Instructor: M. Ahsan

Automate Backups Using RSYNC + CRON

Want to set up a backup for your files or sync files across different machines? Not a big deal! The rsync command have got your back. The rsync or remote sync helps you backup your whole system. It is a utility that helps you sync from a source directory to a destination directory within your local system. It not only helps you sync within your local system but also sync across different machines such as your local machine and a remote machine.

In addition to rsync you have the crontab as well that will help you automate your backups rather than doing it manually all the time. Cron is a software utility that helps you schedule commands as per your need. Using Cron you don't have to execute commands manually all the time. You can simply automate them according to your time preferences. So, without further ado, let's get started and get introduced to some basic useful and important rsync and crontab commands.

Installig RSYNC:

The first step is to install rsync using the commands below:

```
sudo apt update
sudo apt-get install rsync
```

Syncing Data within Local Machine:

Next, for syncing files across different directories, navigate to the location where your directories, that needs to be synced, are stored. In my case it's in the home directory.

```
cd ~
```

Next, for syncing data, use the command:

General Syntax:

```
rsync [-options] SOURCE DESTINATION
```

In my case:

```
rsync Original/* Backup/
```

In the above command, Original is the name of my source directory that I want to sync. Original/* will copy all the files in my source directory (Original). Backup is the name of my destination directory. The files copied from my source directory (Original) will be copied to my destination directory (Backup). If we re-execute the same command so it won't copy any files that are already copied before but if we have placed any new files in the source directory before re-execution so it will only copy the new files into the destination directory without copying the ones that have already been synced(copied) before.

For syncing not only the files within a directory but all the content of the directory, including sub directories, use the command:

```
rsync -rv Original/ Backup/
```

The **-v** option will give you information about what files are being transferred and a brief summary at the end.

For syncing a whole directory within another directory, use the command:

```
rsync -rv Original Backup/
```

This will copy the source directory as a whole into the destination Backup.

If you want to check what files you will be copying, use the command:

```
rsync -av –dry-run Original/ Backup/
```

Due to the dry run option it will not copy the content but just show you what you are about to copy, before actually copying it.

If you have some files present in the destination directory which are not there in the source directory so running the below command will delete all those files:

```
rsync -rv –delete Original/ Backup/
```

Syncing Data Across different Machines:

Using the rsync command you can push data into a remote machine or pull from it.

1) For pushing data into remote machine:

General Syntax:

```
rsync [-options] SOURCE user@x.x.x.x:DESTINATION
```

In my case:

```
rsync -rv Original/ zawster@192.168.1.11:~/Downloads/
```

Or if you want to compress the data while transferring, you can use the same command with some modifications:

```
rsync -zaP Original/ zawster@192.168.1.11:~/Downloads/
```

The **-z** option is used for compressing the data and **-P** is used for checking the progress of data transfer.

If you don't have set key-based authentication for your remote machine, it will ask you for a password.

2) For pulling data from remote machine:

General Syntax:

```
rsync [-options] user@x.x.x.x:SOURCE DESTINATION
```

In my case:

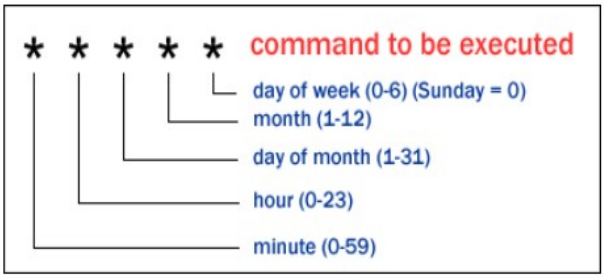
```
rsync -rv zawster@192.168.1.11:Original/ ~/Downloads/
```

Automated Backups:

In order to automate your backups, you first need to understand how Cron works. The general syntax for automating a task is given below:

```
* * * * * <command to be executed>
```

Now for details, look at the picture below:



- The first star is for minutes. Here we specify a number between 0 and 59 to define the exact minute when the scheduled task will run.
- The second star is for hour and here we can enter a number between 0 and 23 to specify the hour at which we want to execute the command.
- The third star is to specify the day of month.
- The fourth star is to specify the month in which we want to schedule our task.
- The fifth star is to specify the day of the week.

Now let's look into the steps that we will follow to automate our backups.

METHOD # 01

To add a task in your Cron Job list, use the command below:

```
crontab -e
```

If you are doing this for the first time it will ask you to choose an editor that you want to use for editing the list of cron tasks.

Once you choose the editor it will take you to a file where you can simply schedule your task.

In my case, I want my data to be synced every Friday at 8:00 AM so I will add the following line in my cron job list.

```
0 7 * * 6 rsync -a Original/ zawster@192.168.1.11:~/Downloads/
```

The above command means that every Saturday at 7:00 AM my local machine should backup my data from source Directory into the destination Directory of remote machine.

Now exit from this file and to make sure your scheduled cron task have been successfully saved, view your crontab list using the command:

```
crontab -l
```

This is how you can automate your tasks.

METHOD # 02:

For method two, first of all create a cron file at the location /etc/cron.d/ .

Use the command below:

```
vi /etc/cron.d/backup.cron
```

Here, backup.cron is the name of my cron file. You can choose any name.

Next add this line of code in your cron file.

```
0 7 * * 6 rsync -av Original/ zawster@192.168.1.11:~/Downloads/ >> /var/log/
cron.log 2>&1
```

This line of code will sync the files every Saturday 7:00 AM and will write the standard output as well standard errors in cron.log file.

Now exit from the backup.cron file.

Next, execute the following commands to bring your cron task into working.

```
touch /var/log/cron.log
crontab /etc/cron.d/backup.cron
cron
```

To view the log of your backups use the command:

cat /var/log/cron.log

You can choose any of the above methods and both should work fine.