



Postgres Enterprise Manager

リリース 7.16

PEM pgBouncer Configuration Guide

2020 年 10 月 10 日

目次

第 1 章	PEM サーバー-PEM エージェント接続管理メカニズム	2
第 2 章	PEM データベースサーバーの準備	4
第 3 章	PgBouncer の設定	6
第 4 章	PEM エージェントの設定	9
4.1	新しい PEM エージェントの構成 (RPM 経由でインストール)	9
4.2	既存の PEM エージェントの構成 (RPM 経由でインストール)	10
第 5 章	結論	12
	索引	13

このドキュメントは、Windows 以外のマシン上の PEM エージェントから PostgresEnterpriseManager™ (PEM) サーバーへの接続数を制限するための接続プーラーとしての pgBouncer の使用に関する詳細情報を提供します。

- PEM データベースサーバーの準備-第 3 章では、pgBouncer で使用する PEM データベースサーバーの準備について説明します。
- pgBouncer の構成-第 4 章では、PEM データベースサーバーで動作するように pgBouncer を構成する方法について詳しく説明します。
- PEM エージェントの設定-第 5 章では、pgBouncer に接続するための PEM エージェントの設定に関する詳細情報を提供します。

PEM7Web インターフェイスの使用に関する詳細については、***PEM 入門ガイド***を参照してください。

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 サーバー-PEM エージェント接続管理メカニズム

各 PEM エージェントは、各ユーザーの SSL 証明書を使用して PEM データベースサーバーに接続します。たとえば、ID#1 のエージェントは、agent1 ユーザーを使用して PEM データベースサーバーに接続します。

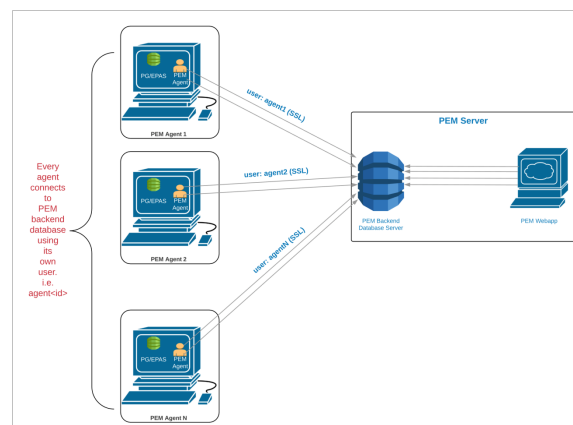
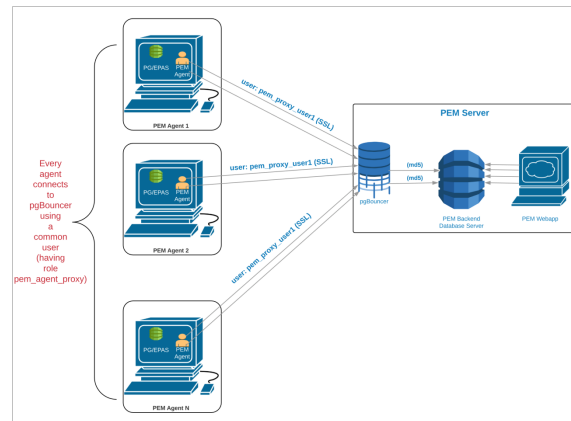


図 1 *pgBouncer* なしで PEM データベースに接続する

PEM バージョン 7.5 より前のバージョンでは、次の制限により、PEM サーバーと PEM エージェント間の接続プーラーの使用が許可されていませんでした。

- PEM エージェントは SSL 証明書を使用して PEM データベースサーバーに接続します。
- PEM データベースサーバーに接続するときに、個々のユーザー識別子を使用します。

EDB は PEM エージェントを変更して、エージェントが（専用のエージェントユーザーではなく）共通のデータベースユーザーを使用して PEM データベースサーバーに接続できるようにしました。

図 2 *pgBouncer* への接続。

接続プールとしてバージョン 1.9.0 以降の PgBouncer バージョンを使用することをお勧めします。バージョン 1.9.0 以降は cert 認証をサポートしています。PEM エージェントは、SSL 証明書を使用して pgBouncer に接続できます。

