Experiment No.6

Aim

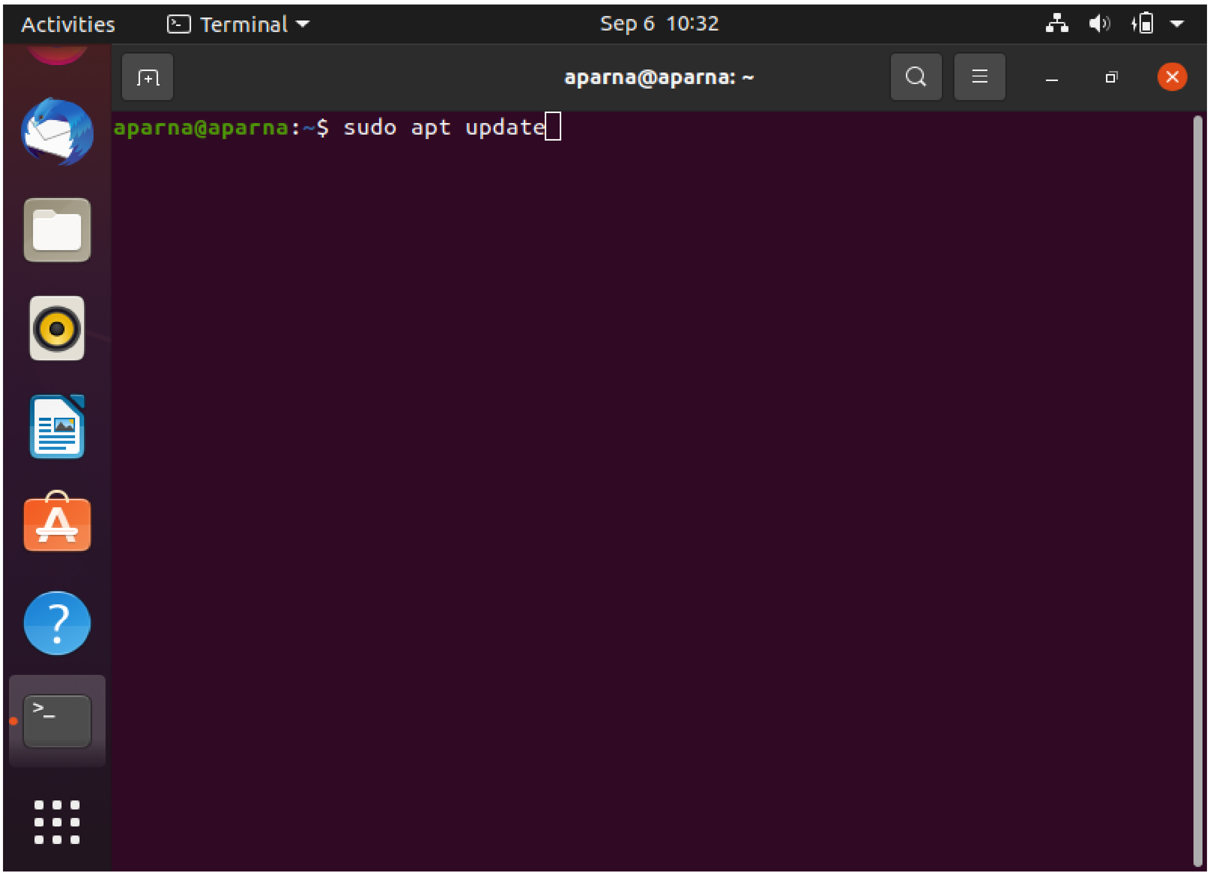
Installation and configuration of common software frame works such as Laravel.

Procedure

Step 1 – Install Apache Web Server

- Let's open up a Terminal and do first thing first update your package list using Sudo apt update command.

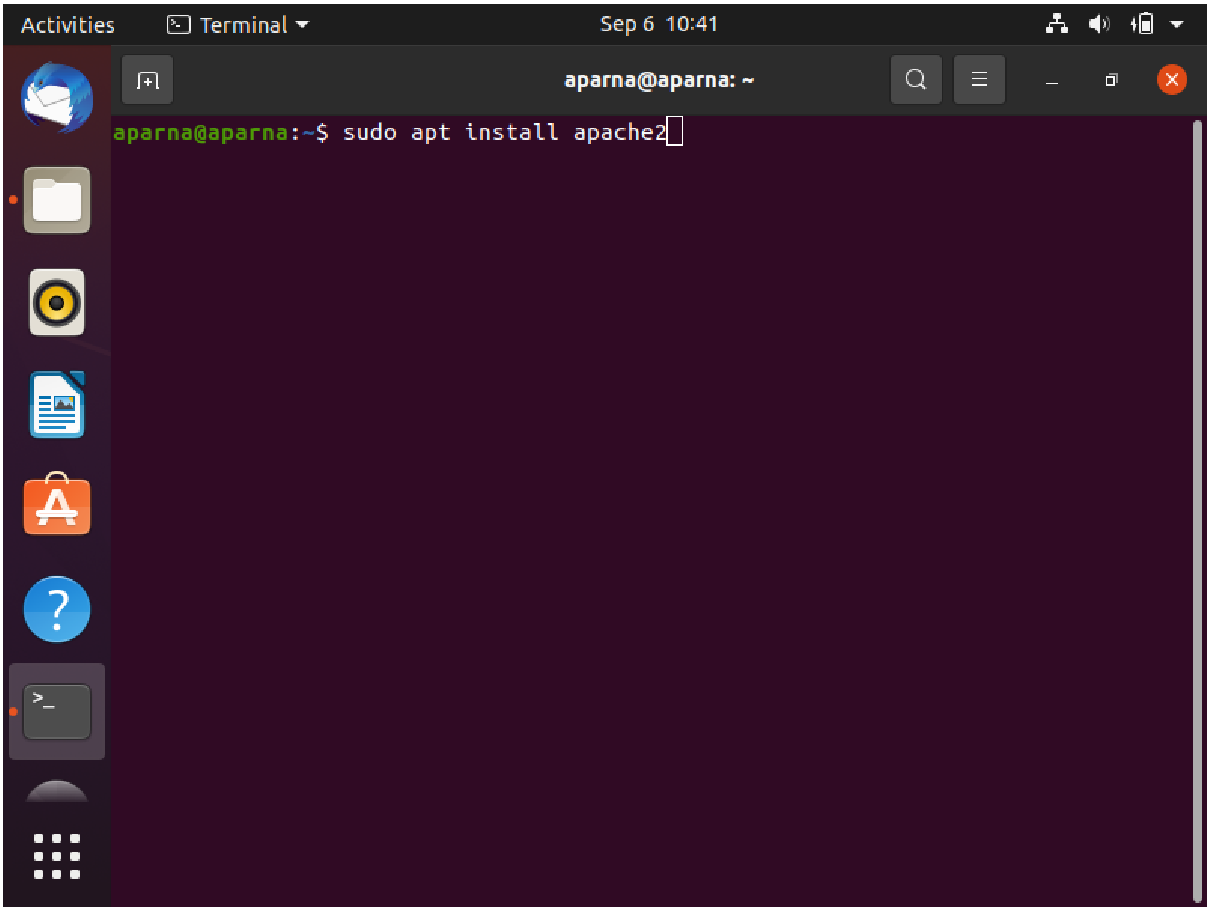
  sudo apt update



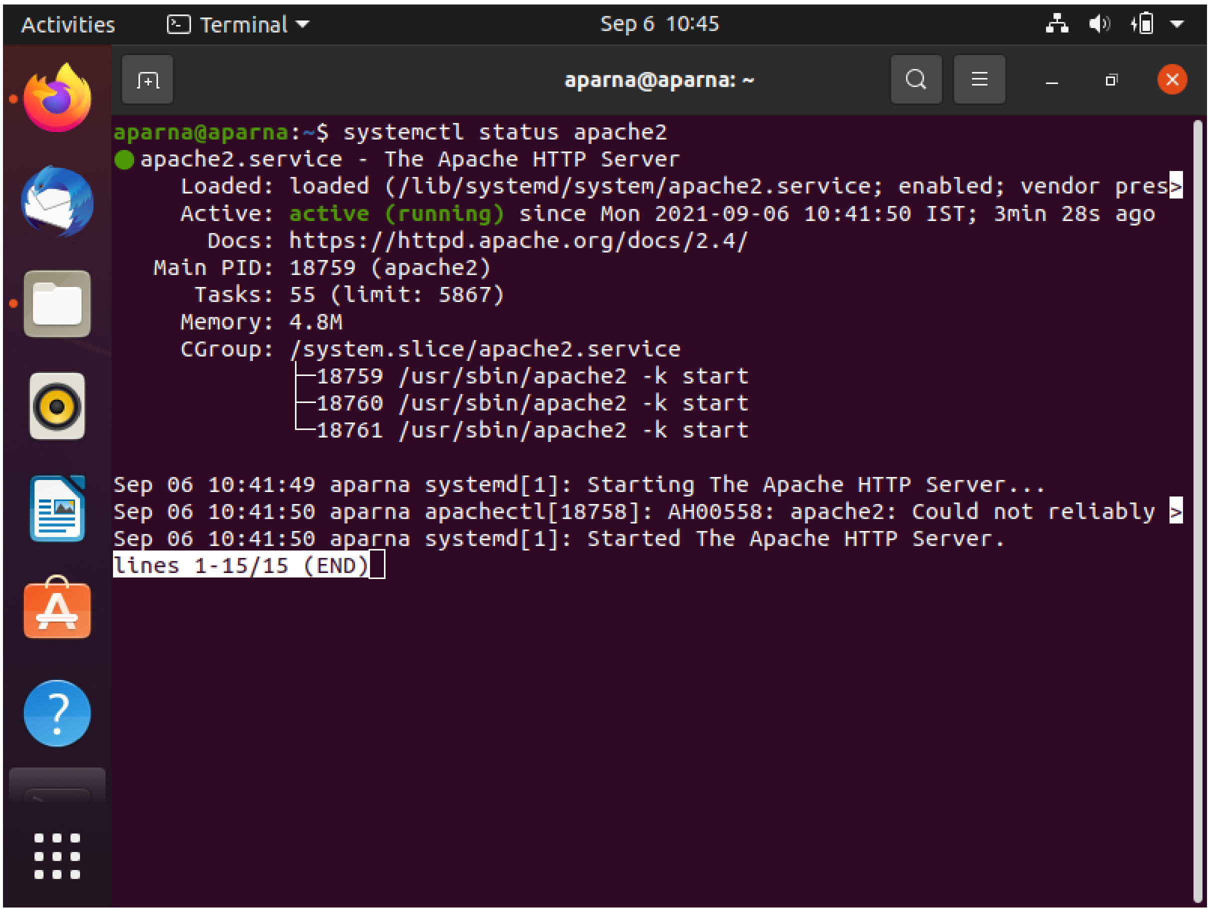
```
aparna@aparna:~$ sudo apt update
```

- After updating your package list install apache webserver. So, go ahead and type sudo apt install apache2 then hit the enter key. Press y key to proceed.

```
sudo apt install apache2
systemctl status apache2
```

