Scheduling Backups

Tomás E. Müller Bravo

1 Introduction

This document briefly explains how to schedule backups of files (NOT system files). It is always good to update and upgrade the packages in your system before you start working at a system level. It is best to work as a super user: "`sudo -s`".

2 RAID for data storage

A RAID stands for "redundant array of independent disks," which is a way of storing data on multiple drives. There are many RAID levels (e.g. RAID 0, 1, 5, 10), but for simplicity, here we use level 1, which uses "data mirroring," where all the data written to the array is written to two disks simultaneously. I read a blog that explains this in more detail [1].

For setting up the RAID, I followed another blog that explains everything in detail [2]. Before setting up the RAID, we can list all attached hard disks with "`fdisk -l`" command or "`lsblk`" command to view a structured overview of all attached storage devices.

Assuming you have two external hard disks under `/dev/sda` and `/dev/sdb`, the following command creates a level 1 RAID:

```
--create --verbose /dev/md1 --level=1 --raid-devices=2 /dev/sda /dev/sdb.
```

You can check if it was correctly created with "`cat /proc/mdstat`".

We cannot use a RAID array for data storage until it contains a valid file system. The following command is used to create a file system in the array: "`mkfs -t ext4 /dev/md1`".

3 Mounting external hard disks

For backup storage, the best option is to permanently mount the RAID array. For this, we should use the UUID instead of its name as the UUID never changes. To know the UUID, use the `blkid` command.

When the system boots, it looks in the `/etc/fstab` file to find out the devices which need to be mounted in the file system automatically. To mount any additional devices automatically we have to make an entry for that device in the file mentioned above:

```
UUID=123abczxxxxx /where/to/mount ext4 defaults 0 0
```

[1] https://www.intego.com/mac-security-blog/understanding-raid-for-data-storage-and-backups/

[2] https://www.computernetworkingnotes.com/linux-tutorials/how-to-configure-raid-in-linux-step-by-step-guide.html

After saving, always check the entries with "`mount -a`" command. This command will mount everything listed in the `/etc/fstab` file. So, if we made any mistake while updating this file, we will get an error as the output of this command.

4 Backups with rsync

The purpose of `rsync` is mainly just to copy files from a source to a destination and only updates if the files from source are new of have been changed. It is good practice to save the output log into a file. The following command is used:

```
rsync --log-file=/var/log/backup.log -av --delete src dest
```

where "`-a`" is "archive mode", which includes a variety of options (such as compressing and recursive), and "`-v`" is simply verbose. The "`--delete`" option just deletes any files in the destination directory that are not in the source directory.

As connecting to remove server usually requires password, this can be stored in a secured file and given as a parameter to rsync. For this, "sshpass" needs to be installed and the following needs to be added to the rsync command:

```
--rsh="sshpass -p `cat /etc/.backup_password` ssh -o StrictHostKeyChecking=no"
```

where `/etc/.backup_password` is a hidden file which can be protected with "`chmod 400`" to make sure only your user can read it.

The following command was used to backup the flows database:

```
rsync --log-file=/var/log/rsync_flows_backup.log -av --delete
    --rsh="sshpass -p `cat /etc/.backup_password`
        ssh -o StrictHostKeyChecking=no"
            max@10.28.1.72:/data/DS1/FLOWS/archive /md_flows/md1/backup/maual
```

5 Cron Jobs Scheduler

As rsync needs to be run manually, we can use cron job scheduler to schedule periodic backups. To create a cronjob, you edit your crontab using the "`crontab -e`" command. The following options are available to schedule jobs: minute - hour - day_of_month - month - day_of_week. Therefore, you have the following syntax: "`m h dom mon dow command`"

As the command from rsync can get really long, this can go into an executable (.sh bash) file. This is also useful to create custom commands for daily, weekly and monthly backups, for instance, as you can give a different name to each backup and output log.

The following jobs were added with "`crontab -e`" to backup the flows database:

```
# daily at 12:00 (noon)
00 12 * * * /etc/run_backup_daily.sh
# weekly at 14:30 on Friday
30 14 * * 5 /etc/run_backup_weekly.sh
# monthly at 08:00 on the 1st
00 08 1 * * /etc/run_backup_monthly.sh
```