**BACKUPS, OFF-SITE BACKUPS, AND AUTOMATION**

***ISEC3700 - ASSIGNMENT 6***

**NSCC INSTITUTE OF TECHNOLOGY**

**5685 LEEDS ST**

**HALIFAX, NS B3K 2T3**

**NOVEMBER 30, 2018**

Contents

[Activity List 2](#_Toc531352500)

[Reference Figures 7](#_Toc531352501)

[References 15](#_Toc531352502)

# Activity List

Here’s an overview of the structure of this assignment. Make sure to replace all instances of rrennehan with your actual username.

| **Activity List** | | | |
| --- | --- | --- | --- |
| **Project: Backups, Off-site backups, and Automation** | | **Date: Nov 30, 2018** | |
| **Activity** | **Instructions** | **Notes** | **Reference Figures** |
| Create a backup of all databases in Ubuntu server | Save them to the /data directory created. Create the /data directory in your home directory if not already created.  Use compression format to store the backups. Include date (MM,DD,YYYY) as part of the backup file name.  The command is: mysqldump -u root –p yourmySQLPasswordHere --all-databases -C | gzip >$HOME/data/mysqldb\_`date + “%m\_%d\_%Y”`.sql.gz  You can optionally set up a script and crontab to run this automatically at certain times. | Enter your actual mySQL password after the -p option  You can set up a file that prevents you from having to enter the password in the mysqldump command. See figures 1 and 2 for more details  My system has a crontab that runs this command once a week | 1-2 |
| Install openssh on both the Ubuntu and CentOS server | For Ubuntu: sudo apt install openssh-server  For CentOS: sudo yum install openssh-server | It’s likely that openssh will already be installed and up to date. | 3-4 |
| Generate passwordless ssh key | Enter command: *ssh-keygen -t rsa*  Keep hitting enter to skip prompts for location and password. Find the ssh keypair in ~/.ssh |  | 5 |
| Copy pub key to remote server (CentOS) | Copy the id\_rsa.pub key from Ubuntu to CentOS.  *ssh-copy-id -i ~/.ssh/id\_rsa.pub USERNAME@REMOTE\_HOST\_IP* | Replace username and remote host ip with that of destination. | 6-7 |
| Create backup directory on CentOS server | *mkdir backup*  For the purposes of this assignment, we will place it in the home directory. |  | 8 |
| Create simple backup script on Ubuntu server | The script will copy everything in /data directory to the /backup on your remote server.  Start by using the touch command to create a databackup.sh file. Use an editor to write the script. Inside the script, write this command at the bare minimum:  *rsync -e 'ssh -p 22' -avzp /home/rrennehan/data REMOTE\_HOST\_IP:/home/rrennehan/backup*  Full script found in reference figures. Don’t forget to apply execution permissions using chmod |  | 9 |
| Test backup script | Run the command: *./databackup.sh* |  | 10-11 |
| Install rdiff-backup on both servers. | Insert the following commands to install rdiff-backup.  On Ubuntu:   * sudo apt-get update * sudo apt-get install librsync-dev rdiff-backup   On CentOS 7:   * sudo yum install librsync-dev rdiff-backup |  | N/A |
| Ensure both servers have same version of rdiff-backup | Use command: *rdiff-backup -V*  The output on both servers should match. |  | 12 |
| Create rdiff-backup for /data to remote /backup | Use following command from CentOS backup server to make backup of data directory from remote Linux Server to the /backup directory.  *rdiff-backup rrennehan@REMOTE\_IP::/home/rrennehan/data /home/rrennehan/backup/rdiffDataBackup* | Replace remote IP with actual IP address. | 13 |
| Show rdiff-backup job statistics | Use the --print-statistics option to show summary statistics after a successful backup.  Command is: *rdiff-backup --print-statistics rrennehan@REMOTE\_IP::/home/rrennehan/data /home/rrennehan/backup/rdiffDataBackup* |  | 14 |
| Show backup increments | Show using -l option a list of increments taken on the new rdiff backup folder. An increment is only counted when there are new files or file alterations after running a backup.  *Command: rdiff-backup -l /home/rrennehan/backup/rdiffDataBackup* |  | 15 |
| Create rdiff-backup script | Enter the working rdiff-backup command into a script so it can be easily run by you and crontab. Full script found in reference figures section |  | 16 |
| Test rdiff-backup script | Be sure to add execution permissions to the script. Run the script by entering its file path in the terminal. |  | 17 |
| Cron job every Tuesday 11:30am for rsync backup from Ubuntu | From Ubuntu, enter the crontab page with command: *sudo crontab -e*  Cron job links to the newly created Rsync backup script. |  | 18 |
| Cron job every Friday at 2:30pm doing an rdiff-backup of remote Ubuntu /data directory to /backup directory | From CentOS, enter the crontab page with command: *sudo crontab -e* |  | 19 |