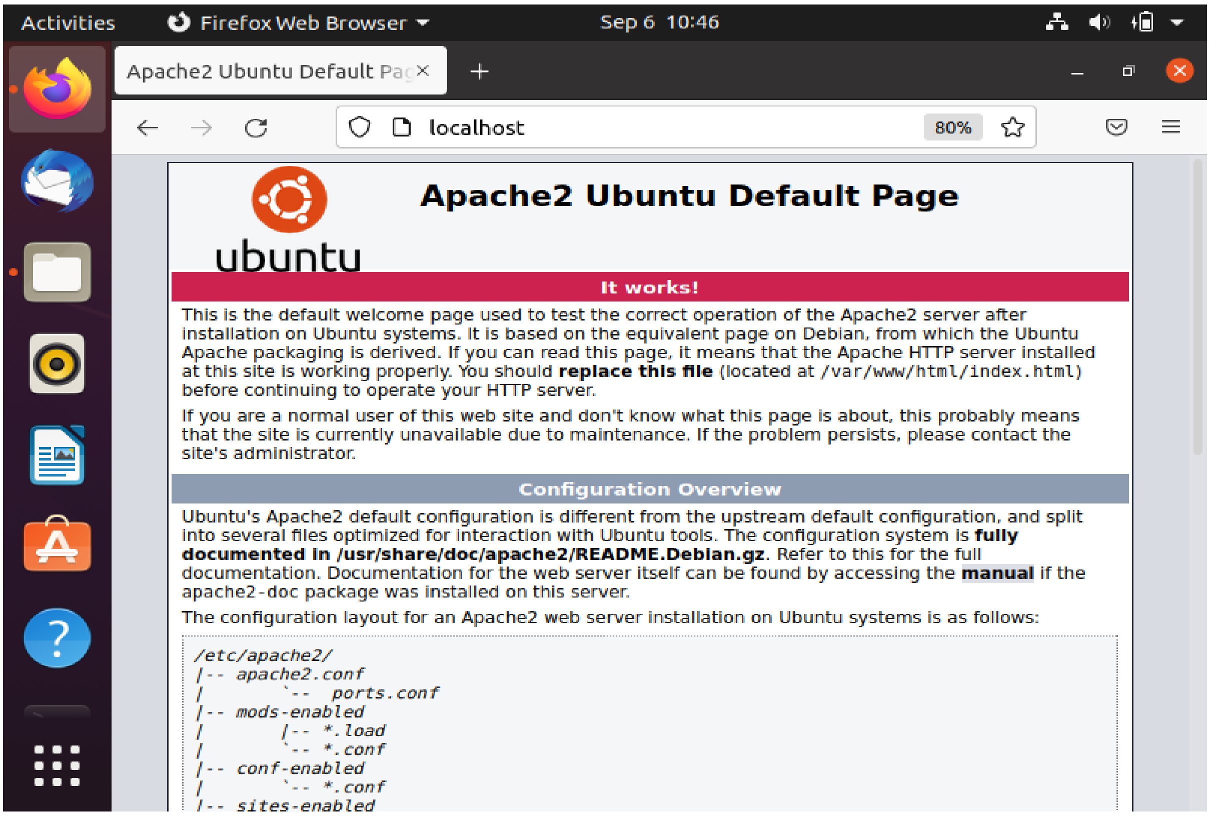


```
aparna@aparna:~$ sudo apt install apache2
```



```
aparna@aparna:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor pres>
     Active: active (running) since Mon 2021-09-06 10:41:50 IST; 3min 28s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 18759 (apache2)
      Tasks: 55 (limit: 5867)
     Memory: 4.8M
     CGroup: /system.slice/apache2.service
             ├─18759 /usr/sbin/apache2 -k start
             ├─18760 /usr/sbin/apache2 -k start
             └─18761 /usr/sbin/apache2 -k start

Sep 06 10:41:49 aparna systemd[1]: Starting The Apache HTTP Server...
Sep 06 10:41:50 aparna apachectl[18758]: AH00558: apache2: Could not reliably >
Sep 06 10:41:50 aparna systemd[1]: Started The Apache HTTP Server.
lines 1-15/15 (END)
```

- Now open up the web browser and type localhost to see the default apache webpage is serving or not



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

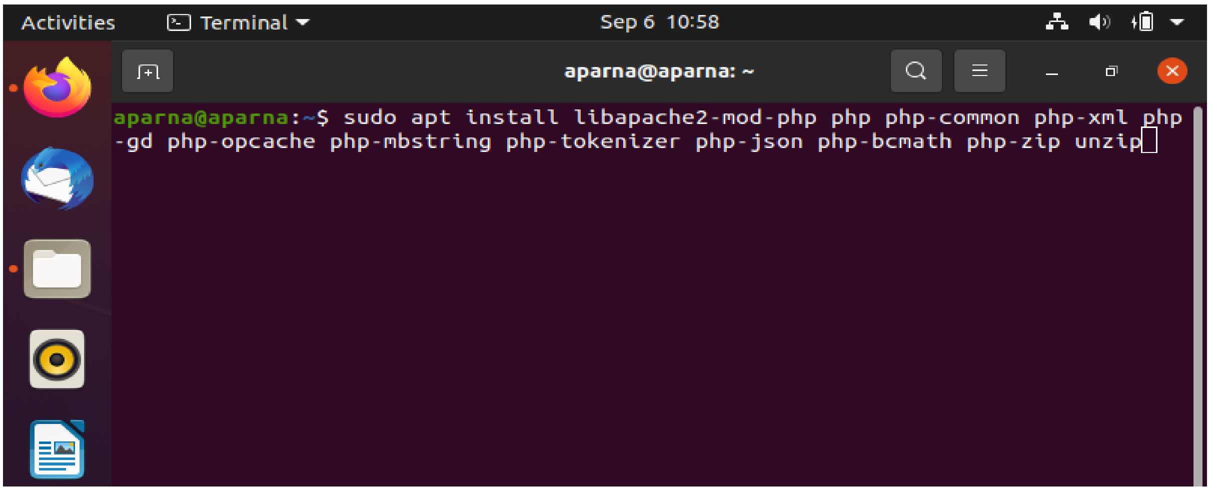
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
```

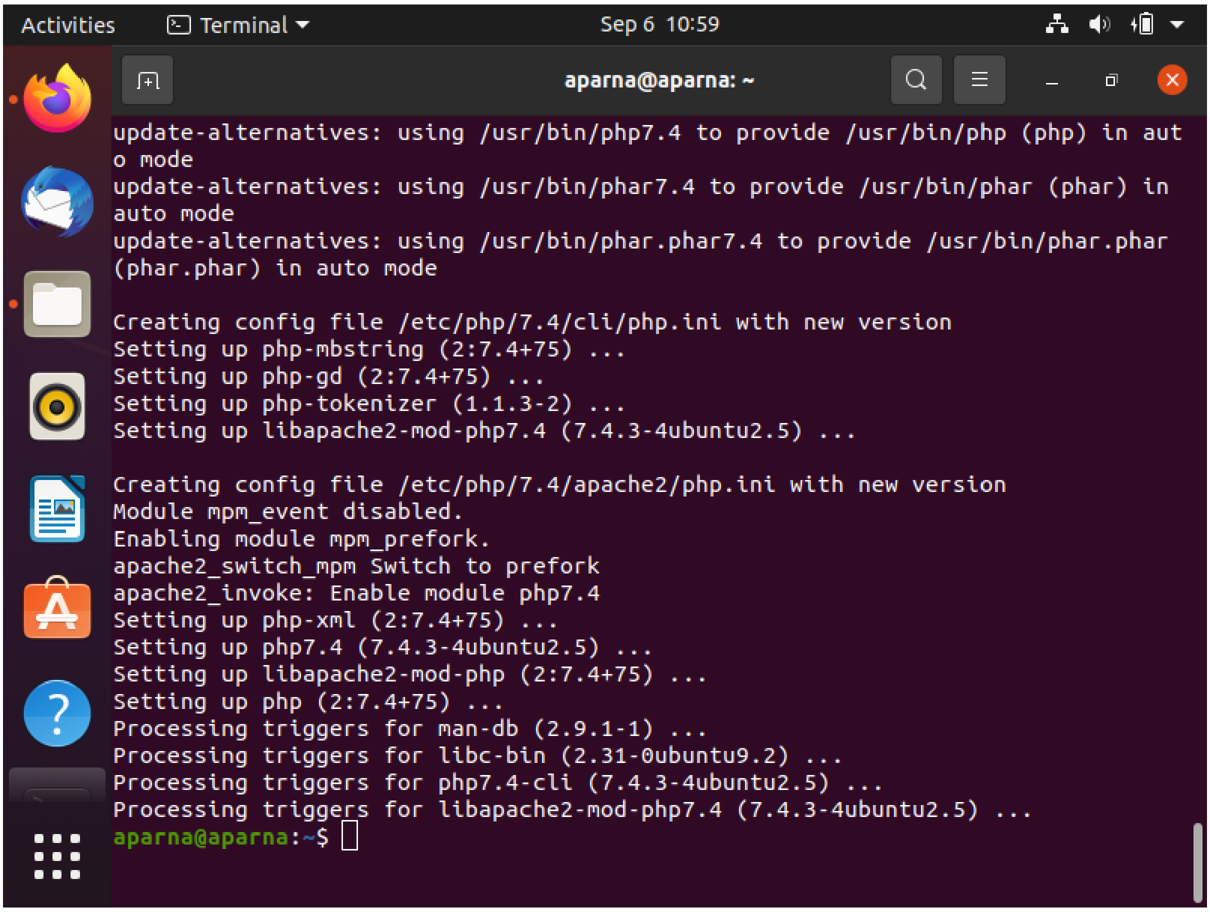
Step 2 – Install and Configure PHP 7.4

- Get back to the terminal and it’s time to install PHP. To install Laravel 8.x, at least you must have PHP >= 7.3 on your system. And by default, the official Ubuntu 20.04 repository provides PHP 7.4 packages. Install PHP 7.4 packages using the apt command below

  ```
  sudo apt install libapache2-mod-php php php-common php-xml
  php-gd php-opcache php-mbstring php-tokenizer php-json php-
  bcmath php-zip unzip
  ```

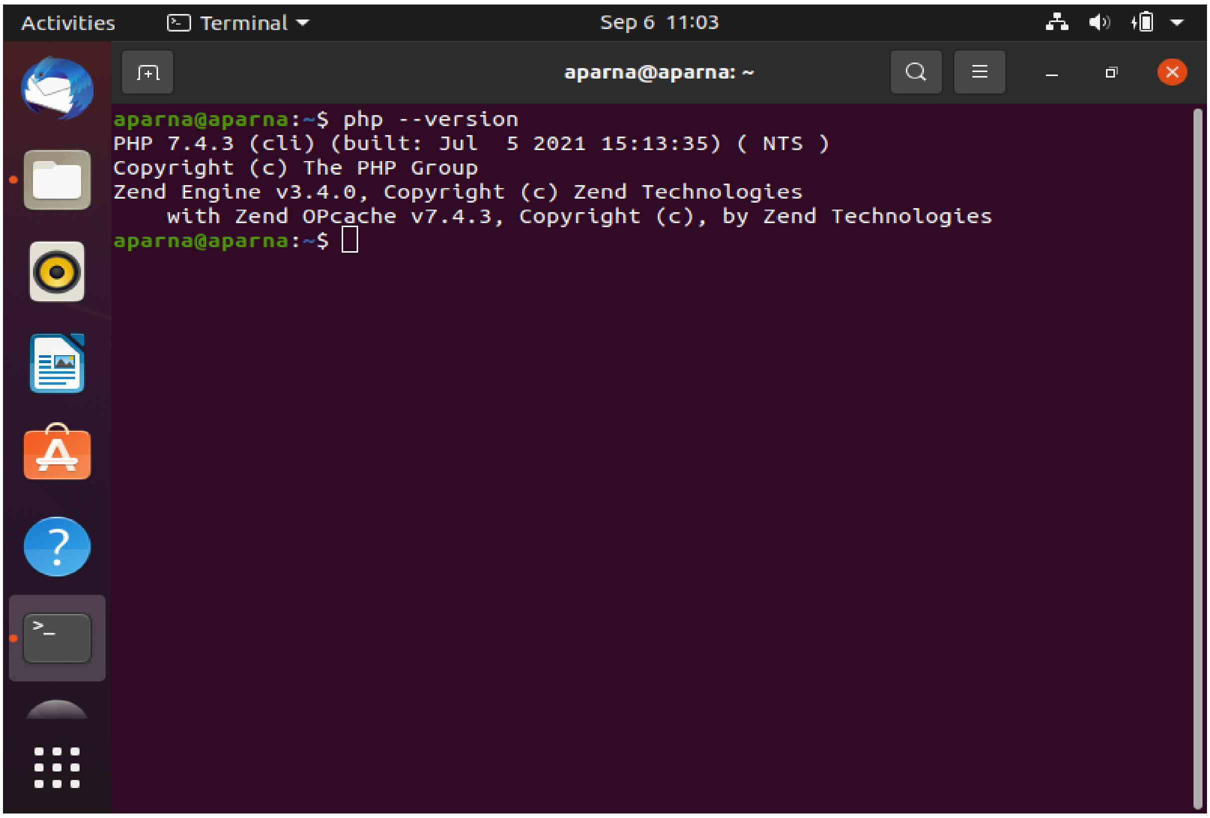


```
aparna@aparna:~$ sudo apt install libapache2-mod-php php php-common php-xml php
-gd php-opcache php-mbstring php-tokenizer php-json php-bcmath php-zip unzip
```



```
update-alternatives: using /usr/bin/php7.4 to provide /usr/bin/php (php) in aut
o mode
update-alternatives: using /usr/bin/phar7.4 to provide /usr/bin/phar (phar) in
auto mode
update-alternatives: using /usr/bin/phar.phar7.4 to provide /usr/bin/phar.phar
(phar.phar) in auto mode

Creating config file /etc/php/7.4/cli/php.ini with new version
Setting up php-mbstring (2:7.4+75) ...
Setting up php-gd (2:7.4+75) ...
Setting up php-tokenizer (1.1.3-2) ...
Setting up libapache2-mod-php7.4 (7.4.3-4ubuntu2.5) ...

