**Ex 3 COMPILING FROM THE SOURCE**

**Date: 29.08.20**

**Aim:**

To study and implement the compiling from the source.

**Description:**

**tar Command :**

The Linux ‘tar’ stands for tape archive, is used to create Archive and extract the Archive files. tar command in Linux is one of the important commands which provides archiving functionality in Linux. We can use Linux tar command to create compressed or uncompressed Archive files and also maintain and modify them.

**Syntax :**

tar [options] [archive-file] [file or directory to be archived]

Options:

**-c :** Creates Archive

**-x :** Extract the archive

**-f :** creates archive with given filename

**-t :** displays or lists files in archive file

**-u :** archives and adds to an existing archive file

**-v :** Displays Verbose Information

**-A :** Concatenates the archive files

**-z :** zip, tells tar command that create tar file using gzip

**-j :** filter archive tar file using tbzip

**-W :** Verify a archive file

**-r :** update or add file or directory in already existed .tar file

# zip Command :

ZIP is a compression and file packaging utility for Unix. Each file is stored in a single .zip {.zip-filename} file with the extension .zip.

* zip is used to compress the files to reduce file size and also used as file package utility. zip is available in many operating systems like unix, linux, windows etc.
* If you have a limited bandwidth between two servers and want to transfer the files faster, then zip the files and transfer.
* The zip program puts one or more compressed files into a single zip archive, along with information about the files (name, path, date, time of last modification, protection, and check information to verify file integrity). An entire directory structure can be packed into a zip archive with a single command.
* Compression ratios of 2:1 to 3:1 are common for text files. zip has one compression method (deflation) and can also store files without compression. zip automatically chooses the better of the two for each file to be compressed.  
   The program is useful for packaging a set of files for distribution; for archiving files; and for saving disk space by temporarily compressing unused files or directories.

**Syntax :**

zip [options] zipfile files\_list

**Syntax for** **Creating a zip file:**

$zip myfile.zip filename.txt

Options :

**-d** : Removes the file from the zip archive

**-u** : Updates the file in the zip archive

**-m** : Deletes the original files after zipping

**-r** : To zip a directory recursively

**-x** : Exclude the files in creating the zip

**-v** : Verbose mode or print diagnostic version info

# gzip Command :

gzip command compresses files. Each single file is compressed into a single file. The compressed file consists of a GNU zip header and deflated data.

If given a file as an argument, gzip compresses the file, adds a “.gz” suffix, and deletes the original file. With no arguments, gzip compresses the standard input and writes the compressed file to standard output.

**Difference between Gzip and zip command in Unix and when to use which command**

* ZIP and GZIP are two very popular methods of compressing files, in order to save space, or to reduce the amount of time needed to transmit the files across the network, or internet.
* In general, GZIP is much better compared to ZIP, in terms of compression, especially when compressing a huge number of files.
* The common practice with GZIP, is to archive all the files into a single tarball before compression. In ZIP files, the individual files are compressed and then added to the archive.
* When you want to pull a single file from a ZIP, it is simply extracted, then decompressed. With GZIP, the whole file needs to be decompressed before you can extract the file you want from the archive.
* When pulling a 1MB file from a 10GB archive, it is quite clear that it would take a lot longer in GZIP, than in ZIP.
* GZIP’s disadvantage in how it operates, is also responsible for GZIP’s advantage. Since the compression algorithm in GZIP compresses one large file instead of multiple smaller ones, it can take advantage of the redundancy in the files to reduce the file size even further.
* If you archive and compress 10 identical files with ZIP and GZIP, the ZIP file would be over 10 times bigger than the resulting GZIP file.

**Syntax :**

gzip [Options] [filenames]

Options:

