PEM Installation Guide on Linux

Version 8.0

Built at 2021-04-07T05:46:58

0 PEM Installation Guide on Linux

Postgres Enterprise Manager (PEM) is designed to assist database administrators, system architects, and performance analysts when administering, monitoring, and tuning PostgreSQL and Advanced Server database servers. PEM has been designed to manage and monitor a single server or multiple servers from a single console, allowing complete control over monitored databases.

This document provides step-by-step instructions to guide you through the installation of Postgres Enterprise Manager on a Linux host.

Throughout this guide, the term *Postgres* refers to either a PostgreSQL or an Advanced Server installation, where either is appropriate.

1 What's New

The following changes have been made to Postgres Enterprise Manager to create version 8.0.1:

Improved the overall performance, usability, and stability of the product by:

- Allowing the dash (-) character in the superuser name that is provided during configuration.
- Gracefully closing the operating system resources during batch probe and command execution to avoid errors.
- Improved support for unicode string handling in pemAgent.
- Fixed a regression restoring BART backup when agent is not bound with the BART server.
- In addition to the above-listed items, a few other minor bugs have also been fixed.

PEM 8.0.1 provides an option to hide or unhide the Dashboard, Statistics, Dependents, and Dependencies tabs.

Features and Functionalities removed in recent versions of PEM:

- PEM 8.0 and later is not supported on CentOS/RHEL 6.x.
- EDB ARK UI Management is not supported by PEM 7.16 and later.
- PEM 7.16 and later is not supported on Internet explorer version 11 and lesser.

2 PEM - Hardware and Software Requirements

Hardware Prerequisites

For optimum performance when monitoring servers and rendering dashboards, we recommend installing PEM on a system with at least:

- 4 CPU cores
- 8 GB of RAM
- 100 GB of Storage

Additional disk space is required for data storage. Please note that resource usage will vary based on which probes are defined and enabled, and the activity level on the monitored databases. Monitoring server resources (as you use PEM) will let you know when you need to expand your initial system configuration.

Software Prerequisites

Platforms and Versions Support

For information about the platforms and versions supported by PEM, visit the EnterpriseDB website at:

https://www.enterprisedb.com/services-support/edb-supported-products-and-platforms

Please note: PEM 8.0.1 is no longer supported on CentOS/RHEL/OEL 6.x platforms. It is strongly recommended that EDB products running on these platforms be migrated to a supported platform.

Modifying the pg_hba.conf File

The `pg_hba.conf` file manages connections for the Postgres server. You must ensure that the `pg_hba.conf` file on each monitored server allows connections from the PEM server, the monitoring PEM Agent, and the host of the PEM-HTTPD server.

For information about modifying the `pg_hba.conf` file, see the PEM Administrator's Guide.

Information about managing authentication is also available in the Postgres core documentation.

Firewall Restrictions

Please note that you must adjust your firewall to allow communication between PEM components.

Supported Locales

Currently, the PEM server and web interface support a locale of `English(US) en_US` and use of a period (.) as a language separator character. Using an alternate locale, or a separator character other than a period may result in errors.

3 PEM Architecture

Postgres Enterprise Manager (PEM) is a tool designed to monitor and manage multiple Postgres servers through a single GUI interface. PEM is capable of monitoring the following areas of the infrastructure:

Note: The term Postgres refers to either PostgreSQL or EDB Postgres Advanced Server.

- Hosts - One or more servers (physical or virtual) and their operating systems.
- Servers - One or more instances of PostgreSQL or EDB Postgres Advanced Server running on a host.
- Databases - One or more databases and the schema objects (tables, indexes, etc.) within them.

PEM consists of a number of individual software components; the individual components are described below.

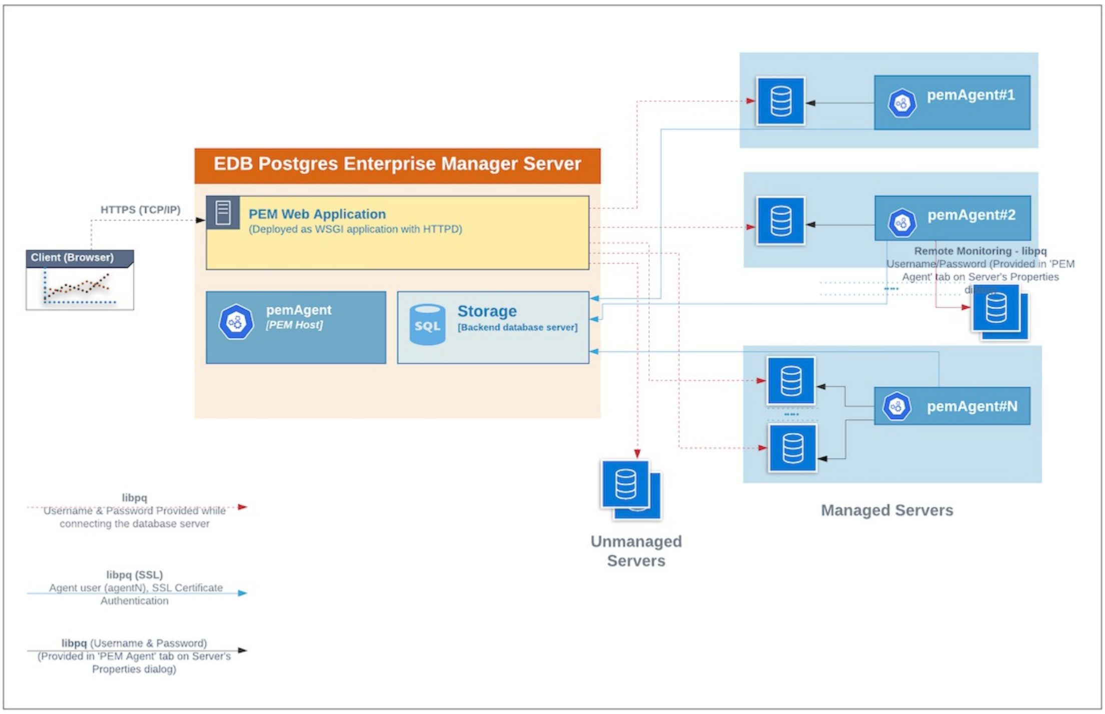
- PEM Server - The PEM Server is used as the data repository for monitoring data and as a server to which both Agents and Clients connect. The PEM server consists of an instance of PostgreSQL and an associated database for storage of monitoring data, and a server that provides web services.
- PEM Agent - The PEM Agent is responsible for executing tasks and reporting statistics from the Agent host and

monitored Postgres instances to the PEM server. A single PEM Agent can monitor multiple installed instances of Postgres that reside on one or many hosts.

- PEM Web Client - The PEM web interface allows you to manage and monitor Postgres servers and utilize PEM extended functionality. The web interface software is installed with the PEM server and is accessed via any supported web browser.
- SQL Profiler - SQL Profiler is a Postgres server plugin to record the monitoring data and query plans to be analysed by the SQL Profiler tool in PEM. This is an optional component of PEM, but the plugin must be installed into each instance of Postgres with which you wish to use the SQL Profiler tool. The SQL Profiler may be used with any supported version of an EDB distribution of a PostgreSQL server or Advanced Server (not just those managed through the PEM server). See the PEM SQL Profiler Configuration Guide for details and supported versions.