Creating config file /etc/php/7.4/apache2/php.ini with new version
Module mpm_event disabled.
Enabling module mpm_prefork.
apache2_switch_mpm Switch to prefork
apache2_invoke: Enable module php7.4
Setting up php-xml (2:7.4+75) ...
Setting up php7.4 (7.4.3-4ubuntu2.5) ...
Setting up libapache2-mod-php (2:7.4+75) ...
Setting up php (2:7.4+75) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
Processing triggers for php7.4-cli (7.4.3-4ubuntu2.5) ...
Processing triggers for libapache2-mod-php7.4 (7.4.3-4ubuntu2.5) ...
aparna@aparna:~$
```

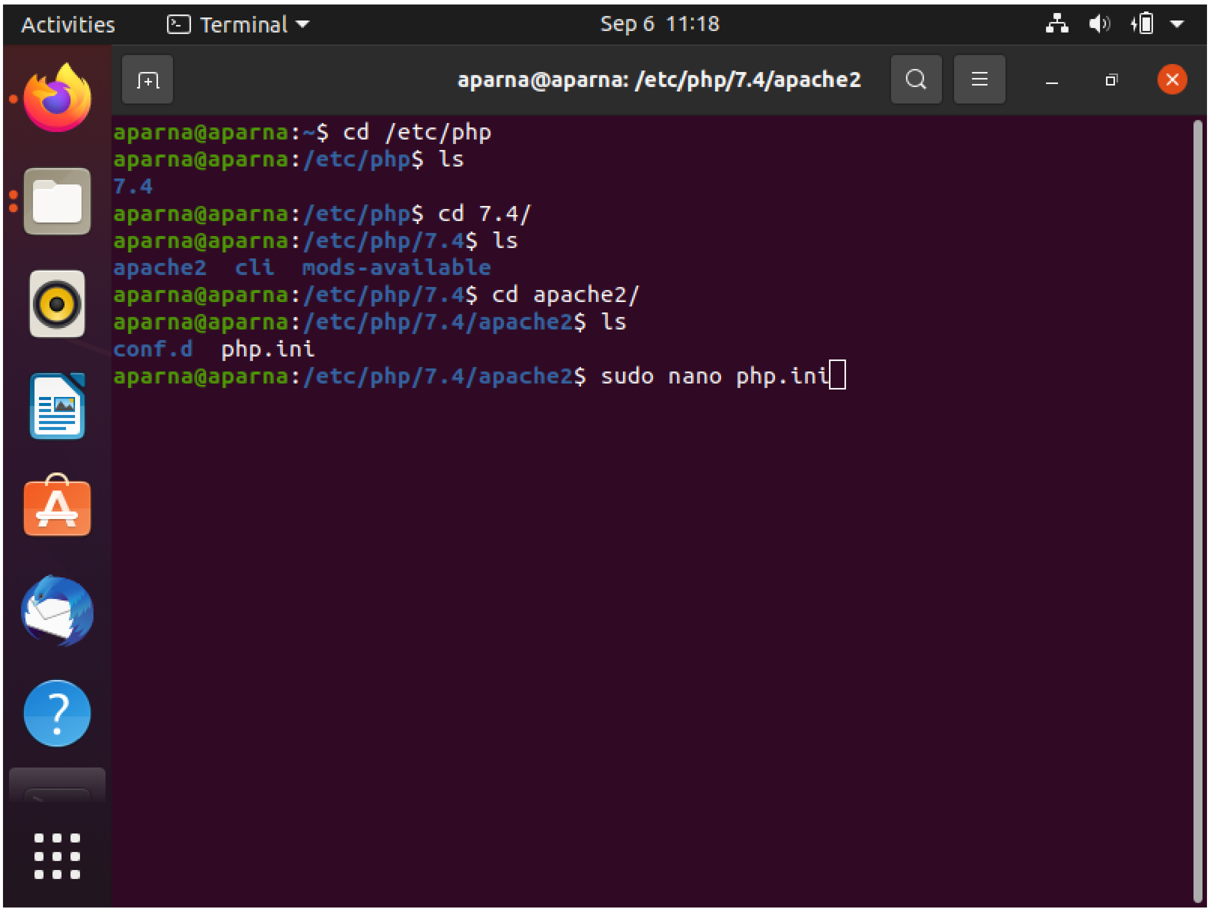
- You can check your PHP version using it.
  php – version



```
aparna@aparna:~$ php --version
PHP 7.4.3 (cli) (built: Jul  5 2021 15:13:35) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies
aparna@aparna:~$
```

- Now go ahead and make tweak changes in PHP ini file and set cgi.fix_pathinfo set to be 0. If this number is kept as a 1, the php interpreter will do its best to process the file that is as near to the requested file as possible. This is a possible security risk. If this number is set to 0, conversely, the interpreter will only process the exact file path—a much safer alternative.

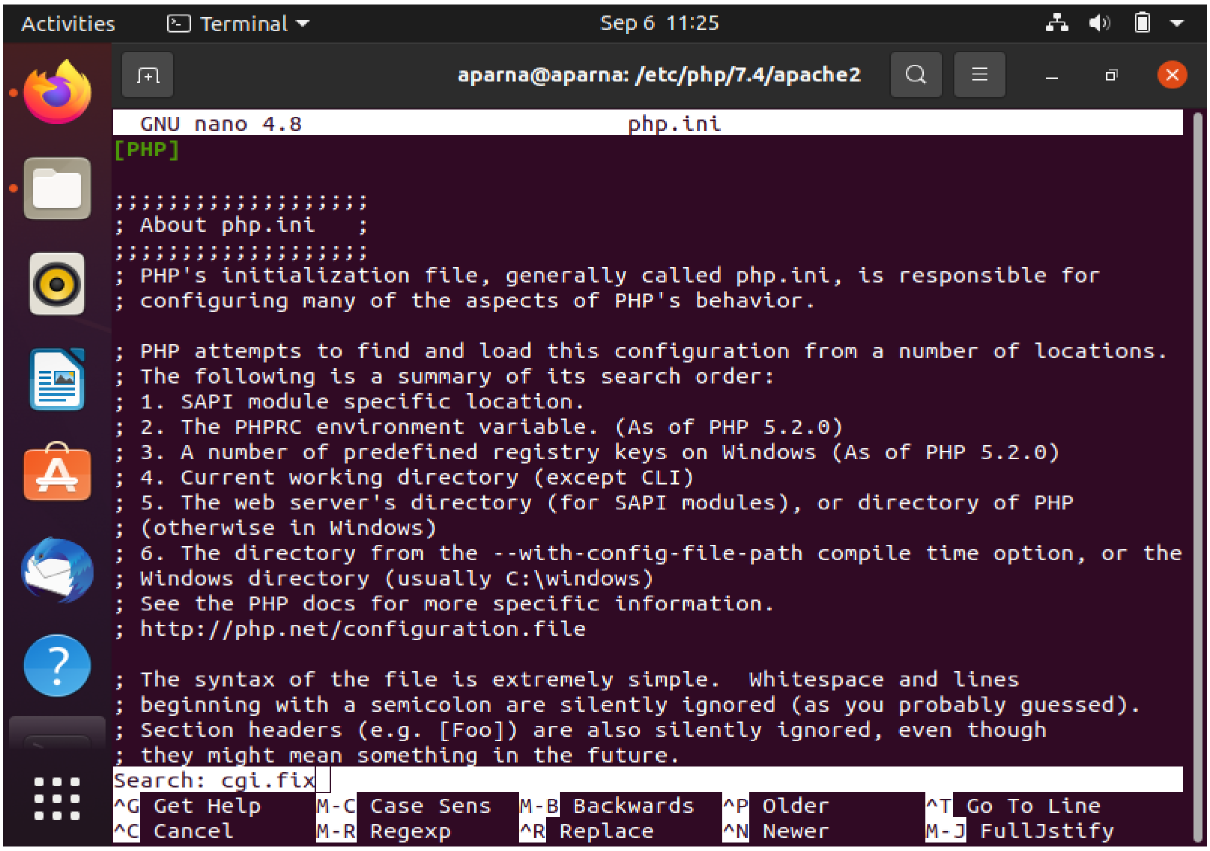
```
cd /etc/php/7.4/apache2

 sudo nano php.ini
```



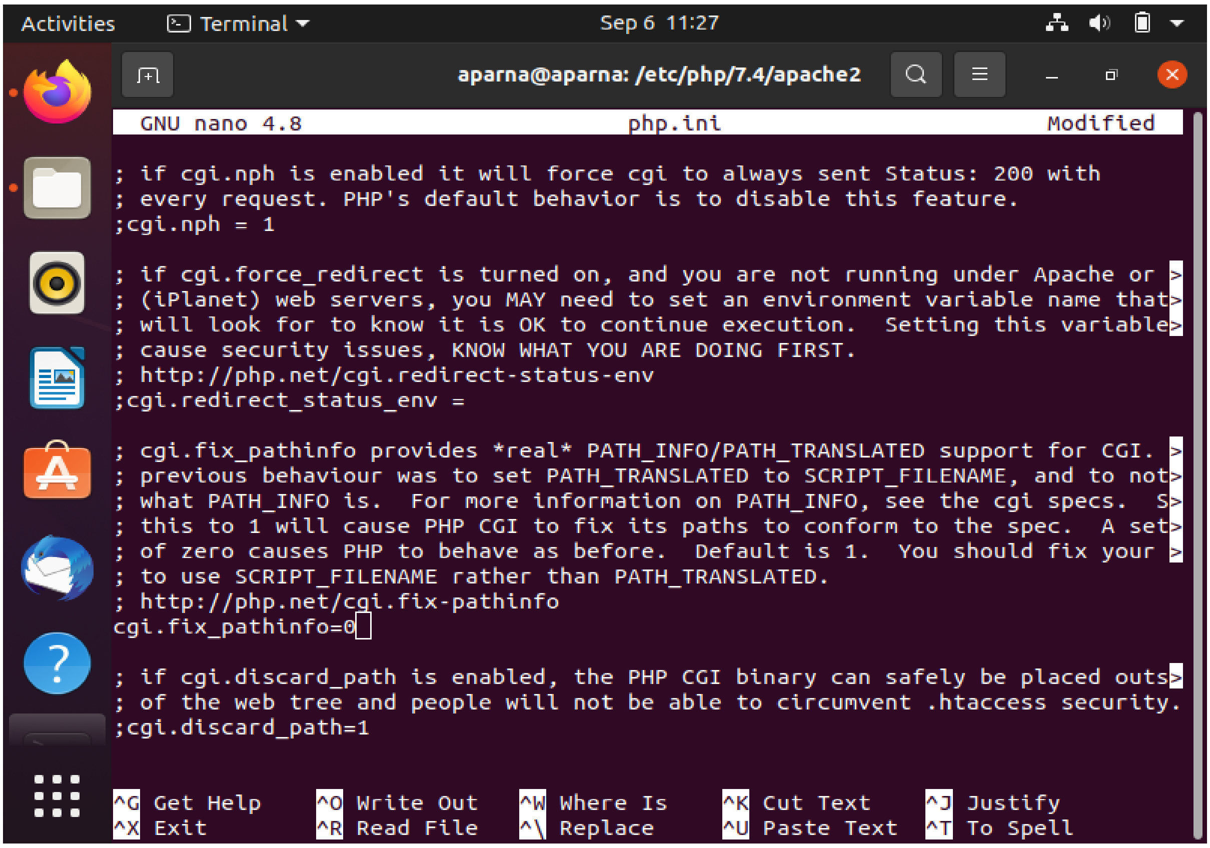
```
aparna@aparna:~$ cd /etc/php
aparna@aparna:/etc/php$ ls
7.4
aparna@aparna:/etc/php$ cd 7.4/
aparna@aparna:/etc/php/7.4$ ls
apache2  cli  mods-available
aparna@aparna:/etc/php/7.4$ cd apache2/
aparna@aparna:/etc/php/7.4/apache2$ ls
conf.d  php.ini
aparna@aparna:/etc/php/7.4/apache2$ sudo nano php.ini
```

- Press ctrl+w and search for the word “cgi.fix” the uncomment the line and set it to 0.

