

pg_rman操作手册

作者: 1061529620@qq.com

1. 安装pg_rman

1.1 脚本介绍

(1) pgrman_conf.json

自定义pgrman的配置文件，包含主机IP、pgdata路径、备份路径(建议pgdata和备份路径在不同磁盘下)、端口、pg用户名和密码。

(2) create.sh

初始化脚本，里面配置了pg_rman的相关参数，并且创建了pg_arch和pg_log目录。

(3) pgrman_install.sh

pg_rman自动安装脚本，需要提前准备pgrman和pg-libs的rpm安装包。最后设置crontab定时任务调用backup.sh脚本进行自动备份

(4) pgrman_backup.sh

自动备份脚本，并且打印日志。

1.2 配置参数

pgrman_conf.json文件需要配置IP，物理机下的pg数据目录（不是容器内的目录），自定义的数据备份目录（建议备份在不同磁盘下），端口，用户名和密码

```
{
  "host": "192.168.119.83",
  "It is recommended that the data directory and backup directory be placed on different disks !!!": "",
  "data_path": "/home/pgdata",
  "backup_path": "/home/pgbackup",
  "port": "5434",
  "user": "postgres",
  "password": "postgres"
}
```

1.3 安装

注意：这边安装默认已经使用create.sh进行了初始化，修改参数

(1) 上传pg_rman安装目录

配置完pgrman_conf.json文件之后，将 pg_rman安装目录上传到需要安装的服务器上，以下的文档备份目录都用/home/pgbackup作为例子
目录结构如下：

```
pg_rman-1.3.8-1.pg10.rhel7.x86_64.rpm
pgrman_backup.sh
pgrman_conf.json
pgrman_install.sh
postgresql10-libs-10.7-1PGDG.rhel7.x86_64.rpm
```

(2) 执行安装脚本

添加脚本可执行的权限

```
chmod +x pgrman_install.sh
```

执行安装脚本

```
./pgrman_install.sh
```

安装之后，会对数据库进行一次压缩的全备份，如果数据量多，用时会比较久。
安装成功如下图：

```
[root@node01 pgrman]# ./pgrman_install.sh
INFO: installation package name: postgresql10-libs-10.7-1PGDG.rhel7.x86_64.rpm
INFO: installation package name: pg_rman-1.3.8-1.pg10.rhel7.x86_64.rpm
warning: postgresql10-libs-10.7-1PGDG.rhel7.x86_64.rpm: Header V4 DSA/SHA1 Signature, key ID 442df0f8: NOKEY
Preparing...##### [100%]
Updating / installing...
 1:postgresql10-libs-10.7-1PGDG.rhel7##### [100%]
INFO: install the postgresql10-libs-10.7-1PGDG.rhel7.x86_64.rpm successfully
Preparing...##### [100%]
Updating / installing...
 1:pg_rman-1.3.8-1.pg10.rhel7##### [100%]
INFO: install the pg_rman-1.3.8-1.pg10.rhel7.x86_64.rpm successfully
INFO: ARCLLOG_PATH is set to '/home/pgdata/pg_arch'
INFO: SRVLOG_PATH is set to '/home/pgdata/pg_log'
INFO: initialize the pg_rman successfully
INFO: If you want to use incremental backup, you must do a full backup
INFO: start first full backup, please wait patiently
INFO: copying database files
INFO: copying archived WAL files
INFO: backup complete
INFO: Please execute 'pg_rman validate' to verify the files are correctly copied.
INFO: start deleting old archived WAL files from ARCLLOG_PATH (keep files = 7, keep days = 7)
INFO: the threshold timestamp calculated by keep days is "2019-10-11 00:00:00"
INFO: start deleting old backup (keep generations = 3 AND keep after = 2019-10-11 00:00:00)
INFO: does not include the backup just taken
INFO: validate: "2019-10-18 14:47:50" backup and archive log files by CRC
INFO: backup "2019-10-18 14:47:50" is valid
INFO: the first full backup successfully
```

```
cd /home/pgbackup
```

目录结构如下:

```
[root@node01 pgbackup]# ll
total 12
drwx----- 3 root root  20 Oct 18 14:47 20191018
drwx----- 4 root root  34 Oct 18 14:47 backup
-rwxr-xr-x 1 root root 2089 Oct 18 14:53 pgrman_backup.sh
-rw-r--r-- 1 root root  221 Oct 18 14:47 pg_rman.ini
-rw-r--r-- 1 root root   40 Oct 18 14:47 system_identifier
drwx----- 2 root root   6 Oct 18 14:47 timeline_history
```

(3) 定时备份脚本

安装成功之后, 会增加两个定时任务, 分别是全量备份和增量备份。可以根据需求自定义修改执行时间。

备份是通过pgrman_backup.sh脚本来进行, 备份结束之后会打印日志。

路径为: /home/pgbackup/backup_log (执行第一次定时任务会自动创建该目录)

