



PEM High Availability Setup Guide

リリース 8.2.0

EnterpriseDB

2021 年 10 月 10 日

目次

第 1 章 High Availability Using EFM

3

このドキュメントでは、EDB Postgres Failover Manager (EFM) を使用して PEM の高可用性 (HA) 環境をセットアップするための詳細情報を提供します。

PEM がサポートするプラットフォームとバージョンについては、[Platform Compatability](#) を参照してください。

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。

第1章 High Availability Using EFM

Postgres Enterprise Manager (PEM) は、データベース管理者、システム設計者、およびパフォーマンスアナリストが Postgres データベースサーバーを管理、モニタリング、およびチューニングするのを支援します。

EDB Postgres Failover Manager (EFM) は、EDB の高可用性ツールであり、プライマリでソフトウェアまたはハードウェアに障害が発生したイベント、Postgres プライマリノードがスタンバイノードに自動的にフェイルオーバーできるようにします。

次のセクションの例では、これらの IP アドレスを使用しています。

- 172.16.161.200-プライマリ PEM
- 172.16.161.201-PEM スタンバイ 1
- 172.16.161.202-PEM スタンバイ 2
- 172.16.161.203-EFM 監視ノード
- 172.16.161.245-PEM VIP (エージェントとユーザーが接続に使用)

製品の初期インストールと構成

1. プライマリおよび 1 つ以上のスタンバイに以下をインストールします

- - (PEM のバックエンドデータベースサーバとして) -
- 上記のリンクを使用して製品文書のインストール手順を参照するか、[EDB Repose website](#) に記載されている手順を参照してください。EDB リポジトリにアクセスするための手順で、`USERNAME:PASSWORD` をユーザー名とパスワードに置き換える必要があります。
- PEM サーバー構成スクリプトは `trust` 認証では動作しないため、データベースサーバが `scram-sha-256` 認証メソッドを使用するように構成されていることを確認してください。
- EFM をインストールするには、`java-1.8.0-openjdk` パッケージをインストールする必要があります。

2. すべてのプライマリおよび 1 つ以上のスタンバイで PEM サーバーを個別に構成します。詳細については、[PEM Installation Guides](#) の「PEM サーバーの構成」セクションを参照してください。

3. すべてのプライマリおよびスタンバイのファイアウォールに次のポートを追加して、アクセスを許可します。

- -8443-PEM サーバー (https) の場合-5444-EPAS 13 の場合-7800-EFM の場合-7908-EFM 管理者の場合

```
* $ sudo firewall-cmd --zone=public --add-port=5444/tcp --permanent
* success
* $ sudo firewall-cmd --zone=public --add-port=8443/tcp --permanent
* success
```

(次のページに続く)

(前のページからの続き)

```
* $ sudo firewall-cmd --zone=public --add-port=7800/tcp --permanent
* success
* $ sudo firewall-cmd --zone=public --add-port=7809/tcp --permanent
* success
* $ sudo firewall-cmd --reload
* success
```

ストリーミングレプリケーションのプライマリノードを設定します。

1. 次のコマンドを使用してレプリケーションロールを作成します

```
* $ /usr/edb/as13/bin/psql -h 172.16.161.200 -p 5444 -U enterprisedb edb -c
"CREATE ROLE repl REPLICATION LOGIN PASSWORD 'password'";
* CREATE ROLE
```

- 任意のパスワードを入力してください。

2. postgresql.conf ファイルで次を構成します。

```
* wal_level = replica
* max_wal_senders = 10
* wal_keep_size = 500
* max_replication_slots = 10
```

ストリーミングレプリケーションのパラメーターの構成の詳細については、[PostgreSQL documentation](#) を参照してください。

!!! Note * 構成パラメーターは、データベースサーバのバージョンによって異なる場合があります。これらのパラメーターの設定に関するヘルプについては、techsupport@enterprisedb.com でサポートにメールを送信できます。

3. ホストベース認証 (/var/lib/edb/as13/data/pg_hba.conf) ファイルに次のエントリーを追加して、レプリケーションユーザがすべてのスタンバイから接続できるようにします。

```
* hostssl replication repl 172.16.161.0/24 scram-sha-256
```

- !!! Note
- IP アドレスに、より適切な cidr レンジを指定できます。

4. pem_user ロールのホストベース認証 (/var/lib/edb/as13/data/pg_hba.conf) ファイルを変更して、scram-sha-256 認証メソッドを使用してすべてのデータベースに接続します。

```
* # Allow local PEM agents and admins to connect to PEM server
* hostssl all +pem_user 172.16.161.201/24 scram-sha-256
* hostssl pem +pem_user 127.0.0.1/32 scram-sha-256
* hostssl pem +pem_agent 127.0.0.1/32 cert
* # Allow remote PEM agents and users to connect to PEM server
```