```
...
cgi.fix_pathinfo=0
...
```



```
  GNU nano 4.8                          php.ini
[PHP]

;;;;;;;;;;;;;;;;;;;;
; About php.ini   ;
;;;;;;;;;;;;;;;;;;;;
; PHP's initialization file, generally called php.ini, is responsible for
; configuring many of the aspects of PHP's behavior.

; PHP attempts to find and load this configuration from a number of locations.
; The following is a summary of its search order:
; 1. SAPI module specific location.
; 2. The PHPRC environment variable. (As of PHP 5.2.0)
; 3. A number of predefined registry keys on Windows (As of PHP 5.2.0)
; 4. Current working directory (except CLI)
; 5. The web server's directory (for SAPI modules), or directory of PHP
; (otherwise in Windows)
; 6. The directory from the --with-config-file-path compile time option, or the
; Windows directory (usually C:\windows)
; See the PHP docs for more specific information.
; http://php.net/configuration.file

; The syntax of the file is extremely simple.  Whitespace and lines
; beginning with a semicolon are silently ignored (as you probably guessed).
; Section headers (e.g. [Foo]) are also silently ignored, even though
; they might mean something in the future.
Search: cgi.fix
^G Get Help    M-C Case Sens   M-B Backwards   ^P Older       ^T Go To Line
^C Cancel      M-R Regexp      ^R Replace      ^N Newer       M-J FullJstify
```

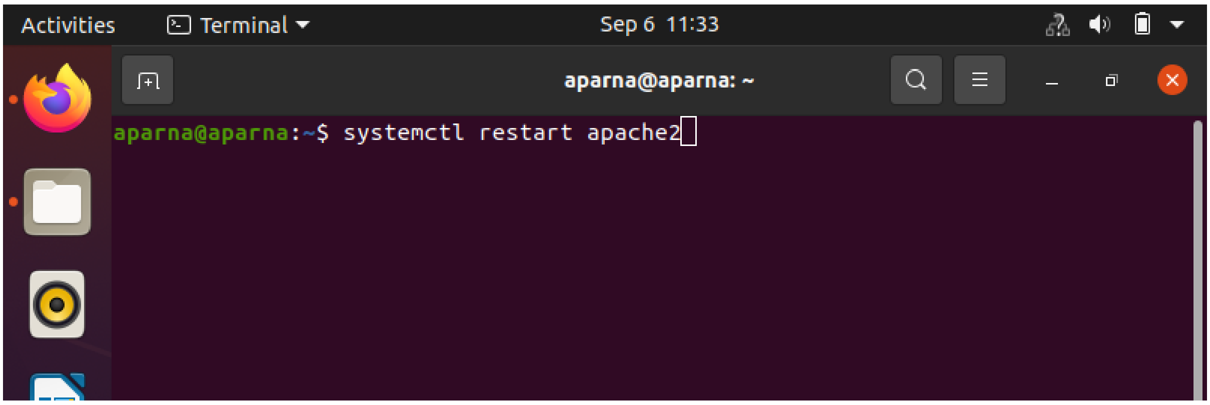


```
  GNU nano 4.8                          php.ini                          Modified

; if cgi.nph is enabled it will force cgi to always sent Status: 200 with
; every request. PHP's default behavior is to disable this feature.
;cgi.nph = 1

; if cgi.force_redirect is turned on, and you are not running under Apache or >
; (iPlanet) web servers, you MAY need to set an environment variable name that>
; will look for to know it is OK to continue execution.  Setting this variable>
; cause security issues, KNOW WHAT YOU ARE DOING FIRST.
; http://php.net/cgi.redirect-status-env
;cgi.redirect_status_env =

; cgi.fix_pathinfo provides *real* PATH_INFO/PATH_TRANSLATED support for CGI. >
; previous behaviour was to set PATH_TRANSLATED to SCRIPT_FILENAME, and to not>
; what PATH_INFO is.  For more information on PATH_INFO, see the cgi specs.  S>
; this to 1 will cause PHP CGI to fix its paths to conform to the spec.  A set>
; of zero causes PHP to behave as before.  Default is 1.  You should fix your >
; to use SCRIPT_FILENAME rather than PATH_TRANSLATED.
; http://php.net/cgi.fix-pathinfo
cgi.fix_pathinfo=0

; if cgi.discard_path is enabled, the PHP CGI binary can safely be placed outs>
; of the web tree and people will not be able to circumvent .htaccess security.
;cgi.discard_path=1

^G Get Help    ^O Write Out    ^W Where Is     ^K Cut Text    ^J Justify
^X Exit        ^R Read File    ^\ Replace      ^U Paste Text  ^T To Spell
```

Press Ctrl + x then y to Save and Exit.

- Now Restart The apache service.

```
systemctl restart apache2
```

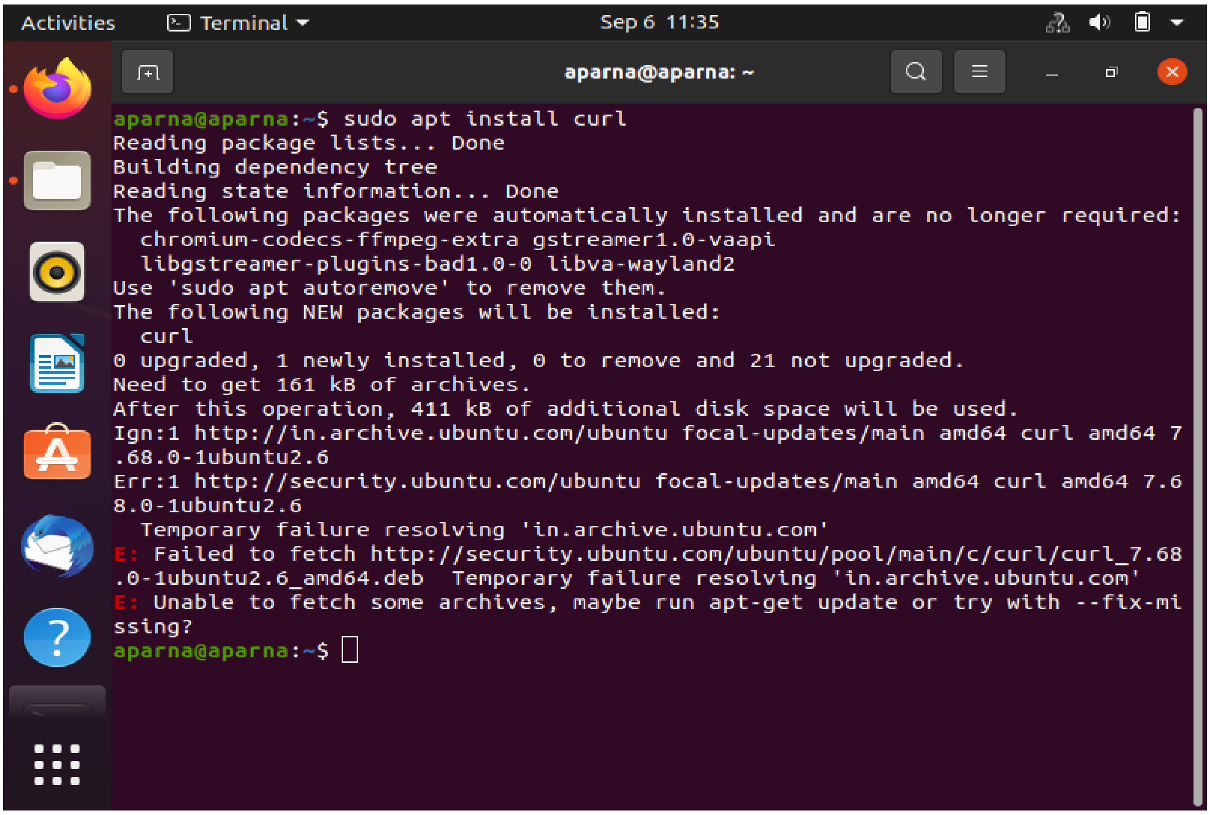


```
aparna@aparna:~$ systemctl restart apache2
```

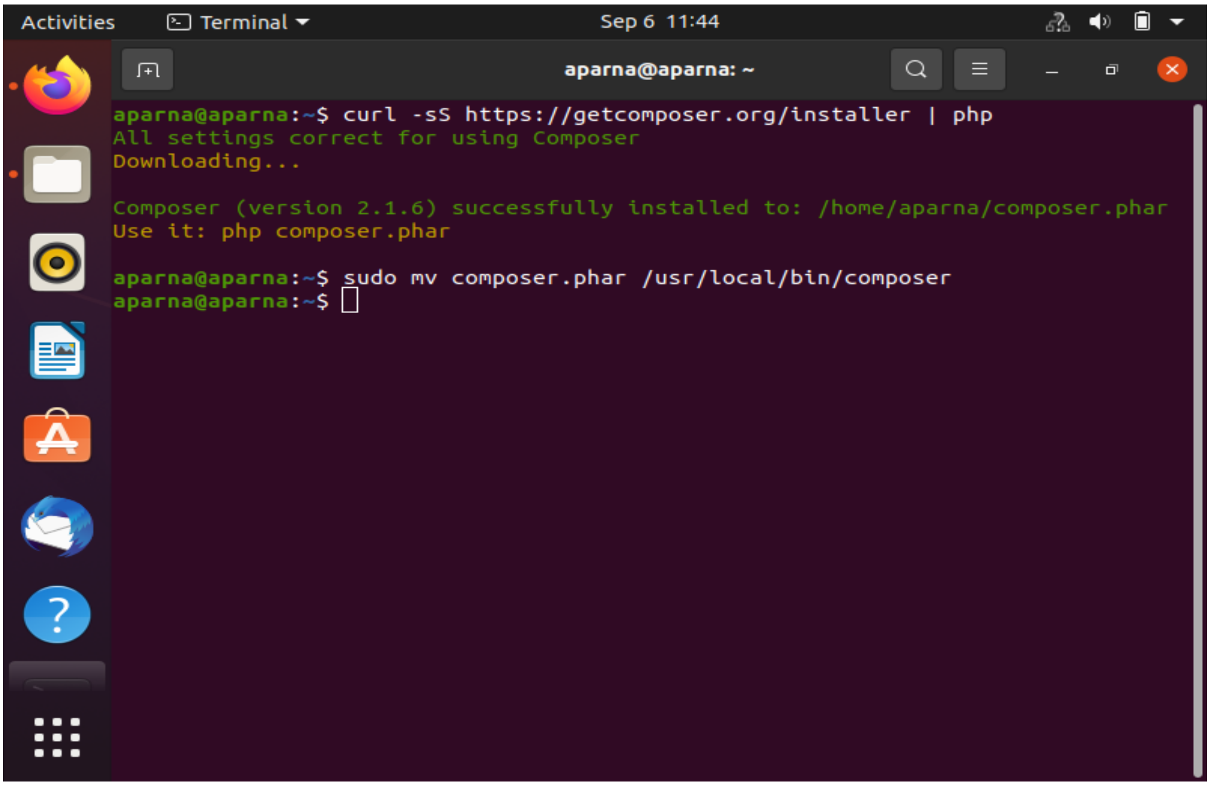
Step 3 – Install Composer PHP Packages Management

- Now it's time to install the composer package manager go ahead and download and install Composer. and move the composer .phar file to usr/local/bin/composer directory.

