**Ex 3 COMPILING FROM THE SOURCE**

**Date: 29.08.20**

**Aim:**

To study and implement the compiling from the source.

**Description:**

**tar**

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.

**zip**

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

**gzip**

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.

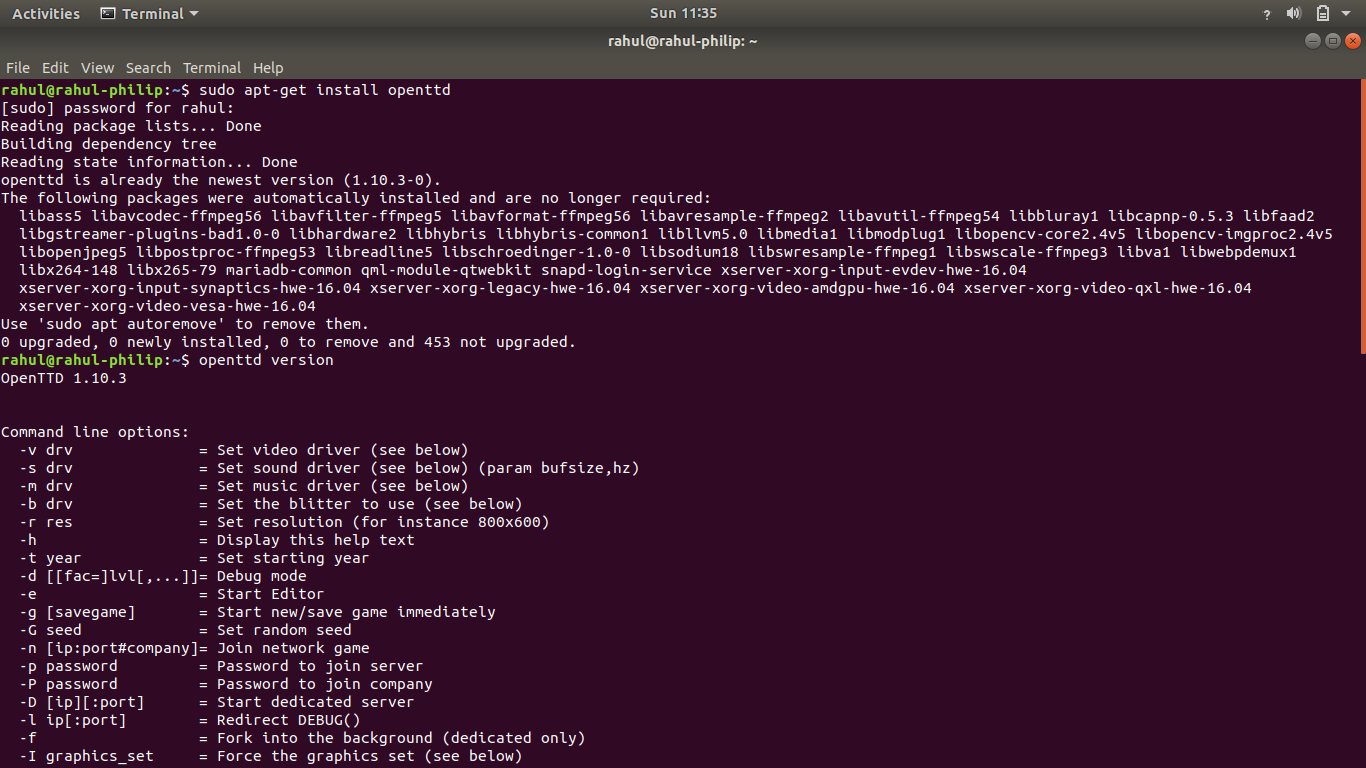
**Commands:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **Command Name** | **Syntax** | **options** |
| **1.** | **rpm** | rpm {rpm-file} | **-a, --all**  Query all packages  **-f**  Query for packages owning given file |
| **2.** | **apt-get** | apt-get [options] source pkg1 [pkg2 ...] | **-a**  It prints all the system information in the order  **-s**  It prints the kernel name.  **-n**  It prints the hostname of the network node  **-r**  It prints the kernel release date  **-v**  It prints the version of the current kernel |

