

Creating a Failover Manager Cluster 4.2

EnterpriseDB

Creating a Failover Manager Cluster

EDB Postgresのフェイルオーバーマネージャー（フェイルオーバーマネージャーは）一次上のソフトウェアまたはハードウェア障害が発生しイベントにスタンバイ・ノードに自動的にフェイルオーバーにPostgresのプライマリノードを可能EnterpriseDBのより高可用性モジュールです。

このクイックスタートガイドでは、テスト環境でのフェールオーバーマネージャークラスターの構成について説明します。あなたは、稼働の展開のためのフェイルオーバーマネージャーを設定する前に[EDB Failover Manager User's Guide](#)を読んで理解する必要があります。

このチュートリアルを実行する前に、いくつかの基本的なインストールおよび構成手順を実行する必要があります。

- 1つのプライマリノードと1つまたは2つのスタンバイノードにデータベースサーバをインストールして初期化する必要があります。Advanced Serverのインストールについては、次を参照してください。

- <https://www.enterprisedb.com/docs/epas/latest/>

Postgresストリーミングレプリケーションを構成し、プライマリノードとスタンバイノード間で実行する必要があります。ストリーミングレプリケーションの構成の詳細については、次を参照してください。

- <https://www.postgresql.org/docs/current/warm-standby.html#STREAMING-REPLICATION>。
- また、各プライマリノードとスタンバイノードにフェールオーバーマネージャーをインストールする必要があります。Advanced Serverのインストール中に、各データベースホストでEnterpriseDBリポジトリを構成しました。EnterpriseDBリポジトリと `yum install` コマンドを使用して、クラスターの各ノードにフェールオーバーマネージャーをインストールできます。

```
yum install edb-efm42
```

インストールプロセス中、インストーラは、`enterprisedb` または `postgres` が所有するクラスターのFailover Managerサービスを制御するスクリプトを呼び出すための十分な特権を持つ `efm` という名前付けのユーザを作成します。次の例では、`efm` 名前付けのクラスターを作成します。

プライマリノードまたはスタンバイノードで構成プロセスを開始します。次に、構成ファイルを他のノードにコピーして、時間を節約します。

ステップ1：作業用構成ファイルの作成

提供されたサンプルファイルをコピーしてEFM構成ファイルを作成し、所有権を修正します。

```
cd /etc/edb/efm-4.2

cp efm.properties.in efm.properties

cp efm.nodes.in efm.nodes

chown efm:efm efm.properties

chown efm:efm efm.nodes
```

ステップ2：暗号化されたパスワードを作成する

プロパティファイルに必要な[encrypted password](#)を作成します。

```
/usr/edb/efm-4.2/bin/efm encrypt efm
```

画面の指示に従って、データベースパスワードの暗号化バージョンを作成します。

ステップ3：efm.propertiesファイルを更新する

`<cluster_name>.properties` ファイル（この例ではefm.propertiesファイル）には、Failover Managerクラスタの接続プロパティと動作を指定するパラメーターが含まれています。プロパティ設定の変更は、フェールオーバーマネージャーの起動時に適用されます。

このチュートリアルで言及されているプロパティは、Failover Managerクラスタの構成に必要な最小限のプロパティです。稼動システムを構成する場合は、* EDB Failover Manager User Guide *を参照して、Failover Managerオプションの詳細を確認してください。

すべてのクラスタースタートで次のプロパティの値を指定します。

Property	Description
db.user	データベースユーザの名前。
db.password.encrypted	データベースユーザの暗号化されたパスワード。
db.port	データベースによって監視されるポート。
db.database	データベースの名前。

