Hanish Patel

CS 497-01

Assignment 3

CS 497 – Assignment 3

**Abstract**

Examine Linux package management and installation through Ubuntu. Understand the package management in Ubuntu and the steps to install a software package. Understand how to backup files, check releases, check the main, universe, restricted, and multiverse repositories, find release information both through command line and GUI, create source lists, and launch applications through command line.

**Introduction**

This is to copy the source list to a new backup file:

Sudo cp /etc/apt/sources.list /etc/apt/sources.list.backup

This will be used to navigate the directory :

Cd

This will be used to view the contents:

Ls

This will be used to read the lists:

Cat

This will be used to print distribution – specific information with short codename:

Lsb\_release -sc

This will be used to find the command to quickly get to the gui:

Man -k

This will be used for a shortcut to the gui:

Xdg-open

This will be used for root:

Sudo -s

This will be used to get updates:

Apt-get update

This will be used to get the package signing key and add it to Ubuntu.

We use wget as the non-iteractive network downloader with quiet ( no output) and -O file:

Wget -q -O – <https://dl.google.com/linux/linux_signing_key.pub> | sudo apt-ey add –

This will make a usage listing of google information to /etc/apt/sources.list.d/google.list to make a stable release:

Sudo sh -c ‘echo “deb [arch=amd64] <https://dl.google.com/linux/chrome/deb/> stable main” >> /etc/apt/sources.list.d/google.list

This will be used to install the google-chrome package:

Sudo apt install google-chrome-stable

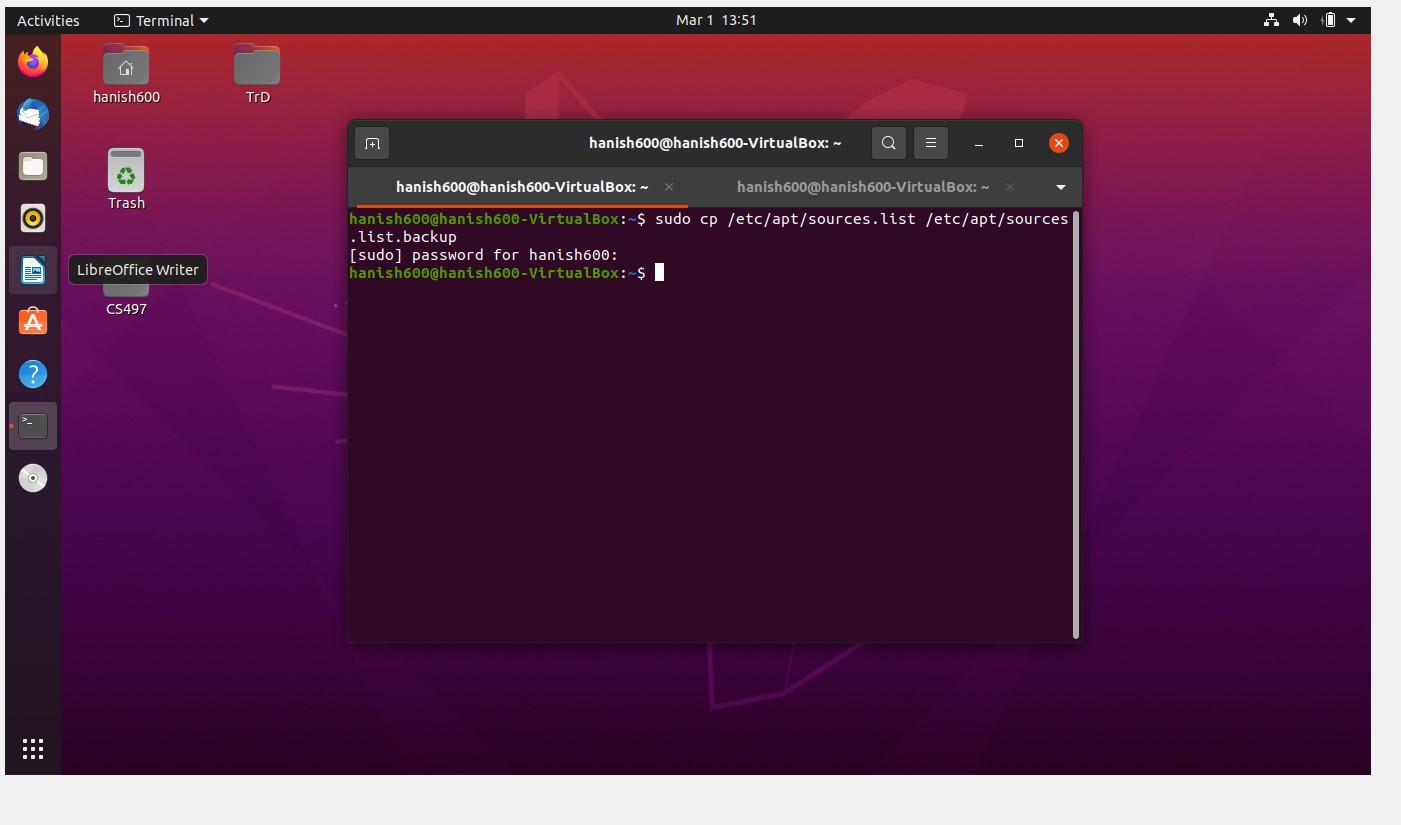
This will be used to run the google chrome in command line:

Google-chrome &

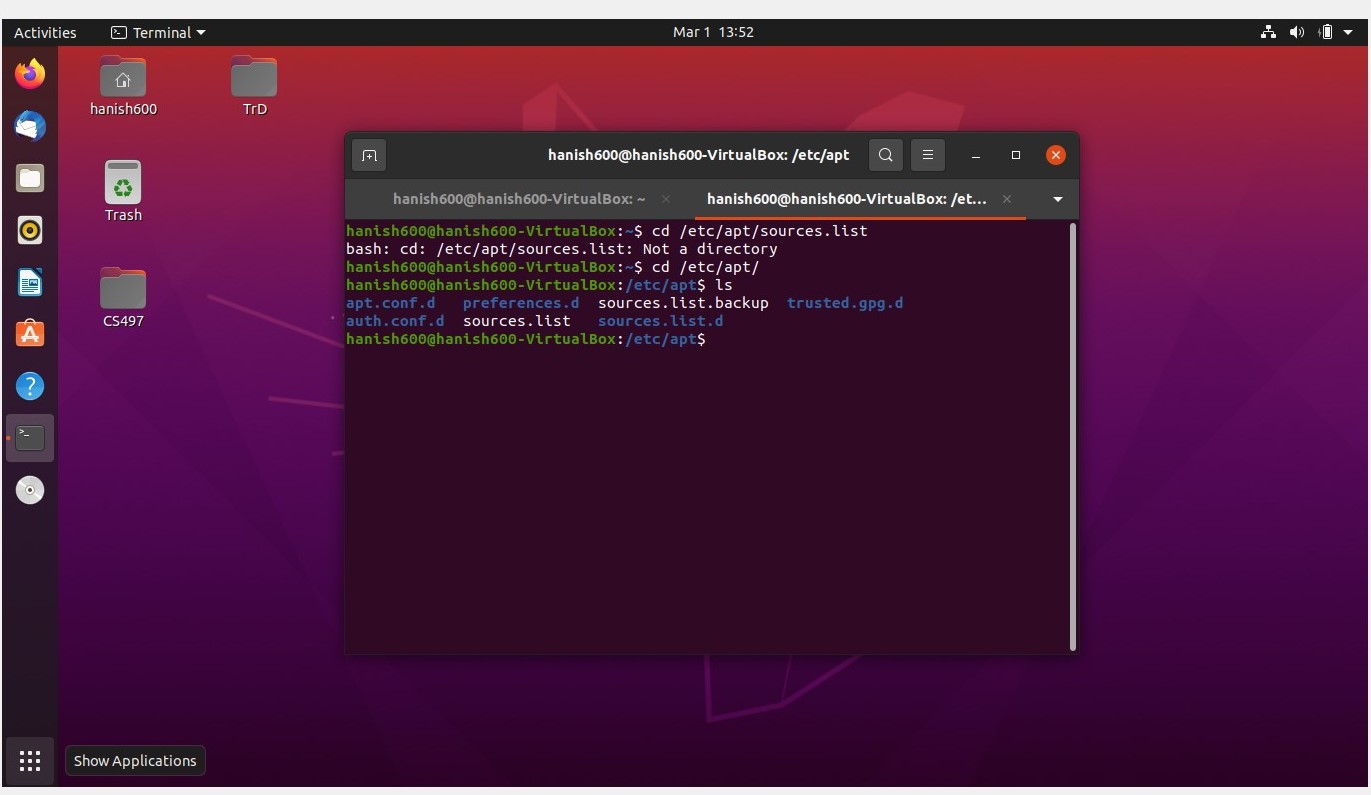
**Summary of Results**

We begin with making a backup of our sources using sudo cp to copy from a source to a destination file.

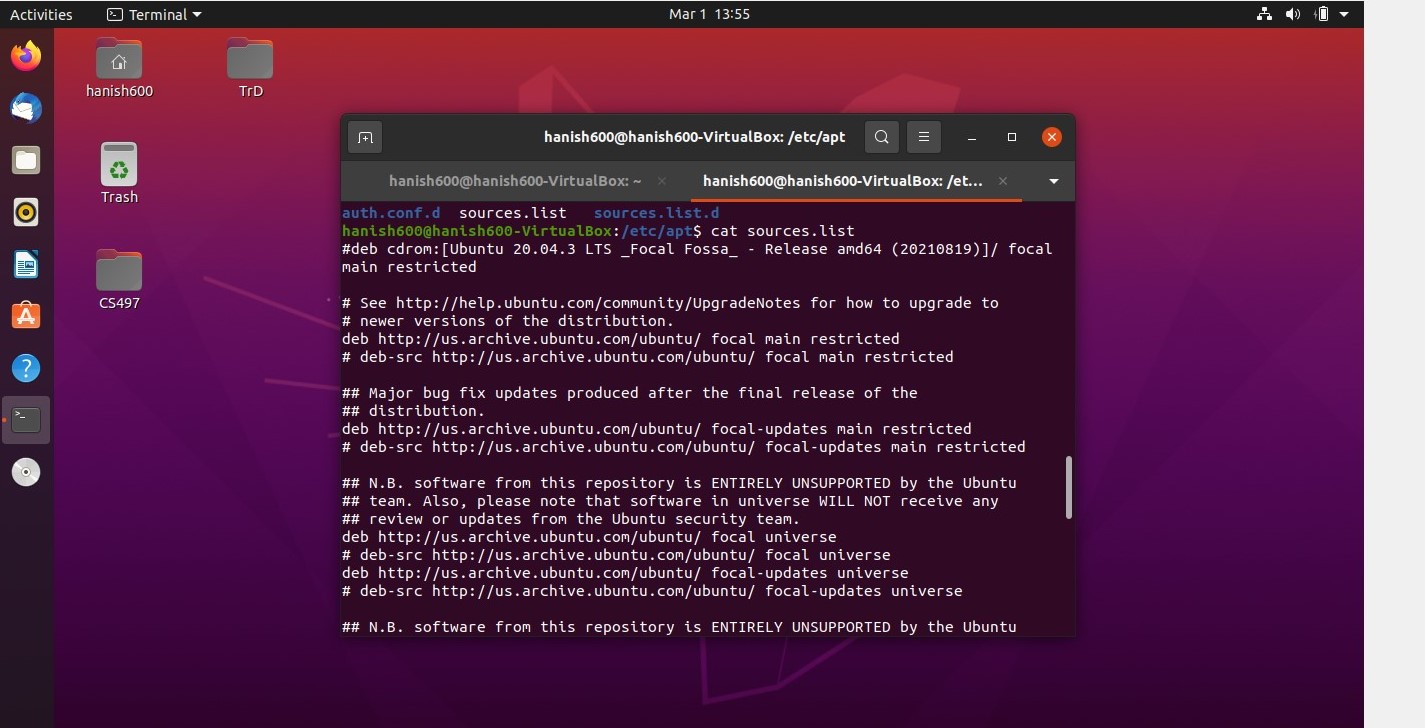
We do a copy sudo cp /etc/apt/sources.list to /etc/apt/sources.list.backup



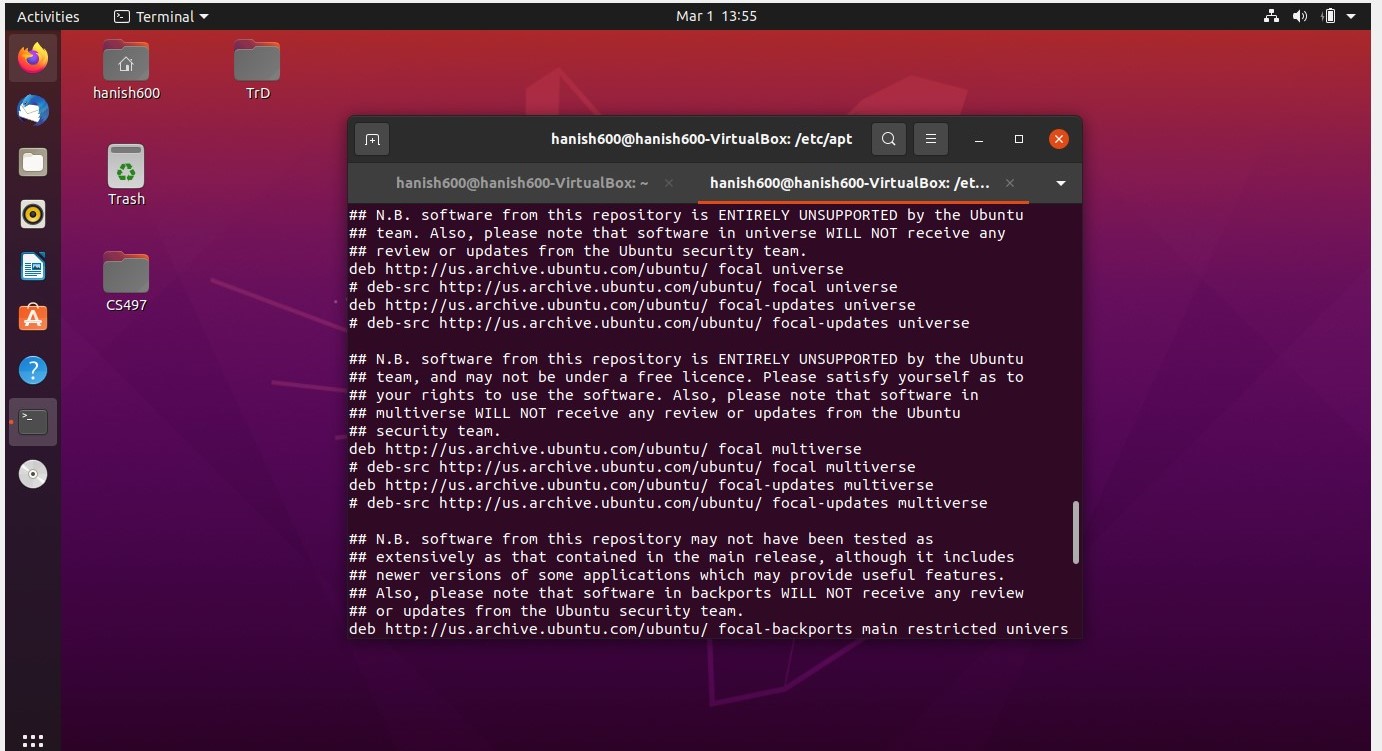
Next, we use cd to navigate the current directory to our newly created file and our old source list. We take a look at the old source list.