PEM データベースサーバーの準備

PgBouncer と連携するように PEM データベースサーバーを構成する必要があります。次の例は、PEM データベースサーバーの構成に必要な手順を示しています。

1. PEM データベースサーバーに pgbouncer という名前の専用ユーザーを作成します。例:

```
pem=# CREATE USER pgbouncer PASSWORD 'ANY_PASSWORD' LOGIN;  
CREATE ROLE
```

2. PEM データベースサーバー上で、pem_admin および pem_agent_pool role メンバーシップを持つ pem_admin1（非スーパーユーザー）という名前のユーザーを作成します。例:

```
pem=# CREATE USER pem_admin1 PASSWORD 'ANY_PASSWORD' LOGIN  
CREATE ROLE;  
CREATE ROLE  
pem=# GRANT pem_admin, pem_agent_pool TO pem_admin1;  
GRANT ROLE
```

3. pem データベースの pgbouncer ユーザーに CONNECT 権限を付与します。例:

```
pem=# GRANT CONNECT ON DATABASE pem TO pgbouncer ;GRANT USAGE ON  
SCHEMA pem TO pgbouncer;  
GRANT
```

4. pem データベースの pem スキーマの pgbouncer ユーザーに USAGE 権限を付与します。例:

```
pem=# GRANT USAGE ON SCHEMA pem TO pgbouncer;  
GRANT
```

5. pem データベースの pem.get_agent_pool_auth(text) 関数で pgbouncer ユーザーに EXECUTE 権限を付与します。例:

```
pem=# GRANT EXECUTE ON FUNCTION pem.get_agent_pool_auth(text) TO
pgbouncer;
GRANT
```

6. `pem.create_proxy_agent_user(vchar)` 関数を使用して、**PEM** データベースサーバーに `pem_agent_user1` という名前のユーザーを作成します。例:

```
pem=# SELECT pem.create_proxy_agent_user('pem_agent_user1');
create_proxy_agent_user
-----
(1 row)
```

この関数は、ランダムなパスワードで同じ名前のユーザーを作成し、ユーザーに `pem_agent` および `pem_agent_pool` ロールを付与します。これにより、**pgBouncer** はエージェントに代わってプロキシユーザーを使用できます。

7. **PEM** データベースサーバーの `pg_hba.conf` ファイルの先頭に次のエントリを追加します。これにより、**pgBouncer** ユーザーは `md5` 認証方法を使用して **pem** データベースに接続できます。例:

```
# Allow the PEM agent proxy user (used by
# pgbouncer) to connect the to PEM server using
# md5

local pem pgbouncer,pem_admin1 md5
```

PgBouncer の設定

PEM データベースサーバーと連携するように PgBouncer を構成する必要があります。この例では、PgBouncer を enterprisedb システムユーザーとして実行します。以下の手順は、pgBouncer（バージョン $\geq$ 1.9）を構成するプロセスの概要を示しています。

1. ターミナルウィンドウを開き、pgBouncer ディレクトリに移動します。
2. pgBouncer の etc ディレクトリの所有者（pgbouncer.ini が存在する場所）を enterprisedb に変更し、ディレクトリのアクセス権を 0700 に変更します。例:

```
$ chown enterprisedb:enterprisedb /etc/edb/pgbouncer1.9
$ chmod 0700 /etc/edb/pgbouncer1.9
```

3. pgbouncer.ini または edb-pgbouncer.ini ファイルの内容を次のように変更します。

```
[databases]
;; Change the pool_size according to maximum connections allowed
;; to the PEM database server as required.
;; 'auth_user' will be used for authenticate the db user (proxy
;; agent user in our case)

pem = port=5444 host=/tmp dbname=pem auth_user=pgbouncer
pool_size=80 pool_mode=transaction
* = port=5444 host=/tmp dbname=pem auth_user=pgbouncer
pool_size=10

[pgbouncer]
logfile = /var/log/edb/pgbouncer1.9/edb-pgbouncer-1.9.log
pidfile = /var/run/edb/pgbouncer1.9/edb-pgbouncer-1.9.pid
listen_addr = *
;; Agent needs to use this port to connect the pem database now
```

