Shell Script to Backup Files

It explains how to create a bash script which can backup the files you need. This backup process will be automatic and it is something that every linux user or admin needs.

Use tar in order to create your backup file

You can use a simple text editor or a command line editor like vi, in order to edit/create your script, let's name it *backup_script.sh.* Remember that .sh is the extension that the bash scripts must have.

The code of your *backup_script.sh* script will be the following:

```
#! /bin/bash                              # First Line of bash script
BackupTime = `date +%b-%d-%y`             #get the current date
#Create a backup file using current date.
Destination = /home/usr/path/backup-$BackupTime.tar.gz
#the folder that contains the files that we want to backup
Source = /home/usr/path/folder
tar –cpzf $Destination $Source            #Create the backup
```

The important command here is the following:

```
        tar –cpzf $Destination $Source
```

But what about *-cpzf* parameter?

c: create

p: preserve permissions for the new files

z: compress the files in order to reduce the size
f: use archive file or device ARCHIVE

Schedule your backup task

We are going to use cron, which is a linux daemon that allows us to run scripts in certain scheduled moments. crontab files automate backups, system maintenance and other useful tasks.
In order to edit the crontab file with the editor you prefer (nano is the easiest), run in a terminal the command

```
crontab -e
```

Let’s understand the contrab line format:

minute(0–59) hour(0–23) day(1–31) month(1–12) weekday(0–6) command

Let’s say that we want to run the script everyday at 12:30 a.m. we would type

29 0 * * * /bin/bash/path/backup_script.sh

29 stands for the 30 minute
0 stands for 12 a.m.
The first * stands for everyday, the second * stands for every month, the third * stands for every week day.