```
sudo apt install curl

curl -sS https://getcomposer.org/installer | php

sudo mv composer.phar /usr/local/bin/composer
```



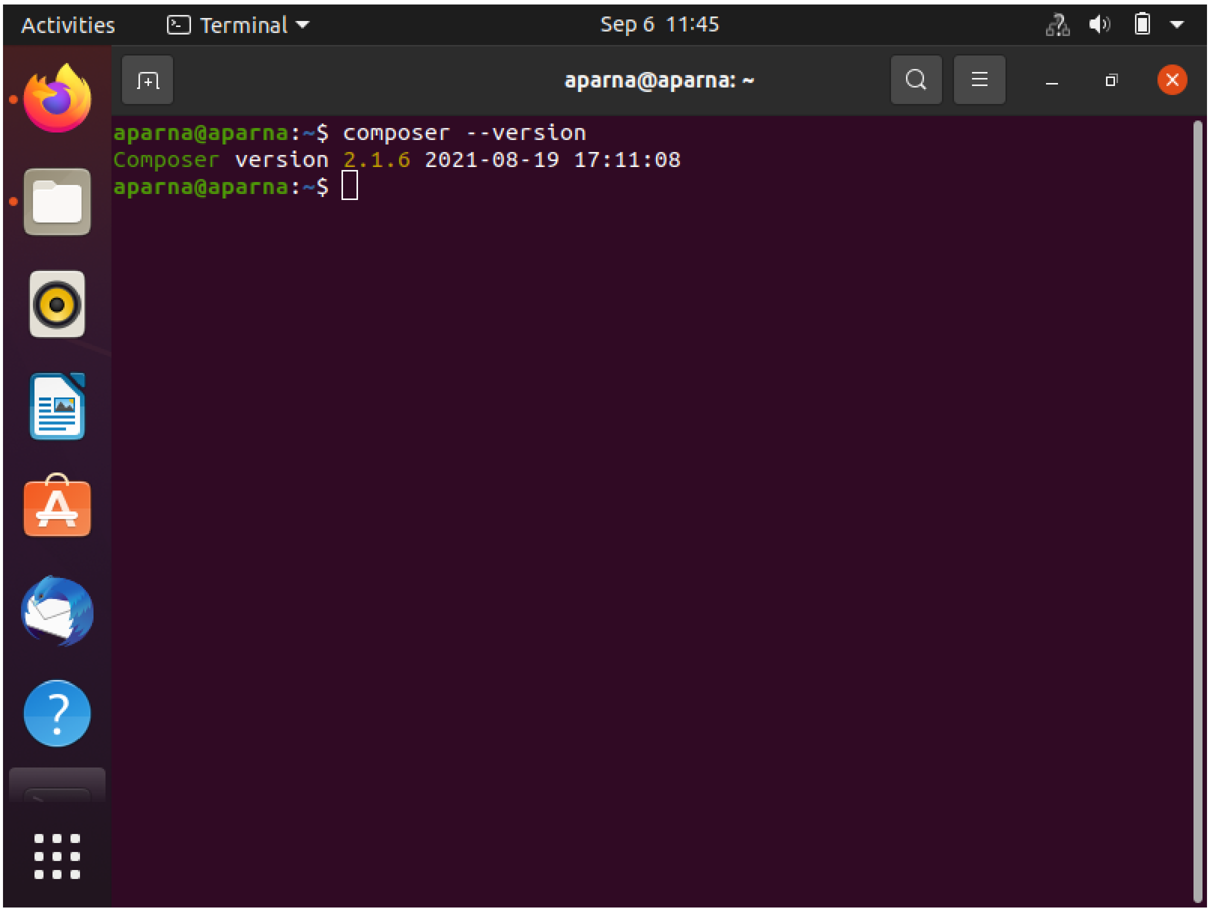
```
aparna@aparna:~$ sudo apt install curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  curl
0 upgraded, 1 newly installed, 0 to remove and 21 not upgraded.
Need to get 161 kB of archives.
After this operation, 411 kB of additional disk space will be used.
Ign:1 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 curl amd64 7
.68.0-1ubuntu2.6
Err:1 http://security.ubuntu.com/ubuntu focal-updates/main amd64 curl amd64 7.6
8.0-1ubuntu2.6
  Temporary failure resolving 'in.archive.ubuntu.com'
E: Failed to fetch http://security.ubuntu.com/ubuntu/pool/main/c/curl/curl_7.68
.0-1ubuntu2.6_amd64.deb  Temporary failure resolving 'in.archive.ubuntu.com'
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-mi
ssing?
aparna@aparna:~$ 
```



```
aparna@aparna:~$ curl -sS https://getcomposer.org/installer | php
All settings correct for using Composer
Downloading...

Composer (version 2.1.6) successfully installed to: /home/aparna/composer.phar
Use it: php composer.phar

aparna@aparna:~$ sudo mv composer.phar /usr/local/bin/composer
aparna@aparna:~$ 
```

- You can check your installed composer version by typing the composer – version.

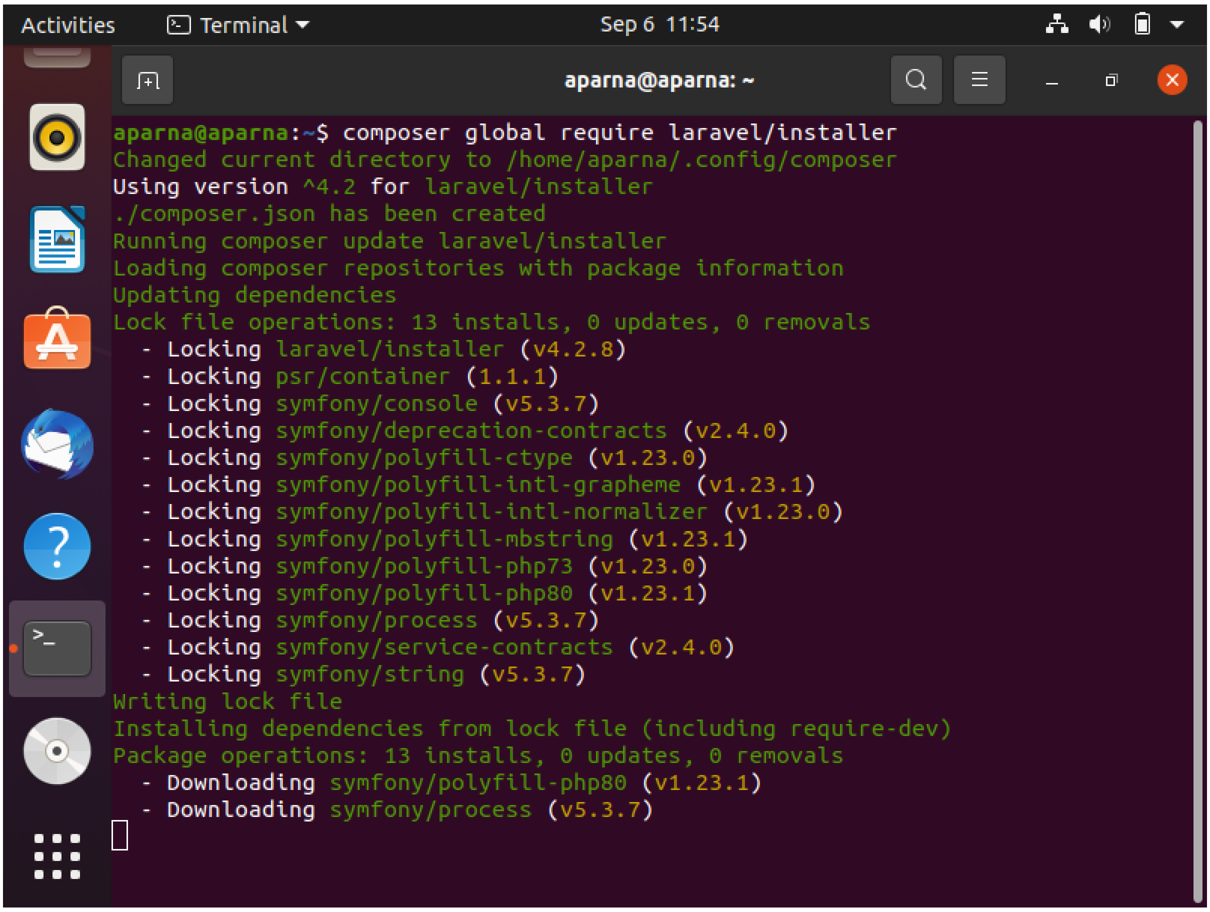
composer –version



```
aparna@aparna:~$ composer --version
Composer version 2.1.6 2021-08-19 17:11:08
aparna@aparna:~$ 
```

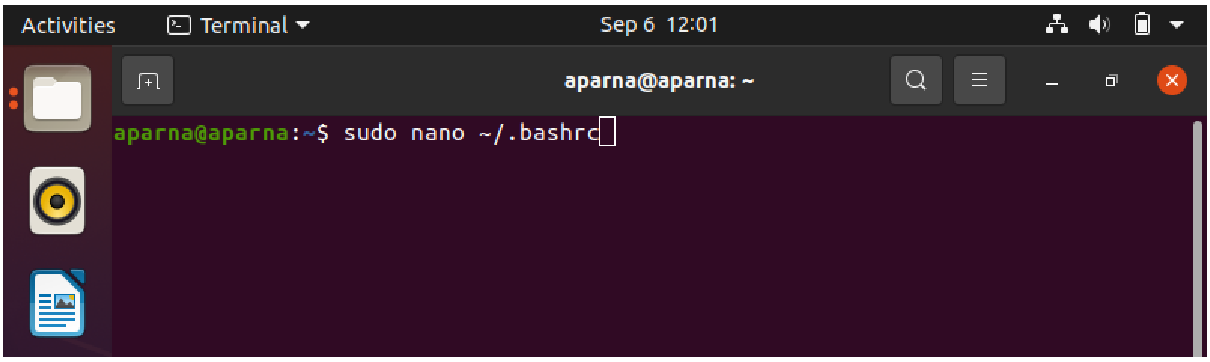
Step 4 – Install Laravel 8.x on Ubuntu 20.04

- Now install Laravel Framework using composer, just type composer global require Laravel/installer It will take a while to complete download its dependencies.
  `composer global require laravel/installer`



```
aparna@aparna:~$ composer global require laravel/installer
Changed current directory to /home/aparna/.config/composer
Using version ^4.2 for laravel/installer
./composer.json has been created
Running composer update laravel/installer
Loading composer repositories with package information
Updating dependencies
Lock file operations: 13 installs, 0 updates, 0 removals
  - Locking laravel/installer (v4.2.8)
  - Locking psr/container (1.1.1)
  - Locking symfony/console (v5.3.7)
  - Locking symfony/deprecation-contracts (v2.4.0)
  - Locking symfony/polyfill-ctype (v1.23.0)
  - Locking symfony/polyfill-intl-grapheme (v1.23.1)
  - Locking symfony/polyfill-intl-normalizer (v1.23.0)
  - Locking symfony/polyfill-mbstring (v1.23.1)
  - Locking symfony/polyfill-php73 (v1.23.0)
  - Locking symfony/polyfill-php80 (v1.23.1)
  - Locking symfony/process (v5.3.7)
  - Locking symfony/service-contracts (v2.4.0)
  - Locking symfony/string (v5.3.7)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 13 installs, 0 updates, 0 removals
  - Downloading symfony/polyfill-php80 (v1.23.1)
  - Downloading symfony/process (v5.3.7)
```

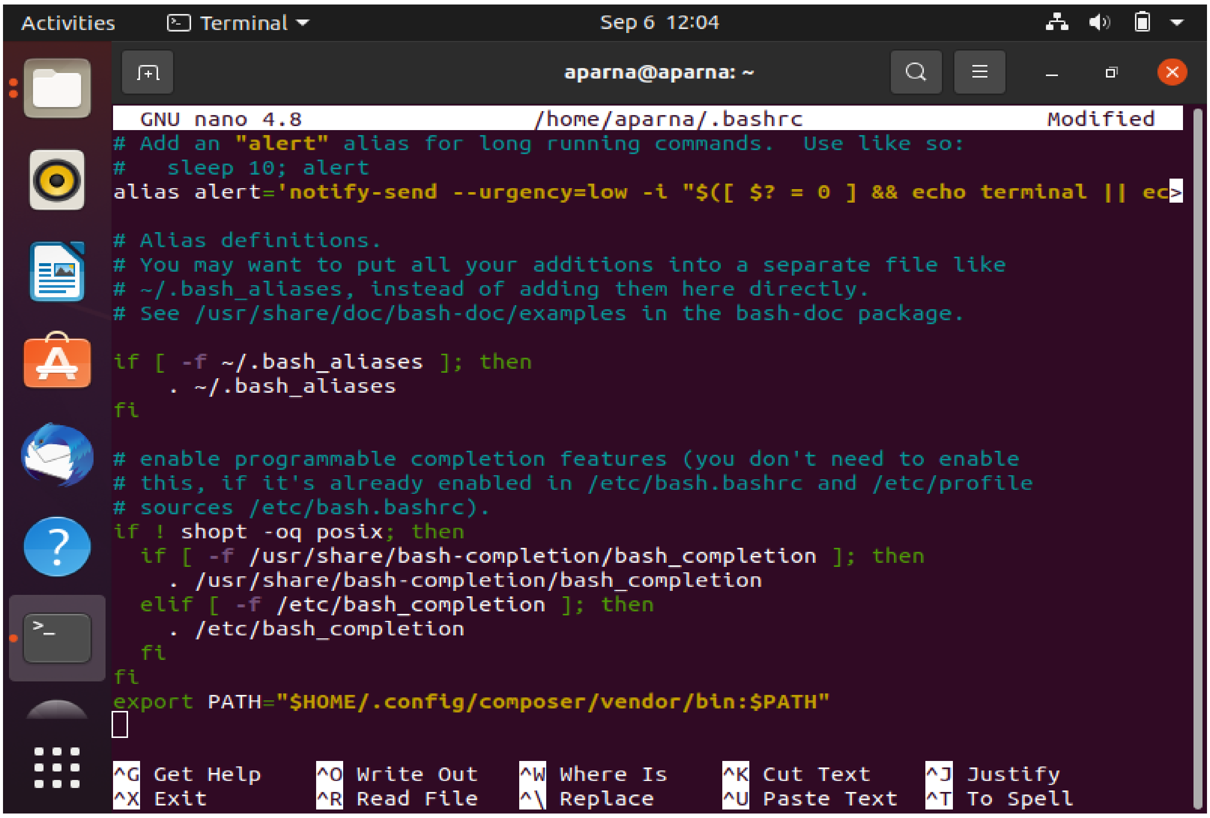
- As you had seen above image, all packages have been installed on the '~/.config/composer' directory. Next, we need to add the 'bin' directory to the PATH environment through the ~/.bashrc configuration. So Now Edit the ~/.bashrc configuration using nano command
  `nano ~/.bashrc`



```
aparna@aparna:~$ sudo nano ~/.bashrc
```

And add the following line at the end of the file.