**Exercise:**

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

Installing Openttd using package manager

Check the version of Openttd****

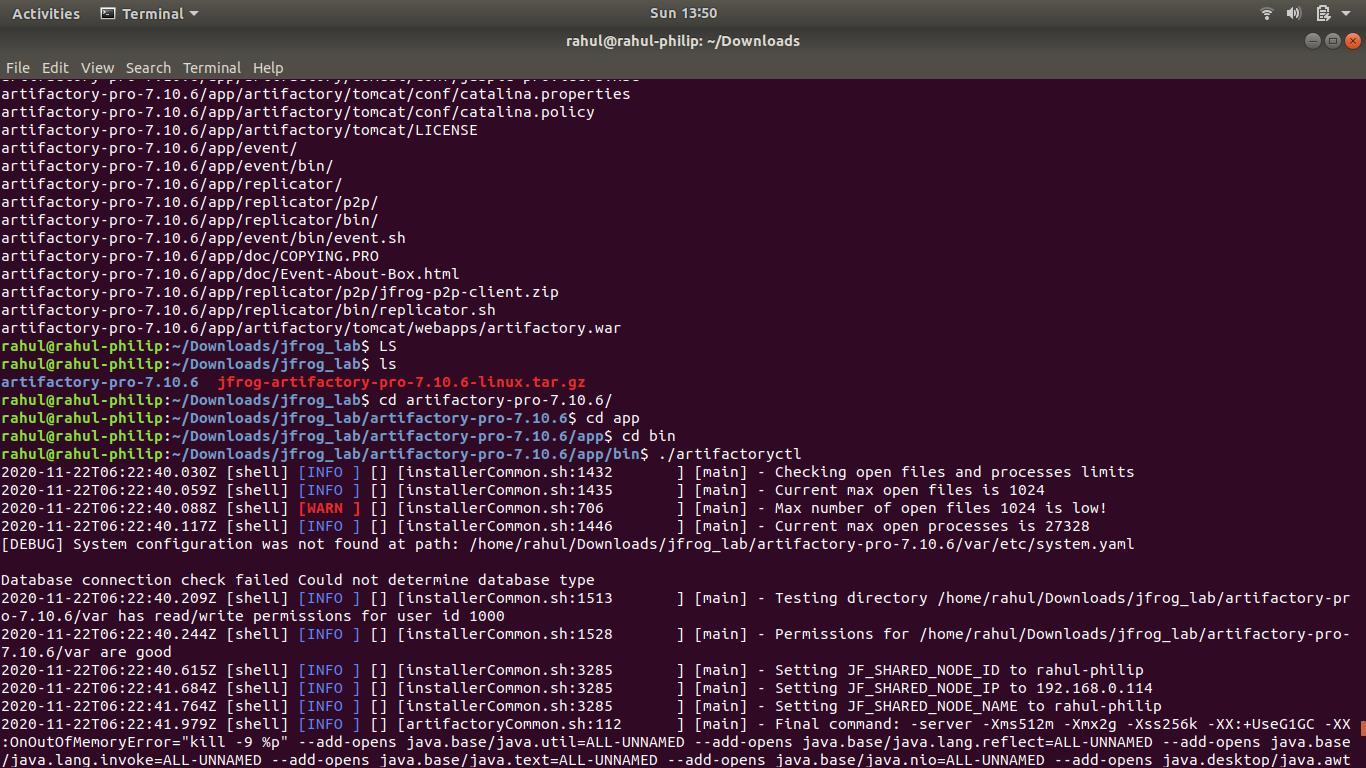