**-f** : Sometimes a file cannot be compressed

**-k** : By default when you compress a file using the “gzip” command you end up with a new file with the extension

**-L** : This option displays the gzip license

**-r** : This option can compress every file in a folder and its subfolders

**-[1-9]** : It allows to change the compression level

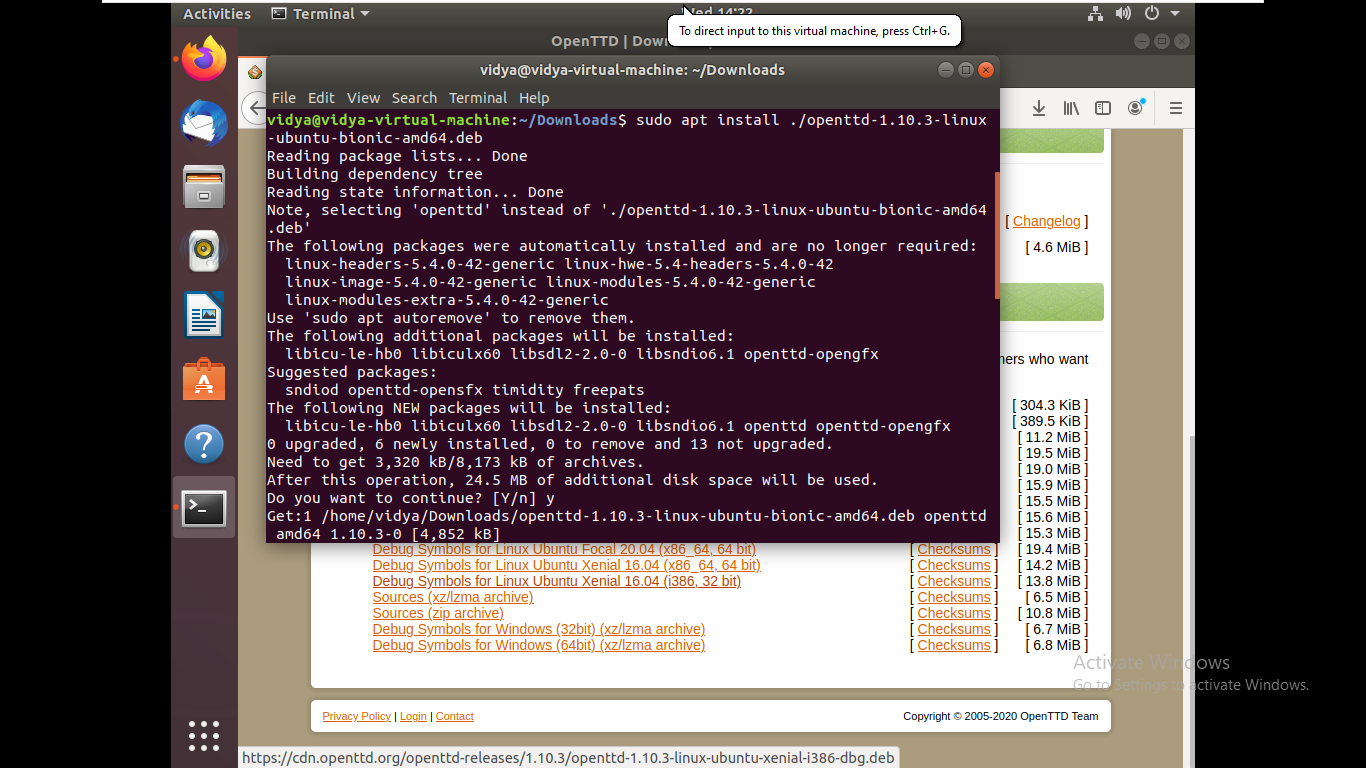
**-v** : his option displays the name and percentage reduction for each file compressed or decompressed