```
...
export PATH="$HOME/.config/composer/vendor/bin:$PATH"
...
```



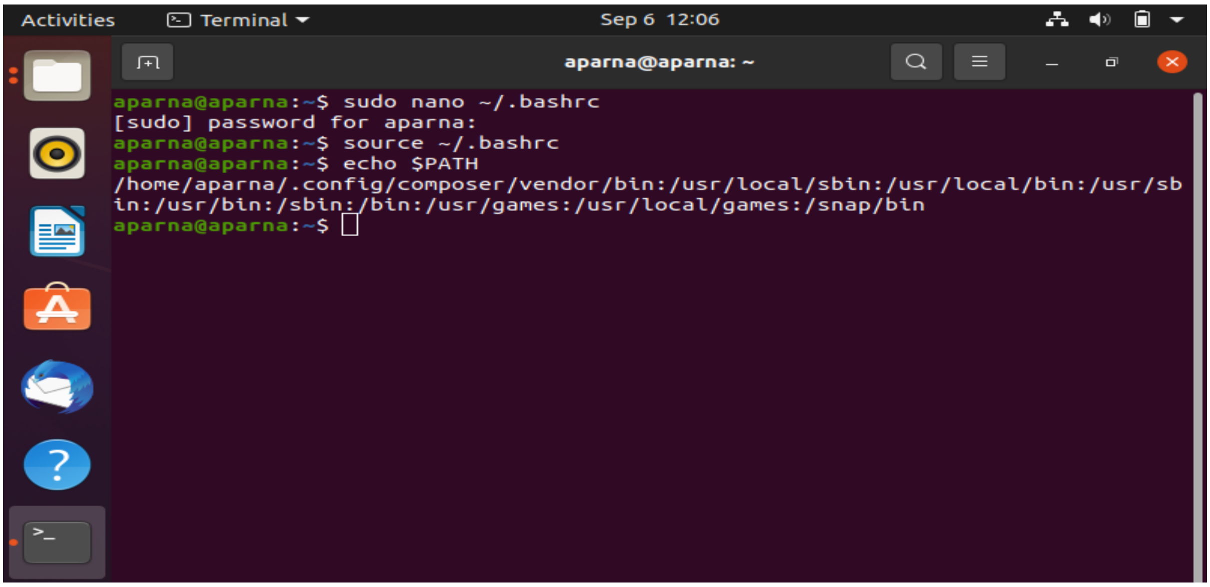
```
  GNU nano 4.8                /home/aparna/.bashrc                     Modified
# Add an "alert" alias for long running commands.  Use like so:
#   sleep 10; alert
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || ec>

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
export PATH="$HOME/.config/composer/vendor/bin:$PATH"


^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell
```

- Now reload your bashrc configuration using the source command.

  ```
  source ~/.bashrc
  ```

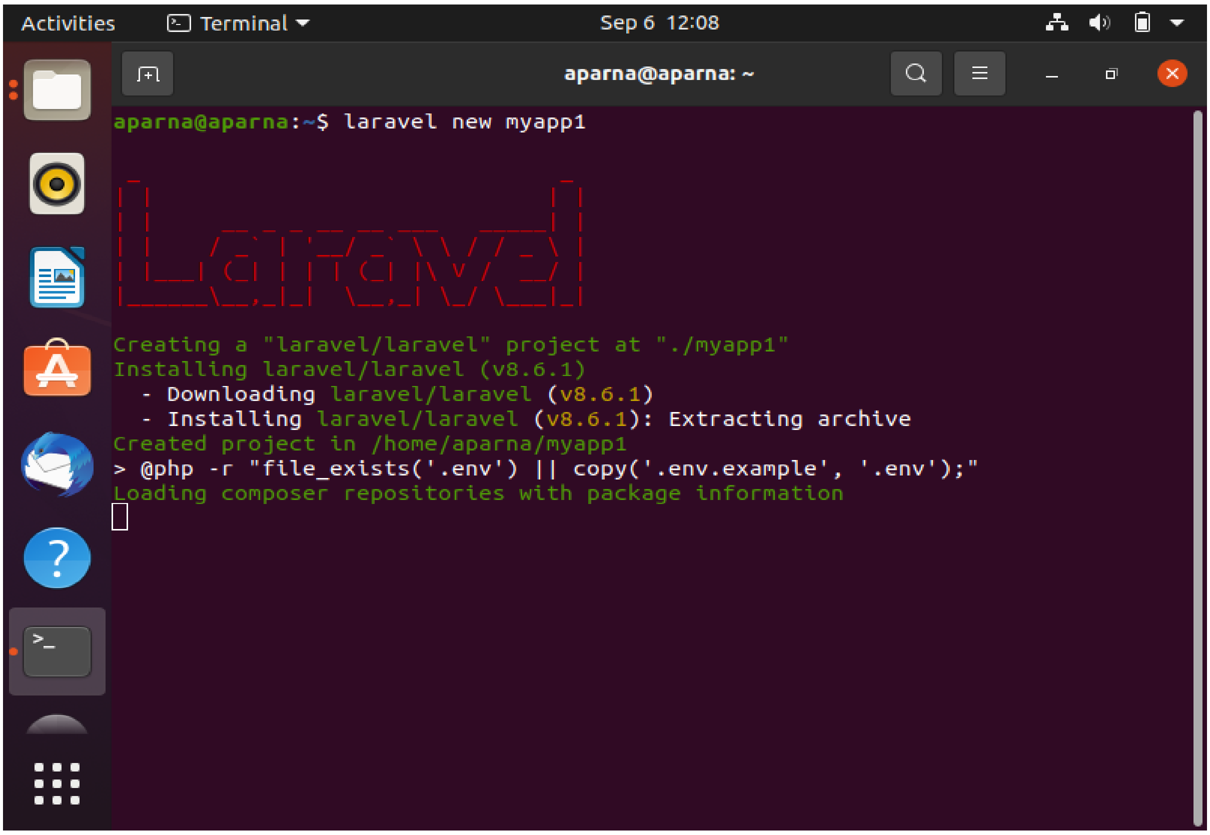
- Now echo $PATH. It will return your “Bin” directory path for the Composer package.

  ```
  echo $PATH
  ```



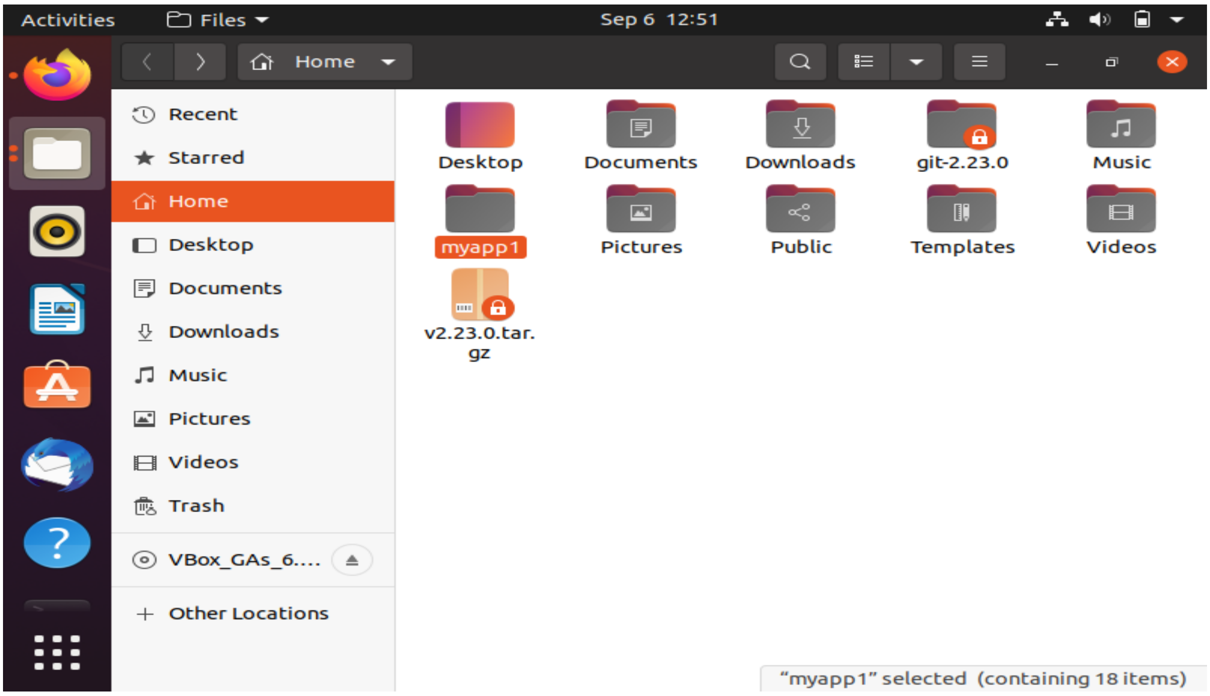
```
aparna@aparna:~$ sudo nano ~/.bashrc
[sudo] password for aparna:
aparna@aparna:~$ source ~/.bashrc
aparna@aparna:~$ echo $PATH
/home/aparna/.config/composer/vendor/bin:/usr/local/sbin:/usr/local/bin:/usr/sb
in:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
aparna@aparna:~$ 
```

- The ‘bin’ directory for the composer packages has been added to the $PATH environment variable. And as a result, you can use the command ‘laravel’ to start and create a new project. Now go ahead and type Laravel new then your project name to start a new Laravel project
  laravel new myapp1



```
aparna@aparna:~$ laravel new myapp1

 _                               _
| |                             | |
| |     __ _ _ __ __ ___   _____| |
| |    / _` | '__/ _` \ \ / / _ \ |
| |___| (_| | | | (_| |\ V /  __/ |
|______\__,_|_|  \__,_| \_/ \___|_|