(次のページに続く)

(前のページからの続き)

```
listen_port = 6432
;; Require to support for the SSL Certificate authentications
;; for PEM Agents
client_tls_sslmode = require
;; These are the root.crt, server.key, server.crt files present
;; in the present under the data directory of the PEM database
;; server, used by the PEM Agents for connections.
client_tls_ca_file = /var/lib/edb/as11/data/root.crt
client_tls_key_file = /var/lib/edb/as11/data/server.key
client_tls_cert_file = /var/lib/edb/as11/data/server.crt
;; Use hba file for client connections
auth_type = hba
;; Authentication file, Reference:
;; https://pgbouncer.github.io/config.html#auth_file
auth_file = /etc/edb/pgbouncer1.9/userlist.txt
;; HBA file
auth_hba_file = /etc/edb/pgbouncer1.9/hba_file
;; Use pem.get_agent_pool_auth(TEXT) function to authenticate
;; the db user (used as a proxy agent user).
auth_query = SELECT * FROM pem.get_agent_pool_auth($1)
;; DB User for administration of the pgbouncer
admin_users = pem_admin1
;; DB User for collecting the statistics of pgbouncer
stats_users = pem_admin1
server_reset_query = DISCARD ALL
;; Change based on the number of agents installed/required
max_client_conn = 500
;; Close server connection if its not been used in this time.
;; Allows to clean unnecessary connections from pool after peak.
server_idle_timeout = 60
```

4. 次のコマンドを使用して、PgBouncer の /etc/edb/pgbouncer1.9/userlist.txt 認証ファイルを作成および更新します。

```
pem=# COPY (
SELECT 'pgbouncer'::TEXT, 'pgbouncer_password'
UNION ALL
SELECT 'pem_admin1'::TEXT, 'pem_admin1_password'
TO '/etc/edb/pgbouncer1.9/userlist.txt'
WITH (FORMAT CSV, DELIMITER ' ', FORCE_QUOTE *);

COPY 2
```

注:スーパーユーザーは、PEM 認証クエリ関数 `pem.get_proxy_auth(text)` を呼び出すことはできません。 `pem_admin` ユーザーがスーパーユーザーの場合、パスワードを認証ファイル (enterprisedb in the above example) に追加する必要があります。

5. 次のコンテンツを含む PgBouncer 用の HBA ファイル (/etc/edb/pgbouncer1.9/hba_file) を作成します。

```
# Use authentication method md5 for the local connections to
# connect pem database & pgbouncer (virtual) database.
local pgbouncer all md5
# Use authentication method md5 for the remote connections to
# connect to pgbouncer (virtual database) using enterprisedb
# user.

host pgbouncer,pem pem_admin1 0.0.0.0/0 md5
```

```
# Use authentication method cert for the TCP/IP connections to
# connect the pem database using pem_agent_user1

hostssl pem pem_agent_user1 0.0.0.0/0 cert
```

6. HBA ファイルの所有者を (/etc/edb/pgbouncer1.9/hba_file) から enterprisedb に変更し、ディレクトリのアクセス権を 0600 に変更します。例:

```
$ chown enterprisedb:enterprisedb /etc/edb/pgbouncer1.9/hba_file
$ chmod 0600 /etc/edb/pgbouncer1.9/hba_file
```

7. PgBouncer サービスを有効にして、サービスを開始します。例:

```
$ systemctl enable edb-pgbouncer-1.9

Created symlink from
/etc/systemd/system/multi-user.target.wants/edb-pgbouncer-1.9.service
to /usr/lib/systemd/system/edb-pgbouncer-1.9.service.

$ systemctl start edb-pgbouncer-1.9
```

第 4

PEM エージェントの設定

RPM パッケージを使用して PEM エージェントをインストールできます。詳細なインストール情報については、次の EDBWeb サイトから入手できる*PEMInstallationGuideLinux*を参照してください。