**-d** : This option allows you to decompress a file using the “gzip” command.

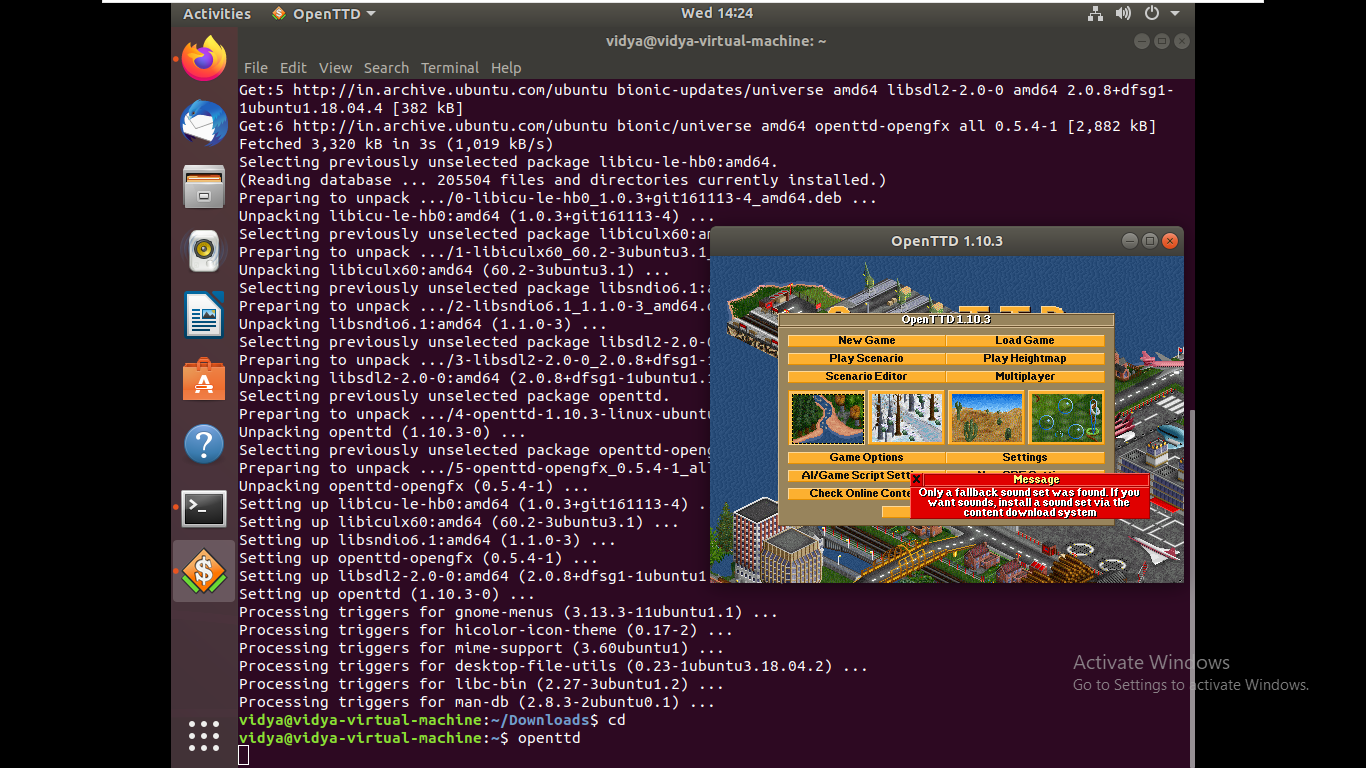
**Exercise**

**1. Compile the source from openttd package**

Installing openttd using dep(Debian) package