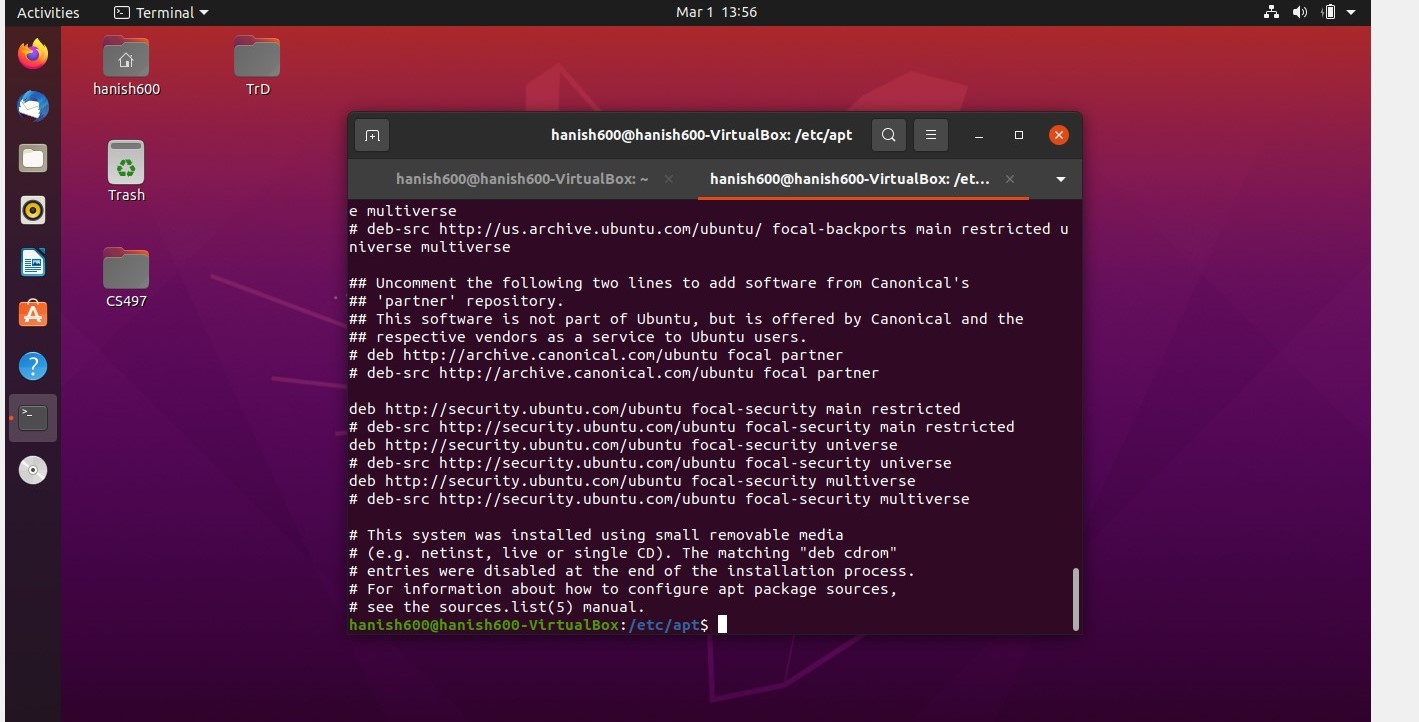


We use ls to list the directories and we see our new backup file along with the original file. We open the original file.



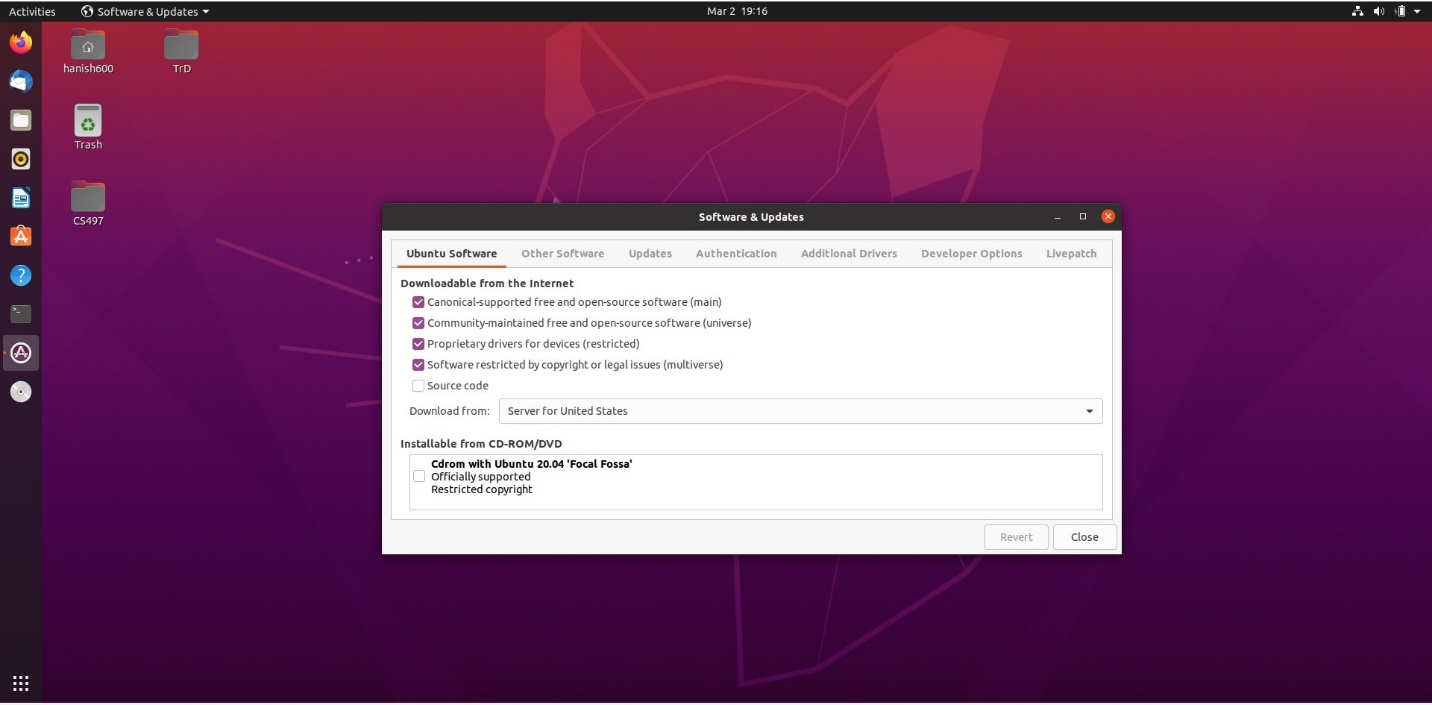
By opening the original source list, we see the main and restricted along with universe and multiverse.



