We use Ubuntu, Debian, CentOS and different operating systems. If anyone asks, “what operating system do you use?” Most of you might say, “I use Linux”. They are not really Linux. Linux is just the name of the kernel. These are actually different Linux distributions.

Now you may ask, “What is a Linux distribution?”

Well, Linux kernel itself can’t do very fancy things. It’s a software that manages the hardware, allocates memory to programs, helps you run the program, and other basic very low level task for you. Let’s say, you want to edit a file with **nano** text editor. Well, the Linux kernel don’t have that. You have to install it on top of the Linux kernel separately in order to use it.

Without useful programs, Linux kernel is of no help to ordinary users. Again, installing programs on top of Linux kernel is not something ordinary people would love to do. So different companies and individuals packaged important tools (or softwares) on top of Linux kernel and packaged it for you. So when you install it, you can start working with the program you need. This is called a Linux based operating system or a Linux distribution. Ubuntu, Debian, CentOS, Fedora and others are Linux distributions or Linux based operating systems. They are not only Linux.

Now, there are so many softwares on Linux that you can’t even count. Including them all in a single operating system package would make the size of the operating system unnecessary large and harder to distribute. So the operating system needs a mechanism to install packages easily as needed. That way, they could include very common utilities and make the installation smaller. It is easier develop, distribute, and download for users and a more modular approach.

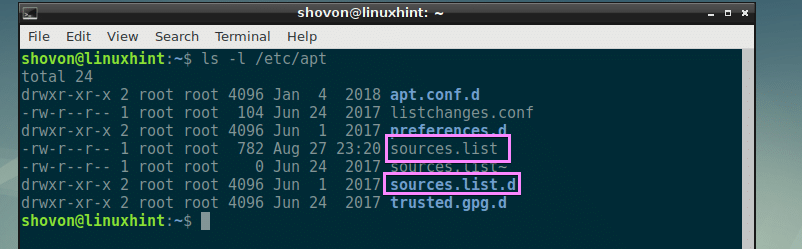
Then the extra packages are hosted on a webserver or FTP server of the Linux distribution from where users can download and install it. These webservers or FTP servers are called package repository.

You also need a way to manage (install, remove, download) these packages from the package repository. So a package manager is included on your favorite Linux distribution.  Ubuntu is based on Debian GNU/Linux distribution. Ubuntu uses the APT (Advanced Package Tool) package manager to manage packages. The APT package manager and all the graphical front ends (Ubuntu Software Center, Muon, aptitude etc) uses the **sources.list** file to learn about which package repository or repositories to use.

In this article, I will show you how the **sources.list** file is used on Ubuntu. Let’s get started.

**Configuration Files:**

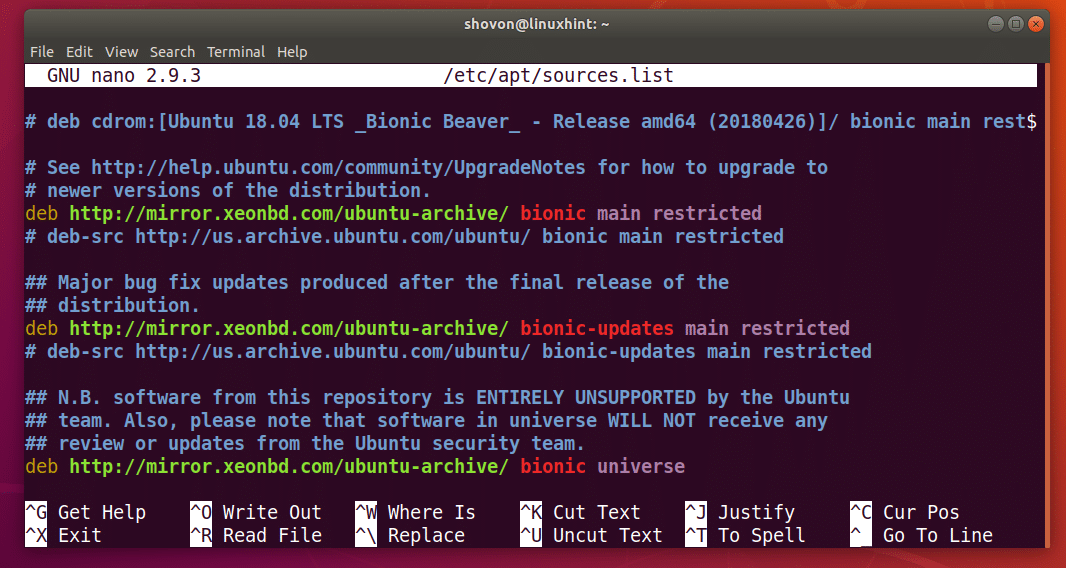
The APT package manager and all its graphical frontends gets the package repository information from **/etc/apt/sources.list** file and files from the **/etc/apt/sources.list.d** directory.