(次のページに続く)

(前のページからの続き)

```
*      hostssl pem      +pem_user      0.0.0.0/0      scram-sha-256
*      hostssl pem      +pem_agent     0.0.0.0/0      cert
```

5. EPAS 13 サーバーを再起動します

```
*      systemctl restart edb-as-13.service
```

ストリーミングレプリケーション用のスタンバイノードのセットアップ

1. すべてのスタンバイノードで PEM バックエンドデータベースサーバ (EPAS 13) サービスと `pemagent` サービスを停止します。

```
*      $ systemctl stop edb-as-13.service
*      $ systemctl stop pemagent.service
```

2. すべてのスタンバイノードで PEM バックエンドデータベースサーバのデータディレクトリを削除します。

```
*      $ sudo su - enterprisedb
*      $ rm -rf /var/lib/edb/as13/data/*
```

3. すべてのスタンバイノードの `enterprisedb` ユーザのホームディレクトリに `.pgpass` ファイルを作成します。

```
*      $ sudo su - enterprisedb
*      $ cat > ~/.pgpass << _EOF_
*      172.16.161.200:5444:replication:repl:CHANGE_ME
*      172.16.161.201:5444:replication:repl:CHANGE_ME
*      172.16.161.202:5444:replication:repl:CHANGE_ME
*      _EOF_
*      $ chmod 600 ~/.pgpass
```

4. `pg_basebackup` を使用して、各スタンバイノードでプライマリノードのバックアップを作成します。

```
*      $ sudo su - enterprisedb /usr/edb/as13/bin/pg_basebackup -h 172.16.161.200 \
*      -D /var/lib/edb/as13/data -U repl -v -P -Fp -R -p 5444
```

バックアップコマンドは、スタンバイノードに `postgresql.auto.conf` および `standby.signal` ファイルを作成します。

5. すべてのスタンバイノードで `postgresql.conf` ファイルの次のパラメータを編集します。

```
*      hot_standby = on
```

6. 各スタンバイノードで PEM バックエンドデータベースサーバ (EPAS 13) を起動します。

```
*      $ systemctl enable edb-as-13
*      $ systemctl start edb-as-13
```

7. /root/.pem から PEM エージェントの証明書とキーをプライマリノードから同じ場所のスタンバイノードにコピーします。PEM エージェント証明書ファイルに次のアクセス許可を設定します。

```
* $ sudo chmod 600 /root/.pem/agent1.key
* $ sudo chmod 640 /root/.pem/agent1.crt
```

!!! Note * 証明書が更新されるたびに、マスタとスタンバイで証明書の同期を手動で維持する必要があります。

EFM をセットアップして、すべてのホストでフェイルオーバーを管理する

1. EFM をサポートするプライマリノードを準備します。

- -データベースユーザに `efm` を作成して、データベースサーバーに接続します。-WAL ログに関連する機能の実行権限とモニタリング権限をユーザに付与します。-`pg_hba.conf` にエントリを追加して、`efm` データベースユーザがすべてのホスト上のすべてのノードからデータベースサーバーに接続できるようにします。-すべてのデータベースサーバーで構成を再読み込みします。

```
* $ cat > /tmp/efm-role.sql << _EOF_
* -- Create a role for EFM
* CREATE ROLE efm LOGIN PASSWORD 'password';
* -- Give privilege to 'efm' user to connect to a database
* GRANT CONNECT ON DATABASE edb TO efm;
* -- Give privilege to 'efm' user to do backup operations
* GRANT EXECUTE ON FUNCTION pg_current_wal_lsn() TO efm;
* GRANT EXECUTE ON FUNCTION pg_last_wal_replay_lsn() TO efm;
* GRANT EXECUTE ON FUNCTION pg_wal_replay_resume() TO efm;
* GRANT EXECUTE ON FUNCTION pg_wal_replay_pause() TO efm;
* GRANT EXECUTE ON FUNCTION pg_reload_conf() TO efm;
* -- Grant monitoring privilege to the 'efm' user
* GRANT pg_monitor TO efm;
* _EOF_
* $ /usr/edb/as13/bin/psql -h 172.16.161.200 -p 5444 -U enterprisedb edb -f /
→tmp/efm-role.sql
* CREATE ROLE
* GRANT
* GRANT
* GRANT
* GRANT
* GRANT
* GRANT
* GRANT ROLE
* $ rm -f /tmp/efm-role.sql
* $ cat > /var/lib/edb/as13/data/pg_hba.conf <<< _EOF_
* hostssl      edb      efm      172.16.161.200/32      scram-sha-256
* hostssl      edb      efm      172.16.161.201/32      scram-sha-256
* hostssl      edb      efm      172.16.161.202/32      scram-sha-256
* hostssl      edb      efm      172.16.161.203/32      scram-sha-256
```