Property	Description
<code>db.service.owner</code>	<code>data</code> ディレクトリの所有者（通常は <code>postgres</code> または <code>enterprisedb</code> ）。データベースがサービスとして実行されている場合にのみ必要です。
<code>db.service.name</code>	データベースサービスの名前（サーバーのリスタートに使用）。データベースがサービスとして実行されている場合にのみ必要です。
<code>db.bin</code>	<code>bin</code> ディレクトリへのパス（ <code>pg_ctl</code> の呼び出しに使用）。
<code>db.recovery.dir</code>	<code>data</code> ディレクトリとは、EFMを見つけるか、 <code>recovery.conf</code> ファイルまたは <code>standby.signal</code> ファイルを作成します。
<code>user.email</code>	メール通知を受信するメールアドレス（通知テキストはエージェントログファイルにもあります）。
<code>bind.address</code>	EFMに使用するノードとポートのローカルアドレス。フォーマットは次のとおりです。 <code>bind.address=1.2.3.4:7800</code>
<code>is.witness</code>	ミラーリング監視ノード上の <code>true</code> と、プライマリまたはスタンバイの場合は <code>false</code>
<code>ping.server.ip</code>	インターネットにアクセスできないネットワークで実行している場合は、 <code>ping.server.ip</code> をネットワークで使用可能なアドレスに設定します。
<code>auto.allow.hosts</code>	テストクラスターでは、 <code>true</code> に設定してスタートアップを簡素化します。稼動環境での使用については、ガイドを参照してください。
<code>stable.nodes.file</code>	テストクラスターでは、 <code>true</code> に設定してスタートアップを簡素化します。稼動環境での使用については、ガイドを参照してください。

ステップ4：efm.nodesファイルを更新する

`<cluster_name>.nodes` ファイル（この例ではefm.nodesファイル）がどのように開始し、最初のノードの場合には、クラスタの残りの部分を見つけたりするために、エージェントを伝えるために、スタートアップに読み込まれ、その後のノードの許可を簡素化するために使

用することができます。クラスタ内の各ノードのアドレスとポートをこのファイルに追加します。1つのノードがメンバーシップコーディネーターとして機能します。リストには、少なくともメンバーシップコーディネーターの住所を含める必要があります。例：

```
1.2.3.4:7800
```

```
1.2.3.5:7800
```

```
1.2.3.6:7800
```

Failover Managerエージェントは `efm.nodes` ファイルのコンテンツを検証しないことにノートしてください。エージェントは、ファイル内の一部のアドレスに到達できないことを予期しています（たとえば、別のエージェントがまだ開始されていないなど）。

ステップ5：他のノードの構成

`efm.properties` および `efm.nodes` ファイルを、サンプルクラスターの他のノードの `/etc/edb/efm-4.2` ディレクトリにコピーします。ファイルをコピーした後、ファイルの所有権を変更して、ファイルが `efm:efm` によって所有されるようにします。

`efm.properties` ファイルは、次のプロパティを除き、すべてのノードで同じにすることができます。

- `bind.address` プロパティを変更して、ノードのローカルアドレスを使用します。
- ノードが監視ノードの場合、`is.witness` を `true` に設定しノード。ノードが監視ノードである場合、ローカルデータベースのインストールに関連するプロパティは無視されます。

ステップ6：EFMクラスターを開始します

任意のノードで、Failover Managerエージェントをスタートします。エージェントの名前付けは `edb-efm-4.2` です。プラットフォーム固有のサービスコマンドを使用して、サービスを制御できます。例、CentOSの/RHEL 7.xまたはCentOSの/RHEL 8.xのホストの使用コマンドで：

```
systemctl start edb-efm-4.2
```

CentOSまたはRHEL 6.xホストでは、次のコマンドを使用します。

```
service edb-efm-4.2 start
```

エージェントの起動後、次のコマンドを実行して、シングルノードクラスターのステータスを確認します。 `Allowed node host` リストに他のノードのアドレスが表示されます。

```
/usr/edb/efm-4.2/bin/efm cluster-status efm
```

他のノードでエージェントを起動します。任意のノードで `efm cluster-status efm` コマンドを実行して、クラスターの状態を確認します。

いずれかのエージェントがスタートに失敗した場合は、スタートアップログを参照して、問題の詳細を確認してください。

```
cat /var/log/efm-4.2/startup-efm.log
```

スイッチオーバーの実行

クラスターステータスの出力に、プライマリとスタンバイが同期していることが示されている場合、次のコマンドでスイッチオーバーを実行できます。

```
/usr/edb/efm-4.2/bin/efm promote efm -switchover
```

このコマンドは、スタンバイをプロモートさせ、プライマリデータベースをクラスター内の新しいスタンバイとして再構成します。スイッチバックするには、再度コマンドを実行します。

オンラインヘルプにすばやくアクセスするには、次のコマンドを呼び出すことができます。

```
/usr/edb/efm-4.2/bin/efm --help
```