In Ubuntu, different package managers edit **/etc/apt/sources.list** file directly. I do not recommend you add custom package repositories there. If you need to add any additional package repository, it’s best to just add these in the **/etc/apt/sources.list.d/**directory. I will show you practically how it’s done later in this article.

**Understanding sources.list File:**

The contents of the **/etc/apt/sources.list** file looks something like this.



Here, lines starting with hash (#) are comments. Comments are used for documentation purposes on this file. Comments are also used to disable a certain package repository here.

For example, you can leave a comment when you add a custom package repository like this.

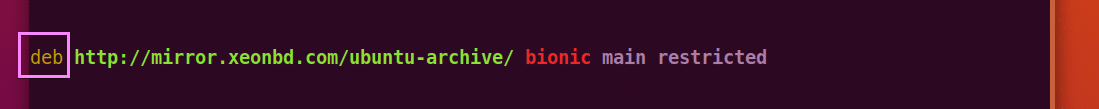
*# This is my local NodeJS v8.x package repository*  
deb http://192.168.10.1/nodejs/8.x stretch main

I am going to call each line used to add a package repository (lines starting with **deb**) on **/etc/apt/sources.list** file and files in the **/etc/apt/sources.list.d/** directory an APT line. You can call it whatever you want.

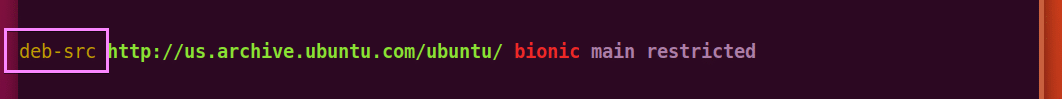
Now let’s talk about how an APT line is formatted. This is an example of an APT line as shown in the screenshot below.



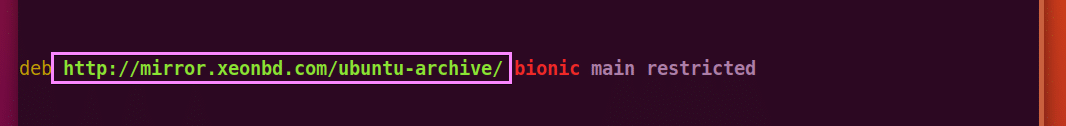
An APT line starts with **deb**, which means this package repository distributes software packages as pre compiled binaries in deb file format.



An APT line may also start with **deb-src**, which means the package repository distributes software packages as source codes, which you will have to compile in your own computer in order to use. By default, all the **deb-src** package repositories are disabled on Ubuntu. I personally prefer them to be disabled as I don’t use them. Installing packages from sources takes a long time depending on the specification of your computer.



Then you have the package repository’s HTTP, HTTPS, or FTP URL. This is where all the package files and package database files are kept. The package manager downloads the package metadata and other information to learn about what packages are available and where to download them.