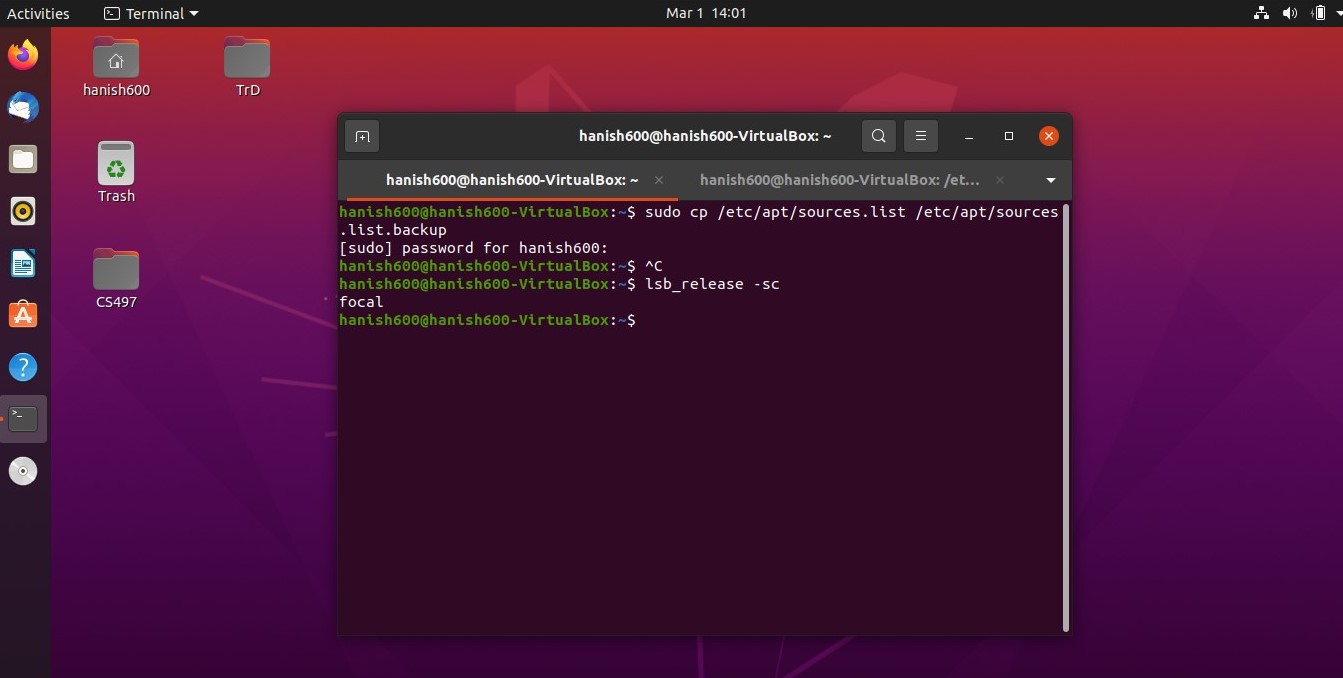


We see we have a ton of information here with some Debian source links.

We can confirm that our repos are enabled by also looking at the gui.

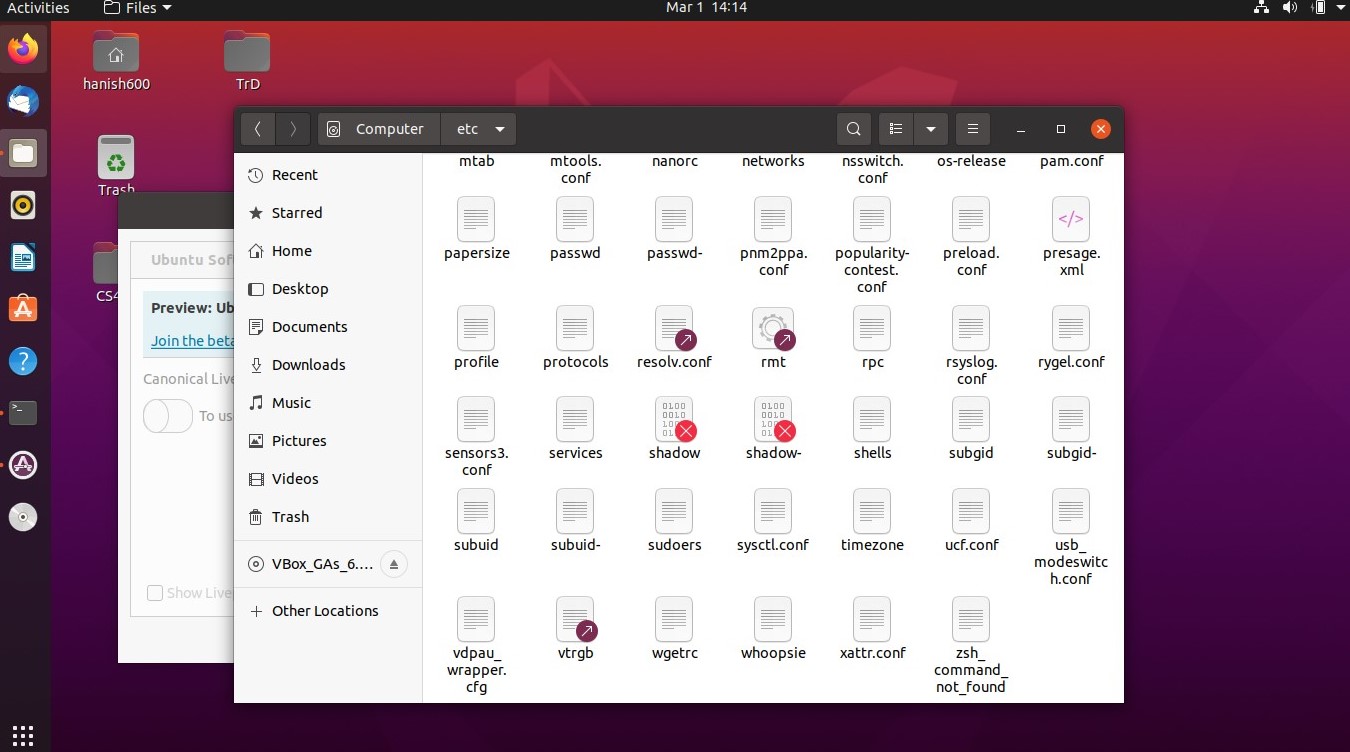


Now we can see the distribution code short name with lsb\_release -sc.