**Output:**

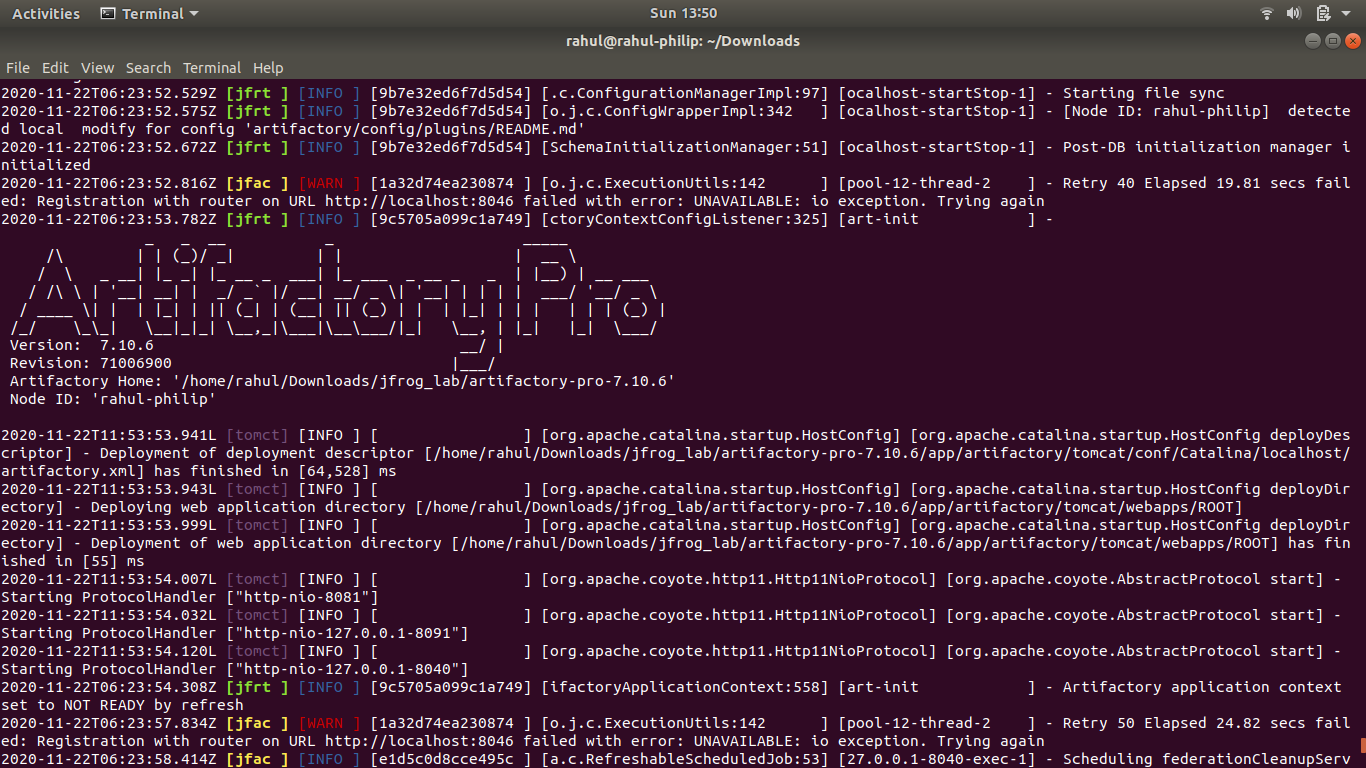
Run the Openttd ****

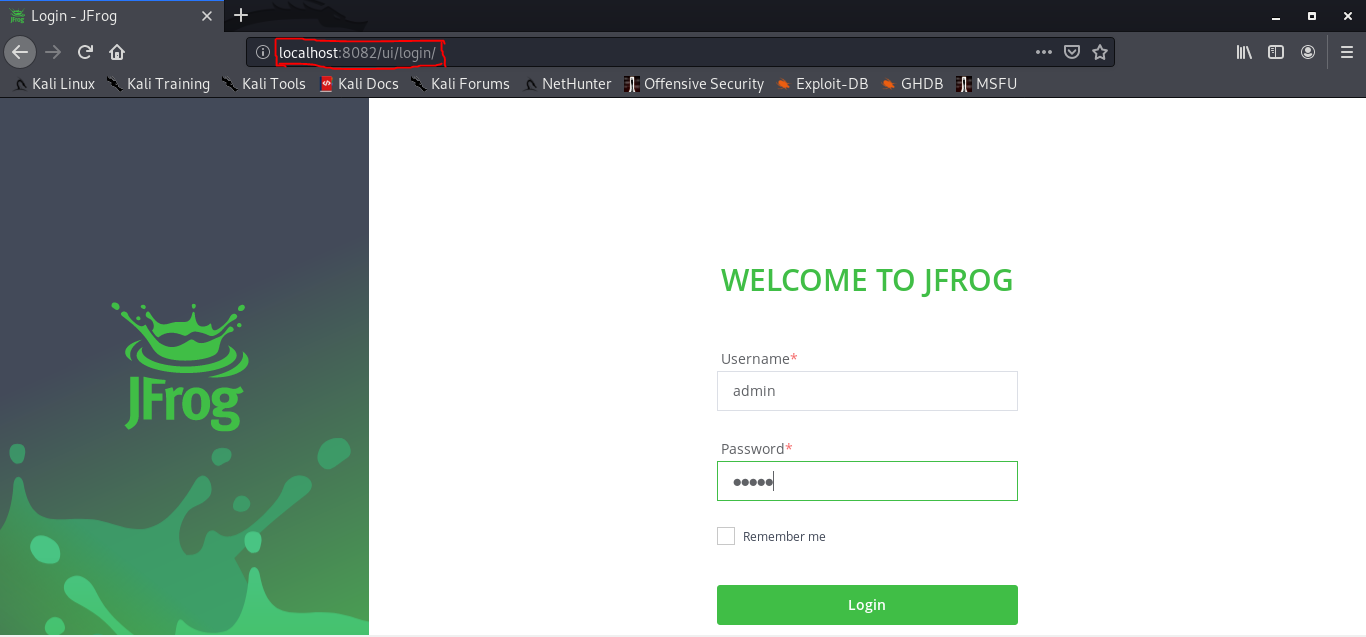
**2. Compile the source from JFrog package**

