

Linux Project Automatic Backup of one server to another server using Rsync & Crontab

Backup today is a lifeline to the future for small, medium, and enterprise organizations. That's because data is the world's most valuable resource and you have to protect it. So to protect our data we can implement security

Of course, to ensure that's true no matter what happens — from natural disasters and human error to ransomware attacks -

A backup plan that includes local and cloud backup

In this project I have implemented ssh to access remote server and used rsync utility to take a backup from one machine to another remote machine.

Rsync

`rsync` is a powerful command-line tool for syncing files and directories between two locations. It's commonly used for backups and mirroring data, thanks to its efficiency and versatility.

`rsync` is a versatile tool that provides efficient file synchronization and copying capabilities. Whether you're performing local backups, syncing files across different systems, or automating routine tasks, `rsync` offers robust features that can be customized to suit various use cases in Linux environments.

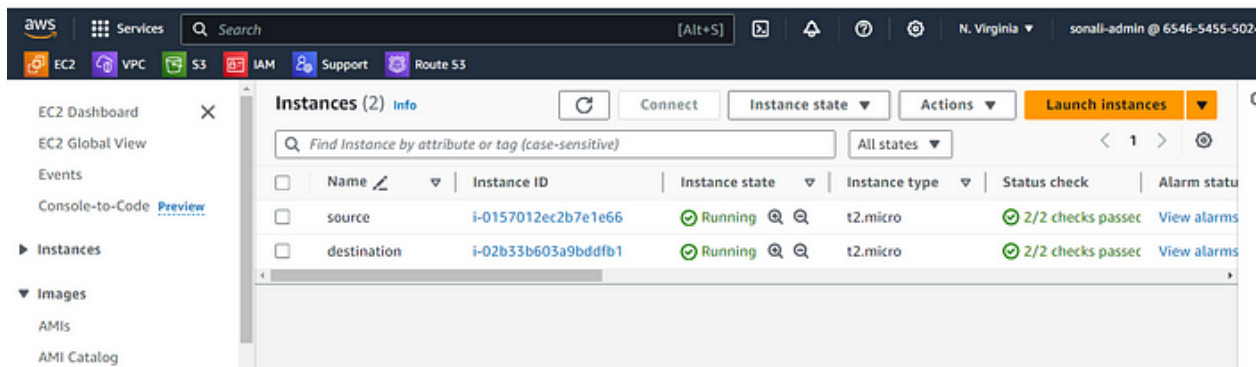
Crontab

The `crontab` command in Linux is used to schedule tasks (commands or scripts) to run automatically at specified intervals or times. Using `crontab`, you can automate repetitive tasks, backups, system maintenance, and more, making it a powerful tool for managing routine tasks on Linux systems.

Introduction

Automating backups in Linux using `cron` and `rsync` is a practical approach to ensure regular and reliable backups of your important data. Here's a step-by-step guide to set it up:

- Launch two instances.
- Connect the instances.



- Check the repo list (yum repolist all or ls /etc/yum.repos.d)
- Install the rsync command (sudo yum install rsync -y)
- On the destination instance also check the repolist and install the rsync command.
- Create one directory in source instance


```

admin@DESKTOP-CLT1FU5 MINGW64 ~/Downloads (master)
$ scp -i "sonali-linux-key.pem" /c/Users/admin/Downloads/sonali-linux-key.pem ec
2-user@ec2-34-207-234-210.compute-1.amazonaws.com:
The authenticity of host 'ec2-34-207-234-210.compute-1.amazonaws.com (34.207.234
.210)' can't be established.
ED25519 key fingerprint is SHA256:I7h0PvnA4xeg5gPJJCAmYQdGz6XT2wG0v4wCFvSncKc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-207-234-210.compute-1.amazonaws.com' (ED25519
) to the list of known hosts.
sonali-linux-key.pem

admin@DESKTOP-CLT1FU5 MINGW64 ~/Downloads (master)
$

```

- Key is copied in Source Instance.



```

#
~\#### Amazon Linux 2023
~~\#####
~~\####
~~\#/ https://aws.amazon.com/linux/amazon-linux-2023
~~V~'-'>
~~~
~~~
~~~
Last login: Fri Aug 2 13:59:56 2024 from 18.206.107.27
[ec2-user@ip-172-31-19-213 ~]$ ls
sonali-linux-key.pem  src_dir
[ec2-user@ip-172-31-19-213 ~]$

```

By writing rsync command manually copied file from source to destination.