(次のページに続く)

(前のページからの続き)

```
*      _EOF_
*      $ /usr/edb/as13/bin/psql -h 172.16.161.200 -p 5444 -U enterprisedb edb -c
"SELECT pg_reload_conf();"

```

2. 各ノードでスクリプトを作成して、PEM エージェントをスタート/停止します。

```
*      $ sudo cat > /usr/local/bin/start-pemagent.sh << _EOF_
*      #!/bin/sh
*      /bin/sudo /bin/systemctl enable pemagent
*      /bin/sudo /bin/systemctl start pemagent
*      _EOF_
*      $ sudo cat > /usr/local/bin/stop-pemagent.sh << _EOF_
*      #!/bin/sh
*      /bin/sudo /bin/systemctl stop pemagent
*      /bin/sudo /bin/systemctl disable pemagent
*      _EOF_
*      $ sudo chmod 770 /usr/local/bin/start-pemagent.sh
*      $ sudo chmod 770 /usr/local/bin/stop-pemagent.sh

```

3. 各ノードで sudoers ファイル (/etc/sudoers.d/efm-pem) を作成して、efm ユーザが pemagent をスタート/停止できるようにします。

```
*      $ sudo cat > /etc/sudoers.d/efm-pem << _EOF_
*      efm    ALL=(ALL)          NOPASSWD:    /bin/systemctl enable pemagent
*      efm    ALL=(ALL)          NOPASSWD:    /bin/systemctl disable pemagent
*      efm    ALL=(ALL)          NOPASSWD:    /bin/systemctl stop pemagent
*      efm    ALL=(ALL)          NOPASSWD:    /bin/systemctl start pemagent
*      _EOF_

```

4. サンプルファイル (/etc/edb/efm-4.1/efm.nodes.in) を使用してすべてのノードで efm.nodes ファイルを作成し、efm OS ユーザに読み書きアクセスを許可します。

```
*      $ sudo cp /etc/edb/efm-4.1/efm.nodes.in /etc/edb/efm-4.1/efm.nodes
*      $ sudo chown efm:efm /etc/edb/efm-4.1/efm.nodes
*      $ sudo chmod 600 /etc/edb/efm-4.1/efm.nodes

```

5. スタンバイノードの/etc/edb/efm-4.1/efm.nodes ファイルにプライマリノードの ip address と EFM PORT を追加します。

```
*      $ sudo cat > /etc/edb/efm-4.1/efm.nodes <<< _EOF_
*      172.16.161.200:7800
*      _EOF_

```

6. サンプルファイル (/etc/edb/efm-4.1/efm.properties.in) を使用してすべてのノードで efm.properties ファイルを作成し、すべてのユーザーに読み取りアクセスを許可します。

```
* $ sudo cp /etc/edb/efm-4.1/efm.properties.in /etc/edb/efm-4.1/efm.properties
* $ sudo chown efm:efm /etc/edb/efm-4.1/efm.properties
* $ sudo chmod a+r /etc/edb/efm-4.1/efm.properties
```

7. efm ユーティリティを使用して、efm ユーザーのパスワードを暗号化します。

```
* $ export EFMPASS=password
* $ /usr/edb/efm-4.1/bin/efm encrypt efm --from-env
* 096666746b05b081d1a98e43d94c9dad
```