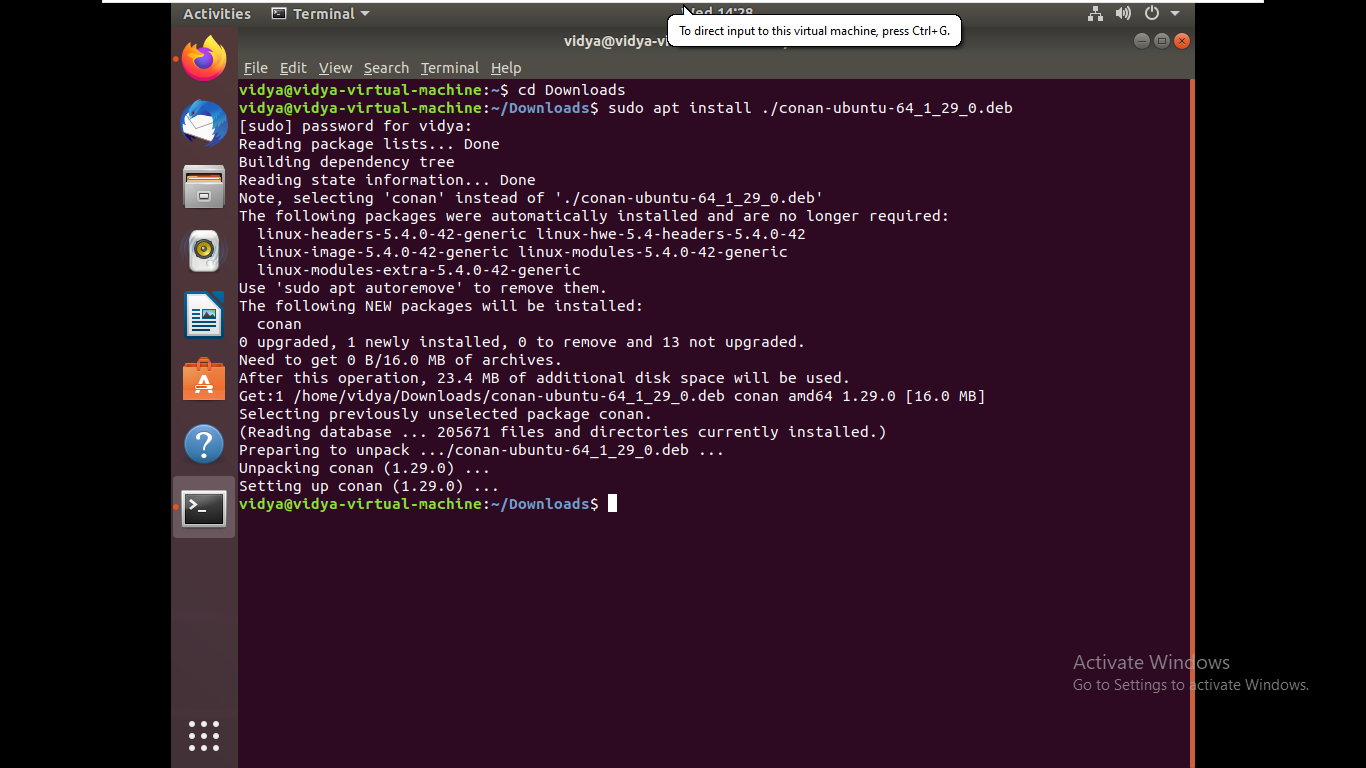


Compiling the source of openttd

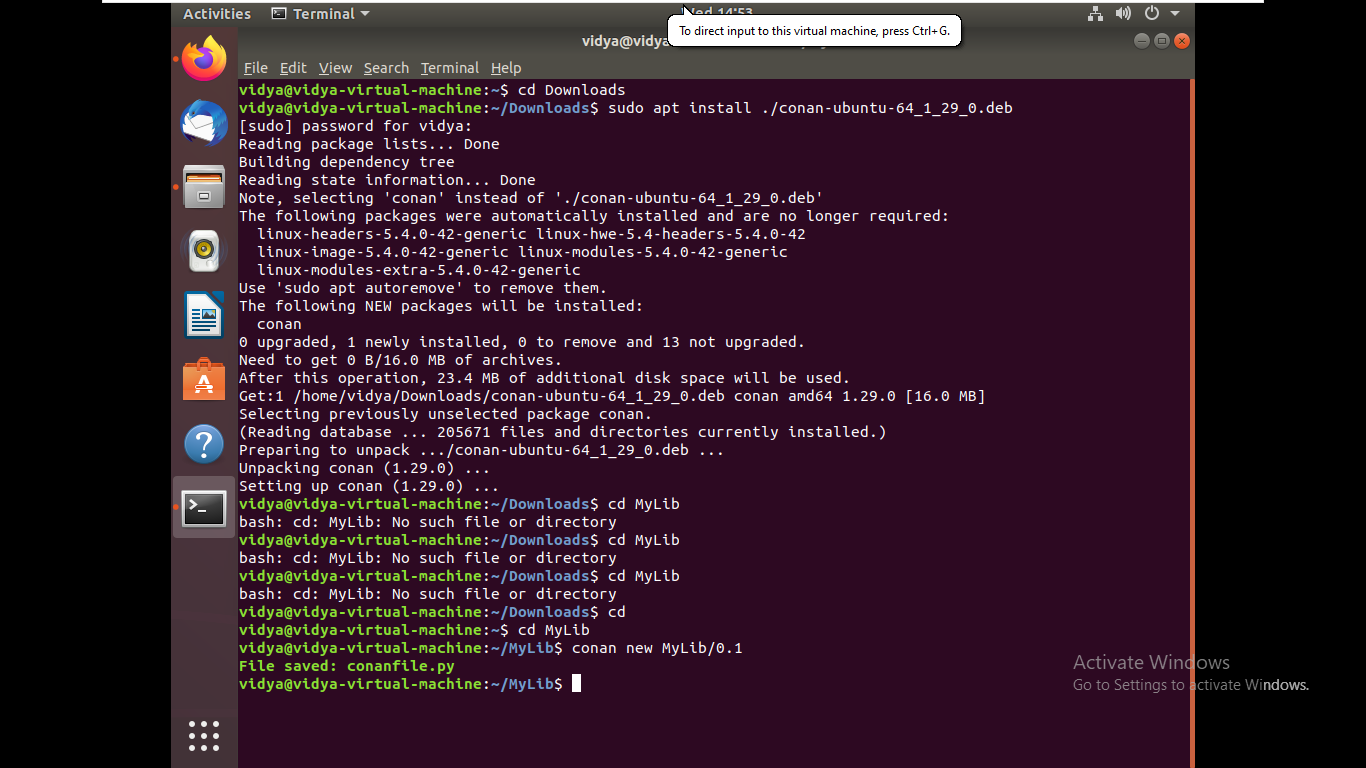


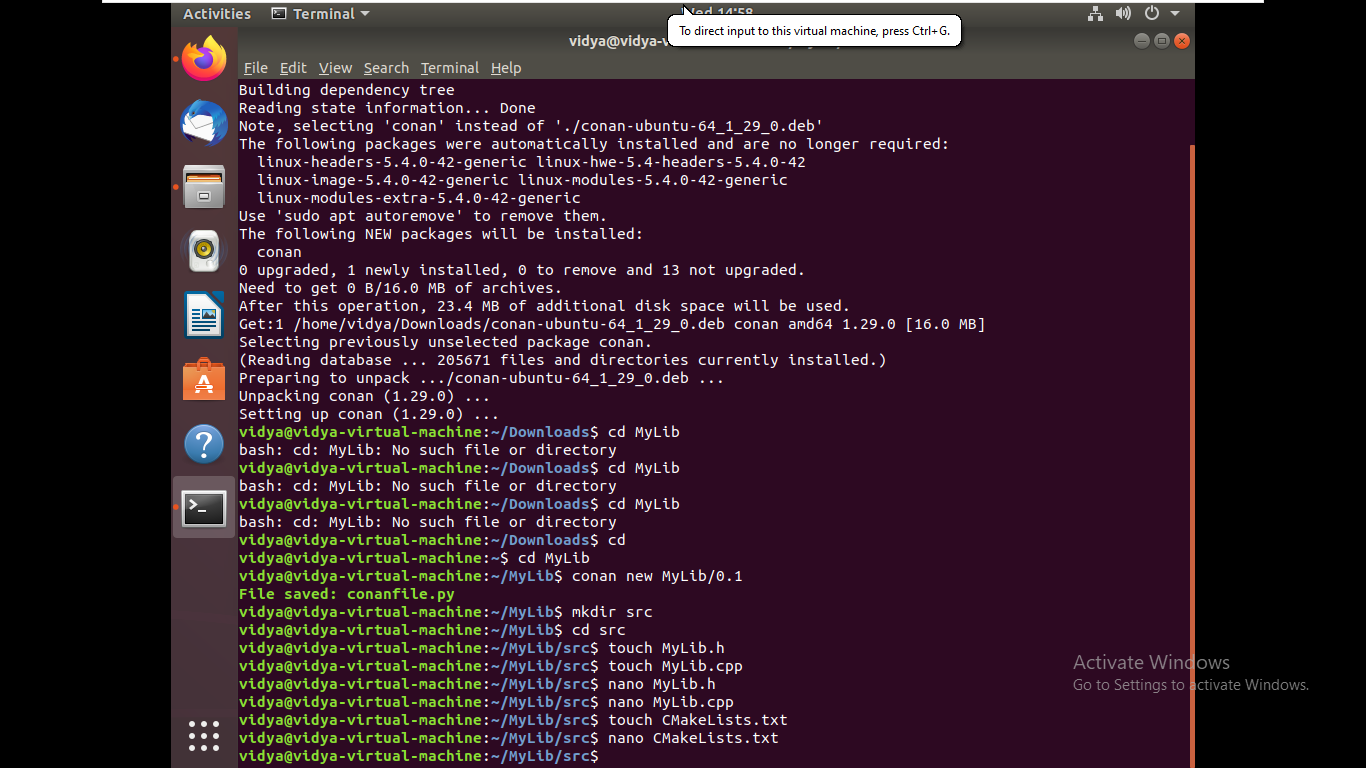
**2. Compile the source from JFrog package**