```

sudo rsync -avz -e "ssh -i /home/ec2-user/sonali-linux-key.pem" /home/ec2-
user/src_dir/* ec2-user@ec2-54-167-114-62.compute-1.amazonaws.com:
/home/ec2-user/dest_dir

```

```
aws | Services | Search | [Alt+S] | N. Virginia | sonali-admin @ 6546-5455-
EC2 VPC S3 IAM Support Route 53
Connection to ec2-54-167-114-62.compute-1.amazonaws.com closed.
[ec2-user@ip-172-31-19-213 ~]$ ls
sonali-linux-key.pem src_dir
[ec2-user@ip-172-31-19-213 ~]$ sudo rsync -avz -e "ssh -i /home/ec2-user/sonali-linux-key.pem" /home/ec2-user/src_dir/* ec2-user@ec2-54-167-114-62.compute-1.amazonaws.com:/home/ec2-user/dest_dir
sending incremental file list
file1
file10
file2
file3
file4
file5
file6
file7
file8
file9

sent 546 bytes  received 206 bytes  1,504.00 bytes/sec
total size is 0  speedup is 0.00
[ec2-user@ip-172-31-19-213 ~]$
```

- File is copied in destination.

```
EC2 VPC S3 IAM Support Route 53
[ec2-user@ip-172-31-16-87 dest_dir]$ ls
file1 file10 file2 file3 file4 file5 file6 file7 file8 file9
[ec2-user@ip-172-31-16-87 dest_dir]$
```

- we deleted all the file which we have copied from source

```
EC2 VPC S3 IAM Support Route 53
[ec2-user@ip-172-31-16-87 dest_dir]$ ls
file1 file10 file2 file3 file4 file5 file6 file7 file8 file9
[ec2-user@ip-172-31-16-87 dest_dir]$ sudo rm -f *
[ec2-user@ip-172-31-16-87 dest_dir]$ ls
[ec2-user@ip-172-31-16-87 dest_dir]$
```

- We are creating backup.sh file to automate the backup.

```
[ec2-user@ip-172-31-19-213 ~]$
[ec2-user@ip-172-31-19-213 ~]$
[ec2-user@ip-172-31-19-213 ~]$ sudo vim /root/backup.sh
```

- backup.sh file

```
#!/bin/bash
sudo /usr/bin/rsync -avz -e "ssh -i /home/ec2-user/sonali-linux-key.pem" /home/ec2-user/src_dir/* ec2-user@
c2-54-167-114-62.compute-1.amazonaws.com:/home/ec2-user/dest_dir

-- INSERT --
```

Changed permission of backup.sh file

```
[ec2-user@ip-172-31-19-213 ~]$  
[ec2-user@ip-172-31-19-213 ~]$ sudo vim /root/backup.sh  
[ec2-user@ip-172-31-19-213 ~]$ sudo chmod +x /root/backup.sh  
[ec2-user@ip-172-31-19-213 ~]$
```

- Creating a cronjob for automate the backup

```
[ec2-user@ip-172-31-19-213 ~]$ sudo crontab -e
```

- Faced This issue

```
[ec2-user@ip-172-31-19-213 ~]$ sudo crontab -e
sudo: crontab: command not found
[ec2-user@ip-172-31-19-213 ~]$
```

- Troubleshoot above error by installing crontab.
- Enable crontab

```

[ec2-user@ip-172-31-19-213 ~]$ sudo crontab -e
sudo: crontab: command not found
[ec2-user@ip-172-31-19-213 ~]$ sudo yum install cronie
Last metadata expiration check: 2:25:24 ago on Fri Aug 2 13:48:08 2024.
Dependencies resolved.
=====
Package                               Architecture      Version           Repository        Size
-----
Installing:
cronie                               x86_64            1.5.7-1.amzn2023.0.2  amazonlinux      115 k
Installing dependencies:
cronie-anacron                       x86_64            1.5.7-1.amzn2023.0.2  amazonlinux      32 k
=====
Transaction Summary
-----
Install 2 Packages

Total download size: 147 k
Installed size: 341 k
Is this ok [y/N]: y
Downloading Packages:
(1/2): cronie-anacron-1.5.7-1.amzn2023.0.2.x86_64.rpm 524 kB/s | 32 kB 00:00
(2/2): cronie-1.5.7-1.amzn2023.0.2.x86_64.rpm       1.7 MB/s | 115 kB 00:00
-----
Total                                           1.1 MB/s | 147 kB 00:00
Running transaction check
Transaction check succeeded.

```

- In Crontab we have written command directly

```

[ec2-user@ip-172-31-19-213 ~]$ sudo crontab -e
crontab: installing new crontab
[ec2-user@ip-172-31-19-213 ~]$ sudo crontab -l
* * * * * sudo rsync -av -e "ssh -i /home/ec2-user/sonali-linux-key.pem" /home/ec2-user/src_dir/* ec2-user@ec2-107-21-72-30.compute-1.amazonaws.com:/home/ec2-user/dest_dir
[ec2-user@ip-172-31-19-213 ~]$

```

- Copied file in destination