# Reference Figures

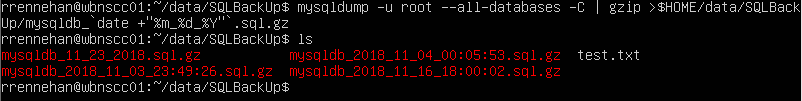


Figure 1: Creating a compressed backup of all databases with date included in the name. The file added is the one made on November 23. It’s important to note that I have a specific hidden file in my home directory that prevents me from having to enter the password in the terminal. Assuming you don’t have such a file created, you’ll need to start the command with *mysqldump -u root -p YourPasswordHere*



Figure 2: If you wish to avoid having to enter your password on the command line for the mysqldump command, make a file with this structure. Ensure it is placed at the root of your home directory. Replace the password 1tsas5cr5t with your actual mySQL password. Same with username. Make sure the filename starts with a period so it is hidden.

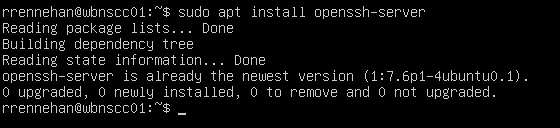


Figure 3: Attempting to install openssh on Ubuntu server. My Ubuntu server already has it installed and up to date

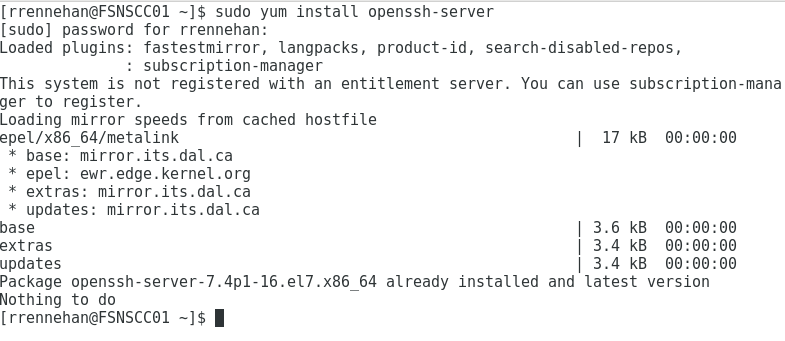


Figure 4: Attempting to install openssh on CentOS server. My CentOS server had it installed by default.

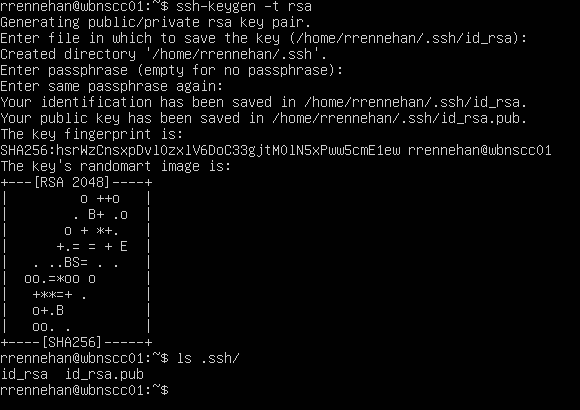


Figure 5: Generating passwordless ssh keys

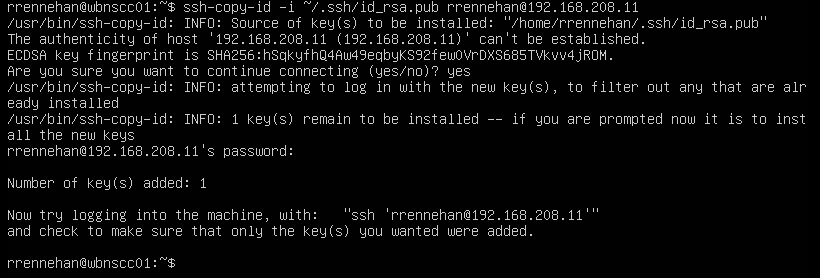


Figure 6: Command to copy ssh key and the result of it

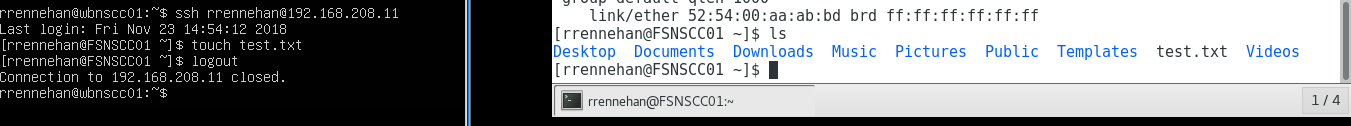


Figure 7: Testing remote login functionality by adding a new file from the remote server.

