は不完全です。
PG HBA	Linux / Windows	Postgres Expert は完全な情報にアクセスできません。
サービス再起動機能	Linux / Windows	監査ログマネージャー、サーバーログマネージャー、ストリーミングレプリケーション、ログ分析エキスパート、および PEM は、要求された変更を適用できない場合があります。
パッケージ展開	Linux / Windows	PEM はダウンロードしたインストールモジュールを実行できません。
バッチタスク	Windows	PEM は、Windows でスケジュールされたバッチジョブを実行できません。
サーバーからデータを収集する（ルートアクセスが必要）	Linux / Windows	スワップ使用率、CPU 使用率、IO 読み取り、IO 書き込みなどの列は、セッションアクティビティダッシュボードで 0 として表示されます。

注釈: 上記のリストは包括的なものではありませんが、制限される機能の種類の概要を提供する必要があります。

PEM エージェントのデータベース特権を制限すると、次の PEM 機能が影響を受ける可能性があります。

プローブ	オペレーティングシステム	影響を受ける PEM 機能
監査ログの収集	Linux / Windows	PEM は PEM データベースから空のデータを受け取ります。
サーバーログ収集	Linux / Windows	PEM はサーバーログ情報を収集できません。
データベース統計	Linux / Windows	データベース/サーバー分析ダッシュボードには、不完全な情報が含まれます。
セッション待機/システム待機	Linux / Windows	セッション/システム待機ダッシュボードには、不完全な情報が含まれます。
ロック情報	Linux / Windows	データベース/サーバー分析ダッシュボードには、不完全な情報が含まれます。
ストリーミングレプリケーション	Linux / Windows	ストリーミングレプリケーションダッシュボードには情報が表示されません。
Slony レプリケーション	Linux / Windows	データベース分析ダッシュボードの Slony 関連のグラフには情報が表示されません。
表領域サイズ	Linux / Windows	サーバー分析ダッシュボードには、完全な情報は表示されません。
xDB レプリケーション	Linux / Windows	PEM は xDB アラートとトラップを送信できません。

プローブが十分な権限を持たないオペレーティングシステムを照会している場合、プローブは permission denied エラーを返すことがあります。

プローブが不十分な特権でデータベースを照会している場合、プローブは permission denied エラーを返すか、返されたデータを空の値として PEM チャートまたはグラフに表示します。

プローブが失敗すると、プローブの名前、プローブが失敗した理由、問題の解決に役立つヒントを含むエントリがログファイルに書き込まれます。

サーバーで発生したプローブ関連のエラーをプローブログダッシュボードで表示したり、PEM ワーカーログファイルのエラーメッセージを確認したりできます。Linux では、ログファイルのデフォルトの場所は次のとおりです。

```
/var/log/pem/worker.log
```

Windows では、ログ情報は Event Viewer で利用できます。

4.2 エージェント構成

多数のユーザー設定可能なパラメーターとレジストリエントリが PEM エージェントの動作を制御します。PEM エージェントのパラメーター設定を変更して、Streaming Replication ウィザードなどの一部の PEM 機能を有効にする必要がある場合があります。PEM エージェント構成ファイルの値を変更した後、変更を適用するには PEM エージェントを再起動する必要があります。

PEM_MAXCONN パラメーターを除き、PEM 機能を有効にするために変更が必要な場合を除き、EnterpriseDB サポートの専門家に相談することなく、以下にリストされている構成パラメーターまたはレジストリエントリを変更しないことを強くお勧めします。

Linux システムでは、PEM 設定オプションは `/usr/edb/pem/agent/etc` にある `agent.cfg` ファイルに保存されます。`agent.cfg` ファイルには次のエントリが含まれます:

パラメータ名	説明	デフォルト値
<code>pem_host</code>	PEM サーバーの IP アドレスまたはホスト名。	127.0.0.1。
<code>pem_port</code>	PEM サーバーと通信するためにエージェントが接続するデータベースサーバーポート。	ポート 5432。
<code>pem_agent</code>	PEM エージェントに割り当てられた一意の識別子。	最初のエージェントは “1”、2 番目のエージェントは “2” などです。
<code>agent_ssl_key</code>	PEM エージェントのキーファイルへの完全なパス。	<code>/root/.pem/agent.key</code>
<code>agent_ssl_cert</code>	PEM エージェントの証明書ファイルへの完全なパス。	<code>/root/.pem/agent.crt</code>
<code>agent_flag_dir</code>	HA サポートに使用されます。別のサーバーの監視を引き継ぐための要求をチェックするディレクトリパスを指定します。要求は、指定されたフラグディレクトリ内のファイルの形式で行われます。	デフォルトでは設定されていません。

次のページに続く

表 1 – 前のページからの続き

パラメータ名	説明	デフォルト値
log_level	ログレベルは、PEM ログファイルに書き込まれるイベントのタイプを指定します。	警告
log_location	PEM ワーカーログファイルの場所を指定します。	127.0.0.1。
agent_log_location	PEM エージェントのログファイルの場所を指定します。	/var/log/pem/agent.log
long_wait	最初の接続試行が失敗した場合に、PEM エージェントが PEM サーバーへの接続を試行する前に待機する最大時間（秒単位）。	30 秒
short_wait	PEM エージェントがキュー内の次のプローブ（実行待ち）をチェックする前に待機する最小時間（秒単位）。	10 秒
alert_threads	エージェントによって生成されるアラートスレッドの数。	PEM サーバーのホストに常駐するエージェントに対して 1 に設定します。他のすべてのエージェントの場合は 0。
enable_smtp	true に設定すると、SMTP 電子メール機能が有効になります。	PEM サーバーホストの場合は true。他のすべての場合は false。
enable_snmp	true に設定すると、SNMP トラップ機能が有効になります。	PEM サーバーホストの場合は true。他のすべての場合は false。
enable_nagios	true に設定すると、Nagios アラートが有効になります。	PEM サーバーホストの場合は true。他のすべての場合は false。
connect_timeout	エージェントが接続を待機する最大時間（秒単位）（10 進整数文字列）。	デフォルトでは設定されていません。0 に設定すると、エージェントは無期限に待機する必要があります。

次のページに続く

表 1 – 前のページからの続き

パラメータ名	説明	デフォルト値
allow_server_restart	TRUE に設定すると、エージェントは、監視するデータベースサーバーを再起動できます。このパラメーターの値に応じて、一部の PEM 機能を有効/無効にすることができます。	偽
allow_package_management	TRUE に設定すると、更新モニターおよびパッケージ管理機能が有効になります。	偽
max_connections	接続スロットルで使用するプロープ接続の最大数。	0（無制限の数）
connection_lifetime	ConnectionLifetime（または connection_lifetime）を使用して、開いているがアイドル状態の接続が保持される最小秒数を指定します。MaxConnections で指定された値に達し、待機中の要求を満たすために（別のデータベースへの）新しい接続が必要な場合、このパラメーターは無視されます。	デフォルトでは、0 に設定されます（エージェントの処理ループの後に接続がアイドル状態になると、接続はドロップされます）。
allow_batch_probes	TRUE に設定すると、ユーザーはカスタムプロープ機能を使用してバッチプロープを作成できます。	偽
heartbeat_connection	TRUE に設定すると、ハートビートの送信に専用の接続が使用されます。	偽
allow_streaming_replication	TRUE に設定すると、ユーザーはストリーミングレプリケーションを設定およびセットアップできます。	偽

次のページに続く

表 1 – 前のページからの続き

パラメータ名	説明	デフォルト値
batch_script_dir	スクリプトファイル（警告用）が保存されるパスを指定します。	/tmp
connection_custom_setup	監視対象サーバーとの新しい接続が確立されたときに呼び出される SQL コードを提供するために使用します。	デフォルトでは設定されていません。
ca_file	CA 証明書が存在するパスを指定します。	デフォルトでは設定されていません。
batch_script_user	バッチ/シェルスクリプトの実行に使用するユーザーの名前を指定します。	なし