<https://www.enterprisedb.com/resources/product-documentation>**https://www.enterprisedb.com/resources/product-documentation?field_document_category_tid=2308&field_version_of_document_tid=3321**>‘__

SNMP 通知の送信を担当する PEM エージェントは、pgBouncer で設定しないでください。たとえば、PEM サーバーと一緒にインストールされたデフォルトの PEM エージェントが SNMP 通知に使用される場合、pgBouncer で構成しないでください。

4.1 新しい PEM エージェントの構成（RPM 経由でインストール）

RPM パッケージを使用して PEM エージェントをインストールした後、特定の PEM データベースサーバーに対して動作するようにパッケージを構成する必要があります。次のコマンドを使用します。

```
$ PGSSLMODE=require PEM_SERVER_PASSWORD=pem_admin1_password
/usr/edb/pem/agent/bin/pemworker --register-agent --pem-server
pem_agent_user1 --display-name "Agent Name"

Postgres Enterprise Manager Agent registered successfully!
```

上記のコマンドで、--pem-agent-user 引数は、/root/.pem ディレクトリの pem_agent_user1 データベースユーザーの SSL 証明書とキーペアを作成するようにエージェントに指示します。

例:

```
/root/.pem/pem_agent_user1.crt  
  
/root/.pem/pem_agent_user1.key
```

PEM エージェントは、キーを `pem_agent_user1` として PEM データベースサーバーに接続するために使用します。また、`/usr/edb/pem/agent/etc/agent.cfg` という名前のエージェント構成ファイルを作成します

`agent.cfg` 設定ファイルで使用される `agent-user` に言及する行があります。

例:

```
$ cat /usr/edb/pem/agent/etc/agent.cfg  
[PEM/agent]  
pem_host=172.16.254.22  
pem_port=6432  
agent_id=12  
agent_user=pem_agent_user1  
agent_ssl_key=/root/.pem/pem_agent_user1.key  
agent_ssl_cert=/root/.pem/pem_agent_user1.crt  
log_level=warning  
log_location=/var/log/pem/worker.log  
agent_log_location=/var/log/pem/agent.log  
long_wait=30  
short_wait=10  
alert_threads=0  
enable_smtp=false  
enable_snmp=false  
allow_server_restart=true  
allow_package_management=false  
allow_streaming_replication=false  
max_connections=0  
connect_timeout=-1  
connection_lifetime=0  
allow_batch_probes=false  
heartbeat_connection=false
```

4.2 既存の PEM エージェントの構成（RPM 経由でインストール）

既存の PEM エージェントを使用している場合は、SSL 証明書とキーファイルをターゲットマシンにコピーして、ファイルを再利用できます。ファイルを修正し、新しいパラメーターを追加し、既存の `agent.cfg` ファイルのいくつかのパラメーターを置き換える必要があります。

エージェントに使用する `agent_user` の行を追加します。例:

```
agent_user=pem_agent_user1
```

ポートを更新して pgBouncer ポートを指定します。例:

```
pem_port=6432
```

証明書とキーパスの場所を更新します。例:

```
agent_ssl_key=/root/.pem/pem_agent_user1.key  
agent_ssl_cert=/root/.pem/pem_agent_user1.crt
```

注:別の方法として、エージェントの自己登録スクリプトを実行できますが、それにより新しいエージェント ID が作成されます。エージェントの自己登録スクリプトを実行する場合は、新しいエージェント ID を既存の ID に置き換え、pem.agent テーブルの新しいエージェント ID のエントリを無効にする必要があります。例:

```
pem=# UPDATE pem.agent SET active = false WHERE id = <new_agent_id>;  
  
UPDATE 1
```

注:既存の SSL 証明書、キーファイル、およびエージェント構成ファイルのバックアップを保存してください。

PEMPgBouncer 構成ガイド

著作権 ©2019-2020EnterpriseDBCorporation。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、EDBPostgresEnterpriseManager 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、入力検証を検証します。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

PEM エージェントの設定, [9](#)
PEM サーバー-PEM エージェント接続管理メカニズム, [2](#)
PEM データベースサーバーの準備, [4](#)
PgBouncer の設定, [6](#)

結論, [12](#)