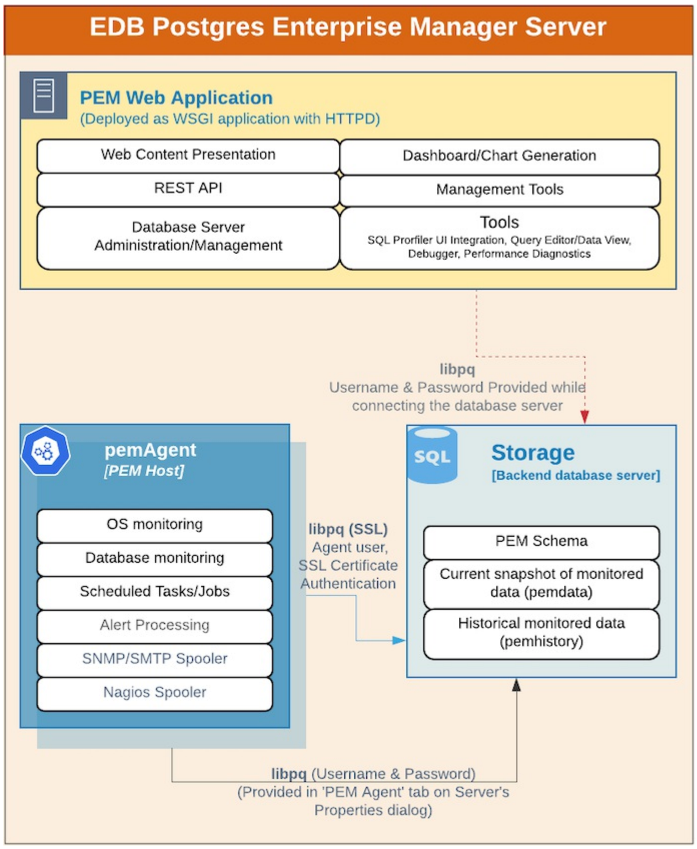
PEM architecture

The following architectural diagram illustrates the relationships between the PEM server, clients, and managed as well as unmanaged Postgres servers.


EDB Postgres Enterprise Manager Server
HTTPS (TCP/IP)
Client (Browser)
PEM Web Application
(Deployed as WSGI application with HTTPD)
pemAgent
[PEM Host]
SQL
Storage
[Backend database server]
pemAgent#1
pemAgent#2
Remote Monitoring - libpq
Username/Password (Provided in 'PEM Agent' tab on Server's Properties dialog)
pemAgent#N
Managed Servers
Unmanaged Servers
libpq
Username & Password Provided while connecting the database server
libpq (SSL)
Agent user (agentN), SSL Certificate Authentication
libpq (Username & Password)
(Provided in 'PEM Agent' tab on Server's Properties dialog)

PEM Architecture

The PEM Server


EDB Postgres Enterprise Manager Server
PEM Web Application
(Deployed as WSGI application with HTTPD)
Web Content Presentation
Dashboard/Chart Generation
REST API
Management Tools
Database Server Administration/Management
Tools
SQL Profiler UI Integration, Query Editor/Data View, Debugger, Performance Diagnostics
libpq
Username & Password Provided while connecting the database server
pemAgent
[PEM Host]
OS monitoring
Database monitoring
Scheduled Tasks/Jobs
Alert Processing
SNMP/SMTP Spooler
Nagios Spooler
libpq (SSL)
Agent user, SSL Certificate Authentication
SQL
Storage
[Backend database server]
PEM Schema
Current snapshot of monitored data (pemdata)
Historical monitored data (pemhistory)
libpq (Username & Password)
(Provided in 'PEM Agent' tab on Server's Properties dialog)

PEM Server

The PEM server consists of an instance of Postgres, an instance of the Apache web-server providing web services to the client, and a PEM Agent. PEM utilizes a server-side cryptographic plugin to generate authentication certificates.

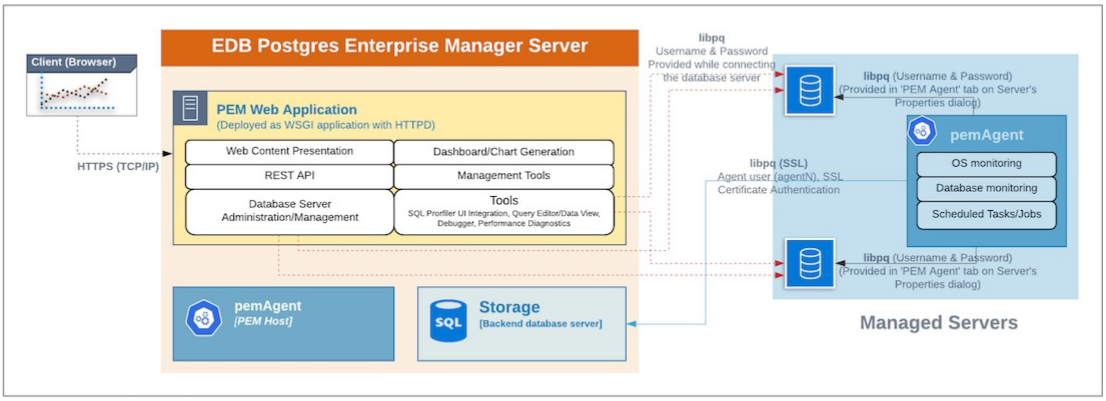
The instance of Postgres (a database server) and an instance of the Apache web-server (HTTPD) can be on the same host or on separate hosts.

- Postgres Instance (Database server) - This is the backend database server. It hosts a database namedpem which acts as the repository for PEM Server. The pem database contains several schemas that store metric data collected from each monitored host, server, and database.
  - pem - This schema is the core of the PEM application. It contains the definitions of configuration functions, tables, or views required by the application.
  - pemdata - This schema stores the current snapshot of the monitored data.
  - pemhistory - This schema stores the historical monitored data.

- Apache Web Server (HTTPD) - The PEM Web Application is deployed as a WSGI application with HTTPD to provide web services to the client. It is comprised of the following:
  - Web content presentation - The presentation layer is created by the Web Application (for example Browser, login page,..).
  - Rest API - The REST API allows integration with other apps and services.
  - Database Server Administration/Management - Database server administration and management activities like CREATE, ALTER, DROP, etc. can be performed for managed as well as unmanaged servers.
  - Dashboard/Chart generation - Internally, the web application includes functionality that generates Dashboards and Charts.
  - Management Tools - The Audit Manager, Capacity Manager, Log Manager, Postgres Expert, Postgres Log Analysis Expert, and the Tuning Wizard are made available in the Web Application.
  - Other tools provide functionality on managed or unmanaged servers:
    - SQL Profiler UI Integration - SQL Profiler generates easily analyzed traces of session content.
    - Query Editor/Data View - The Query editor allows you to query, edit, and view data.
    - Debugger - The Debugger helps you debug queries.
    - Performance Diagnostics - Performance Diagnostics help you analyze the performance of Postgres instances.

We recommend that you use a dedicated machine to host production instances of the PEM backend database. The host may be subject to high levels of data throughput, depending on the number of database servers that are being monitored and the workloads the servers are processing.