8. プロパティファイルで次のパラメータを編集します。

```
* 1. db.user=efm
* 2. db.password.encrypted=096666746b05b081d1a98e43d94c9dad
* 3. db.port=5444
* 4. db.database=edb
* 5. db.service.owner=enterprisedb
* 6. db.service.name=edb-as-13
* 7. db.bin=/usr/edb/as13/bin
* 8. db.data.dir=/var/lib/edb/as13/data
* 9. jdbc.sslmode=require
* 10. user.email=username@example.com
* 11. from.email=node1@efm-pem
* 12. notification.level=INFO
* 13. notification.text.prefix=[PEM/EFM]
* 14. bind.address=172.16.161.200:7800
* 15. admin.port=7809
* 16. is.witness=false
* 17. local.period=10
* 18. local.timeout=60
* 19. local.timeout.final=10
* 20. remote.timeout=10
* 21. node.timeout=50
* 22. encrypt.agent.messages=true
* 23. stop.isolated.primary=true
* 24. stop.failed.primary=true
* 25. primary.shutdown.as.failure=false
* 26. update.physical.slots.period=0
* 27. ping.server.ip=8.8.8.8
* 28. ping.server.command=/bin/ping -q -c3 -w5
* 29. auto.allow.hosts=false
* 30. stable.nodes.file=false
* 31. db.reuse.connection.count=0
* 32. auto.failover=true
* 33. auto.reconfigure=true
* 34. promotable=true
```

(次のページに続く)

(前のページからの続き)

```
* 35. use.replay.tiebreaker=true
* 36. standby.restart.delay=0
* 37. reconfigure.num.sync=false
* 38. reconfigure.sync.primary=false
* 39. minimum.standbys=0
* 40. recovery.check.period=1
* 41. restart.connection.timeout=60
* 42. auto.resume.period=0
* 43. virtual.ip=172.16.161.245
* 44. virtual.ip.interface=ens33
* 45. virtual.ip.prefix=24
* 46. virtual.ip.single=true
* 47. check.vip.before.promotion=true
* 48. pgpool.enable=false
* 49. sudo.command=sudo
* 50. sudo.user.command=sudo -u %u
* 51. syslog.host=localhost
* 52. syslog.port=514
* 53. syslog.protocol=UDP
* 54. syslog.facility=LOCAL1
* 55. file.log.enabled=true
* 56. syslog.enabled=false
* 57. jgroups.loglevel=INFO
* 58. efm.loglevel=INFO
* 59. jvm.options=-Xmx128m
* 60. script.fence=/usr/local/bin/stop-pemagent.sh
* 61. script.post.promotion=/usr/local/bin/start-pemagent.sh
```

9. 監視ノードの `is.witness` 設定パラメータの値を `true` に設定します。

```
* is.witness=true
```

10. プライマリノードで EFM サービスを有効にしてスタートします。

```
* * $ systemctl enable edb-efm-4.1
* * $ systemctl start edb-efm-4.1
```

11. スタンバイがプライマリノードで開始されたクラスターに結合できるようにします。

```
* /usr/edb/efm-4.1/bin/efm allow-node efm 172.16.161.201
* /usr/edb/efm-4.1/bin/efm allow-node efm 172.16.161.202
* /usr/edb/efm-4.1/bin/efm allow-node efm 172.16.161.203
```

12. スタンバイノードと EFM 監視ノードで EFM サービスを有効にしてスタートします。

```
* * $ systemctl enable edb-efm-4.1
* * $ systemctl start edb-efm-4.1
```

13. 任意のノードから EFM クラスターステータスを確認します。

```
* $ sudo /usr/edb/efm-4.1/bin/efm cluster-status efm
* Cluster Status: efm
* Agent Type Address DB VIP
* -----
* Primary 172.16.161.200 UP 172.16.161.245*
* Standby 172.16.161.201 UP 172.16.161.245
* Standby 172.16.161.202 UP 172.16.161.245
* Witness 172.16.161.203 N/A 172.16.161.245
* Allowed node host list:
* 172.16.161.200 172.16.161.201 172.16.161.202 172.16.161.203
* Membership coordinator: 172.16.161.200
* Standby priority host list:
* 172.16.161.201 172.16.161.202
* Promote Status:
* DB Type Address WAL Received LSN WAL Replayed LSN
* -----
* Info
* -----
* Primary 172.16.161.200 0/F7A3808
* Standby 172.16.161.201 0/F7A3808 0/F7A3808
* Standby 172.16.161.202 0/F7A3808 0/F7A3808
* Standby database(s) in sync with primary. It is safe to promote.
```

このステータスは、EFM が正常にセットアップされ、PEM サーバーのフェイルオーバーを管理していることを確認します。

監視対象のデータベースサーバーは、VIP アドレスを使用して PEM エージェントを登録する必要があります。また、VIP アドレスを使用して PEM Web クライアントにアクセスします。

フェイルオーバーの場合、スタンバイのいずれかがプライマリノードとして昇格されます。PEM エージェントは、新しいプライマリノードに自動的に接続します。すべてのユーザセッションが失われ、再度ログする必要があります。このプロシージャを使用して、障害が発生したプライマリノードを新しいスタンバイに交換できます。