



Creating a Failover Manager Cluster

Version 4.2

0 Creating a Failover Manager Cluster

3

0 Creating a Failover Manager Cluster

EDB Postgres Failover Manager (Failover Manager) is a high-availability module from EnterpriseDB that enables a Postgres Primary node to automatically failover to a Standby node in the event of a software or hardware failure on the Primary.

This quick start guide describes configuring a Failover Manager cluster in a test environment. You should read and understand the [EDB Failover Manager User's Guide](#) before configuring Failover Manager for a production deployment.

You must perform some basic installation and configuration steps before performing this tutorial:

- You must install and initialize a database server on one primary and one or two standby nodes; for information about installing Advanced Server, visit:

<https://www.enterprisedb.com/docs/p/edb-postgres-advanced-server>

- Postgres streaming replication must be configured and running between the primary and standby nodes. For detailed information about configuring streaming replication, visit:

<https://www.postgresql.org/docs/current/warm-standby.html#STREAMING-REPLICATION>.

- You must also install Failover Manager on each primary and standby node. During Advanced Server installation, you configured an EnterpriseDB repository on each database host. You can use the EnterpriseDB repository and the `yum install` command to install Failover Manager on each node of the cluster:

```
yum install edb-efm41
```

During the installation process, the installer will create a user named **efm** that has sufficient privileges to invoke scripts that control the Failover Manager service for clusters owned by **enterprisedb** or **postgres**. The example that follows creates a cluster named **efm**.

Start the configuration process on a primary or standby node. Then, copy the configuration files to other nodes to save time.

Step 1: Create Working Configuration Files

Copy the provided sample files to create EFM configuration files, and correct the ownership:

```
cd /etc/edb/efm-4.1  
  
cp efm.properties.in efm.properties  
  
cp efm.nodes.in efm.nodes  
  
chown efm:efm efm.properties  
  
chown efm:efm efm.nodes
```

Step 2: Create an Encrypted Password

Create the **encrypted password** needed for the properties file:

```
/usr/edb/efm-4.1/bin/efm encrypt efm
```

Follow the onscreen instructions to produce the encrypted version of your database password.

Step 3: Update the efm.properties File

The `<cluster_name>.properties` file (efm.properties file in this example) contains parameters that specify connection properties and behaviors for your Failover Manager cluster. Modifications to property settings are applied when Failover Manager starts.

The properties mentioned in this tutorial are the minimal properties required to configure a Failover Manager cluster. If you are configuring a production system, please review the *EDB Failover Manager Guide* for detailed information about Failover Manager options.

Provide values for the following properties on all cluster nodes:

Property	Description
<code>db.user</code>	The name of the database user.
<code>db.password.encrypted</code>	The encrypted password of the database user.
<code>db.port</code>	The port monitored by the database.
<code>db.database</code>	The name of the database.
<code>db.service.owner</code>	The owner of the <code>data</code> directory (usually <code>postgres</code> or <code>enterprisedb</code>). Required only if the database is running as a service.
<code>db.service.name</code>	The name of the database service (used to restart the server). Required only if the database is running as a service.
<code>db.bin</code>	The path to the <code>bin</code> directory (used for calls to <code>pg_ctl</code>).
<code>db.recovery.dir</code>	The <code>data</code> directory in which EFM will find or create the <code>recovery.conf</code> file or the <code>standby.signal</code> file.

Property	Description
<code>user.email</code>	An email address at which to receive email notifications (notification text is also in the agent log file).
<code>bind.address</code>	The local address of the node and the port to use for EFM. The format is: <code>bind.address=1.2.3.4:7800</code>
<code>is.witness</code>	<code>true</code> on a witness node and <code>false</code> if it is a primary or standby.
<code>ping.server.ip</code>	If you are running on a network without Internet access, set <code>ping.server.ip</code> to an address that is available on your network.
<code>auto.allow.hosts</code>	On a test cluster, set to <code>true</code> to simplify startup; for production usage, consult the user's guide.
<code>stable.nodes.file</code>	On a test cluster, set to <code>true</code> to simplify startup; for production usage, consult the user's guide.

Step 4: Update the efm.nodes File

The `<cluster_name>.nodes` file (efm.nodes file in this example) is read at startup to tell an agent how to find the rest of the cluster or, in the case of the first node started, can be used to simplify authorization of subsequent nodes. Add the addresses and ports of each node in the cluster to this file. One node will act as the membership coordinator; the list should include at least the membership coordinator's address. For example:

```
1.2.3.4:7800
```

```
1.2.3.5:7800
```

1.2.3.6:7800

Please note that the Failover Manager agent will not verify the content of the `efm.nodes` file; the agent expects that some of the addresses in the file cannot be reached (e.g. that another agent hasn't been started yet).

Step 5: Configure the Other Nodes

Copy the `efm.properties` and `efm.nodes` files to the `/etc/edb/efm-4.1` directory on the other nodes in your sample cluster. After copying the files, change the file ownership so the files are owned by `efm:efm`. The `efm.properties` file can be the same on every node, except for the following properties:

- Modify the `bind.address` property to use the node's local address.
- Set `is.witness` to `true` if the node is a witness node. If the node is a witness node, the properties relating to a local database installation will be ignored.

Step 6: Start the EFM Cluster

On any node, start the Failover Manager agent. The agent is named `edb-efm-4.1`; you can use your platform-specific service command to control the service. For example, on a CentOS/RHEL 7.x or CentOS/RHEL 8.x host use the command:

```
systemctl start edb-efm-4.1
```

On a CentOS or RHEL 6.x host use the command:

```
service edb-efm-4.1 start
```

After the agent starts, run the following command to see the status of the single-node cluster. You should see the addresses of the other nodes in the `Allowed node host` list.

```
/usr/edb/efm-4.1/bin/efm cluster-status efm
```

Start the agent on the other nodes. Run the `efm cluster-status efm` command on any node to see the cluster status.

If any agent fails to start, see the startup log for information about what went wrong:

```
cat /var/log/efm-4.1/startup-efm.log
```

Performing a Switchover

If the cluster status output shows that the primary and standby(s) are in sync, you can perform a switchover with the following command:

```
/usr/edb/efm-4.1/bin/efm promote efm -switchover
```

The command will promote a standby and reconfigure the primary database as a new standby in the cluster. To switch back, run the command again.

For quick access to online help, you can invoke the following command:

```
/usr/edb/efm-4.1/bin/efm --help
```