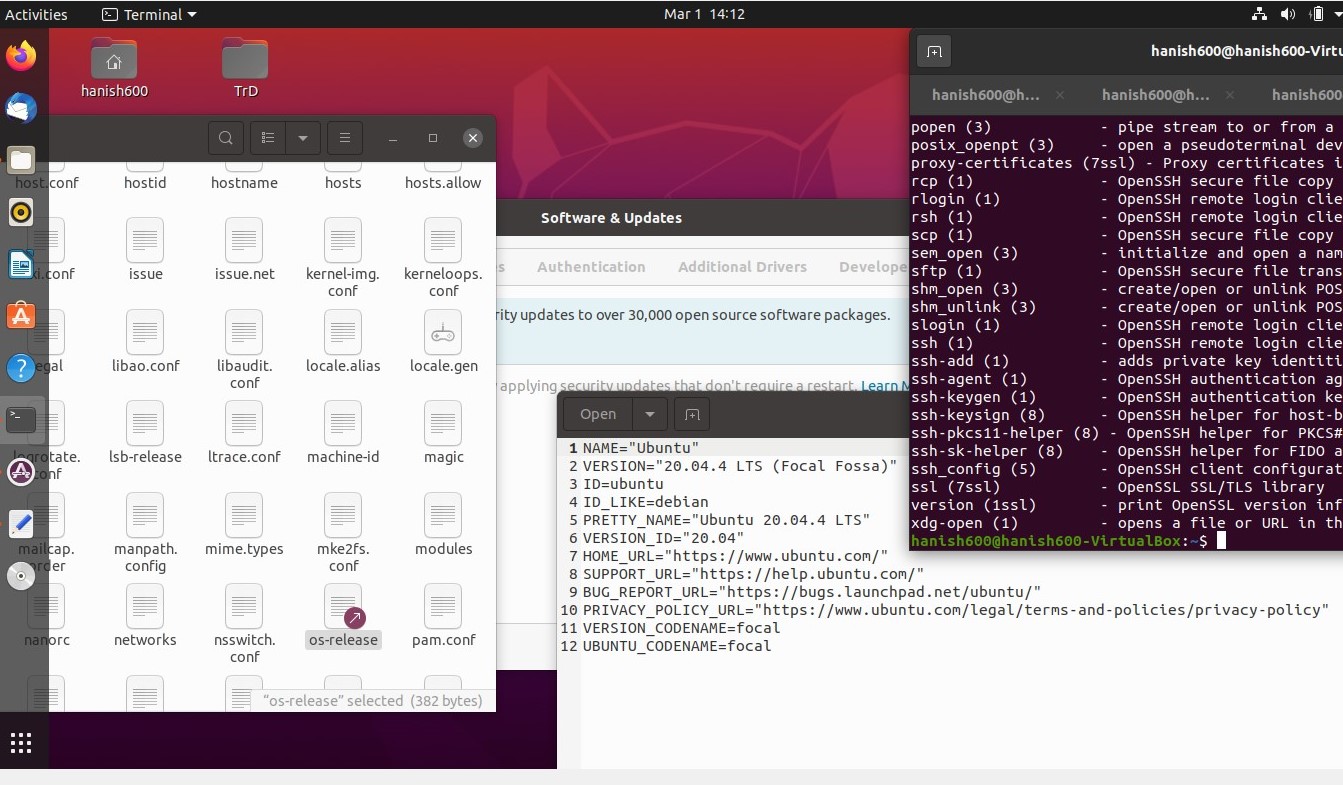
If we want to see this through the gui, we have two different options.

The first option is to go to other locations -> computer -> etc -> os-release.

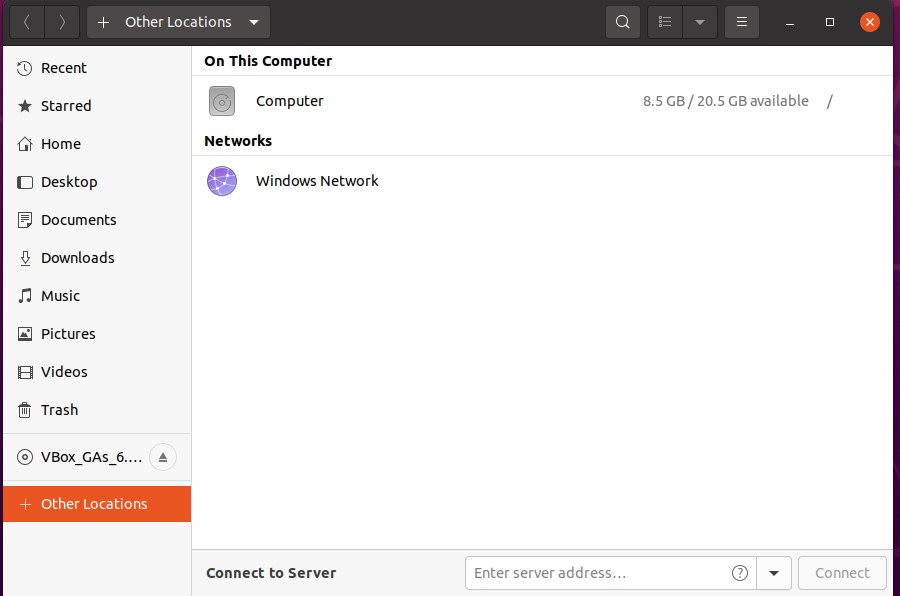
The quicker and second option is to use that man -k to see how to open the directory and use the command to go directly to the etc folder in the gui.

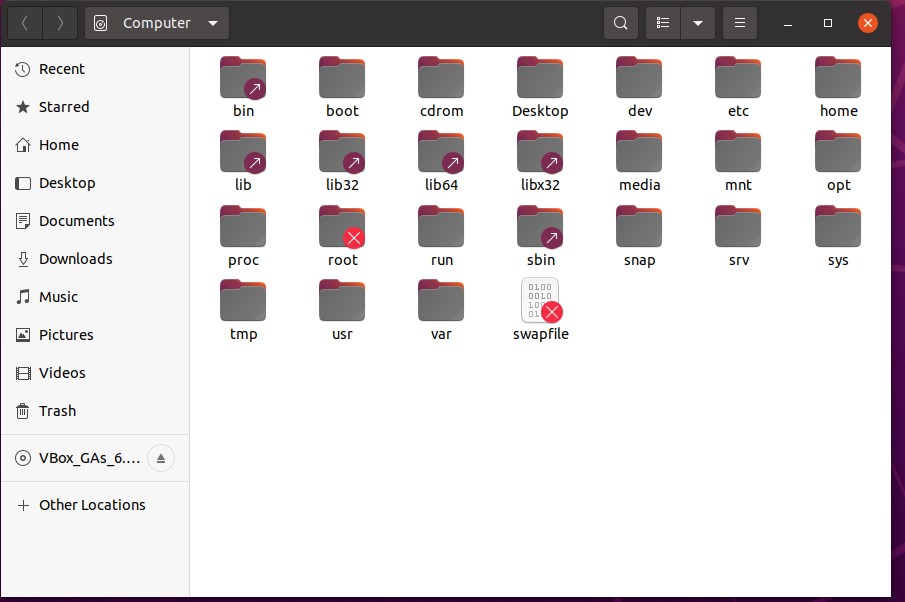
We find the command to quickly get to our gui and we used both ways to see it’s possible to find this file in both the GUI and through both the command line and GUI.

After trying xdg-open [filename] and the GUI we get to this.