2. 卸载pgrman

2.1 删除环境变量

```
使用root用户
vi ~/.bashrc
将以下几列全部删除
export PG_RMAN=/usr/pgsql-10
export PATH=$PATH:$PG_RMAN/bin
export BACKUP_PATH=/home/pgbackup
export PGUSER=postgres
export PGPASSWORD=postgres
export PGPORT=5434
export PGHOSTADDR=192.168.14.32
export PGDATA=/home/pgdata
export ARCLLOG_PATH=/home/pgdata/pg_arch
export SRVLOG_PATH=/home/pgdata/pg_log
```

2.2 卸载rpm

```
rpm -e pg_rman-1.3.8-1.pg10.rhel7.x86_64
rpm -e postgresql10-libs-10.7-1PGDG.rhel7.x86_64
```

2.3 删除安装目录

```
rm -rf /usr/pgsql-10
/home/pgbackup目录是在pgrman_conf.json配置文件中自定义配置的, 如果里面的备份文件有需要, 那么该目录可以先提前备份
rm -rf /home/pgbackup
```

2.4 删除定时任务

在安装pg_rman的时候会自动配置定时任务, 所以需要删除下列两行

```
crontab -e
删除下列两行
0 */6 * * * /home/pgbackup/pg_rman_backup.sh incremental
0 4 * * * /home/pgbackup/pg_rman_backup.sh full
```

3. 备份恢复

如果因为某些原因, 导致数据丢失, 需要将数据库恢复到前面的某一个时间点。

3.1 停止数据库

需要停止pg之后, 才能执行备份恢复命令

```
kubect1 exec -it viid-pg-0 /bin/bash
su - postgres
/usr/lib/postgresql/10/bin/pg_ctl stop -D /var/lib/postgresql/data
kubect1 scale sts viid-pg --replicas 0
```

3.2 恢复数据库

(1) 恢复

数据库停止之后, 退出容器, 在物理机上执行pg_rman恢复命令, 且需要指定恢复时间戳, 可以通过pg_rman show命令来查看备份集的时间段, 选择EndTime 2019-10-18 14:53:47(可以大于等于)就行, 备份恢复最好就是指定时间点进行恢复。如下图:

```
[root@node01 pgrman]# pg_rman show
=====
StartTime                EndTime                Mode    Size    TLI    Status
=====
2019-10-18 14:47:50      2019-10-18 14:53:47    FULL    1905MB    1    OK
```

执行恢复命令

```
pg_rman restore --recovery-target-time "2019-10-18 14:53:47" --hard-copy
```

结果如下图:

```
[root@node01 pgrman]# pg_rman restore --recovery-target-time "2019-10-18 14:53:47" --hard-copy
INFO: the recovery target timeline ID is not given
INFO: use timeline ID of current database cluster as recovery target: 1
INFO: calculating timeline branches to be used to recovery target point
INFO: searching latest full backup which can be used as restore start point
INFO: found the full backup can be used as base in recovery: "2019-10-18 14:47:50"
INFO: copying online WAL files and server log files
INFO: clearing restore destination
INFO: validate: "2019-10-18 14:47:50" backup and archive log files by SIZE
INFO: backup "2019-10-18 14:47:50" is valid
INFO: restoring database files from the full mode backup "2019-10-18 14:47:50"
INFO: searching incremental backup to be restored
INFO: searching backup which contained archived WAL files to be restored
INFO: backup "2019-10-18 14:47:50" is valid
INFO: restoring WAL files from backup "2019-10-18 14:47:50"
INFO: restoring online WAL files and server log files
INFO: generating recovery.conf
INFO: restore complete
HINT: Recovery will start automatically when the PostgreSQL server is started.
```

(2) 修改recovery.conf文件

执行pg_rman恢复命令之后会在\$PGDATA目录下生成recovery.conf文件, 需要修改一下该恢复文件。

源文件内容如下:

```
[root@node01 pgdata]# more recovery.conf
# recovery.conf generated by pg_rman 1.3.8
restore_command = 'cp /home/pgdata/pg_arch/%f %p'
recovery_target_time = '2019-10-18 14:53:47'
recovery_target_timeline = '1'
```

需要修改两个点

```
vi recovery.conf
1.将restore_command = 'cp /home/pgdata/pg_arch/%f %p'中的
/home/pgdata($PGDATA)替换成容器中的路径, 改为:
restore_command = 'cp /var/lib/postgresql/data/pg_arch/%f %p'

2.在文件末尾添加;
```

```
recovery_target_action = 'promote'
```

(3) 修改目录所属用户

```
cd /home/pgdata/  
chown polkitd:input *
```

3.3 重启数据库

```
kubect1 scale sts viid-pg --replicas 1
```