



RECM

Version 1.0

Recovery Manager for PostgreSQL

Quick User guide
(Revision 2021-12)

Quick User guide of RECM

Recm (RECOVERY Manager) is an open-source administration tool for disaster recovery of PostgreSQL databases. It allows you to perform backups of multiple servers in any business environments to facilitate the maintenance of your backups, and reduce the risk to loose data at recovery time.

Table of contents

| | |
|---|----|
| I PREREQUISITES..... | 3 |
| I.a) Software Requirements..... | 3 |
| I.b) Configuration requirements..... | 3 |
| II INSTALLATION..... | 4 |
| III INITIALISATION..... | 5 |
| III.a) Create the deposit..... | 5 |
| IV REGISTER CLUSTER..... | 6 |
| IV.a) Prerequisites..... | 6 |
| IV.b) Register a cluster(Server instance)..... | 6 |
| IV.c) Configure cluster..... | 7 |
| V TAKE BACKUPS..... | 8 |
| V.a) FULL backups..... | 8 |
| V.b) WAL backup | 8 |
| V.c) CONFIG backup | 9 |
| V.d) META backup | 9 |
| VI DUPLICATE CLUSTER..... | 10 |
| VI.a Duplicate cluster locally..... | 10 |
| VI.b Duplicate cluster from a different source cluster..... | 10 |
| VI.b Create a 'ReadOnly' standby | 11 |
| VI.b.1) Create the user in the cluster..... | 11 |
| VI.b.2) Prepare the primary configuration..... | 11 |
| VI.b.3) Perform the duplicate for STANDBY..... | 11 |
| VII RESTORE A CLUSTER..... | 12 |

I PREREQUISITES

RECM is build in C, with gcc and is available in the RPM format.

I.a) Software Requirements

Packages required for RECM (dependencies)

- libzip (minimum Version 1.5)
- libpq (minimum Version 11)
- recm

I.b) Configuration requirements

Component used (recommended)

- NFS file server accessible for all PostgreSQL engine
- dedicated PostgreSQL database

You can perform backups locally, if you want. But if you need to perform, for example, duplicates from your production to your test, you need to have backups accessible from both environment.

To have RECM running, you need to have a database that will be used as 'repository' that we call 'DEPOSIT'. This database will retain all informations about backup,restore point and WAL history, to help you to restore a database,schema or table at any time you need.

It is recommended to have a dedicated database for this purpose.

You can estimate the size required for the database, you have to know that one DB, for a retention of 7 days for full and wal backups will use approximatively 10M per DB.

II INSTALLATION

To install RECM, download package and launch the installer as follow:

The package is available for 'CENTOS' for the moment. But it is up to you to use the sources.

```
#> dnf install <source_directory>/recm-1.0-1_x86_64.rpm
```

III INITIALISATION

To have RECM running, you need to have a database that will be used as 'repository' we call 'deposit'.

III.a) Create the deposit

The deposit is a PostgreSQL database that maintain all informations needed to perform backup and restore. This deposit is MANDATORY.

```
RECM> connect cluster /host=192.168.1.62/port=5432/usr=postgres/pwd=postgres/db=postgres
recm-inf: Connected to cluster 'REPO12'
RECM#> create deposit /usr=recmadm/pwd=recmadm/db=recm_db;
.....
recm-inf: Deposit 'REPO12' created.
RECM#>
```

IV REGISTER CLUSTER

The following steps are required before to be able to take your first backup.

IV.a) Prerequisites

Cluster configuration parameters :

| Parameter | Mandatory | Recommandation |
|-----------------|-----------|---|
| archive_command | Yes | This parameter copy WAL files onto a directory(usually local). In the 'cp' command, add the option '-p' to preserve the TIME of the file. This will give a more precise granularity at restore time. Example: <code>cp -p %p /var/lib/postgres/13/WALfiles/%f</code> |
| cluster_name | Yes | The 'cluster_name' parameter is mandatory for RECM. |
| archive_mode | Yes | 'on' |
| wal_level | Yes | 'replica' |

IV.b) Register a cluster(Server instance)

To register a cluster, you need to connect first onto the cluster, connect to the deposit and than register the cluster. After those operations, the 'connect target' can be used.

```
RECM#> connect cluster/host=192.168.1.62/port=5432/usr=postgres/pwd=postgres/db=postgres
recm-inf: Connected to cluster 'CLU12'

RECM#> connect deposit /host=192.168.1.62/port=5555/usr=recmadm/pwd=recmadm/db=recm_db;
recm-inf: Connected to deposit 'REPO12'
recm-err(c04): Not registerd into this deposit. Use 'register' command.

RECM#> register cluster;
recm-inf: Registering cluster 'CLU12' (IP 192.168.1.62)
recm-inf: Cluster CID will be 1.
recm-inf: Cluster 'CLU12' registered in deposit 'REPO12' (cid=1)
RECM#>
```

IV.c) Configure cluster

The minimum configuration parameters you need to setup, after registration are :

| Cluster recm Parameter | Description |
|------------------------|--|
| waldir=<PATH> | Where WAL files are copied (The PostgreSQL parameter 'archive_command' contain the path. |
| Backupdir=<PATH> | Destination folder where backups are stored |

```
RECM#> modify cluster/waldir="/Library/PostgreSQL/12/WALfiles";
recm-inf: WAL directory must correspond to the 'archive_command' definition.
RECM#> modify cluster/backupdir="/Volumes/pg_backups";
RECM#> modify cluster/compression=9;
RECM#> modify cluster/delwal=10;
RECM#> modify cluster/maxsize=10G;
RECM#> modify cluster/maxfiles=200;
RECM#> modify cluster/full="count:3";
RECM#> modify cluster/cfg="days:7";
RECM#> modify cluster/meta="days:30";
RECM#>
```

At this step, your cluster is ready for been backuped.