64 ビット Windows システムでは、PEM レジストリエントリは次の場所にあります。

HKEY_LOCAL_MACHINE/Software/Wow6432Node/EnterpriseDB/PEM/agent

レジストリには次のエントリが含まれます。

パラメータ名	説明	デフォルト値
PEM_HOST	PEM サーバーの IP アドレスまたはホスト名。	127.0.0.1。
PEM_PORT	PEM サーバーと通信するためにエージェントが接続するデータベースサーバーポート。	ポート 5432。
AgentID	PEM エージェントに割り当てられた一意の識別子。	最初のエージェントは “1”、2 番目のエージェントは “2” などです。
AgentKeyPath	PEM エージェントのキーファイルへの完全なパス。	% APPDATA %/ Roaming / pem / agent.key。
AgentCrtPath	PEM エージェントの証明書ファイルへの完全なパス。	% APPDATA %/ Roaming / pem / agent.crt

次のページに続く

表 2 – 前のページからの続き

AgentFlagDir	HA サポートに使用されます。別のサーバーの監視を引き継ぐための要求をチェックするディレクトリパスを指定します。要求は、指定されたフラグディレクトリ内のファイルの形式で行われます。	デフォルトでは設定されていません。
LogLevel	ログレベルは、PEM ログファイルに書き込まれるイベントのタイプを指定します。	警告
LongWait	最初の接続試行が失敗した場合に、PEM エージェントが PEM サーバーへの接続を試行する前に待機する最大時間（秒単位）。	30 秒
shortWait	PEM エージェントがキュー内の次のプローブ（実行待ち）をチェックする前に待機する最小時間（秒単位）。	10 秒
AlertThreads	エージェントによって生成されるアラートスレッドの数。	PEM サーバーのホストに常駐するエージェントに対して 1 に設定します。他のすべてのエージェントの場合は 0。
EnableSMTP	true に設定すると、SMTP 電子メール機能が有効になります。	PEM サーバーホストの場合は true。他のすべての場合は false。
EnableSNMP	true に設定すると、SNMP トラップ機能が有効になります。	PEM サーバーホストの場合は true。他のすべての場合は false。
ConnectTimeout	エージェントが接続を待機する最大時間（秒単位）（10 進整数文字列）。	デフォルトでは設定されていません。0 に設定すると、エージェントは無期限に待機します。

次のページに続く

表 2 – 前のページからの続き

AllowServerRestart	TRUE に設定すると、エージェントは、監視するデータベースサーバーを再起動できます。このパラメーターの値に応じて、一部の PEM 機能を有効/無効にすることができます。	本当
AllowPackageManagement	TRUE に設定すると、更新モニターおよびパッケージ管理機能が有効になります。	偽
MaxConnections	接続スロットルで使用されるプローブ接続の最大数。	0（無制限の数）
ConnectionLifetime	ConnectionLifetime（または connection_lifetime）を使用して、開いているがアイドル状態の接続が保持される最小秒数を指定します。MaxConnections で指定された値に達し、待機中の要求を満たすために（別のデータベースへの）新しい接続が必要な場合、このパラメーターは無視されます。	デフォルトでは、0 に設定されます（エージェントの処理ループの後に接続がアイドル状態になると、接続はドロップされます）。
AllowBatchProbes	TRUE に設定すると、ユーザーはカスタムプローブ機能を使用してバッチプローブを作成できます。	偽
HeartbeatConnection	TRUE に設定すると、ハートビートの送信に専用の接続が使用されます。	偽
AllowStreamingReplication	TRUE に設定すると、ユーザーはストリーミングレプリケーションを設定およびセットアップできます。	偽

次のページに続く

表 2 – 前のページからの続き

BatchScriptDir	スクリプトファイル（警告用）が保存されるパスを指定します。	/tmp
ConnectionCustomSetup	監視対象サーバーとの新しい接続が確立されたときに呼び出される SQL コードを提供するために使用します。	デフォルトでは設定されていません。
ca_file	CA 証明書が存在するパスを指定します。	デフォルトでは設定されていません。
AllowBatchJobSteps	true に設定すると、バッチ/シェルスクリプトは管理者ユーザーアカウントを使用して実行されます。	なし