Installing conan c++ package using dep(Debian) package

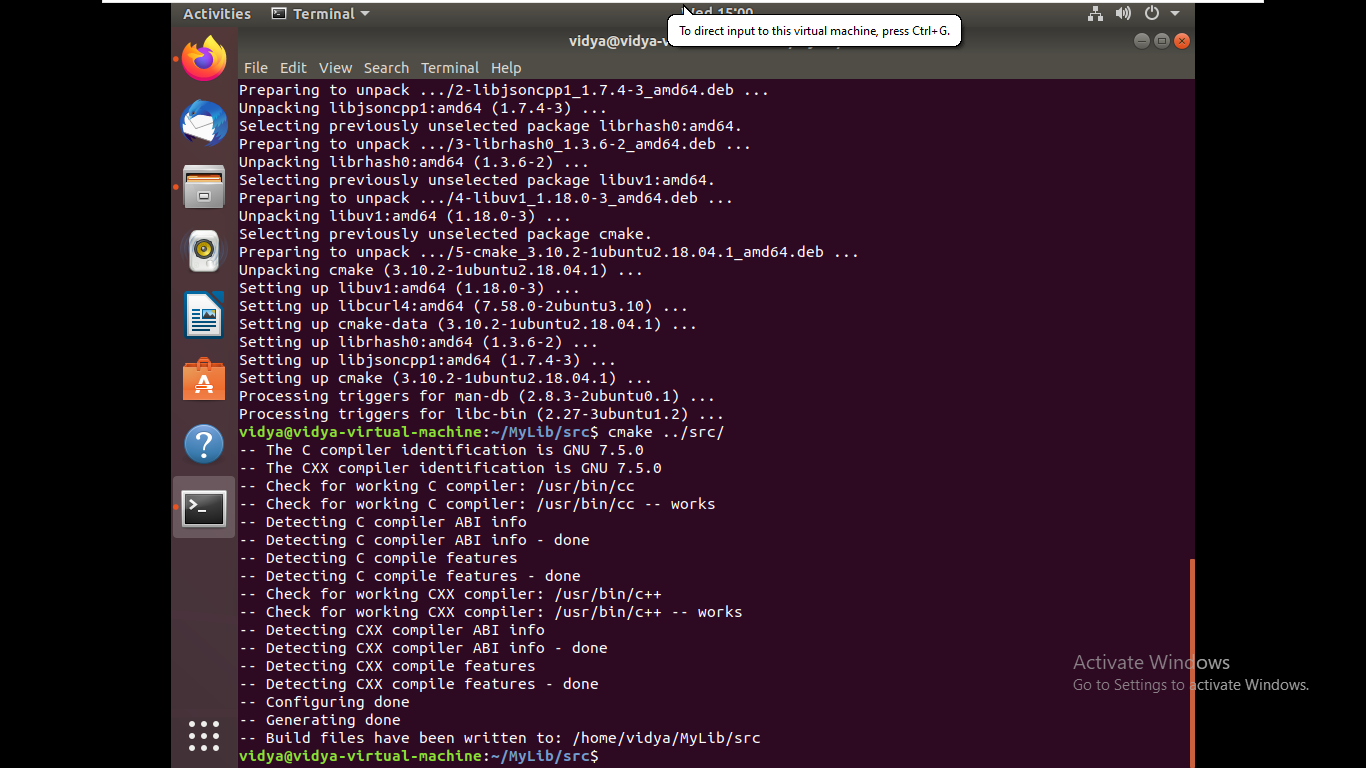
****

creating a sample package

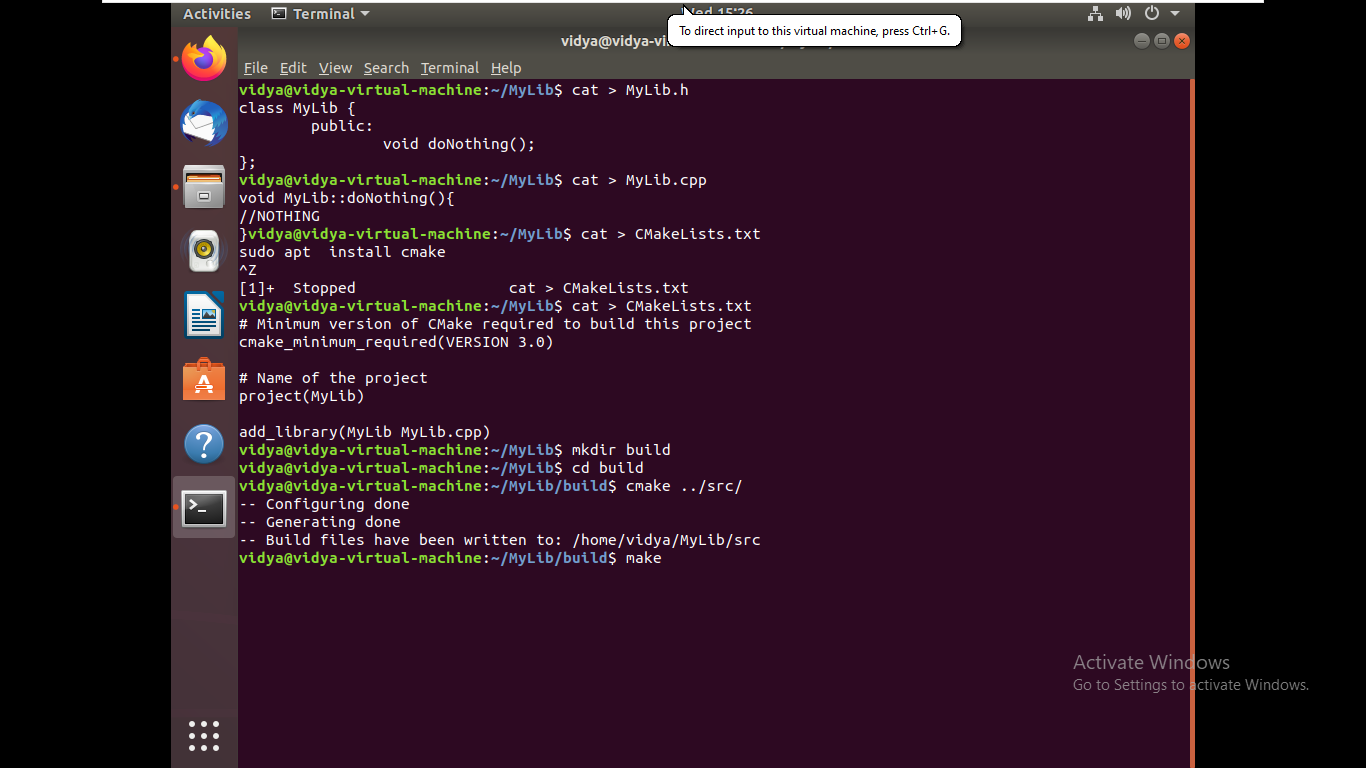
****

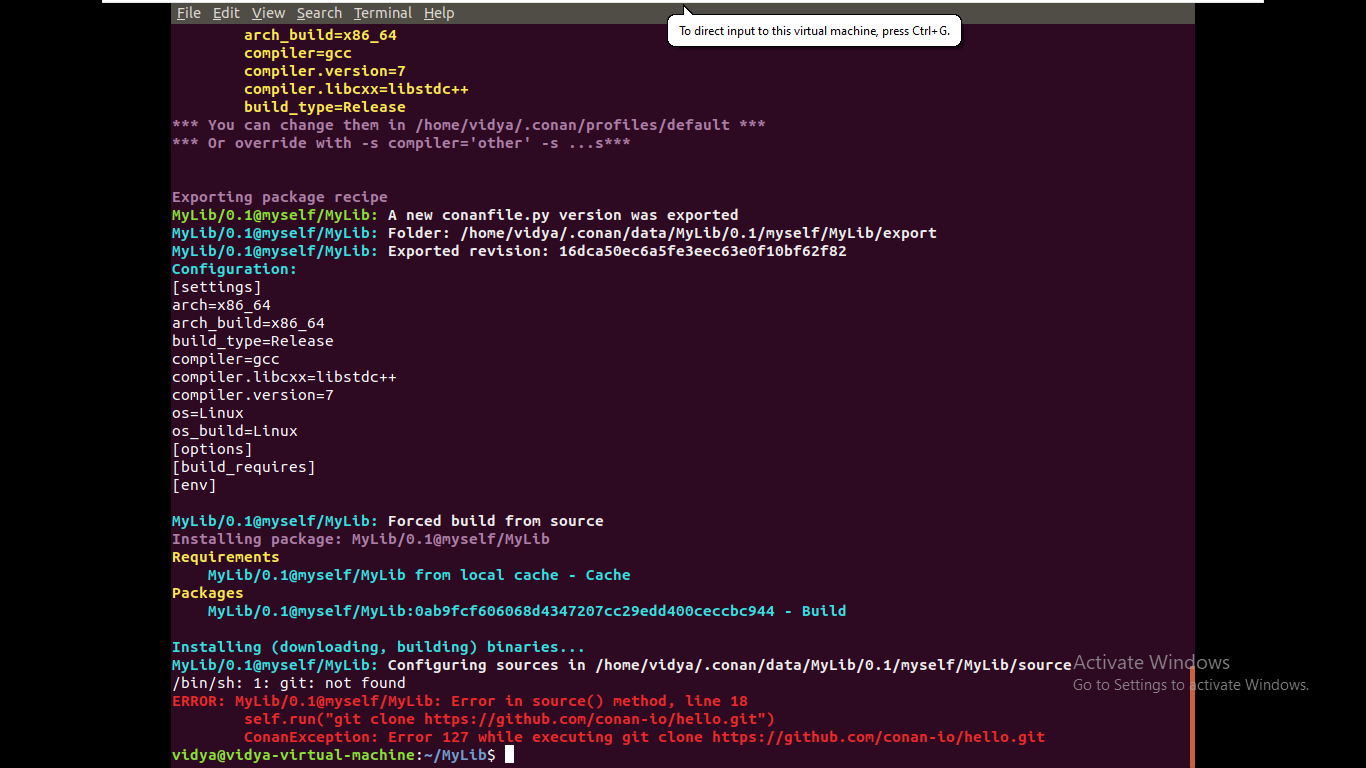