```

[ec2-user@ip-172-31-16-87 dest_dir]$ ls
file1 file10 file2 file3 file4 file5 file6 file7 file8 file9
[ec2-user@ip-172-31-16-87 dest_dir]$

```

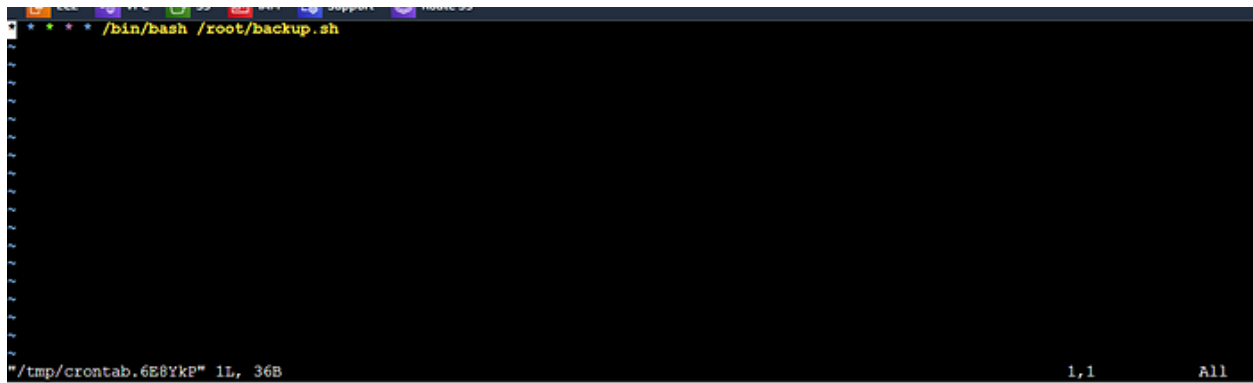
- Assigned Executable permission to backup.sh file. Permission changed .

```

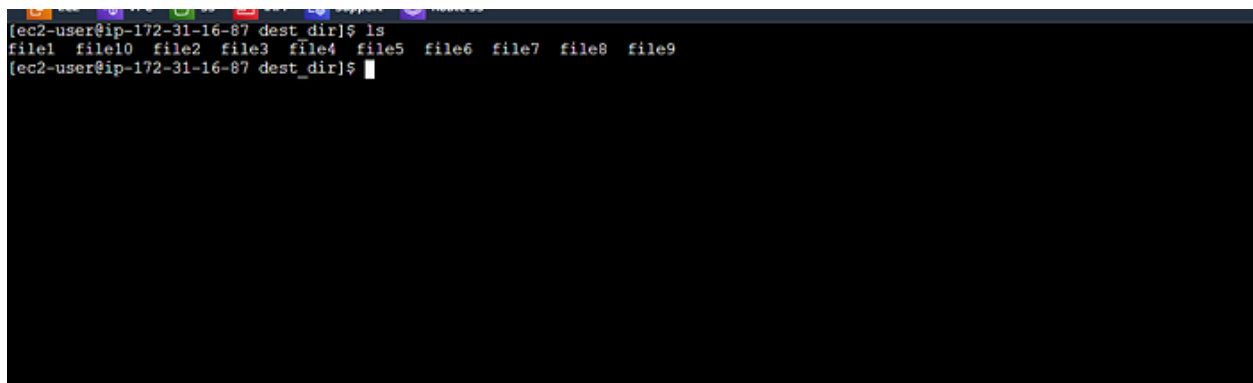
-rwxr-xr-x. 1 root root 175 Aug 3 07:51 backup.sh
drwxr-xr-x. 2 root root 137 Aug 3 07:06 ec2-user@e
[root@ip-172-31-19-213 ~]#
logout
[ec2-user@ip-172-31-19-213 ~]$ sudo crontab -l
* * * * * /bin/bash /root/backup.sh
[ec2-user@ip-172-31-19-213 ~]$
[ec2-user@ip-172-31-19-213 ~]$
[ec2-user@ip-172-31-19-213 ~]$
[ec2-user@ip-172-31-19-213 ~]$
[ec2-user@ip-172-31-19-213 ~]$

```

- Crontab written with scripting file.

A terminal window with a dark background. The title bar shows several application icons. The terminal prompt is `*/tmp/crontab.6E8YkP`. The user has entered `1L, 36B` and the terminal shows `1,1` and `All` at the bottom right.

- Copied file.

A terminal window with a dark background. The prompt is `[ec2-user@ip-172-31-16-87 dest_dir]$`. The user has entered `ls` and the terminal shows the output: `file1 file10 file2 file3 file4 file5 file6 file7 file8 file9`. The prompt is now `[ec2-user@ip-172-31-16-87 dest_dir]$`.

Summary

Following these steps, you can effectively automate backups using `cron` and `rsync` on your Linux system. This setup provides a robust solution for maintaining regular backups of critical files.