Download and Extract the JFrog file

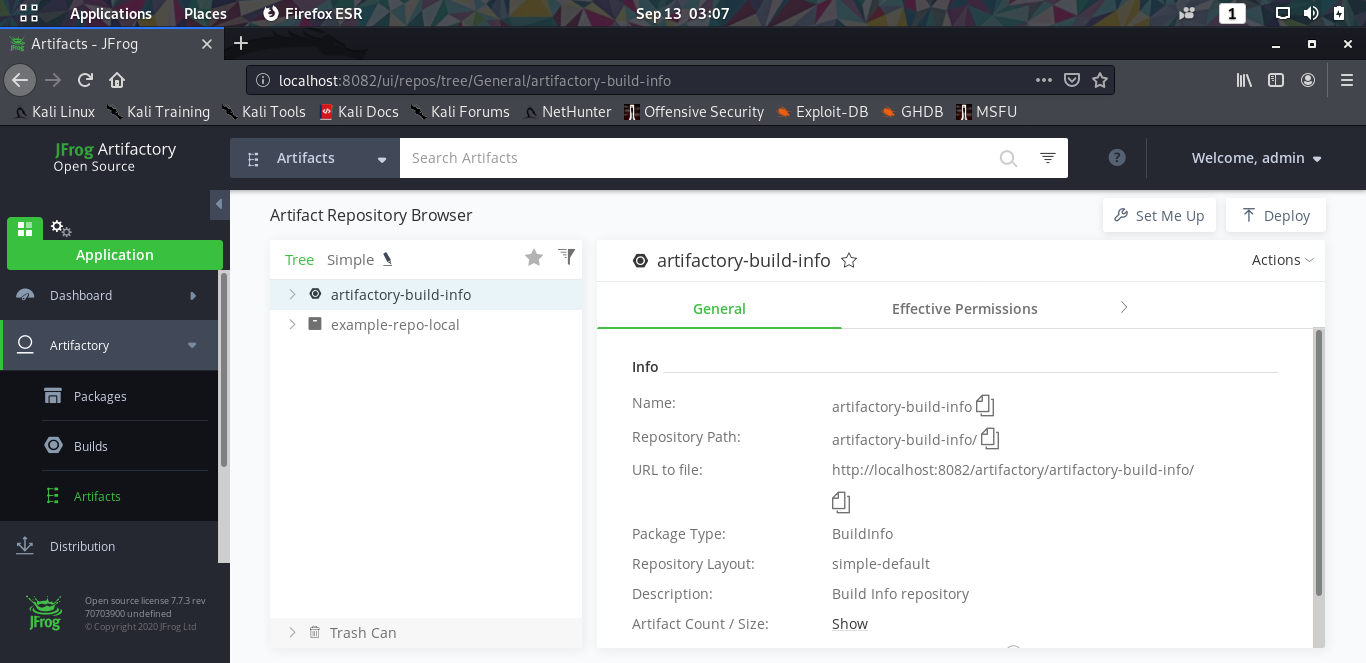
Compiling the Source of JFrog

****

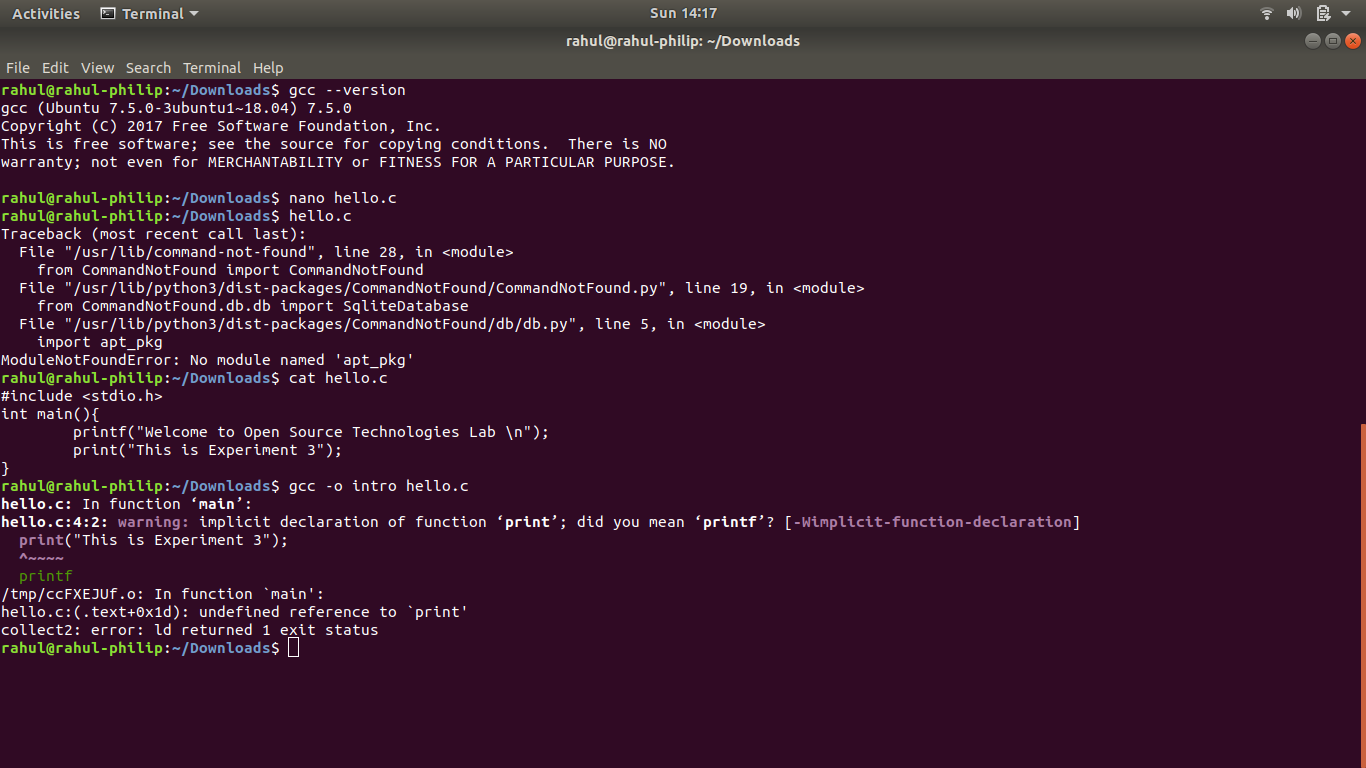


