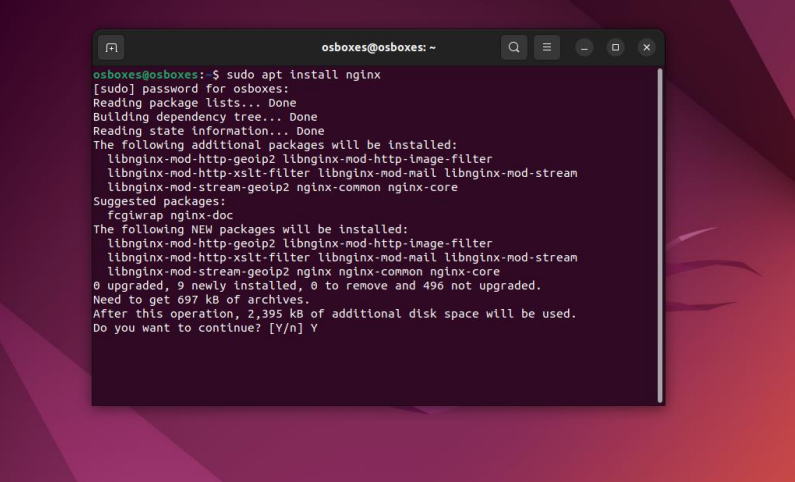


Step 1: Install Nginx

A screenshot of a Linux desktop environment. The background is a dark purple wallpaper with a stylized bird or wing design. On the left side, there is a vertical dock with several application icons: a Firefox browser icon, a LibreOffice Writer icon, a file manager icon, a terminal icon, a software center icon, a help icon, and a power button icon. At the top of the screen, there is a status bar showing the date and time as 'Jul 30 02:37' and some system icons. In the center of the screen, a terminal window is open. The terminal has a title bar that says 'osboxes@osboxes: ~'. Inside the terminal, the following text is displayed:

```
osboxes@osboxes:~$ sudo apt install nginx
[sudo] password for osboxes:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libnginx-mod-http-geoip2 libnginx-mod-http-image-filter
  libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
  libnginx-mod-stream-geoip2 nginx-common nginx-core
Suggested packages:
  fcgiwrap nginx-doc
The following NEW packages will be installed:
  libnginx-mod-http-geoip2 libnginx-mod-http-image-filter
  libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
  libnginx-mod-stream-geoip2 nginx nginx-common nginx-core
0 upgraded, 9 newly installed, 0 to remove and 496 not upgraded.
Need to get 697 kB of archives.
After this operation, 2,395 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```



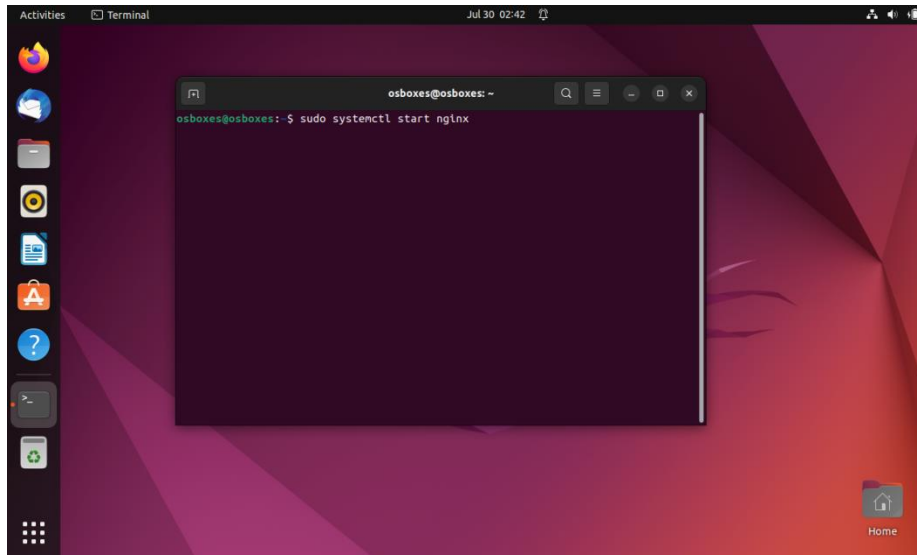
The screenshot shows a Linux desktop environment with a dark purple background. On the left side, there is a vertical dock containing several application icons: a web browser, a file manager, a terminal, and a recycling bin. At the top of the screen, there is a status bar with the text 'Activities', 'Terminal', and a clock showing 'Jul 30 02:37'. In the center of the screen, a terminal window is open, displaying the command 'sudo apt install nginx' and its output. The output shows the password prompt, the package lists, the dependency tree, the state information, the additional packages to be installed, the suggested packages, the new packages to be installed, the disk space requirements, and the confirmation to continue.

