**# vi pg\_backup.config**

##############################

## POSTGRESQL BACKUP CONFIG ##

##############################

# Optional system user to run backups as. If the user the script is running as doesn't match this

# the script terminates. Leave blank to skip check.

BACKUP\_USER=

# Optional hostname to adhere to pg\_hba policies. Will default to "localhost" if none specified.

HOSTNAME=

# Optional username to connect to database as. Will default to "postgres" if none specified.

USERNAME=

# This dir will be created if it doesn't exist. This must be writable by the user the script is

# running as.

BACKUP\_DIR=/var/lib/pgsql/11/backups/data/

# List of strings to match against in database name, separated by space or comma, for which we only

# wish to keep a backup of the schema, not the data. Any database names which contain any of these

# values will be considered candidates. (e.g. "system\_log" will match "dev\_system\_log\_2010-01")

SCHEMA\_ONLY\_LIST=""

# Will produce a custom-format backup if set to "yes"

ENABLE\_CUSTOM\_BACKUPS=yes

# Will produce a gzipped plain-format backup if set to "yes"

ENABLE\_PLAIN\_BACKUPS=yes

# Will produce gzipped sql file containing the cluster globals, like users and passwords, if set to "yes"

ENABLE\_GLOBALS\_BACKUPS=yes

#### SETTINGS FOR ROTATED BACKUPS ####

# Which day to take the weekly backup from (1-7 = Monday-Sunday)

DAY\_OF\_WEEK\_TO\_KEEP=5

# Number of days to keep daily backups

DAYS\_TO\_KEEP=14

# How many weeks to keep weekly backups

WEEKS\_TO\_KEEP=5

######################################

**#vi pg\_backup.sh**

#!/bin/bash

###########################

####### LOAD CONFIG #######

###########################

while [ $# -gt 0 ]; do

 case $1 in

 -c)

 if [ -r "$2" ]; then

 source "$2"

 shift 2

 else

 ${ECHO} "Unreadable config file \"$2\"" 1>&2

 exit 1

 fi

 ;;

 \*)

 ${ECHO} "Unknown Option \"$1\"" 1>&2

 exit 2

 ;;

 esac

done

if [ $# = 0 ]; then

# SCRIPTPATH=$(cd ${0%/\*} && pwd -P)

 SCRIPTPATH=$(cd /var/lib/pgsql/11/backups && pwd -P)

 source $SCRIPTPATH/pg\_backup.config

fi;

###########################

#### PRE-BACKUP CHECKS ####

###########################

# Make sure we're running as the required backup user

if [ "$BACKUP\_USER" != "" -a "$(id -un)" != "$BACKUP\_USER" ]; then

 echo "This script must be run as $BACKUP\_USER. Exiting." 1>&2

 exit 1;

fi;

###########################

### INITIALISE DEFAULTS ###

###########################

if [ ! $HOSTNAME ]; then

 HOSTNAME="localhost"

fi;

if [ ! $USERNAME ]; then

 USERNAME="postgres"

fi;

###########################

#### START THE BACKUPS ####

###########################

FINAL\_BACKUP\_DIR=$BACKUP\_DIR"`date +\%Y-\%m-\%d`/"

echo "Making backup directory in $FINAL\_BACKUP\_DIR"

if ! mkdir -p $FINAL\_BACKUP\_DIR; then

 echo "Cannot create backup directory in $FINAL\_BACKUP\_DIR. Go and fix it!" 1>&2

 exit 1;

fi;