****

Making up the package

****

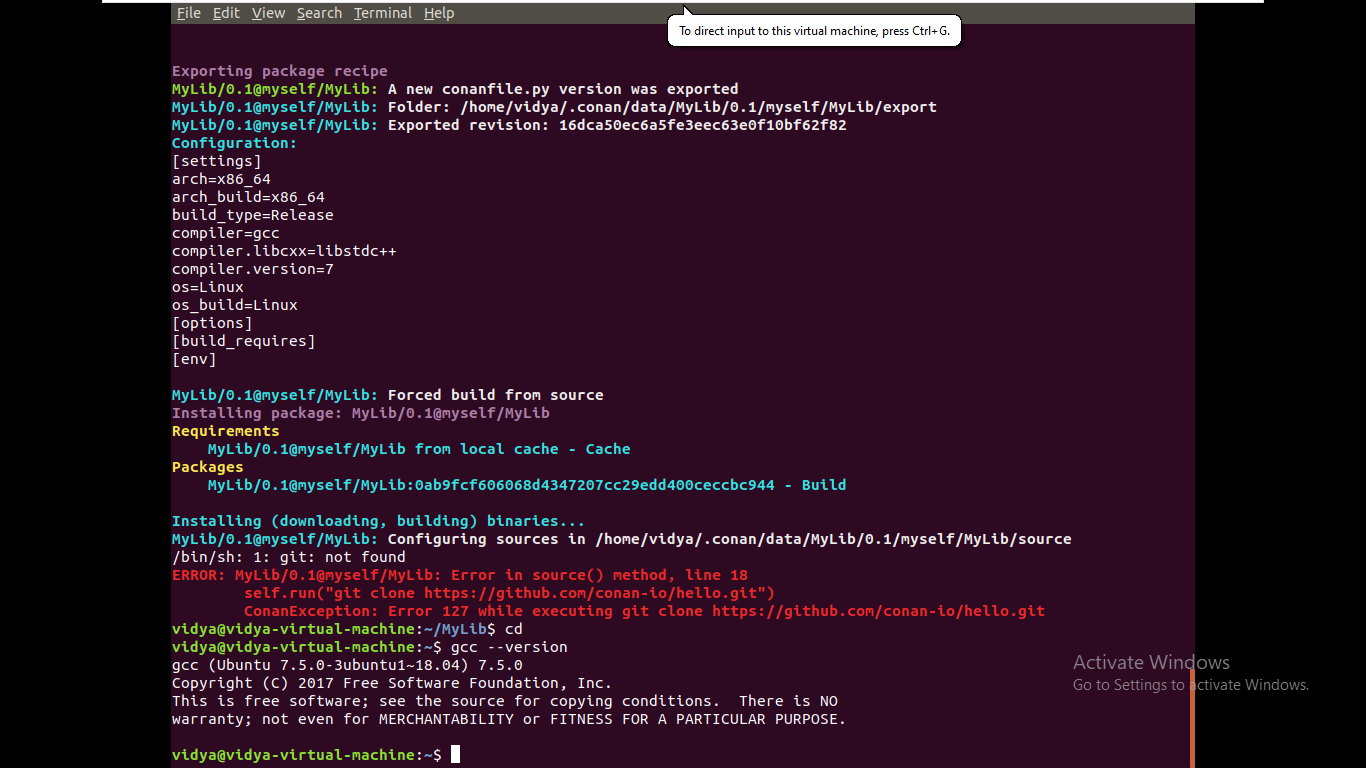
Build package output

****

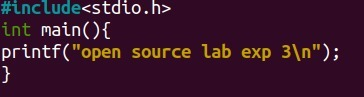
****

**3. Compile the source from gcc**

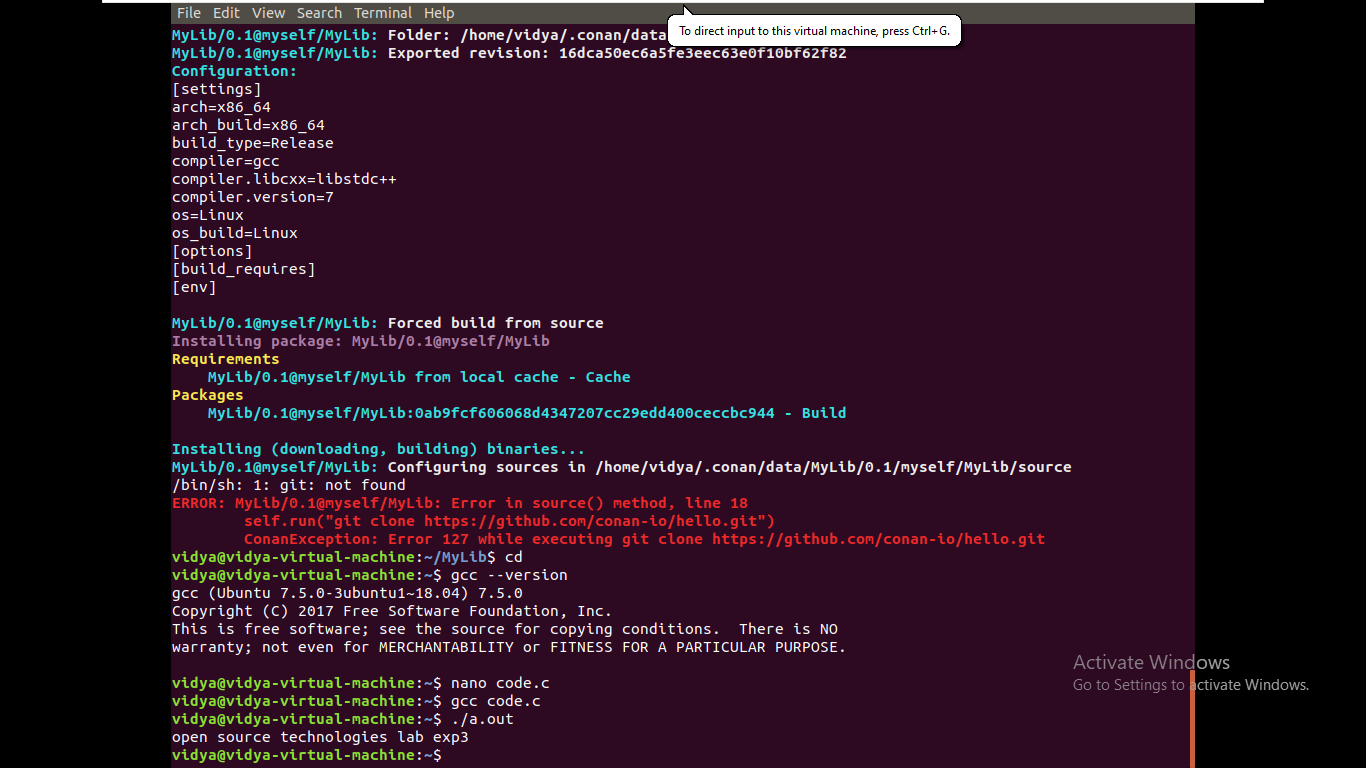