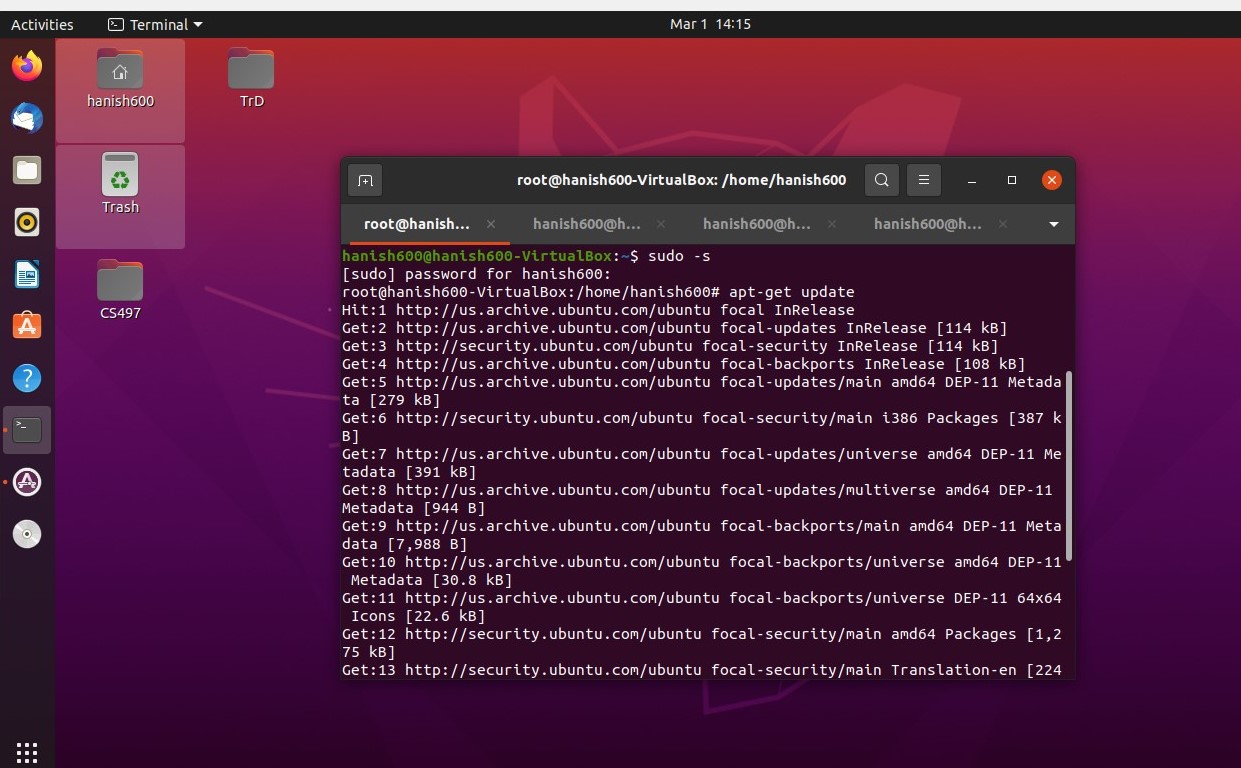
We can also just use the gui by doing other locations we will see “computer” and if we click that we can navigate through the files.





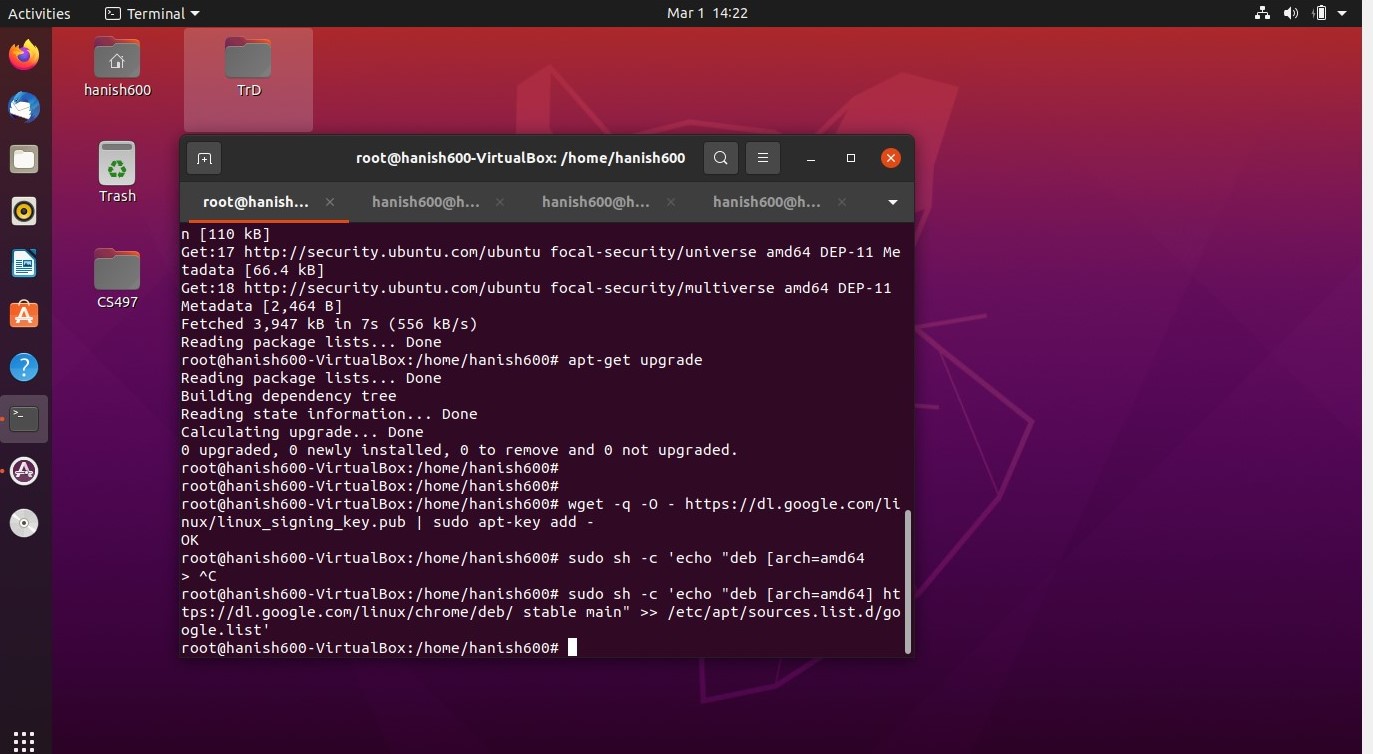
We get to the same folder /etc here.