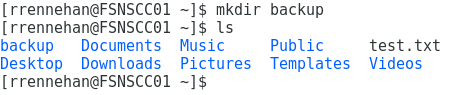


Figure 8: Creating backup directory in CentOS home directory

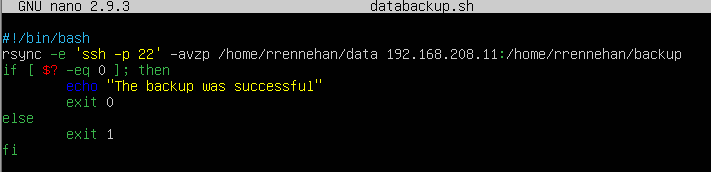


Figure 9: Script to backup data directory to remote backup directory

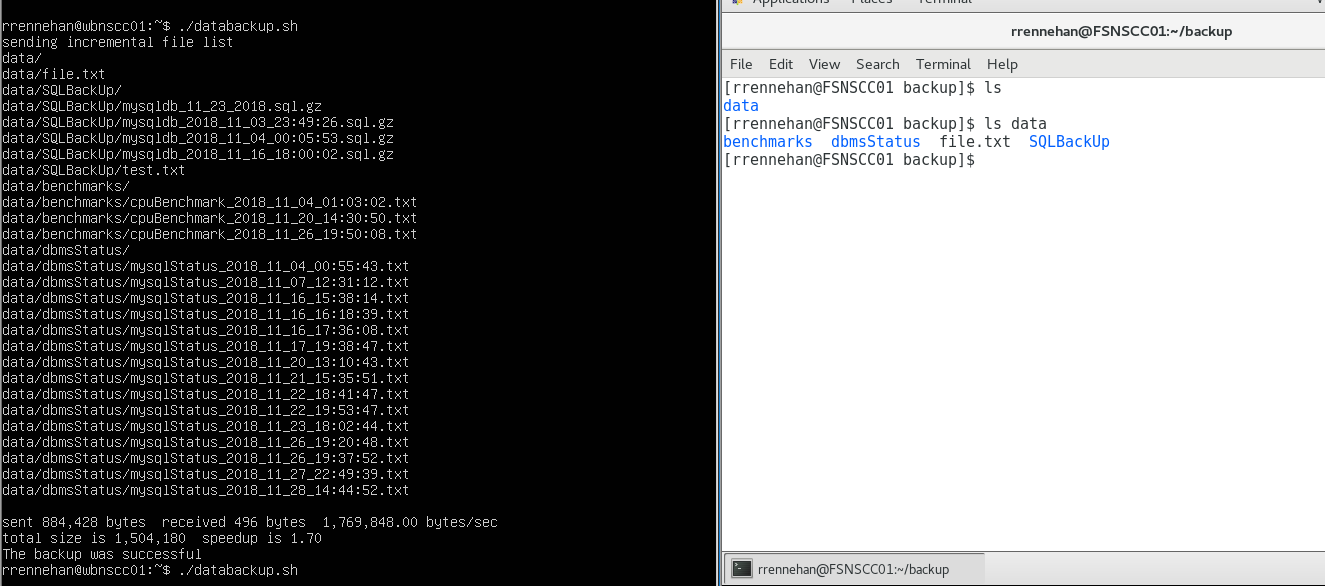


Figure 10: Backup script in action (/data vs /backup)

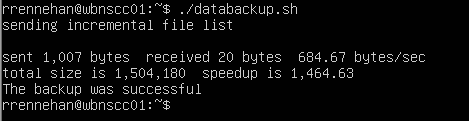


Figure 11: Running the backup script a second time in a row. Notice how rsync does not show files being copied if they already exist in the remote server

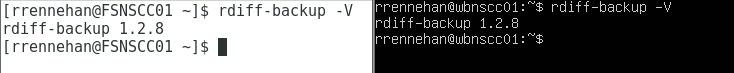


Figure 12: Comparing rdiff-backup versions installed on CentOS (left) and Ubuntu (right)

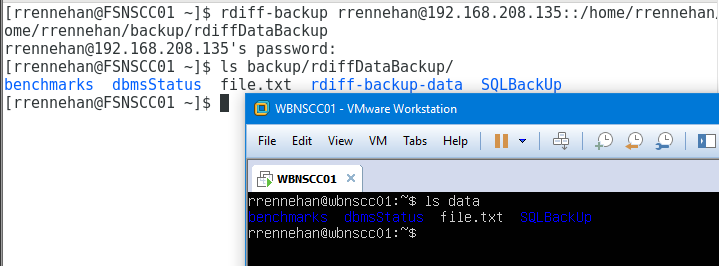


Figure 13: Result after running rdiff-backup command to backup data directory. Comparing backup to the data directory