4.3 エージェントのプロパティ

PEM Agent Properties ダイアログは、ダイアログが開かれた PEM エージェントに関する情報を提供します。ダイアログを開くには、PEM クライアントツリーコントロールでエージェント名を右クリックし、コンテキストメニューから Properties を選択します。

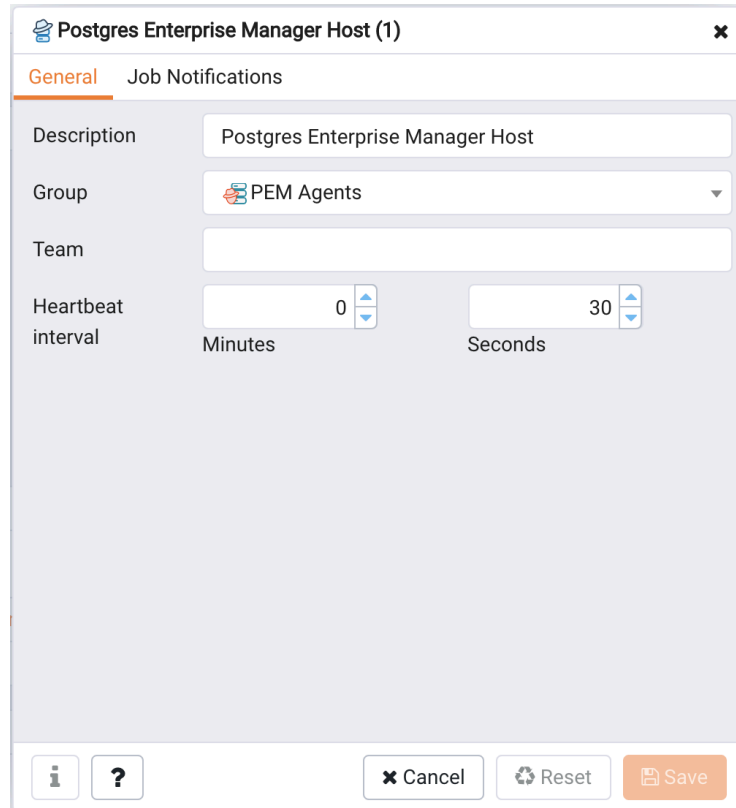


図 1 The PEM Agent Properties dialog

PEM エージェントプロパティダイアログのフィールドを使用して、PEM エージェントに関する情報を確認または変更します。

- Description フィールドには、PEM エージェントの変更可能な説明が表示されます。この説明は、PEM クライアントのツリーコントロールに表示されます。
- グループを使用して、PEM クライアントツリーコントロールでサーバーとエージェントを整理できます。Group ドロップダウンリストボックスを使用して、エージェントが表示されるグループを選択します。
- Team フィールドを使用して、エージェントによって監視されているサーバーにアクセスできるグループロールの名前を指定します。このエージェントによって監視されるサーバーは、接続さ

れたチームメンバーの PEM クライアントツリーコントロールに表示されます。これは便利な機能であることに注意してください。[チーム] フィールドは真の分離を提供しないため、セキュリティの目的で使わないでください。

- Heartbeat interval フィールドは、PEM エージェントから PEM サーバーへのレポート間の経過時間を表示します。間隔を変更するには、Minutes または Seconds フィールドの横にあるセクターを使用します。

第 5

PEM エージェントのトラブルシューティング

5.1 削除された PEM エージェントの復元

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

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

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

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

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

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

ここで、x は手順 1 で使用したクエリの出力に表示された識別子です。

3. PEM Web クライアントを更新します。

削除されたエージェントは再び復元されます。ただし、その特定のエージェントにバインドされたサーバーはダウンしているように見える場合があります。この問題を解決するには、サーバーの PEM エー

