# Demo Environment

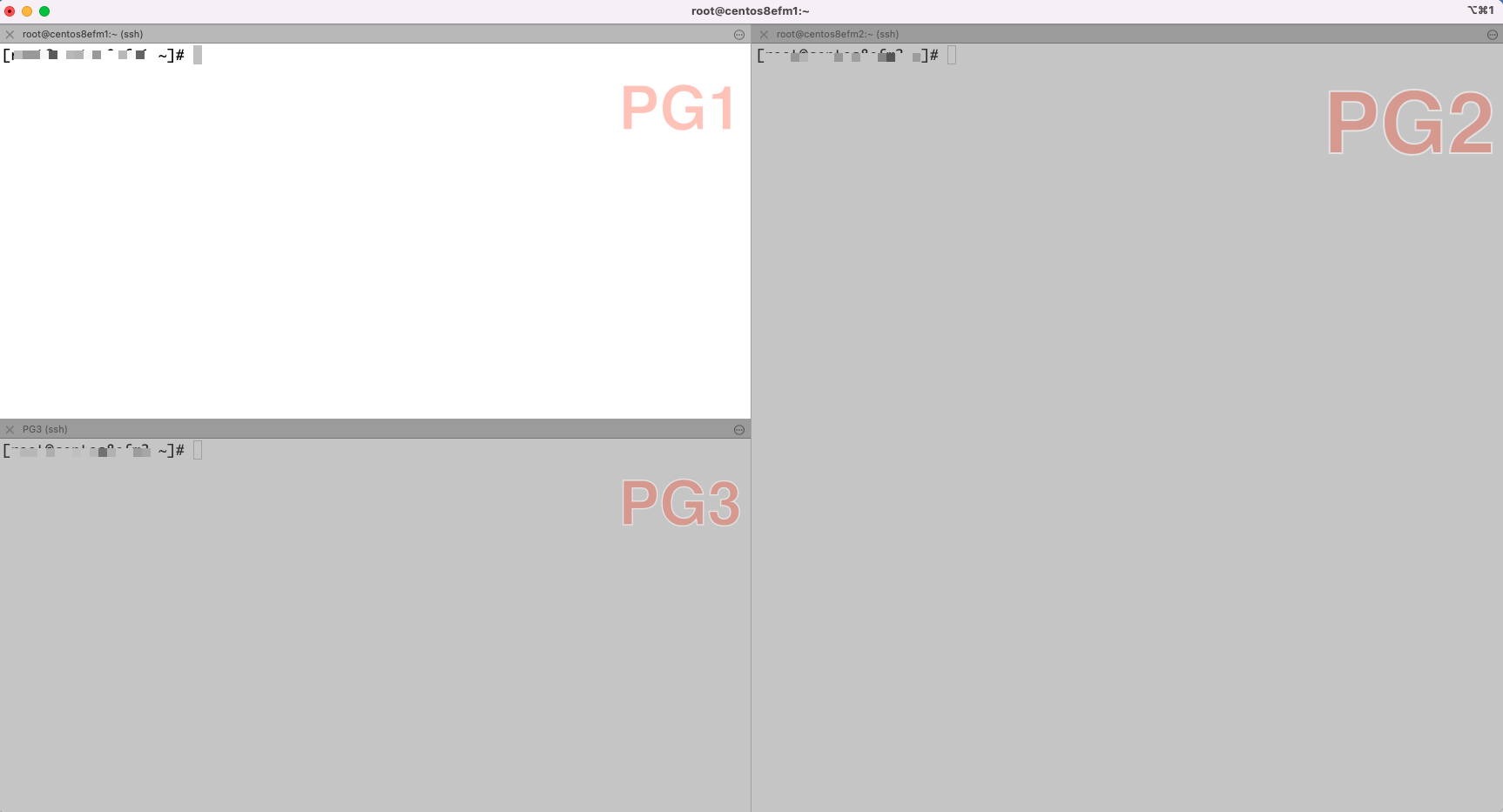
This demo is recorded as a video and is available on [YouTube](https://www.youtube.com/watch?v=mWjnNM5g2YU&t=24s)