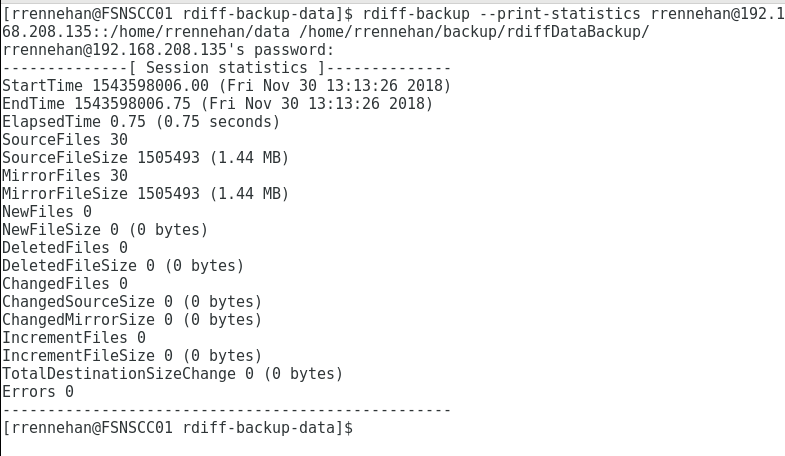


Figure 14: Adding the --print-statistics option

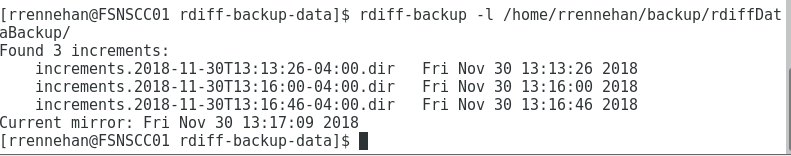


Figure 15: Checking amount of increments done on rdiff-backup folder



Figure 16: My rdiffBackup script. Because my Ubuntu machine swaps between two IP addresses, I try the second if the first fails.

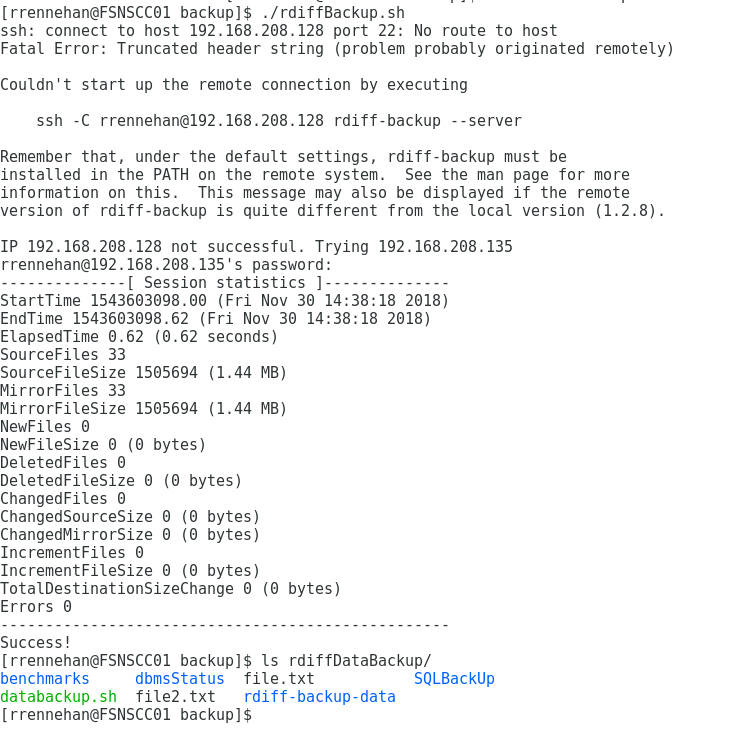


Figure 17: Running my rdiffBackup.sh script. My Ubuntu machine swaps between two IP addresses, so after the first one fails, it tries the second



Figure 18: Ubuntu Crontab to remotely backup data on Tuesdays 11:30 AM

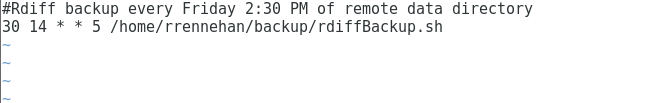


Figure 19: CentOS Crontab to backup remote data directory on Fridays 2:30 PM

# References

No academic references