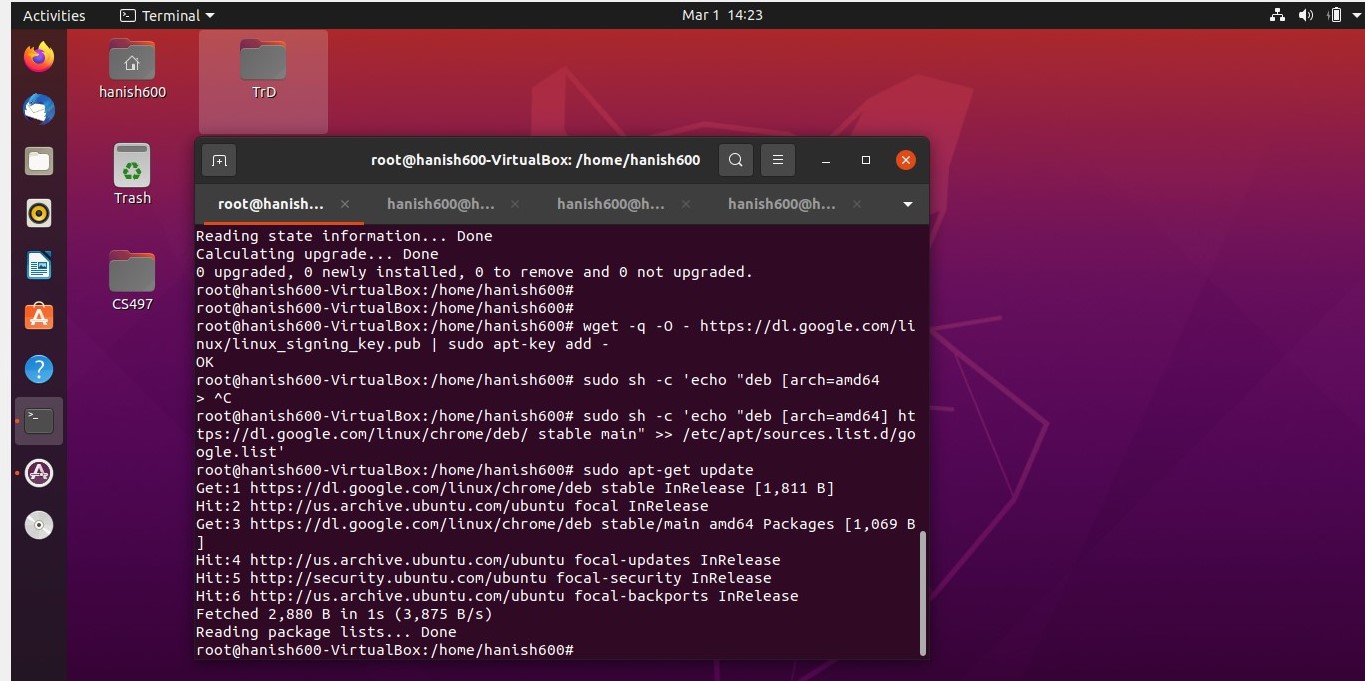
Next we use sudo -s to get root and start to get updates with apt-get update.



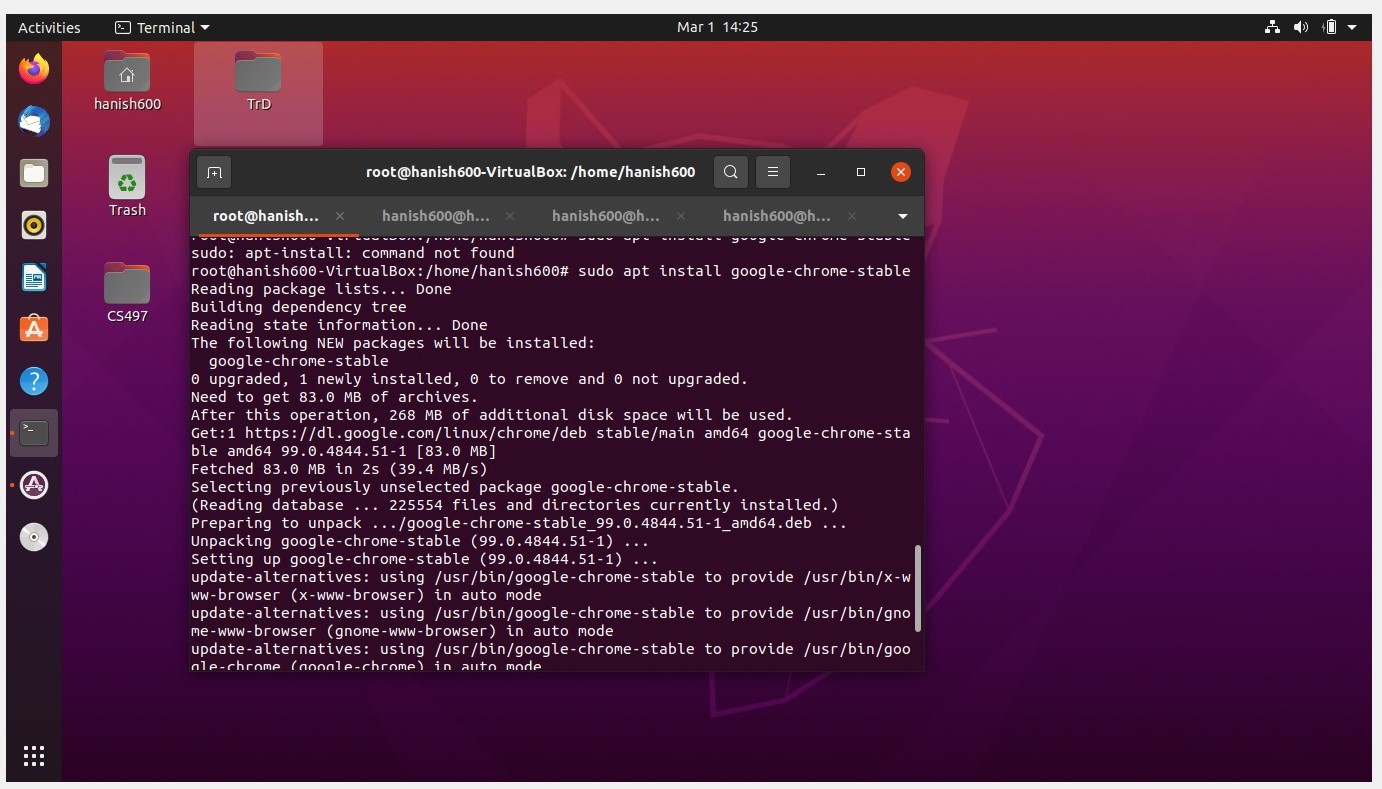
We can then add a package signing key by using the non-iteractive network downloader wget with -q quiet (no output) and -o for file.

We also use the sudo sh -c ‘echo “deb [arch=amd64] <https://dl.google.com/linux/chrome/deb/> stable main” >> /etc/apt/sources.list.d/google.list’ to make a usage listing with google’s information and stable version to /etc/apt/sources.list.d/google.list