The PEM Agent


Client (Browser)
HTTPS (TCP/IP)
EDB Postgres Enterprise Manager Server
PEM Web Application
(Deployed as WSGI application with HTTPD)
Web Content Presentation
Dashboard/Chart Generation
REST API
Management Tools
Database Server Administration/Management
Tools
SQL Profiler UI Integration, Query Editor/Data View, Debugger, Performance Diagnostics
pemAgent
[PEM Host]
SQL
Storage
[Backend database server]
libpq
Username & Password
Provided while connecting the database server
libpq (SSL)
Agent user (agentN), SSL Certificate Authentication
libpq (Username & Password)
(Provided in 'PEM Agent' tab on Server's Properties dialog)
pemAgent
OS monitoring
Database monitoring
Scheduled Tasks/Jobs
libpq (Username & Password)
(Provided in 'PEM Agent' tab on Server's Properties dialog)
Managed Servers

PEM Agent

The PEM Agent is responsible for the collection of monitoring data from the machine and operating system, as well as from each of the Postgres instances to which they are bound. Each PEM Agent can monitor one physical or virtual machine and is capable of monitoring multiple database servers locally - installed on the same system, or remotely - installed on other systems. It is also responsible for executing other tasks that may be scheduled by the user (for example, server shutdowns, SQL Profiler traces, user-defined jobs).

A PEM Agent is installed by default on the PEM Server along with the installation of the PEM Server. It is generally referred to as a PEM Agent on the PEM Host. Separately, the PEM Agent can also be installed on the other servers hosting the Postgres instances to be monitored using PEM.

Whether monitoring locally or remotely, the PEM Agent connects to the PEM Server using PostgreSQL's libpq, using SSL certificate-based authentication. The PEM Agent installer in Windows and pemworker CLI in Linux is responsible for registering each agent with the PEM Server, and generating and installing the required certificates.

Please note that there is only one-way traffic between the PEM Agent and PEM Server; the PEM Agent always connects to the PEM Server.

The PEM Agent must be able to connect to each database server that it monitors. This connection is made over a TCP/IP connection (or optionally a Unix Domain Socket on Unix hosts), and may optionally use SSL. The user must configure the connection and authentication to the monitored server.

Once configured, each agent collects statistics and other information on the host and each database server and database that it monitors. Each piece of information is known as a metric and is collected by a probe. Most probes will collect multiple metrics at once for efficiency. Examples of the metrics collected include:

- Disk I/O statistics
- Network statistics
- Database server version string
- Database server configuration option (GUC) values
- Table access statistics
- Table and index sizes

A list of PEM probes can be found here.

By default, the PEM Agent bound to the database server collects the OS/Database monitoring statistics and also runs any scheduled tasks/jobs for that particular database server, storing data in the pem database on the PEM server.

The Alert processing, SNMP/SMTP spoolers, and Nagios Spooler data is stored in the `pem` database on the PEM server and is then processed by the PEM Agent on the PEM Host by default. However, processing by other PEM Agents can be enabled by adjusting the SNMP/SMTP and Nagios parameters of the PEM Agents.

To see more information about these parameters see Server Configuration.

The PEM Web Client

The PEM client is a web-based application that runs in supported browsers. The client's web interface connects to the PEM server and allows direct management of managed or unmanaged servers, and the databases and schemas that reside on them.

The client allows you to use PEM functionality that makes use of the data logged on the server through features such as the dashboards, the Postgres Log Analysis Expert, and Capacity Manager.

The SQL Profiler Plugin

You are not required to install the SQL Profiler plugin on every server, but you must install and configure the plugin on each server on which you wish to use the SQL Profiler. You may also want to install and configure SQL Profiler on un-monitored development servers. For ad-hoc use also, you may temporarily install the SQL Profiler plugin.

The plugin is installed with the EDB Postgres Advanced Server distribution but must be installed separately for use with PostgreSQL. The SQL Profiler installer is available from the EDB website.

SQL Profiler may be used on servers that are not managed through PEM, but to perform scheduled traces, a server must have the plugin installed, and must be managed by an installed and configured PEM agent.

For more information about using SQL Profiler, see the PEM SQL Profiler Guide.

4 Installing Postgres Enterprise Manager

The `edb-pem` package for Linux platforms installs the PEM Server, a PEM Agent, and the required software to connect to the PEM web interface with a supported browser.

The PEM server uses a Postgres installation and backend database to manage data. The `pem` backend database is created when you configure PEM.

For detailed information about installing the PEM Server, see Installing the PEM Server on Linux. For information about configuring a PEM Server see Configuring the PEM Server on Linux.

The PEM Agent that is installed with the PEM server is capable of monitoring multiple servers that reside on the same host, or on remote hosts. Please note that the PEM functionality on servers monitored by a remote Agent may be limited. For detailed information about remote monitoring functionality see the PEM Agent Privileges.

For detailed information about installing and configuring a PEM Agent, see Installing the PEM Agent on Linux.

4.1 Prerequisites for Installing the PEM Server on Linux Platforms

1. Install a backend database.

   When installing a PEM server on a Linux host, you must first install a backend database cluster which will hold the `pem` database. The PEM server's backend database may be installed via package for Linux. The backend database must be one of the following versions:

   - EDB Postgres Advanced Server version 11 or above
   - PostgreSQL version 11 or above

   For detailed information about installing an Advanced Server or PostgreSQL database, please see the product documentation at the EDB website.

2. Configure Postgres authentication on the backend database.

   The `pg_hba.conf` file on the backend database can be configured to use any supported authentication methods (for example: md5, trust,...) for connections. For information about modifying the `pg_hba.conf` file, see the PostgreSQL core documentation.

3. If you are using a PostgreSQL database, use the following command to install the `hstore contrib module`:

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

   Where, `x` is the server version.

4. Ensure that the `sslutils` extension is installed.

   - On an Advanced Server backend database, the `sslutils` extension is installed by default.
   - If you are using a PostgreSQL backend database, ensure you have access to the PostgreSQL community repository, and use the command:

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

   Where, `x` is the server version.

   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 `ssltuils` 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`.

5. Adjust your firewall restrictions.

   If you are using a firewall, you must allow access to port `8443` on the PEM backend database:

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

   firewall-cmd --reload
   ```

6. Request credentials that allow you to access the EDB repositories: To install the PEM Server, you must have credentials that allow access to the EDB repository. To request credentials for the repository, contact EDB . When using commands in the sections that follow, replace the `username` and `password` placeholders with the credentials provided by EDB

7. PEM is dependent on third-party components from the vendor repository, including the python3, libboost, openssl, snmp++, libcurl, etc. To ensure these components are up to date, you should update your operating system using following platform-specific commands. Minimum version require for openssl is 1.0.2k.

   To upgrade packages on a CentOS or RHEL 7.x host

   ```
   yum upgrade
   ```

   To upgrade packages on a CentOS or RHEL 8.x host

   ```
   dnf upgrade
   ```

   To upgrade packages on a Debian or Ubuntu host

   ```
   apt-get update
   ```

   To upgrade packages on a SLES host

   ```
   zypper update
   ```

4.2 Web Server Hosting Preferences

During the PEM server installation, you can specify your hosting preferences for the Apache Web Server(PEM-HTTPD):

To install the PEM Server and Apache Web Server (PEM-HTTPD) on the same host

Follow the installation steps; while running the configuration script, select the `Web Services and Database` option to install PEM Server and Apache Web Server on the same host.

To install the PEM Server and Apache Web Server (PEM-HTTPD) on separate hosts

Follow the installation steps on both the hosts. While running the configuration script, first configure the PEM Server host by selecting the `Database` option on first host and then configure an Apache Web Server (PEM-HTTPD) by selecting the `Web Services` option on the second host.

For detailed information about configuring a PEM Server, see Configuring the PEM Server on Linux Platforms.

4.3 Installing the PEM Server on Linux Platforms

Before following the detailed instructions that install the PEM server on your specific platform, you must perform the prerequisite steps detailed in Prerequisites for installing PEM Server.

Installing the PEM Server on a CentOS or RHEL Host

On a CentOS or RHEL system, you can use the `yum` package manager or `dnf` command to install a PEM Server; the installation tool you use will be dependent on the version of the host operating system. Before installing the server, you must ensure that your system contains the required prerequisite software listed below.

Install Version-Specific Software

Follow the version-specific instructions listed below to prepare your host system.

- On a CentOS or RHEL 7.x host:

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

- On a CentOS or RHEL 8.x host:

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

!!! Note You may need to enable the `[extras]` repository definition in the `CentOS-Base.repo` file (located in `/etc/yum.repos.d`).

If you are a Red Hat Network user

You must enable the `rhel-<x>-server-extras-rpms` repository, where `x` specifies the RHEL version.

You must also enable the `rhel-<x>-server-optional-rpms` repository to use EPEL packages, where *x* specifies the RHEL version. You can make the repository accessible by enabling the `RHEL optional subchannel` for `RHN-Classic`. If you have a certificate-based subscription, then you must also enable `rhel-<x>-server-eus-optional-rpms` repository to use EPEL packages; please see the `Red Hat Subscription Management Guide` for more information about the required repositories.

Install and Configure the edb.repo File

To create an EnterpriseDB repository configuration file, assume superuser privileges and invoke the following command:

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

The repository configuration file is named `edb.repo`. The file resides in `/etc/yum.repos.d`. After creating the `edb.repo` file, use the following command to replace the `USERNAME` and `PASSWORD` placeholders in the `baseurl` specification with the <username> and <password> of a registered EDB user:

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

Install the PEM Server

After meeting the platform-specific prerequisites listed above, you can use yum or dnf to install the PEM Server:

- On CentOS or RHEL 7.x or 8.x, you can use `yum` to install the PEM server :

  ```
  yum install edb-pem
  ```

- On CentOS or RHEL 8.x, you can use `dnf` to install the PEM Server:

  ```
  dnf install edb-pem
  ```

If you are doing a fresh installation of the PEM Server on a CentOS or RHEL 7.x host, the installer will also install `edb-python3-mod_wsgi` packages along with the installation as per requirement of the operating system.

If you are upgrading the PEM Server on a CentOS or RHEL 7.x host, the `mod_wsgi system` package will be replaced by the `edb-python3-mod_wsgi` package as per the requirement of the operating system.

When you install an RPM package that is signed by a source that is not recognized by your system, yum may ask for your permission to import the key to your local server. If prompted, and you are satisfied that the packages come from a trustworthy source, enter `y`, and press `Return` to continue.

During the installation, yum may encounter a dependency that it cannot resolve. If it does, it will provide a list of the required dependencies that you must manually resolve.

If you want to install PEM server on a machine that is in isolated network, you must first create PEM repository on that machine. For more information about creating PEM repository on an isolated network, see Creating a PEM repository in an Isolated Network.

After installing PEM Server using `yum` or `dnf`, you need to configure the PEM Server. For more detailed information see Configuring the PEM Server on Linux platforms.

Installing the PEM Server on a Debian or Ubuntu Host

The following steps will walk you through using the EDB `apt` repository to install a Debian package.

!!! Note You can also visit https://repos.enterprisedb.com/ and select the platform and product to view the steps for installation.

1. Log in as root on your Debian or Ubuntu host:

   ```
   sudo su -
   ```

2. Configure the EDB repository:

   - For Debian 9:

     ```
     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'
     ```

   Where `username:password` is to be replaced by the credentials provided by EDB.

   - For Debian 10:

     a. Set up the EnterpriseDB repository:

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

     b. Substitute your EnterpriseDB credentials for the `username` and `password` placeholders in the following command:

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

3. Add support to your system for secure APT repositories:

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

4. Add the EBD signing key:

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

5. Update the repository metadata:

   ```
   apt-get update
   ```

6. Use the following command to install the Debian package for the PEM server:

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

When the installation completes, you must configure the PEM Server. For detailed information see Configuring the PEM Server on Linux Platforms.

Installing the PEM Server on a SLES Host

Use the following command to add the EDB repository configuration files to your SLES host:

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

The command creates a repository configuration file named `edb.repo` in the `/etc/zypp/repos.d` directory. Modify the repository configuration file, adding the username and password of a registered EDB user.

Before installing PEM, you must install prerequisite packages. Invoke the following commands, replacing *sp_no* with the service pack that you are using (i.e. SP4):

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

SUSEConnect -p sle-sdk/12/x86_64

zypper addrepo
https://download.opensuse.org/repositories/Apache:Modules/SLE_12_<sp_no>/Apache:Modul
o

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

Then, refresh the repository and install the PEM server:

```
zypper refresh

zypper install edb-pem
```

After installing PEM Server using `zypper`, you must configure the PEM Server. For detailed information see Configuring the PEM Server on Linux Platforms.

4.4 Creating a PEM Repository on an Isolated Network

You can create a local repository to act as a host for the PEM RPM packages if the server on which you wish to upgrade PEM cannot directly access the EnterpriseDB repository. Please note that this is a high-level overview of the steps required; you may need to modify the process for your individual network. To create and use a local repository, you must:

1. Use the following commands on a system with Internet access to download the dependencies for PEM:

   ```
   yum install yum-plugin-downloadonly

   mkdir /<pem_dir>

   yum install --downloadonly --downloaddir=/<pem_dir>/ edb-pem

   mkdir /<epel_dir>

   yum install --downloadonly --downloaddir=/<epel_dir>/ epel-release*
   ```

   Where `<pem_dir>` and `<epel_dir>` are the local directories that you create for downloading the RPMs.

2. Copy the directories `/<pem_dir>` and `/<epel_dir>` to the machine that is in the isolated network.

3. Create the repositories:

   ```
   createrepo /<pem_dir>