ジェントプロパティを変更して、バインドされたエージェントを再度追加する必要があります。変更が成功すると、サーバーは正常に実行されていると表示されます。

5.2 PEM サーバーの再構成

特定の状況では、PEM サーバーのアンインストール、再インストール、および PEM サーバーの再構成が必要になる場合があります。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 サーバーを再度構成します。

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

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

PEM Web インターフェースを使用して、ステータスが Down または Unknown の PEM エージェントを削除するのは、そのようなエージェントの数が多い場合は困難です。このような状況では、コマンドラインインターフェイスを使用して、ダウンまたは不明なエージェントを削除することができます。

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

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

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

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

第 6

PEM エージェントのアンインストール

PEM インストールディレクトリにあるアンインストーラーを使用して、システムから PEM エージェントを削除します。デフォルトでは、PEM エージェントのアンインストーラーは次の場所にあります。

成分	PEM エージェント	アンインストーラー名
uninstall-pemagent	デフォルトの場所	/usr/edb/pem/agent

エージェントを削除するには、スーパーユーザー権限を引き受け、ターミナルウィンドウを開き、アンインストーラーが存在するディレクトリに移動します。次のようにインストーラーを起動します。

```
./uninstall-<agent_name>
```

Where *agent_name* is the name of the agent that you wish to remove.

PEM インストールが Windows ホストにある場合、Windows の Uninstall a Program アプレットを使用して PEM コンポーネントを削除できます。Uninstall a Program アプレットを開くには、Windows Control Panel の Uninstall a Program サブメニューをナビゲートし、Programs and Features を選択します。Uninstall a Program ウィンドウが開いたら、削除する PEM コンポーネントの名前を強調表示し、Uninstall/Change ボタンをクリックします。Windows ポップアップが開き、コンポーネントの削除を確認するプロンプトが表示されます。Yes をクリックしてコンポーネントを削除します。

結論

EDB Postgres Enterprise Manager エージェントユーザーガイド

Copyright©2007-2020 EnterpriseDB Corporation。無断複写・転載を禁じます。

EnterpriseDB®Corporation 34 Crosby Drive, Suite 201, Bedford, MA 01730, USA

T +1 781 357 3390 F +1 978 467 1307 E info@enterprisedb.com www.enterprisedb.com

- EDB は、EDB Postgres Enterprise Manager が存在する場合、ログオン UI の設計、コーディングのベストプラクティスの確立、レビュー、および入力検証の検証を行います。EDB は追加の入力コンポーネントに対して同じアプローチに従いますが、製品の性質上、限定された検証が可能な信頼できるユーザーによって入力および送信されるフリーフォーム SQL、WMI またはその他の文字列を受け入れる必要がある場合があります。このような場合、ユーザーが誤った入力やその他の危険な入力を防ぐことはできません。
- EDB は、将来、認証された信頼できるユーザーからのフリーフォーム SQL、WMI またはその他の潜在的に危険な入力を受け入れる製品に機能を追加する権利を留保しますが、そのような機能はすべて、可能な限り最小のリスクを提供するように設計およびテストされます可能であれば、スーパーユーザーまたは同等の特権が必要です。
- EDB は、すべての潜在的な脅威を予測できることを保証しません。したがって、すべての潜在的な脆弱性が対処または検討されたことをプロセスが完全に保証することはできません。

索引

Agent Configuration, 32
Agent Privileges, 29
Agent Properties, 39

deleting agents with Down status, 44
deleting agents with Unknown status, 44

Installing agent on CentOS, 17
Installing agent on Debian, 21
Installing agent on RHEL, 17
Installing agent on SLES, 20
Installing agent on Ubuntu, 21
Installing agent on Windows, 5
Invoking graphical installer from CLI, 15
Invoking graphical installer in unattended mode, 15

PEM architecture, 2
PEM components, 2
PEM overview, 2
PEM エージェントのアンインストール, 45
PEM エージェントのインストール, 4
PEM エージェントのトラブルシューティング, 41
PEM エージェントの管理, 29

reconfiguring an agent, 43
restoring a deleted agent, 41

エージェントの登録, 22

結論, 46