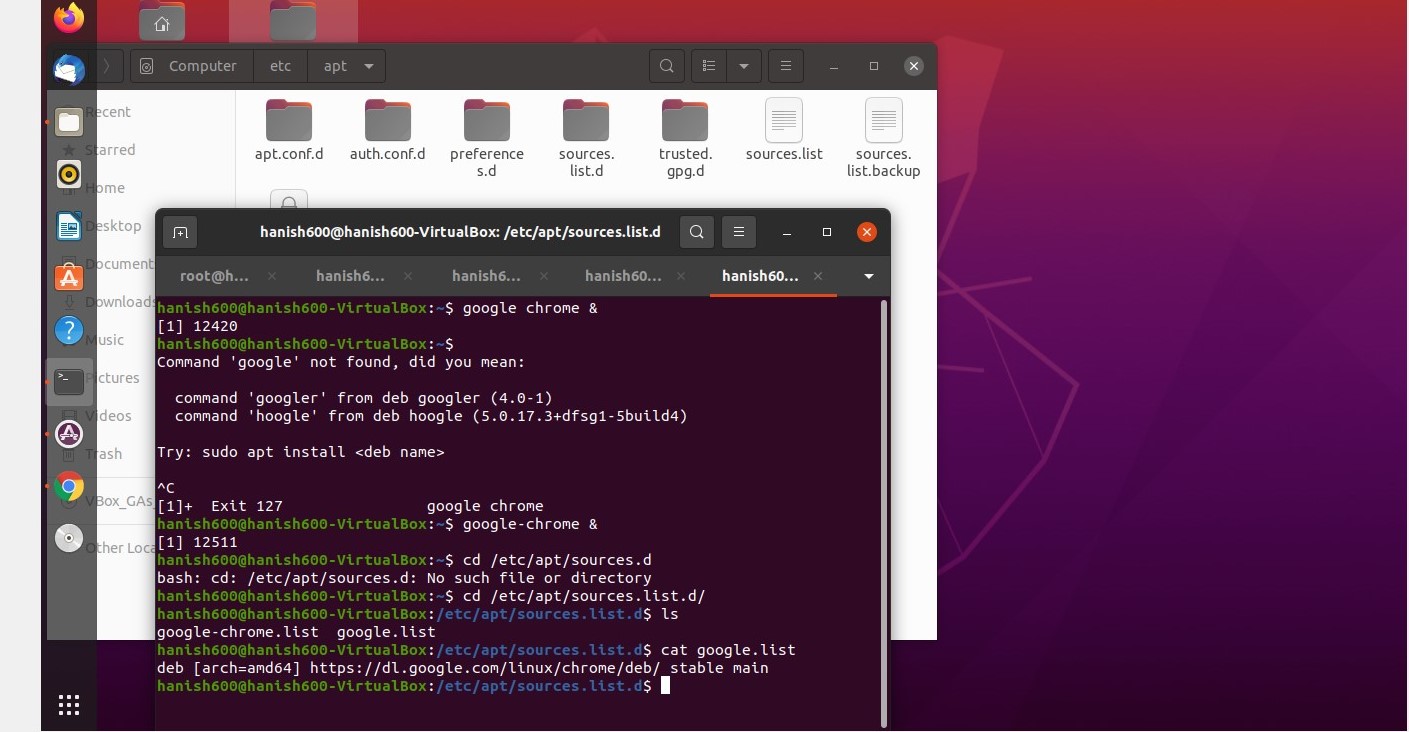


Now that we have the package, we can run apt-get update to automatically update the local cache. 

We can now use sudo apt install google-chrome-stable to get google chrome.

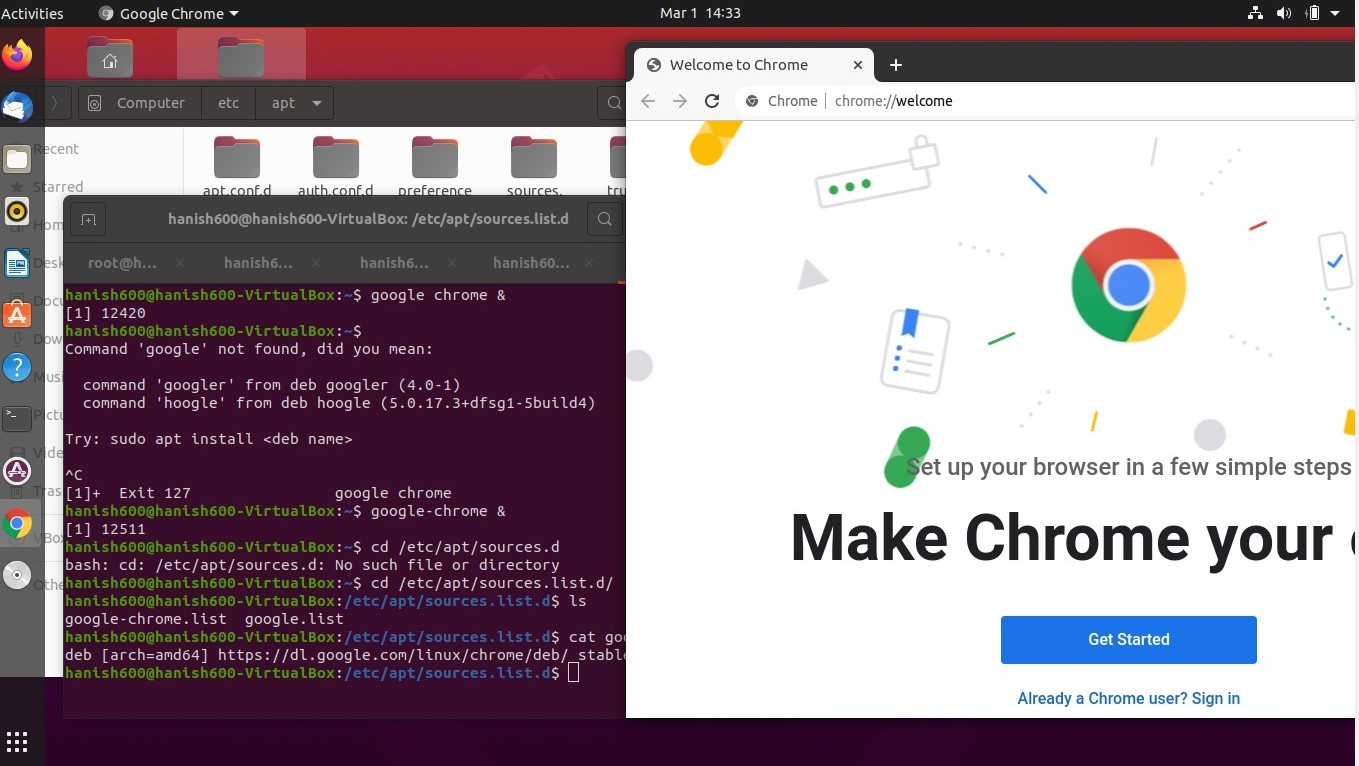


We can run google chrome with google-chrome & and we can navigate to read the google.list file with cat.



We find the usage information when we previously used the sudo sh -c command.

As we can see google-chrome is open.



**Conclusion**

We observed the Linux package management is a useful tool. We are able to add source lists and keys, get updates, download files, and use those packages. We learned how to find the version release through our GUI and the command line. We learned we can launch an application through the command line with the &. We also learned that the google list will contain the package information and the link to the package. We also learned the source contains a lot of information about our packages and repos. It also describes how we installed our Ubuntu through “a small removable media.”

In apt-get we have commands for update,upgrade,dist-upgrade,apt-get clean, and more including ones found in YUM. The distros using apt-get can include Ubuntu, Debian, Linux Mint, etc. In apt-get you must use apt-get update to get the new packages. Yum automatically refreshes the lists of packages. YUM has the commands for install,remove,update,list,search,info, and many more including ones found in apt-get. YUM stands for yellowdog updater, modified and APT stands for Advanced Packaging Tool. Yum is used in distributions such as RHEL and CentOS. It is a high-level abstraction. The newer version of Yum is DNF. Most of these package management tools work the same way. Both support tar packages. Tar is universally compiled. There The common package in RedHat is rpm while in Debian is deb. Rpm has pre-compiled files. is even an aptitude software for deb packages which works in the same way but offers better functionality than apt-get.