   createrepo /<epel_dir>
   ```

4. Create a repository configuration file called `/etc/yum.repos.d/pem.repo` with connection information that specifies:

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

5. Create a repository configuration file called `/etc/yum.repos.d/epel.repo` with connection information that specifies:

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

6. After specifying the location and connection information for your local repository, you can use `yum` commands to install or upgrade PEM server:

   To install PEM server:

   ```
   yum install edb-pem
   ```

   To upgrade PEM server:

   ```
   yum upgrade edb-pem
   ```

For more information about creating a local yum repository, visit: https://wiki.centos.org/HowTos/CreateLocalRepos

4.5 Configuring the PEM Server on Linux Platforms

Before configuring the PEM server, ensure that the `sslutils` extension and `hstore` contrib module are installed on your backend database.

- For an Advanced Server backend database, the `sslutils` extension and `hstore` contrib module are by default installed along with Advanced Server.
- For a PostgreSQL backend database, ensure you have access to the PostgreSQL community repository, and then install `sslutils` extension and `hstore` contrib module with the command:

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

  Where, `x` is the server version.

The PEM server package includes a script (`configure-pem-server.sh`) to help automate the configuration process for Linux platform installations. The script is installed in the `/usr/edb/pem/bin` directory. To invoke the script, use the command:

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

When invoking the script, you can include command line options to specify configuration properties; the script will prompt you for values that you omit on the command line. The accepted options are:

!!! Note If you are using the SSL certificates then make sure that all the SSL certificates are inside the data directory the backend database server. If the certificates are not inside the data directory then the PEM Server's configure script may fail as it looks in to the data directory while configuring the PEM Server.

Option	Description
`-acp`	Defines PEM Agent certificate path. The default is `/root/.pem`.
`-ci`	CIDR formatted network address range that Agents will connect to the server from, to be added to the server's `pg_hba.conf` file. For example, `192.168.1.0/24`. The default is `0.0.0.0/0`.
`-dbi`	The directory for the database server installation. For example, `/usr/edb/as12` for Advanced Server or `/usr/pgsql-12` for PostgreSQL.
`-ds`	The unit file name of the PEM database server. For Advanced Server, the default file name is `edb-as-12`; for PostgreSQL, it is `postgresql-12`.
`-ho`	The host address of the PEM database server.
`-p`	The port number of the PEM database server.
`-ps`	The service name of the pemagent; the default value is `pemagent`.
`-sp`	The superuser password of the PEM database server. This value is required.
`-su`	The superuser name of the PEM database server.
`-t`	The installation type: Specify 1 if the configuration is for web services and backend database, 2 if you are configuring web services, or 3 if you are configuring the backend database. If you specify 3, please note that the database must reside on the local host.
`-un`	To unregister the PEM Server.

If you do not provide configuration properties on the command line, you will be prompted for values by the script. When you invoke the script, choose from:

1. `Web Services and Database` -Select this option if the web server and database both reside on the same host as the PEM server.

2. `Web Services` -Select this option if the web server resides on a different host than the PEM server.

3. `Database` -Select this option to configure the PEM backend database for use by the PEM server. Please note that the specified database must reside on the local host.

!!! Note If the web server (PEM-HTTPD) and the backend database (PEM Server) reside on separate hosts, configure the database server first (option 3), and then web services (option 2). The script will exit if the backend database is not configured before web services.

After selecting a configuration option, the script will proceed to prompt you for configuration properties. When the script completes, it will create the objects required by the PEM server, or perform the configuration steps required. `1` To view script-related help, use the command:

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

After configuring the PEM server, you can access the PEM web interface in your browser. Navigate to:

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

4.6 Installing a PEM Agent on Linux Platforms

A PEM Agent may monitor one or more servers on one or more hosts. For comprehensive information about managing a PEM Agent, see the PEM Agent User Guide.

Installing a PEM Agent on a CentOS or RHEL Host

On a CentOS or RHEL system, you can use the `yum` package manager or `dnf` command to install a PEM Agent; the installation tool you use will be dependent on the version of the host operating system. Before installing the agent, you must ensure that your system contains the required prerequisite software listed below.

Install Platform-Specific Software

- On a CentOS or RHEL 7.x host:

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

- On a CentOS or RHEL 8.x host:

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

!!! Note You may need to enable the `[extras]` repository definition in the `CentOS-Base.repo` file (located in `/etc/yum.repos.d`).

If you are a Red Hat Network user

You must enable the `rhel-<x>-server-extras-rpms` repository, where `x` specifies the RHEL version.

You must also enable the `rhel-<x>-server-optional-rpms` repository to use EPEL packages, where *x* specifies the RHEL version. You can make the repository accessible by enabling the `RHEL optional subchannel` for `RHN-Classic`. If you have a certificate-based subscription, then you must also enable `rhel-<x>-server-eus-optional-rpms` repository to use EPEL packages; please see the `Red Hat Subscription Management Guide` for more information about the required repositories.

Install and Configure the edb.repo File

To create an EnterpriseDB repository configuration file, assume superuser privileges and invoke the following command:

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

The repository configuration file is named `edb.repo`. The file resides in `/etc/yum.repos.d`. After creating the `edb.repo` file, use the following command to replace the `USERNAME` and `PASSWORD` placeholders in the `baseurl` specification with the <username> and <password> of a registered EDB user:

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

Installing the PEM Agent

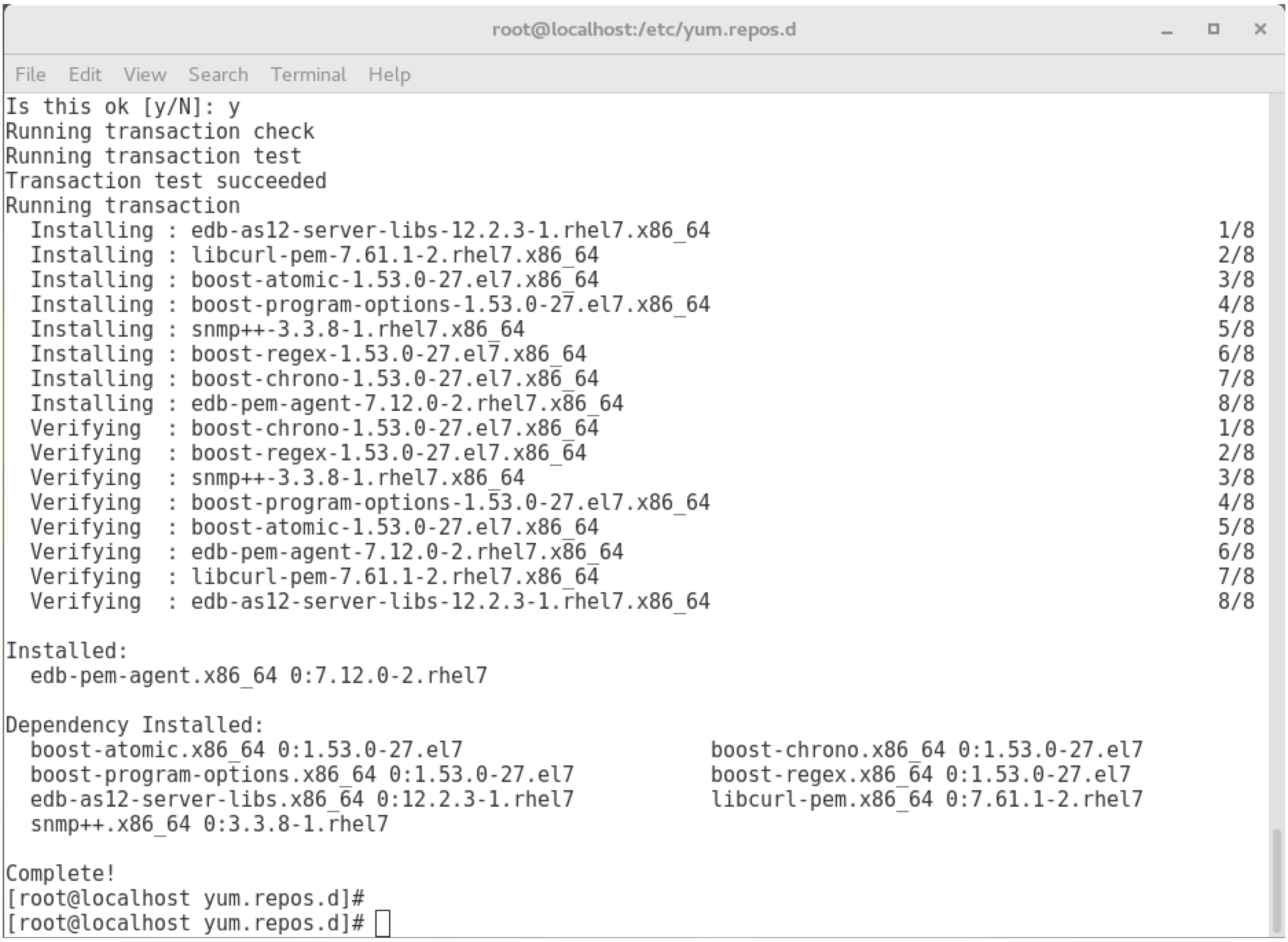
- Use the `yum install` command to install the PEM Agent on CentOS or RHEL 7.x or 8.x:

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

- Also, you can use `dnf` command to install the PEM Agent on CentOS or RHEL 8.x:

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

When the installation is complete, you can review a list of the installed packages and dependencies.


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]#
```

When you install an RPM package that is signed by a source that is not recognized by your system, yum may ask for your permission to import the key to your local server. If prompted, and you are satisfied that the packages come from a trustworthy source, enter `y`, and press `Return` to continue.