#######################

### GLOBALS BACKUPS ###

#######################

echo -e "\n\nPerforming globals backup"

echo -e "--------------------------------------------\n"

if [ $ENABLE\_GLOBALS\_BACKUPS = "yes" ]

then

 echo "Globals backup"

 if ! pg\_dumpall -g -h "$HOSTNAME" -U "$USERNAME" | gzip > $FINAL\_BACKUP\_DIR"globals".sql.gz.in\_progress; then

 echo "[!!ERROR!!] Failed to produce globals backup" 1>&2

 else

 mv $FINAL\_BACKUP\_DIR"globals".sql.gz.in\_progress $FINAL\_BACKUP\_DIR"globals".sql.gz

 fi

else

 echo "None"

fi

###########################

### SCHEMA-ONLY BACKUPS ###

###########################

for SCHEMA\_ONLY\_DB in ${SCHEMA\_ONLY\_LIST//,/ }

do

 SCHEMA\_ONLY\_CLAUSE="$SCHEMA\_ONLY\_CLAUSE or datname ~ '$SCHEMA\_ONLY\_DB'"

done

SCHEMA\_ONLY\_QUERY="select datname from pg\_database where false $SCHEMA\_ONLY\_CLAUSE order by datname;"

echo -e "\n\nPerforming schema-only backups"

echo -e "--------------------------------------------\n"

SCHEMA\_ONLY\_DB\_LIST=`psql -h "$HOSTNAME" -U "$USERNAME" -At -c "$SCHEMA\_ONLY\_QUERY" postgres`

echo -e "The following databases were matched for schema-only backup:\n${SCHEMA\_ONLY\_DB\_LIST}\n"

for DATABASE in $SCHEMA\_ONLY\_DB\_LIST

do

 echo "Schema-only backup of $DATABASE"

 if ! pg\_dump -Fp -s -h "$HOSTNAME" -U "$USERNAME" "$DATABASE" | gzip > $FINAL\_BACKUP\_DIR"$DATABASE"\_SCHEMA.sql.gz.in\_progress; then

 echo "[!!ERROR!!] Failed to backup database schema of $DATABASE" 1>&2

 else

 mv $FINAL\_BACKUP\_DIR"$DATABASE"\_SCHEMA.sql.gz.in\_progress $FINAL\_BACKUP\_DIR"$DATABASE"\_SCHEMA.sql.gz

 fi

done

###########################

###### FULL BACKUPS #######

###########################

for SCHEMA\_ONLY\_DB in ${SCHEMA\_ONLY\_LIST//,/ }

do

 EXCLUDE\_SCHEMA\_ONLY\_CLAUSE="$EXCLUDE\_SCHEMA\_ONLY\_CLAUSE and datname !~ '$SCHEMA\_ONLY\_DB'"

done

FULL\_BACKUP\_QUERY="select datname from pg\_database where not datistemplate and datallowconn $EXCLUDE\_SCHEMA\_ONLY\_CLAUSE order by datname;"

echo -e "\n\nPerforming full backups"

echo -e "--------------------------------------------\n"

for DATABASE in `psql -h "$HOSTNAME" -U "$USERNAME" -At -c "$FULL\_BACKUP\_QUERY" postgres`

do

 if [ $ENABLE\_PLAIN\_BACKUPS = "yes" ]

 then

 echo "Plain backup of $DATABASE"

 if ! pg\_dump -Fp -h "$HOSTNAME" -U "$USERNAME" "$DATABASE" | gzip > $FINAL\_BACKUP\_DIR"$DATABASE".sql.gz.in\_progress; then

 echo "[!!ERROR!!] Failed to produce plain backup database $DATABASE" 1>&2

 else

 mv $FINAL\_BACKUP\_DIR"$DATABASE".sql.gz.in\_progress $FINAL\_BACKUP\_DIR"$DATABASE".sql.gz

 fi

 fi

 if [ $ENABLE\_CUSTOM\_BACKUPS = "yes" ]

 then

 echo "Custom backup of $DATABASE"

 if ! pg\_dump -Fc -h "$HOSTNAME" -U "$USERNAME" "$DATABASE" -f $FINAL\_BACKUP\_DIR"$DATABASE".custom.in\_progress; then

 echo "[!!ERROR!!] Failed to produce custom backup database $DATABASE" 1>&2

 else

 mv $FINAL\_BACKUP\_DIR"$DATABASE".custom.in\_progress $FINAL\_BACKUP\_DIR"$DATABASE".custom

 fi

 fi

done

echo -e "\nAll database backups complete!"