Checking the version of gcc



**Program :**

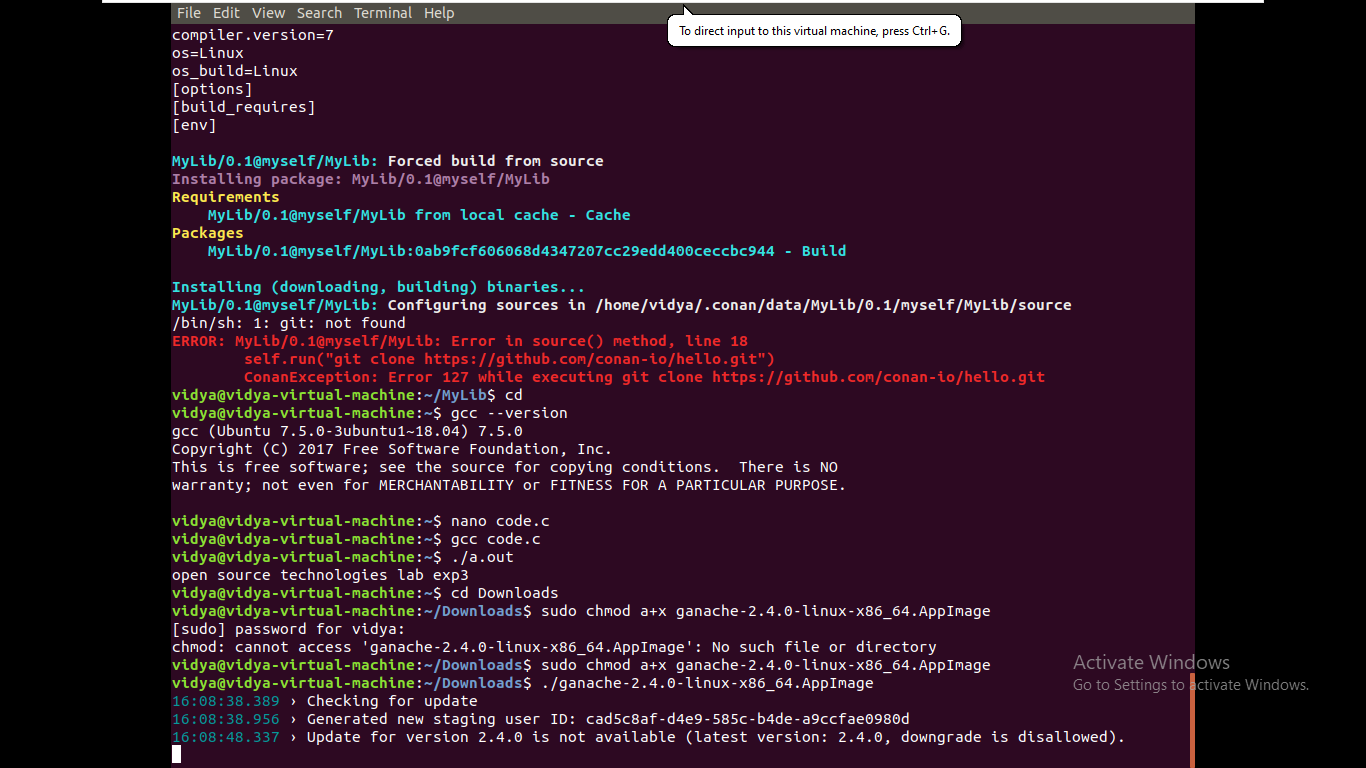


**Output :**

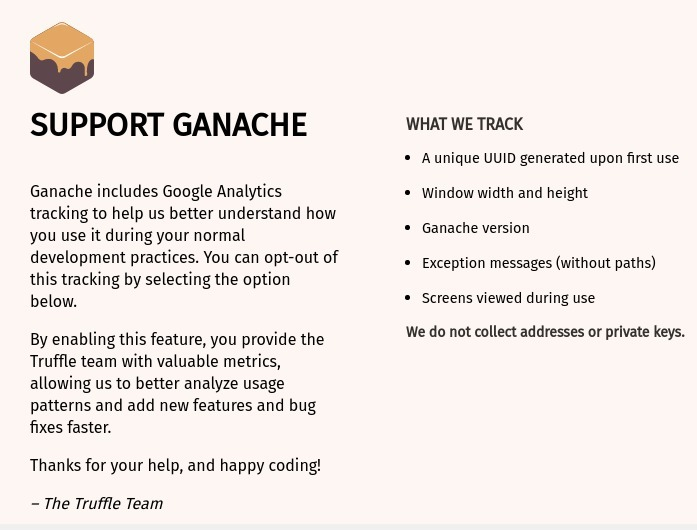


**4. Compile the source from any open source package**

Installing ganache(open source package) using AppImage



Compiling an open source package



**Results:**

The compiling from the source is studied and executed.

**Video :**

<http://www.youtube.com/watch?v=pnJgAH2E4mk>