During the installation, yum may encounter a dependency that it cannot resolve. If it does, it will provide a list of the required dependencies that you must manually resolve.

After installing PEM Agent using `yum` or `dnf`, you need to register the PEM Agent. For detailed information see Registering a PEM Agent.

Installing a PEM Agent on a Debian or Ubuntu Host

To install PEM on a Debian or Ubuntu host, you must have credentials that allow access to the EDB repository. To request credentials for the repository, contact EDB .

The following steps will walk you through using the EDB apt repository to install a Debian package. When using the commands, replace the username and password with the credentials provided by EDB.

1. Log in as root:

   ```
   sudo su -
   ```

2. Configure the EDB repository:

   - For Debian 9:

```
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'
```

Where `username:password` is to be replaced by the credentials provided by EDB.

- For Debian 10:

  a. Set up the EnterpriseDB repository:

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

  b. Substitute your EnterpriseDB credentials for the `username` and `password` placeholders in the following command:

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

3. Add support to your system for secure APT repositories:

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

4. Add the EBD signing key:

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

5. Update the repository metadata:

   ```
   apt-get update
   ```

6. Use the following command to install the Debian package for the PEM Agent:

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

After installing PEM Agent using `apt-get`, you need to register the PEM Agent. For more detailed information see Registering a PEM Agent.

Installing a PEM Agent on a SLES Host

Use the following command to add the EDB repository configuration files to your SLES host:

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

The command creates a repository configuration file named `edb.repo` in the `/etc/zypp/repos.d` directory. Modify the repository configuration file, adding the username and password of a registered EDB user.

Before installing PEM, you must install prerequisite packages. Use the following commands replacing *sp_no* with the service pack that you are using (i.e. SP2 or SP3):

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

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

```
zypper addrepo
https://download.opensuse.org/repositories/Apache:Modules/SLE_12_<sp_no>/Apache:Modul
o

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

Then, you can refresh the repository and add a PEM agent:

```
zypper refresh

zypper install edb-pem-agent
```

After installing the PEM Agent, you must register the agent. For more detailed information see Registering a PEM Agent.

4.7 Registering a PEM Agent

Each PEM Agent must be *registered* with the PEM Server. The registration process provides the PEM server with the information it needs to communicate with the Agent. You can use the `pemworker` utility to register the Agent if you use the package to install a PEM Agent.

The PEM Agent package places the PEM Agent in the `/usr/edb/pem/agent/bin` directory. To register an Agent, include the `--register-agent` keywords along with registration details when invoking the `pemworker` utility:

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

Append command line options to the command string when invoking the pemworker utility. Each option should be followed by a corresponding value:

Option	Description
`--pem-server`	Specifies the IP address of the PEM backend database server. This parameter is required.
`--pem-port`	Specifies the port of the PEM backend database server. The default value is 5432.
`--pem-user`	Specifies the name of the Database user (having superuser privileges) of the PEM backend database server. This parameter is required.
`--pem-agent-user`	Specifies the Agent user to connect the PEM server backend database server.
`--cert-path`	Specifies the complete path to the directory in which certificates will be created. If you do not provide a path, certificates will be created in: On Linux, ~/.pem On Windows, %APPDATA%/pem
`--config-dir`	Specifies the directory path where configuration file can be found. The default is the <pemworker path>/../etc.
`--display-name`	Specifies a user-friendly name for the Agent that will be displayed in the PEM Browser tree control. The default is the system hostname.
`--force-registration`	Include the force_registration clause to instruct the PEM server to register the Agent with the arguments provided; this clause is useful if you are overriding an existing Agent configuration. The default value is Yes.
`--group`	The name of the group in which the Agent will be displayed.

Option	Description
`--team`	The name of the database role, on the PEM backend database server, that should have access to the monitored database server.
`--owner`	The name of the database user, on the PEM backend database server, who will own the Agent.
`--allow_server_restart`	Enable the allow-server_restart parameter to allow PEM to restart the monitored server. The default value is True.
`--allow-batch-probes`	Enable the allow-batch-probes parameter to allow PEM to run batch probes on this Agent. The default value is False.
`--batch-script-user`	Specifies the operating system user that should be used for executing the batch/shell scripts. The default value is none; the scripts will not be executed if you leave this parameter blank or the specified user does not exist.
`--enable-heartbeat-connection`	Enable the enable-heartbeat-connection parameter to create a dedicated heartbeat connection between PEM Agent and server to update the active status. The default value is False.
`--enable-smtp`	When set to true for multiple PEM Agents (7.13 or lesser) it may send more duplicate emails. Whereas for PEM Agents (7.14 or higher) it may send lesser duplicate emails.
`--enable-snmp`	When set to true for multiple PEM Agents (7.13 or lesser) it may send more duplicate traps. Whereas for PEM Agents (7.14 or higher) it may send lesser duplicate traps.
`-o`	Specify if you want to override the configuration file options.

If you want to use any PEM feature for which a database server `restart` is required by the pemagent (such as Audit Manager, Log Manager, or the Tuning Wizard), then you must set the value for `allow_server_restart` to `true` in the `agent.cfg` file.

!!! Note When configuring a shell/batch script run by a PEM Agent that has PEM version 7.11 or later installed, the user for the `batch_script_user parameter` must be specified. 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 the `root` user to run the scripts, then the `batch_script_user` parameter value must be set to `root`.

Before any changes are made on the PEM database, the connecting agent is authenticated with the PEM database server. When invoking the `pemworker` utility, you must provide the password associated with the PEM server administrative user role (`postgres`). There are three ways to specify the administrative password; you can:

- set the `PEM_MONITORED_SERVER_PASSWORD` environment variable.
- provide the password on the command line with the `PGPASSWORD` keyword.
- create an entry in the `.pgpass` file.

Failure to provide the password will result in a password authentication error; you will be prompted for any other required but omitted information. When the registration is complete, the server will confirm that the Agent has been successfully registered.

Setting PEM Agent Configuration Parameters

The PEM Agent native package creates a sample configuration file named `agent.cfg.sample` in the `/usr/edb/pem/agent/etc` directory. When you register the PEM Agent, the `pemworker` program creates the actual Agent configuration file (named `agent.cfg`). Modify the `agent.cfg` file, changing the following configuration parameter value to `true`:

```
heartbeat_connection = true
```

By default, `heartbeat_connection` value is `false` but you can override the value during pemagent registration with `pemworker` utility using the `--enable-heartbeat-connection` option.

Then, use a platform-specific command to start the PEM Agent service;

On a CentOS or RHEL 7.x or 8.x host, use `systemctl` to start the service:

```
systemctl start pemagent
```

The service will confirm that it is starting the Agent; when the Agent is registered and started, it will be displayed on the `Global Overview` and in the `Object browser` of the PEM web interface.

For information about using the `pemworker` utility to register a server, please see the PEM Administrator's Guide

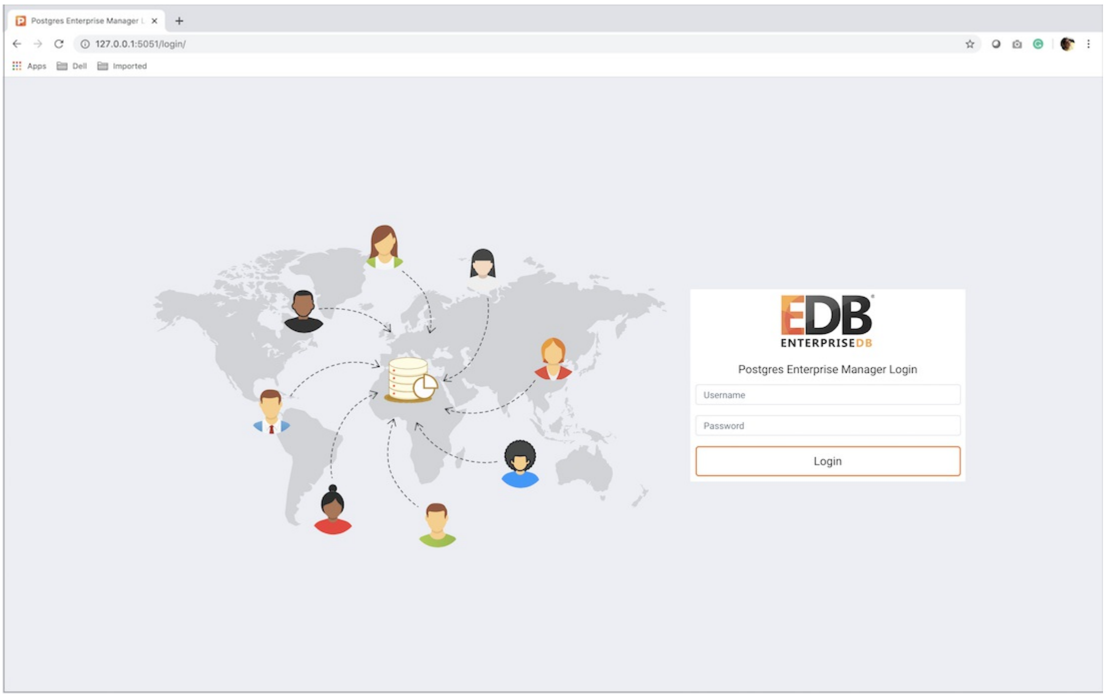
5 The PEM Web Interface

After installing a PEM server and Agent, you can configure PEM to start monitoring and managing PostgreSQL or Advanced Server instances. The PEM server installer installs the PEM web interface. You can use the interface to review information about objects that reside on monitored servers, or to review statistical information gathered by the PEM server.

After installing and configuring PEM, you can use your browser to access the PEM web interface. Open your browser, and navigate to:

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

Where `ip_address_of_PEM_host` specifies the IP address of the host of the PEM server. The `Postgres Enterprise Manager Web Login` window opens:

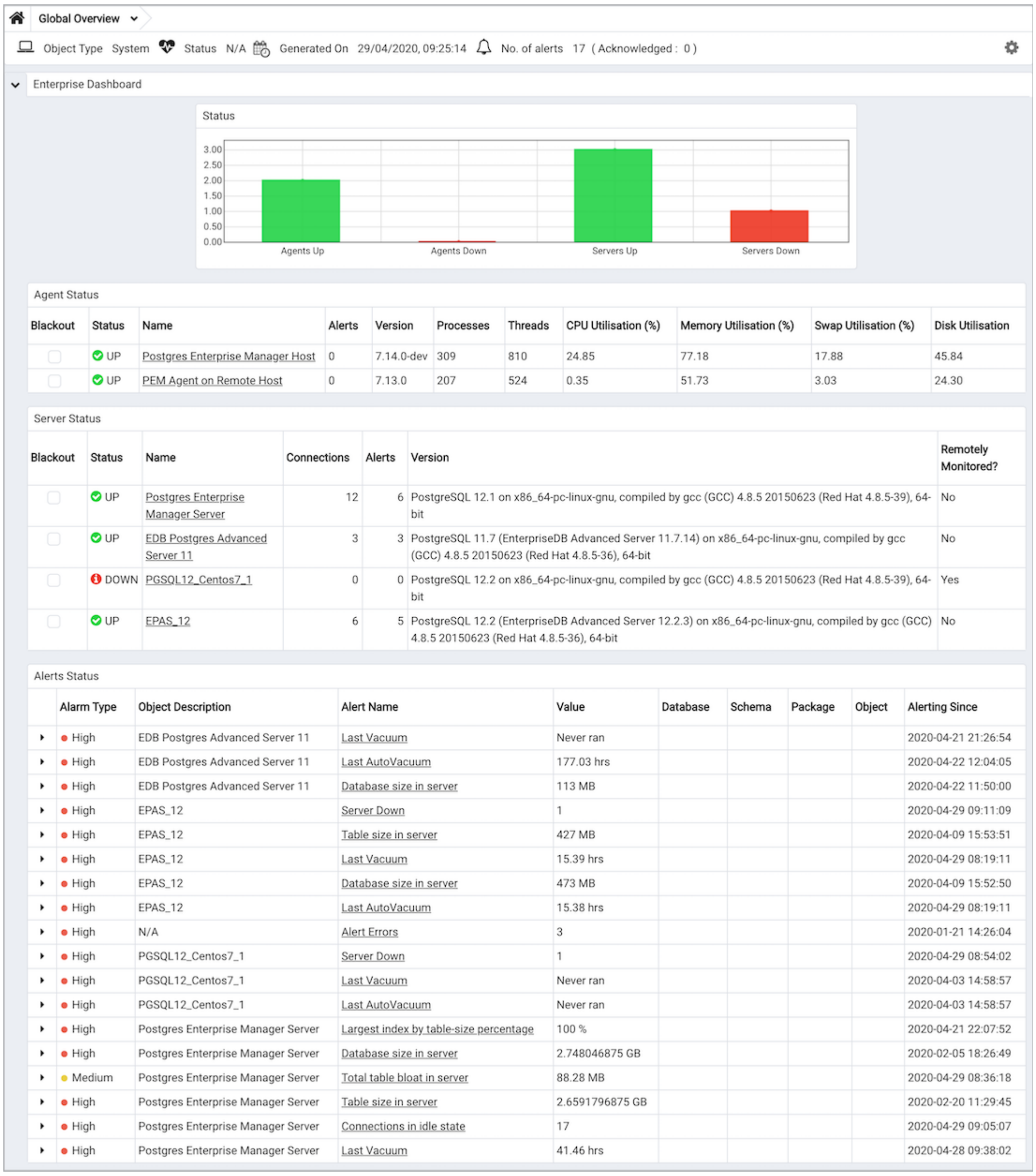


Use the fields on the `Postgres Enterprise Manager Login` window to authenticate yourself with the PEM server:

- Provide the name of a `pem` database user in the `Username` field. For the first user connecting, this will be the name provided when installing the PEM server.

- Provide the password associated with the user in the `Password` field.

Click the `Login` button to connect to the PEM server.



Object Type System Status N/A Generated On 29/04/2020, 09:25:14 No. of alerts 17 (Acknowledged : 0)

Agent Status

Blackout	Status	Name	Alerts	Version	Processes	Threads	CPU Utilisation (%)	Memory Utilisation (%)	Swap Utilisation (%)	Disk Utilisation
□	✓ UP	Postgres Enterprise Manager Host	0	7.14.0-dev	309	810	24.85	77.18	17.88	45.84
□	✓ UP	PEM Agent on Remote Host	0	7.13.0	207	524	0.35	51.73	3.03	24.30

Server Status

Blackout	Status	Name	Connections	Alerts	Version	Remotely Monitored?
□	✓ UP	Postgres Enterprise Manager Server	12	6	PostgreSQL 12.1 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-39), 64-bit	No
□	✓ UP	EDB Postgres Advanced Server 11	3	3	PostgreSQL 11.7 (EnterpriseDB Advanced Server 11.7.14) on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-36), 64-bit	No
□	! DOWN	PGSQL12_Centos7_1	0	0	PostgreSQL 12.2 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-39), 64-bit	Yes
□	✓ UP	EPAS_12	6	5	PostgreSQL 12.2 (EnterpriseDB Advanced Server 12.2.3) on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-36), 64-bit	No

Alerts Status

	Alarm Type	Object Description	Alert Name	Value	Database	Schema	Package	Object	Alerting Since
▸	● High	EDB Postgres Advanced Server 11	Last Vacuum	Never ran					2020-04-21 21:26:54
▸	● High	EDB Postgres Advanced Server 11	Last AutoVacuum	177.03 hrs					2020-04-22 12:04:05
▸	● High	EDB Postgres Advanced Server 11	Database size in server	113 MB					2020-04-22 11:50:00
▸	● High	EPAS_12	Server Down	1					2020-04-29 09:11:09
▸	● High	EPAS_12	Table size in server	427 MB					2020-04-09 15:53:51
▸	● High	EPAS_12	Last Vacuum	15.39 hrs					2020-04-29 08:19:11
▸	● High	EPAS_12	Database size in server	473 MB					2020-04-09 15:52:50
▸	● High	EPAS_12	Last AutoVacuum	15.38 hrs					2020-04-29 08:19:11
▸	● High	N/A	Alert Errors	3					2020-01-21 14:26:04
▸	● High	PGSQL12_Centos7_1	Server Down	1					2020-04-29 08:54:02
▸	● High	PGSQL12_Centos7_1	Last Vacuum	Never ran					2020-04-03 14:58:57
▸	● High	PGSQL12_Centos7_1	Last AutoVacuum	Never ran					2020-04-03 14:58:57
▸	● High	Postgres Enterprise Manager Server	Largest index by table-size percentage	100 %					2020-04-21 22:07:52
▸	● High	Postgres Enterprise Manager Server	Database size in server	2.748046875 GB					2020-02-05 18:26:49
▸	● Medium	Postgres Enterprise Manager Server	Total table bloat in server	88.28 MB					2020-04-29 08:36:18
▸	● High	Postgres Enterprise Manager Server	Table size in server	2.6591796875 GB					2020-02-20 11:29:45
▸	● High	Postgres Enterprise Manager Server	Connections in idle state	17					2020-04-29 09:05:07
▸	● High	Postgres Enterprise Manager Server	Last Vacuum	41.46 hrs					2020-04-28 09:38:02

Before you can use the PEM web interface to manage or monitor a database server, you must *register* the server with the PEM server. When you register a server, you describe the connection to the server, provide authentication information for the connection, and specify any management preferences (optionally binding an Agent).

A server may be managed or unmanaged:

- A `managed` server is bound to a PEM Agent. The PEM Agent will monitor the server to which it is bound, and perform tasks or report statistics for display on the PEM dashboards. A managed server has access to extended PEM functionality such as Custom Alerting; when registering a server, you can also allow a managed server to be restarted by PEM as required.
- An `unmanaged` server is not bound to a PEM Agent; you can create database objects on an unmanaged server, but extended PEM functionality (such as Custom Alerting) is not supported on an unmanaged server.

You must also ensure the `pg_hba.conf` file of the server that you are registering allows connections from the host of the PEM web interface.

To access online help information about the PEM web interface, select `Help` from the menu bar. Additional information is available from the EDB website.

6 Uninstalling Postgres Enterprise Manager Components

The process of uninstalling the PEM server or Agent is platform-specific. The name of the package for PEM server is `edb-pem` and for PEM Agent is `edb-pem-agent`.

If you uninstall the PEM server package from a host, the PEM Agent package installed on the same host doesn't get uninstalled. But if you uninstall the PEM Agent package, then the PEM server package installed on the same host also gets uninstalled.

Uninstalling PEM components from CentOS or RHEL Hosts

You can use variations of the `rpm`, `yum remove`, or `yum erase`, commands to remove the installed packages from CentOS or RHEL 7.x or 8.x hosts. Also you use `dnf remove` command to remove the installed package from CentOS or RHEL 8.x host. Note that removing a package does not damage the PEM data directory.

- Include the `-e` option when invoking the rpm command to remove an installed package; the command syntax is:

  ```
  rpm -e package_name
  ```

- You can use the `yum remove` command to remove the PEM Server or Agent package installed by yum. To remove a package, open a terminal window, assume superuser privileges, and enter the command:

  ```
  yum remove package_name
  ```

- You can use the `yum erase` command to remove the pem server or Agent package along with the `edb-pem` and `edb-pem-docs` dependencies. To remove a package, open a terminal window, assume superuser privileges, and enter the command:

  ```
  yum erase package_name
  ```

Where `package_name` is the name of the package that you would like to remove.

- You can use `dnf remove` command to remove the pem server or Agent along with the `edb-pem` and `edb-pem-docs` dependencies on CentOS or RHEL 8.x hosts. To remove a package, open a terminal window, assume superuser privileges, and enter the command:

  ```
  dnf remove package_name
  ```

Uninstalling PEM components from Debian or Ubuntu hosts

You can use `apt-get remove` or `apt-get purge` command to uninstall the PEM server or Agent package from a Debian or Ubuntu host:

- To uninstall PEM server or Agent from a Debian or Ubuntu host without impacting the configuration files and data directories, invoke the following command:

```
apt-get remove package_name
```

- To uninstall PEM server or Agent along with the configuration files and data directory, invoke the following command:

  ```
  apt-get purge package_name
  ```

Where `package_name` is the name of the package that you would like to remove.

Uninstalling PEM components from SLES hosts

To uninstall PEM server or Agent from a SLES host, invoke the following command:

```
zypper remove package_name
```

Where `package_name` is the name of the package that you would like to remove.

7 Reference - Linux Service Script

- A service script allows the PEM server to start, stop or restart the server if necessary when performing configuration management, certificate management, or other administrative tasks.
- The Postgres server on which the PEM server resides must contain a service script. Postgres installers in Windows generated by EDB create a service script for you; if you are using a Postgres server from another source like native packages, you must provide a service script.

  On CentOS or RHEL 7.x or 8.x, the service script resides in the `/usr/lib/systemd/system` directory.
- Service scripts are platform-specific.
- For information about customizing a Postgres service, visit:

  https://www.postgresql.org/docs/current/static/server-start.html