In the demo environment, I have configured the EFM cluster with a master and a standby.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Hostname | EPAS Version | EFM Version | Role |
| VM1 | pg1 | 14 | 4.4 | Master |
| VM2 | pg2 | 14 | 4.4 | Standby |
| VM3 | pg3 | 14 | 4.4 | Client |

In the first step we will start three terminal windows here, with Master at the top left, Standby at the right and the Client at the bottom left.

|  |  |  |
| --- | --- | --- |
| pg1 - Master | pg2 - Standby | pg3 - Client |
| cd <working\_directory>  ssh –F ssh\_config pg1 | cd <working\_directory>  ssh –F ssh\_config pg2 | cd <working\_directory>  ssh –F ssh\_config pg3 |



On the standby I now start monitoring the cluster, running the command “efm cluster-status efmdemo” at regular intervals

|  |  |  |
| --- | --- | --- |
| pg1 - Master | pg2 - Standby | pg3 - Client |
|  | sudo watch /usr/edb/efm-4.4/bin/efm cluster-status efmdemo |  |

As we can see, my master and standby databases are online and synchronized: the WAL LSN number is the same on both: primary and standby databases.

In the third session (pg3 - client - bottom left) I connect to both databases using psql. The read\_write attribute defines that I always connect to a primary database because only the master database can receive and process the write transactions.

We can query the emp table. And check if we are on the master:

|  |  |  |
| --- | --- | --- |
| pg1 - Master | pg2 - Standby | pg3 - Client |
|  |  | sudo -i -u enterprisedb psql postgresql://pg1,pg2,pg3/edb?target\_session\_attrs=read-write  SELECT \* from EMP;  SELECT inet\_server\_addr(); |

# Failover Demo

Now we shut down the primary database and see what happens:

|  |  |  |
| --- | --- | --- |
| pg1 - Master | pg2 - Standby | pg3 - Client |
| sudo systemctl stop postgres.service | *<EFM Cluster Monitor Session>* |  |

We see (in the pg2 window) that the failover manager has detected that the primary database is not available and performs the failover. The standby database now acts as the primary.

Let's see, the client side.

We re-execute the query and see that the psql session has reconnected. Re-executing the SQL command gives us a result that we have connected to the new master database.

And of course, we can query the emp table.

|  |  |  |
| --- | --- | --- |
| pg1 - Master | pg2 - Standby | pg3 - Client |
|  | *<EFM Cluster Monitor Session>* | SELECT inet\_server\_addr();  SELECT inet\_server\_addr();  SELECT \* from EMP; |

But the situation is not good. The old master database does not exist. And we only have one database running without any protection.

In this step, I start the old primary database as the new standby. The tool pg\_rewind can be used for this. I have automated the steps and summarized them in the script efm\_reconfigure\_node.sh.

Now we run the script and see that the old primary database is included in the cluster as the new standby.

|  |  |  |
| --- | --- | --- |
| pg1 - Master | pg2 - Standby | pg3 - Client |
| sudo /usr/edb/efm-4.4/bin/efm\_reconfigure\_node.sh | *<EFM Cluster Monitor Session>* |  |

In some situations, due to network bandwidth, WAL replication may take some time. To speed it up, execute the command "SELECT pg\_swicth\_wal();" in the client session:

|  |  |  |
| --- | --- | --- |
| pg1 - Master | pg2 - Standby | pg3 - Client |
|  | *<EFM Cluster Monitor Session>* | select pg\_switch\_wal(); |

# Switchover Demo

And as the last action, I show the switchover. In the operation, the roles are switched: standby database becomes primary, and primary: becomes standby.

|  |  |  |
| --- | --- | --- |
| pg1 - Master | pg2 - Standby | pg3 - Client |
| sudo /usr/edb/efm-4.4/bin/efm promote efmdemo -switchover | *<EFM Cluster Monitor Session>* |  |