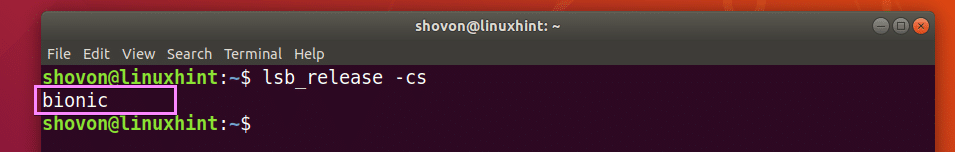


Then you have to type in the short codename of your Ubuntu operating system. It is different for each version of Ubuntu. For example, in Ubuntu 18.04 LTS, it is **bionic**.

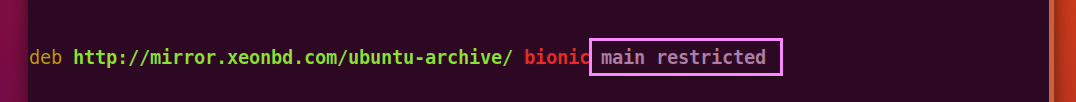


You can find out what it is for your distribution with the following command:

*$*lsb\_release -cs



Then you place a space separated list of different sections of that package repository. The packages of a package repository may be logically divided into many groups as you can see in the marked section of this article below. The Ubuntu package repository is divided into **main**, **restricted**, **universe** and **multiverse** sections. In this example, I added only **main** and **restricted** sections of the **bionic** package repository.



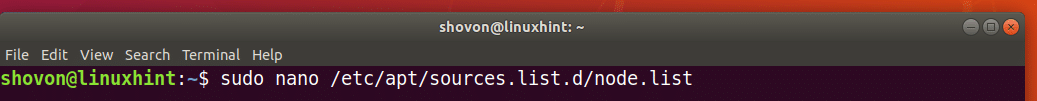
That’s basically all you need to know about **sources.list** file on Ubuntu.

**Adding Your Own Package Repository on Ubuntu:**

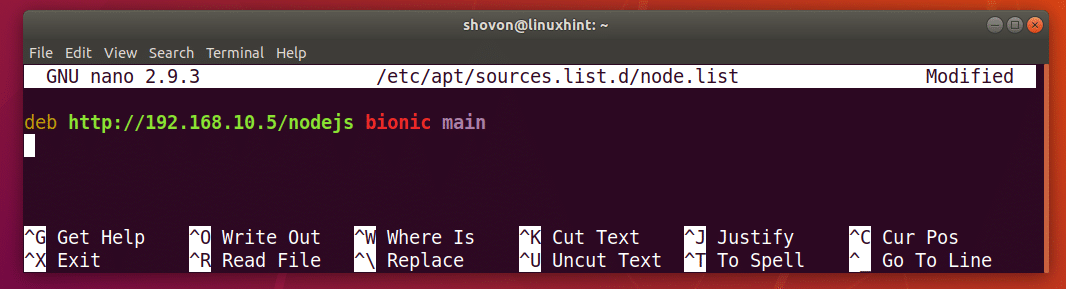
Let’s say, you want to add your own package repository on Ubuntu. Let’s say, it is hosted on your local network and available at **http://192.168.10.5/nodejs** and it’s a mirror of the NodeJS package repository.

First, create a new file **node.list** in the **/etc/apt/sources.list.d/** directory with the following command:

*$*sudo nano /etc/apt/sources.list.d/node.list

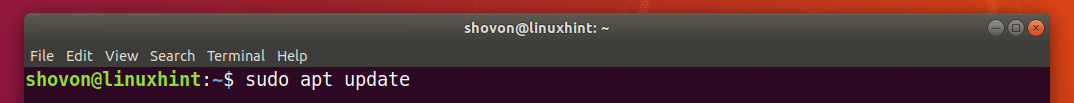


Now add the following line and save the file by pressing **<Ctrl>** + **x** and then press **y** and then press **<Enter>**.



Now for the changes to take effect, update the APT package repository cache with the following command:

*$*sudo apt update



Now you can install packages from the package repository you added. Thanks for reading this article.