Run the JFrog in the Browser and Enter Valid Credentials ****

**Output:**

****

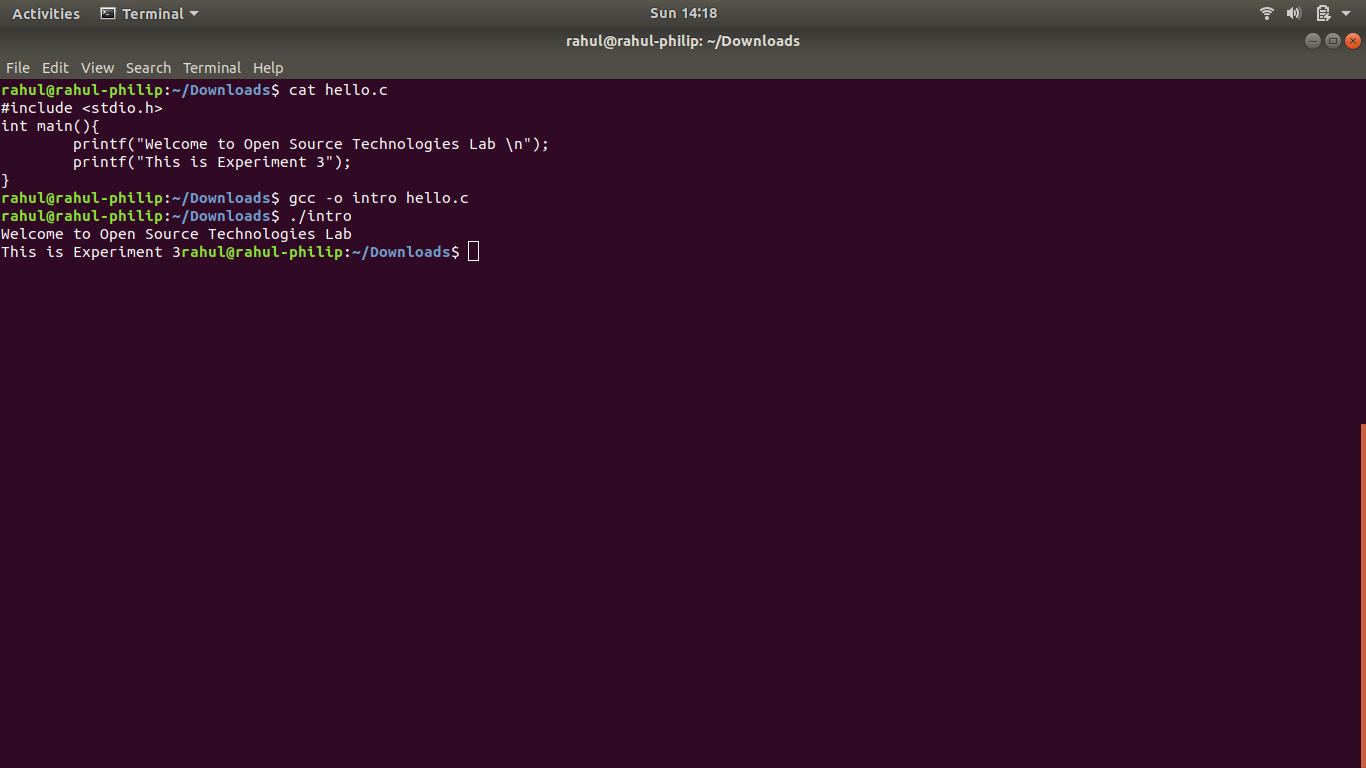
**3. Compile the Source from gcc**