Creating a "laravel/laravel" project at "./myapp1"
Installing laravel/laravel (v8.6.1)
  - Downloading laravel/laravel (v8.6.1)
  - Installing laravel/laravel (v8.6.1): Extracting archive
Created project in /home/aparna/myapp1
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
```

- Here you can see the installation of my new project myapp1 finished. You can also see inside my home directory a new directory has been created with my project name.



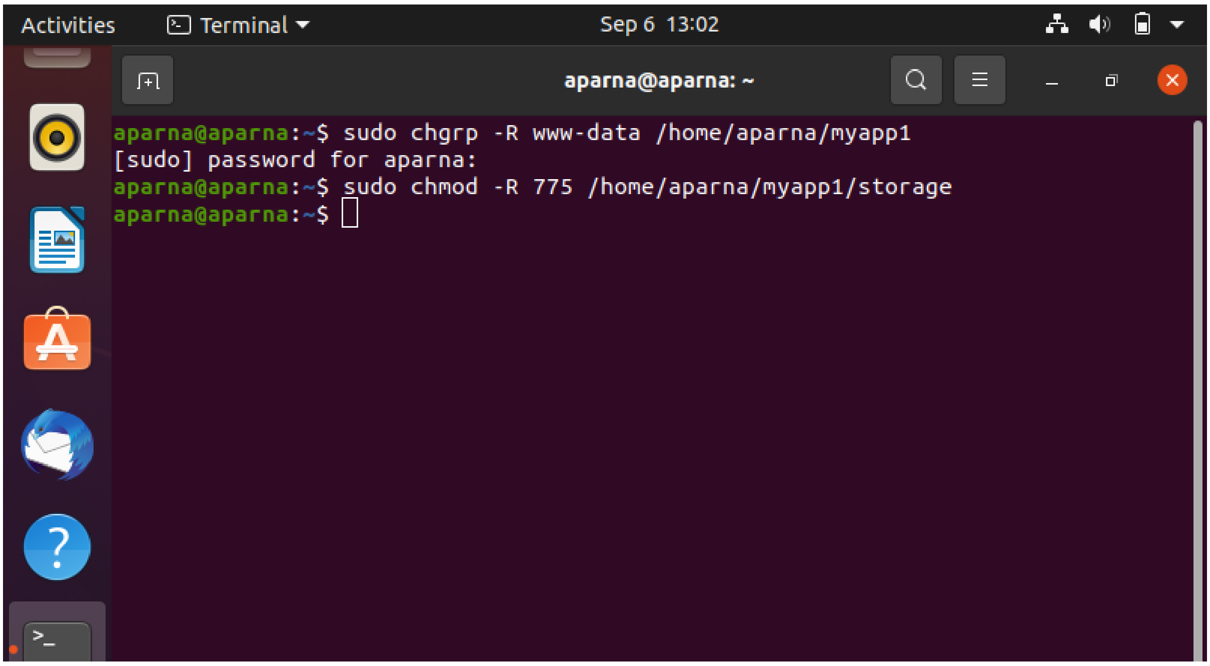
“myapp1” selected (containing 18 items)

Step 5 – Finally Configure Apache for Laravel and test it.

- First, add your project directory to www-data group use the following command

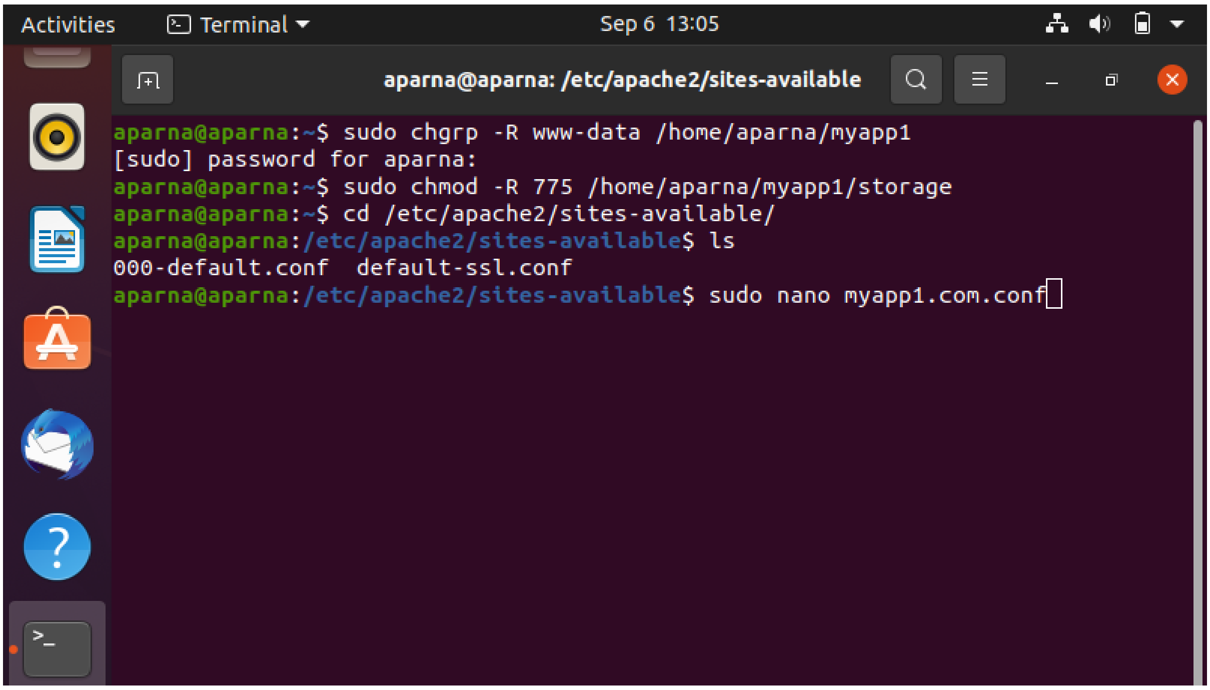
  `sudo chgrp -R www-data /home/aparna/myapp1`

- Also, you need to change access permission 775 of the storage directory under your project. So, go ahead and use the following command.
  `sudo chmod -R 775 /home/aparna/myapp1/storage`



```
aparna@aparna:~$ sudo chgrp -R www-data /home/aparna/myapp1
[sudo] password for aparna:
aparna@aparna:~$ sudo chmod -R 775 /home/aparna/myapp1/storage
aparna@aparna:~$ 
```

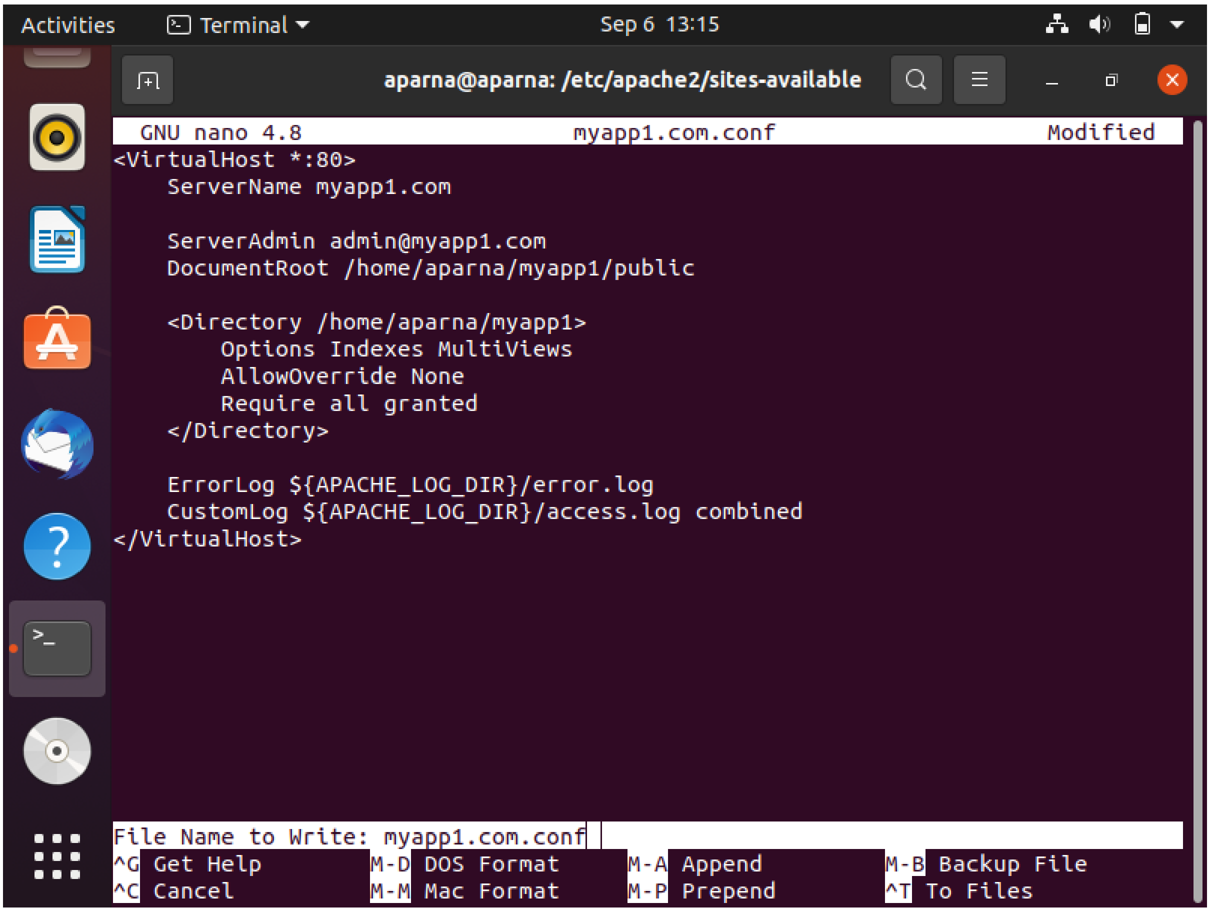
- Now create an apache vhost configuration go to the following directory and create a vhost config file using nano file editor.
  `cd /etc/apache2/sites-available/`
  `sudo nano myapp1.com.conf`



```
aparna@aparna:~$ sudo chgrp -R www-data /home/aparna/myapp1
[sudo] password for aparna:
aparna@aparna:~$ sudo chmod -R 775 /home/aparna/myapp1/storage
aparna@aparna:~$ cd /etc/apache2/sites-available/
aparna@aparna:/etc/apache2/sites-available$ ls
000-default.conf  default-ssl.conf
aparna@aparna:/etc/apache2/sites-available$ sudo nano myapp1.com.conf
```

And paste the following line inside the file.

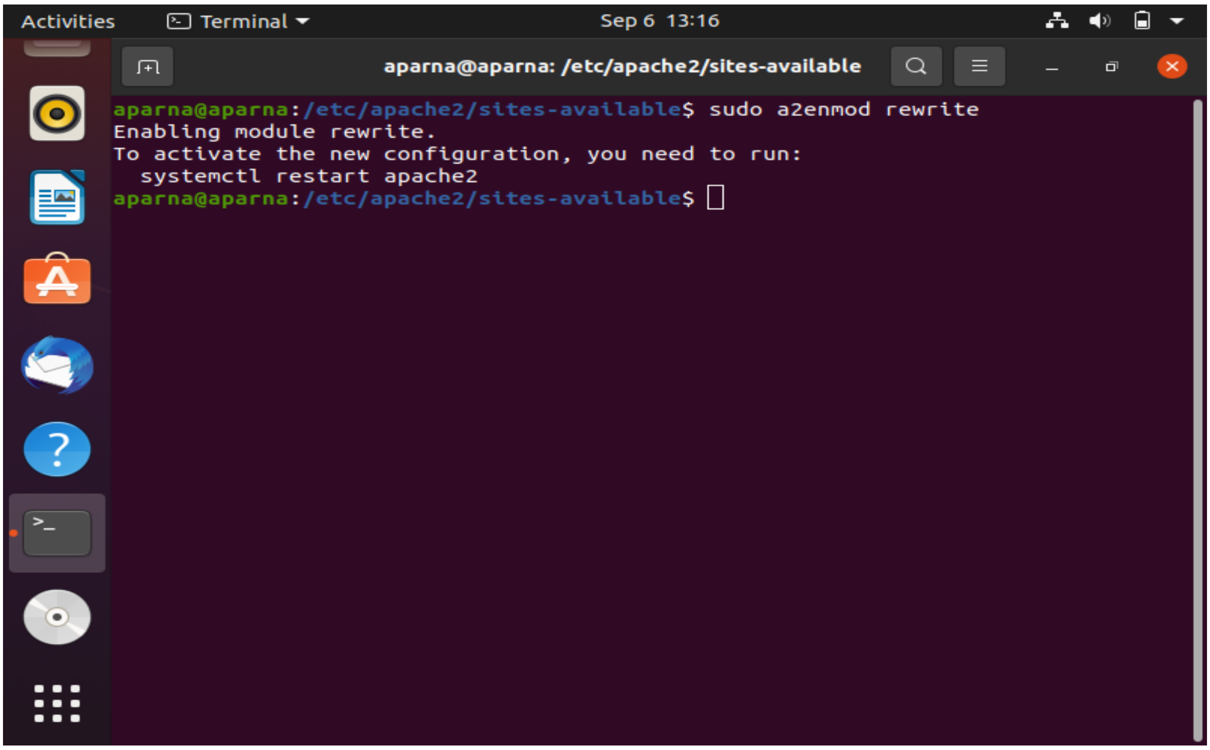
```
<VirtualHost *:80>
    ServerName myapp1.com

