Backup Admin Guide

The Backup Solution

The backup is initiated with rsnapshot on the backup server (`potato-backup.colab.duke.edu`). The configuration can be found at `/etc/rsnapshot.conf`. rsnapshot runs a script at `/home/backupuser/rsync-wrapper.sh` on the production server (`potato.colab.duke.edu`), which dumps the database and transfers it to the backup server.

On the backup server, a script `/home/backupuser/run_rsnapshot.sh` calls rsnapshot, and send an email notification based the return status of the rsnapshot run. The script is scheduled with crontab to run on a scheduled basis.

Step-by-Step Set Up

On production server

As root

- Ensure `rsync` is installed
- Create a backupuser on the production server (vcm specific instruction)

```
sudo adduser backupuser
sudo apt install whois
mkpasswd --method=sha-512
sudo usermod -p '<password>' backupuser
```

- `sudo visudo` and add this line so that the backup user can run `rsync` without password with root permission

```
%backupuser ALL=(ALL) NOPASSWD: /usr/bin/rsync
```

- Create the same user in postgres

```
su postgres
psql
create user backupuser superuser password '<password>';
alter user backupuser set default_transaction_read_only = on;
```

As backupuser

- Confirm that the backup user can run `pg_dump`
- add `rsync-wrapper.sh` at ~

```
#!/bin/sh

pg_dump bus > potato/bus.sql
/usr/bin/sudo /usr/bin/rsync "$@"
```

Set execution permission

```
chmod ug+x /home/backup/rsync-wrapper.sh
```

On backup server

As root

- Create a backupuser on the backup server (vcm specific instruction)

```
sudo adduser backupuser
sudo apt install whois
mkpasswd --method=sha-512
sudo usermod -p '<password>' backupuser
```

- Install rsnaptshot

```
sudo apt install rsnapshot
```

- Config rsnapshot with this file

```
#####################################################
# rsnapshot.conf - rsnapshot configuration file #
#####################################################
#                                                   #
# PLEASE BE AWARE OF THE FOLLOWING RULE:            #
#                                                   #
# This file requires tabs between elements          #
#                                                   #
#####################################################

########################
# CONFIG FILE VERSION #
########################

config_version        1.2

############################
# SNAPSHOT ROOT DIRECTORY #
############################

# All snapshots will be stored under this root directory.
#
snapshot_root        /mnt/backup
```

```
#################################
# EXTERNAL PROGRAM DEPENDENCIES #
#################################

# LINUX USERS:   Be sure to uncomment "cmd_cp". This gives you extra
features.
# EVERYONE ELSE: Leave "cmd_cp" commented out for compatibility.
#
# See the README file or the man page for more details.
#
cmd_cp                /bin/cp

# uncomment this to use the rm program instead of the built-in perl
routine.
#
cmd_rm                /bin/rm

# rsync must be enabled for anything to work. This is the only command
that
# must be enabled.
#
cmd_rsync        /usr/bin/rsync

# Uncomment this to enable remote ssh backups over rsync.
#
cmd_ssh        /usr/bin/ssh

# Comment this out to disable syslog support.
#
cmd_logger        /usr/bin/logger


########################################
#     BACKUP LEVELS / INTERVALS        #
# Must be unique and in ascending order #
# e.g. alpha, beta, gamma, etc.        #
########################################

retain        daily        7
retain        weekly        4
retain        monthly        12

############################################
#              GLOBAL OPTIONS              #
# All are optional, with sensible defaults #
############################################

# Verbose level, 1 through 5.
# 1     Quiet           Print fatal errors only
# 2     Default         Print errors and warnings only
```

```
# 3     Verbose         Show equivalent shell commands being executed
# 4     Extra Verbose   Show extra verbose information
# 5     Debug mode      Everything
#
verbose               5

# Same as "verbose" above, but controls the amount of data sent to the
# logfile, if one is being used. The default is 3.
# If you want the rsync output, you have to set it to 4
#
loglevel        5

# ssh has no args passed by default, but you can specify some here.
#
rsync_long_args        -evaAX --rsync-path=/home/backupuser/rsync-
wrapper.sh
ssh_args        -i /home/backupuser/.ssh/id_ed25519

################################
### BACKUP POINTS / SCRIPTS ###
################################
backup        backupuser@potato.colab.duke.edu:/home/backupuser
/potato        ./
```

- Make a backup directory and transfer the ownership to bakcupuser

```
mkdir -p /mnt/backup
chown backupuser:backupuser /mnt/backup
chmod 770 /mnt/backup
```

As backupuser

- Create a pair of ssh-key and add it to the production server

```
ssh-keygen -t ed25519
ssh-copy-id backupuser@potato.colab.duke.edu
```

  Test with

```
ssh backupuser@potato.colab.duke.edu
```

- Add the main/mailing script at `~/run_rsnapshot.sh`

```
OUTPUT=`rsnapshot $@`
if [ $? -ne 0 ]
then
    printf "Here's your backup log: ${OUTPUT}" | /usr/bin/mail -s
"[Potato] Beta Server Backup Failed" zz160@duke.edu
else
    printf "Here's your backup log: ${OUTPUT}" | /usr/bin/mail -s
"[Potato] Beta Server Backup Succeeded" zz160@duke.edu
fi
```

- Configure crontab with `crotab -e` to run `/home/backupuser/run_rsnapshot.sh` on a regular basis

```
0 5 * * * /home/backupuser/run_rsnapshot.sh daily
0 4 * * 1 /home/backupuser/run_rsnapshot.sh weekly
0 3 1 * * /home/backupuser/run_rsnapshot.sh monthly
```

Manual Testing

run `rsnapshot <daily/weekly/montly>` if you want to test the backup, but don't want to send out an email notification.

run `/home/backupuser//home/backupuser/run_rsnapshot.sh <daily/weekly/monthly>` if you want to spam me.

Disaster Recovery

Retrieve Database Backup

Access backup database at

```
ssh backupuser@potato-backup.colab.duke.edu
```

All databases are stored at `/mnt/backup`. `/daily/weekly/monthly.x` has a daily/weekly/monthly backup. The smaller `x` is, the more recent the backup is.

Restore Database

Access backup database at

```
ssh backupuser@potato.colab.duke.edu
```

To download `daily.0`, for example, to the current directory, run

```
scp backupuser@potato-backup.colab.duke.edu:/mnt/backup/daily.0/potato
/bus.sql .
```

Switch to user `postgres`

```
su postgres
```

If the database is not empty, you may encounter some problems when restoring the database. The easiest way is to recreate the database

```
dropdb bus; createdb bus
```

And then restore the database

```
psql bus < bus.sql
```

Either check the database on the command line, or log in to the website as admin to verify that all data have been restored.