Check the Version of gcc 

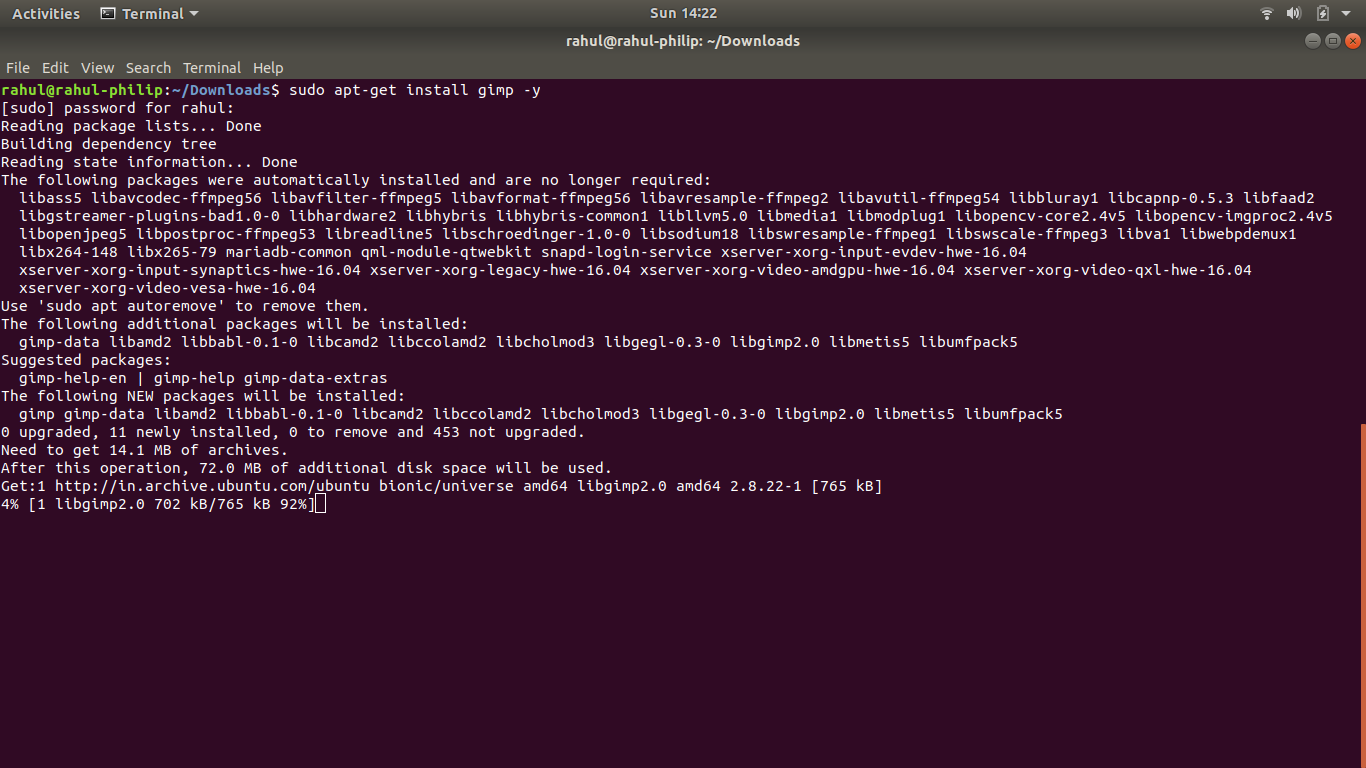
Sample Program

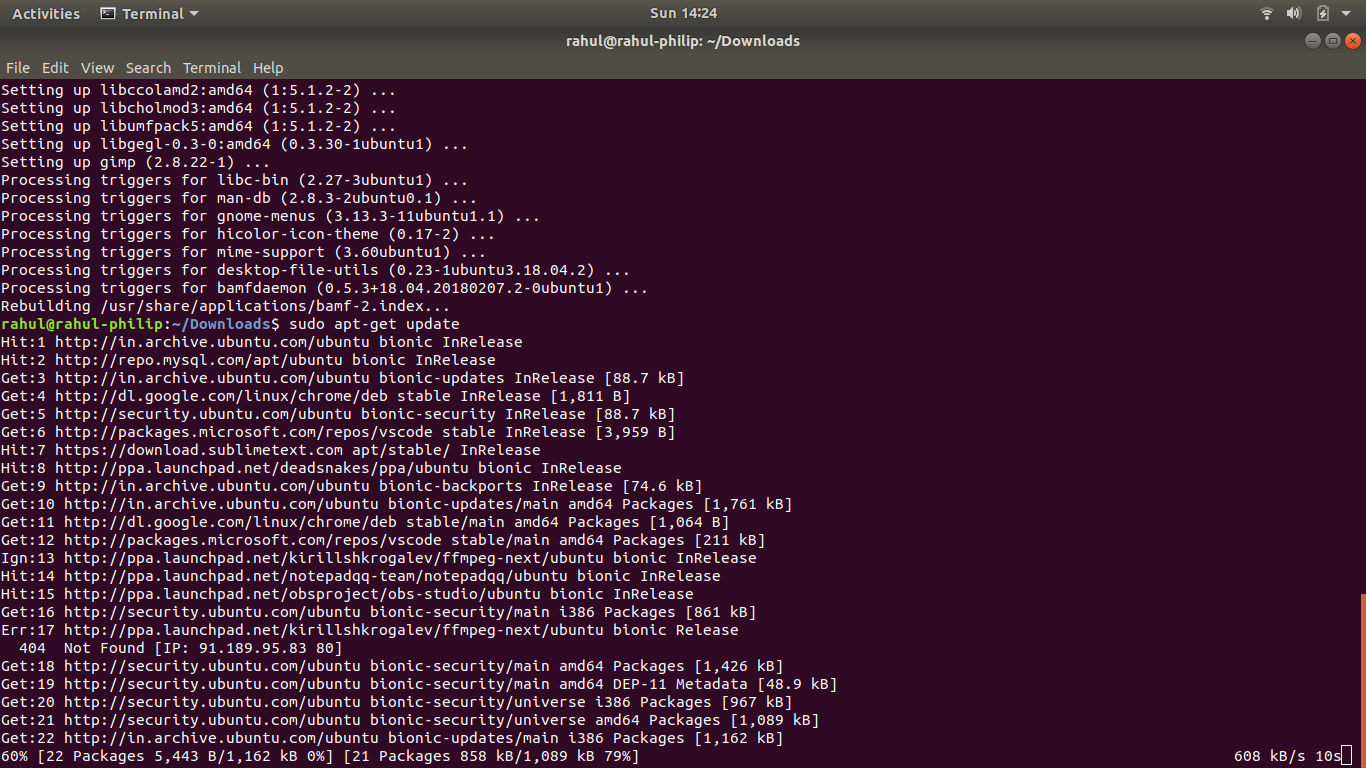
**Output:**

Compile and Run the C program using gcc