    ServerAdmin admin@myapp1.com
    DocumentRoot /home/aparna/myapp1/public

    <Directory /home/aparna/myapp1>
        Options Indexes MultiViews
        AllowOverride None
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```



```
  GNU nano 4.8                  myapp1.com.conf                       Modified
<VirtualHost *:80>
    ServerName myapp1.com

    ServerAdmin admin@myapp1.com
    DocumentRoot /home/aparna/myapp1/public

    <Directory /home/aparna/myapp1>
        Options Indexes MultiViews
        AllowOverride None
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

File Name to Write: myapp1.com.conf
^G Get Help          M-D DOS Format       M-A Append           M-B Backup File
^C Cancel            M-M Mac Format       M-P Prepend          ^T To Files
```

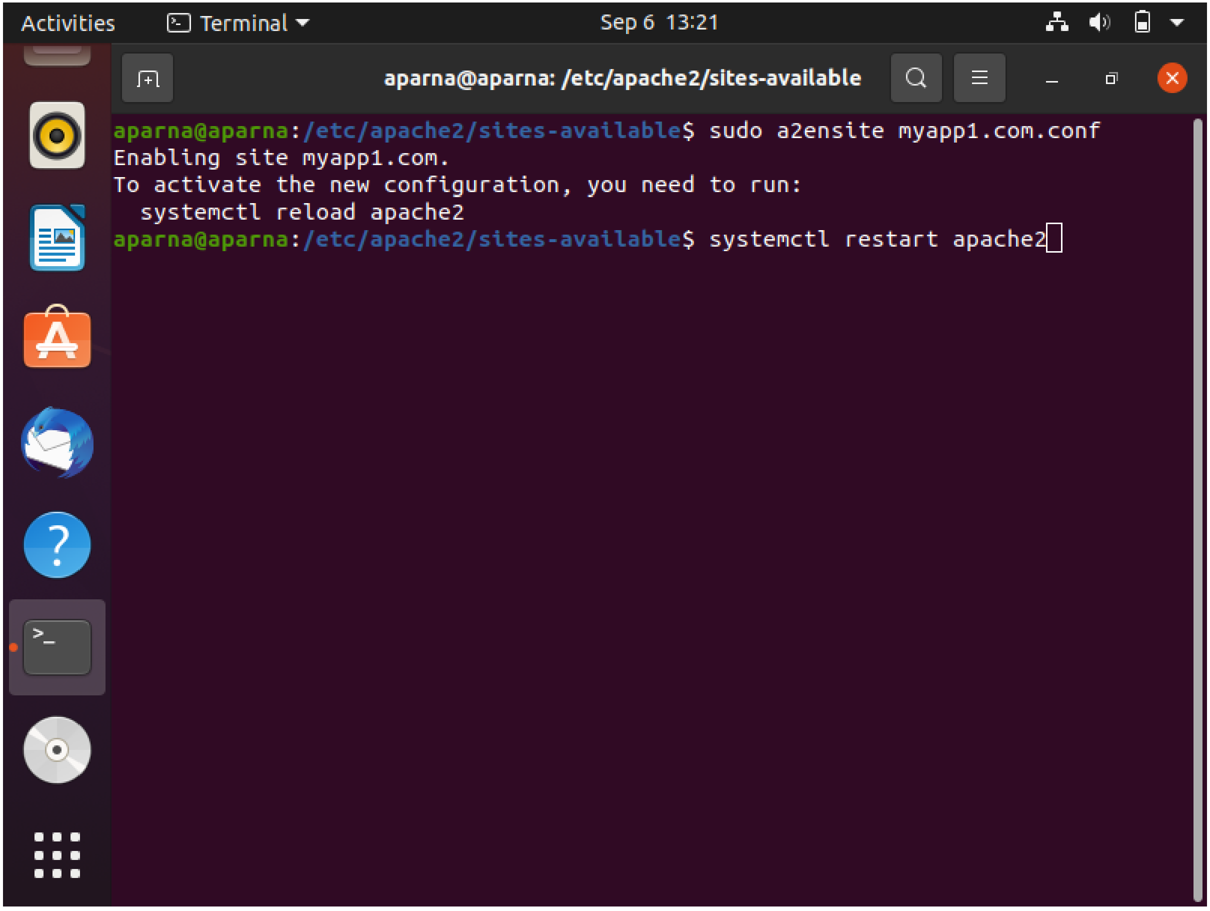
- Now enable mod rewrite for apache2 just type

  ```
  sudo a2enmod rewrite
  ```



```
aparna@aparna:/etc/apache2/sites-available$ sudo a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
  systemctl restart apache2
aparna@aparna:/etc/apache2/sites-available$ 
```

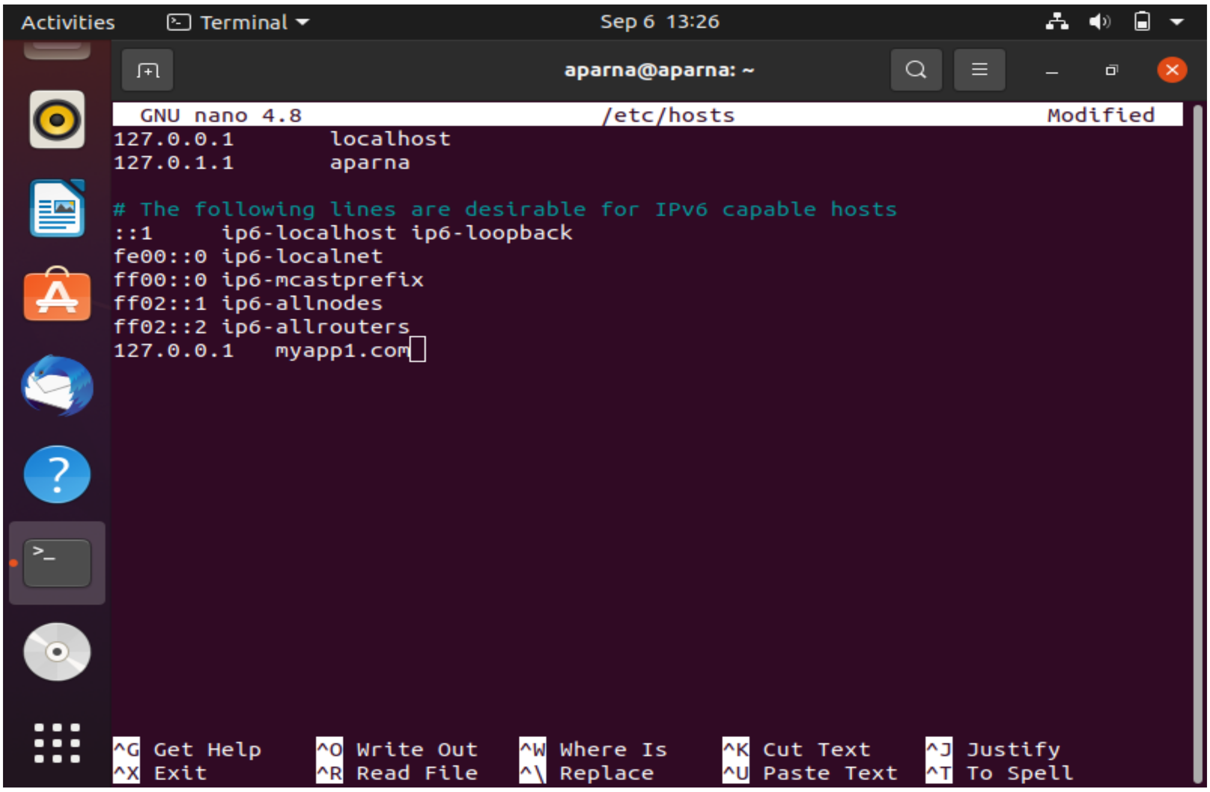
- Now enable your site, just type
  sudo a2ensite myapp1.com.conf
- Finally, Restart the apache service, type
  systemctl restart apache2



```
aparna@aparna:/etc/apache2/sites-available$ sudo a2ensite myapp1.com.conf
Enabling site myapp1.com.
To activate the new configuration, you need to run:
  systemctl reload apache2
aparna@aparna:/etc/apache2/sites-available$ systemctl restart apache2
```

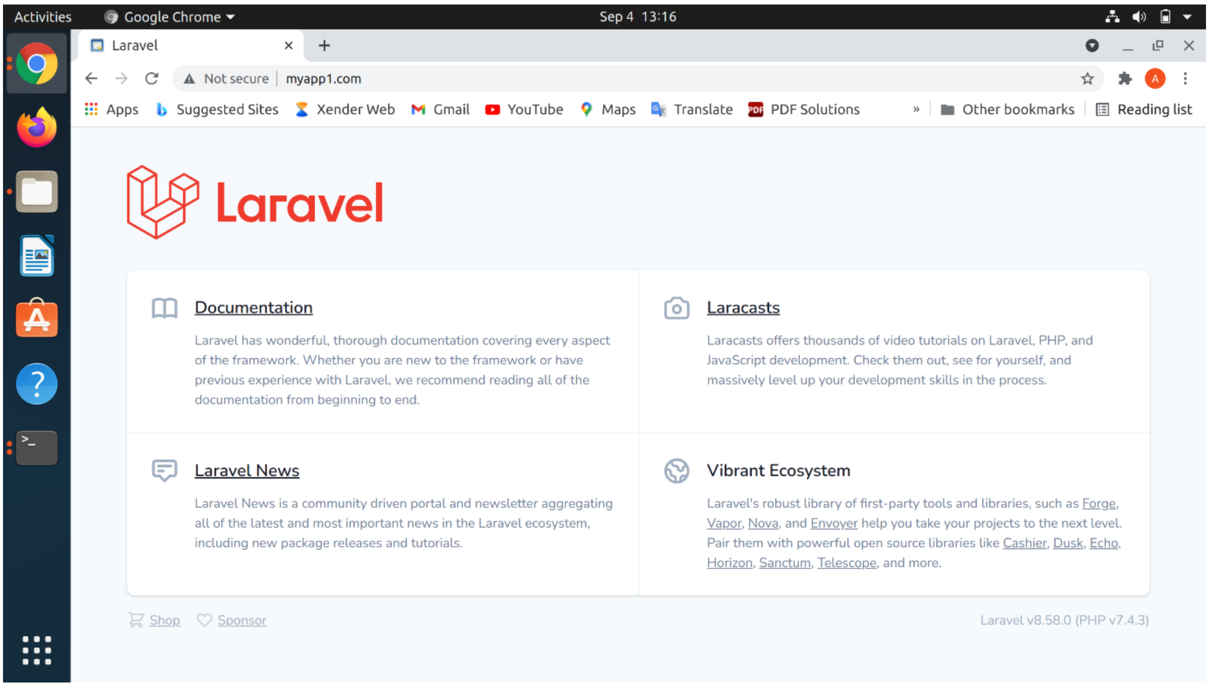
- As you are in a local environment you need a local dns resolver for your site. Go ahead and edit /etc/hosts file, add a dns record for your site then save the file.

```
sudo nano /etc/hosts
...
127.0.0.1    myapp1.com
...
```



```
  GNU nano 4.8                    /etc/hosts                          Modified
127.0.0.1       localhost
127.0.1.1       aparna

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
127.0.0.1    myapp1.com









^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell
```

- Now get back to the web browser and open a tab then type your project hostname.



Documentation

Laravel has wonderful, thorough documentation covering every aspect of the framework. Whether you are new to the framework or have previous experience with Laravel, we recommend reading all of the documentation from beginning to end.

Laracasts

Laracasts offers thousands of video tutorials on Laravel, PHP, and JavaScript development. Check them out, see for yourself, and massively level up your development skills in the process.

Laravel News

Laravel News is a community driven portal and newsletter aggregating all of the latest and most important news in the Laravel ecosystem, including new package releases and tutorials.

Vibrant Ecosystem

Laravel's robust library of first-party tools and libraries, such as Forge, Vapor, Nova, and Envoyer help you take your projects to the next level. Pair them with powerful open source libraries like Cashier, Dusk, Echo, Horizon, Sanctum, Telescope, and more.

Shop Sponsor

Laravel v8.58.0 (PHP v7.4.3)