V TAKE BACKUPS

V.a) FULL backups

Perform a FULL backup.

```
RECM#> backup full;
recm-inf: Backup UID : 000161ccc769387fe1e8
recm-inf: Total pieces ..... : 12
recm-inf: Total files backuped.. : 2364
recm-inf: Total Backup size .... : 717719965 (684 MB)
recm-inf: Total bytes skipped... : 16753      (0 MB)
recm-inf: Total files skipped... : 1
NOTICE: all required WAL segments have been archived
RECM#>
```

V.b) WAL backup

WAL files can be cleared automatically.

```
RECM#> backup wal/verbose
recm-inf: Scanning directory '/Library/PostgreSQL/12/WALfiles'
recm-inf: Starting WAL backup of '13/C7000060'
recm-inf: Copying WAL file '/Library/PostgreSQL/12/WALfiles/0000002300000013000000C3'
recm-inf: Creating piece '/Volumes/pg_backups/000161ccc7d51ee944e0_1_WAL.recm'
recm-inf: Copying WAL file '/Library/PostgreSQL/12/WALfiles/0000002300000013000000C4'
recm-inf: Copying WAL file '/Library/PostgreSQL/12/WALfiles/0000002300000013000000C5'
recm-inf: Copying WAL file '/Library/PostgreSQL/12/WALfiles/0000002300000013000000C6'
recm-inf: Backup Piece /Volumes/pg_backups/000161ccc7d51ee944e0_1_WAL.recm FileSize=66746
(Compression ratio: 99.90%)
recm-inf: deleting WAL '/Library/PostgreSQL/12/WALfiles/0000002300000013000000C3'
recm-inf: deleting WAL '/Library/PostgreSQL/12/WALfiles/0000002300000013000000C4'
recm-inf: Backup UID : 000161ccc7d51ee944e0
recm-inf: Total pieces ..... : 1
recm-inf: Total files backuped.. : 5
recm-inf: Total files deleted... : 2          (Files aged 10 minute(s) or more)
recm-inf: Total Backup size .... : 67109204 (64 MB)
RECM#>
```


V.c) CONFIG backup

Backup configuration files

```
RECM#> backup config/verbose
recm-inf: Creating piece '/Volumes/pg_backups/000161ccc82f19ab6d28_1_CFG.recm'
recm-inf: Add file '/Library/PostgreSQL/12/data/postgresql.auto.conf'
recm-inf: Add file '/Library/PostgreSQL/12/data/pg_hba.conf'
recm-inf: Add file '/Library/PostgreSQL/12/data/pg_ident.conf'
recm-inf: Add file '/Library/PostgreSQL/12/data/postgresql.conf'
recm-inf: Backup Piece /Volumes/pg_backups/000161ccc82f19ab6d28_1_CFG.recm FileSize=3976
(Compression ratio: 51.27%)
recm-inf: Backup UID : 000161ccc82f19ab6d28
recm-inf: Total pieces ..... : 1
recm-inf: Total files backedup.. : 4
recm-inf: Total Backup size .... : 8159      (7 KB)
RECM#>
```

V.d) META backup

Create METADATA files (generated by pg_dump utility)

```
RECM#> backup meta/verbose
recm-inf: Creating meta sql file '/pgdata/metadata_CLU12_demo.sql' for database 'demo'
recm-inf: Creating piece '/Volumes/pg_backups/000161ccc85d28269c38_1_META.recm'
recm-inf: Creating meta sql file '/pgdata/metadata_CLU12_depo_prod.sql' for database
'depo_prod'
recm-inf: Creating meta sql file '/pgdata/recm_meta/metadata_CLU12_recmdb.sql' for
database 'recmdb'
recm-inf: Creating meta sql file '/pgdata/recm_meta/metadata_CLU12_recmrepo.sql' for
database 'recmrepo'
recm-inf: Backup Piece /Volumes/pg_backups/000161ccc85d28269c38_1_META.recm FileSize=6162
(Compression ratio: 79.09%)
recm-inf: Backup UID : 000161ccc85d28269c38
recm-inf: Total pieces ..... : 1
recm-inf: Total files backedup.. : 4
recm-inf: Total Backup size .... : 29466     (28 KB)
RECM#>
```


VII RESTORE A CLUSTER

To avoid losing critical data, you need to perform the following actions before to proceed to a restore :

1. exit from recm (if connected)
2. stop the PostgreSQL engine
3. execute recm and execute the following commands

[connect the deposit only](#)

```
RECM> connect deposit/name=REPO12
```

[set source location](#)

```
RECM> set source/dir="/Volumes/pg_backups"
```

[Launch the restore](#)

```
RECM> restore full /dir="/Library/PostgreSQL/12/data"/port=5432/cid=1/keepdata
```