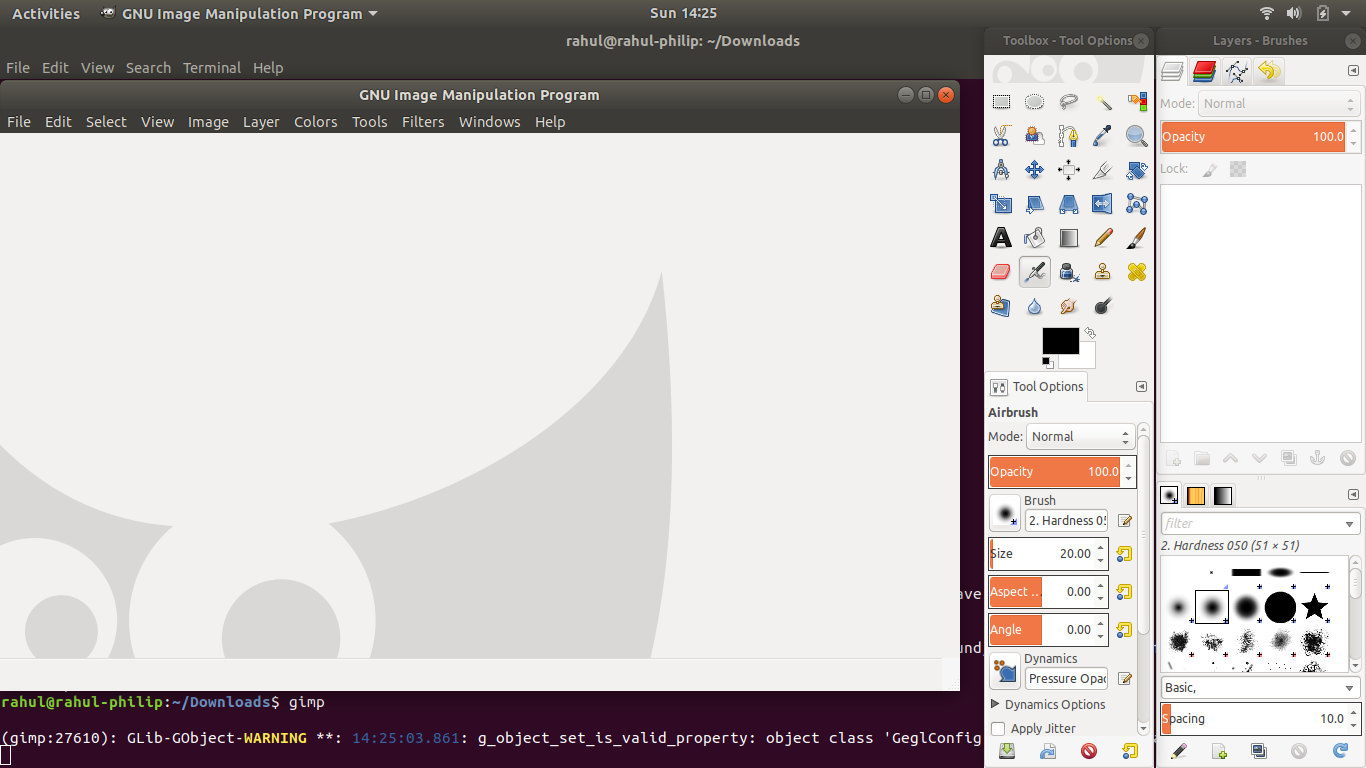
**4. Compile the source from any open source package(GIMP)**

Installing GIMP using package manager 

Update the System to Compile GIMP 

**Output:**

Run the GIMP



**Results:**

The compiling from the source is studied and executed.

**Video Link:**