```
osboxes@osboxes: ~
osboxes@osboxes:~$ sudo apt install nginx
[sudo] password for osboxes:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libnginx-mod-http-geoip2 libnginx-mod-http-image-filter
  libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
  libnginx-mod-stream-geoip2 nginx-common nginx-core
Suggested packages:
  fcgiwrap nginx-doc
The following NEW packages will be installed:
  libnginx-mod-http-geoip2 libnginx-mod-http-image-filter
  libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
  libnginx-mod-stream-geoip2 nginx nginx-common nginx-core
0 upgraded, 9 newly installed, 0 to remove and 496 not upgraded.
Need to get 697 kB of archives.
After this operation, 2,395 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Step 2: Start Nginx

After Nginx is installed, start the Nginx service with the following command:

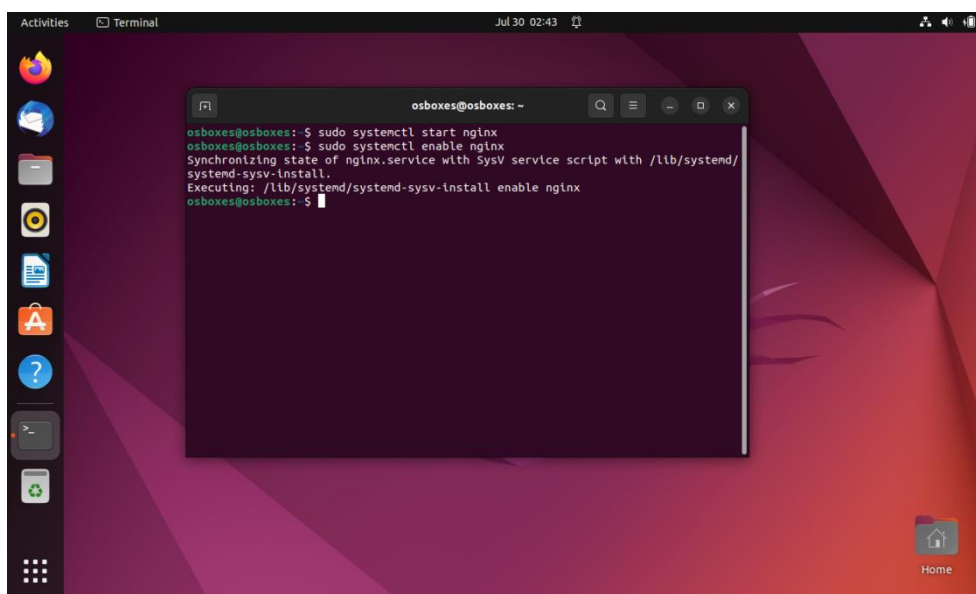
sudo systemctl start nginx



Step 3: Enable Nginx

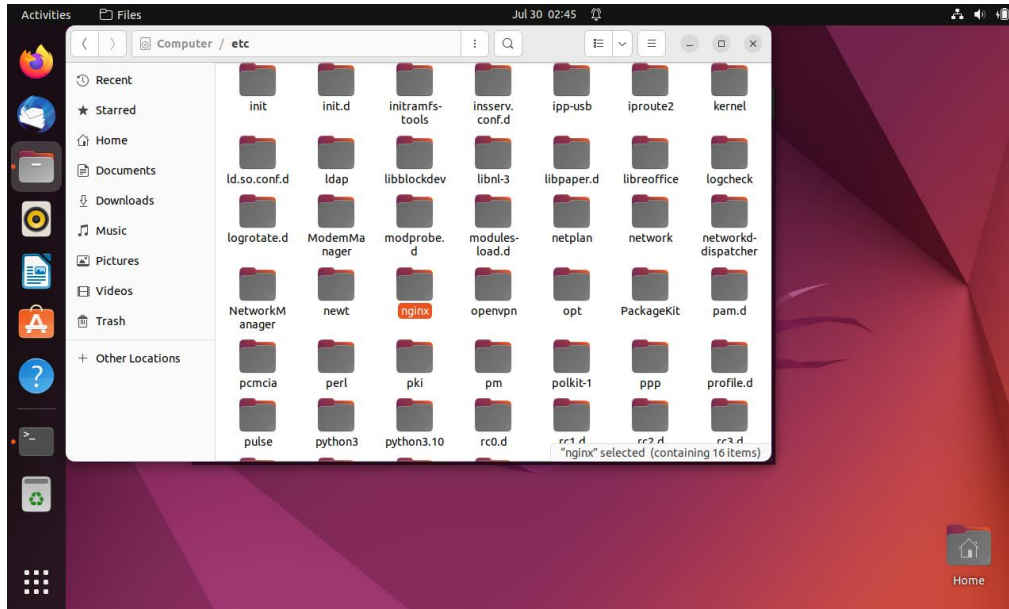
To ensure that Nginx starts automatically when the system boots, run the following command:

sudo systemctl enable nginx



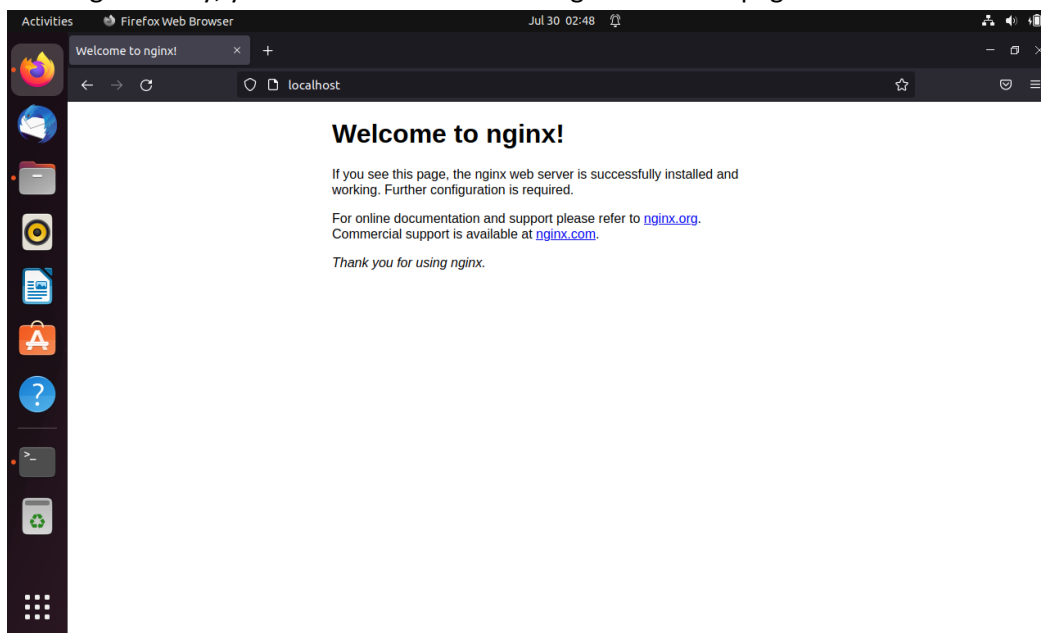
Step 4: Verify Nginx Installation

You can check if Nginx is installed by browsing to the Nginx directory. Open the Files, go to "Other Locations," then navigate to "Computer" and finally "etc." There, you should find the Nginx folder.



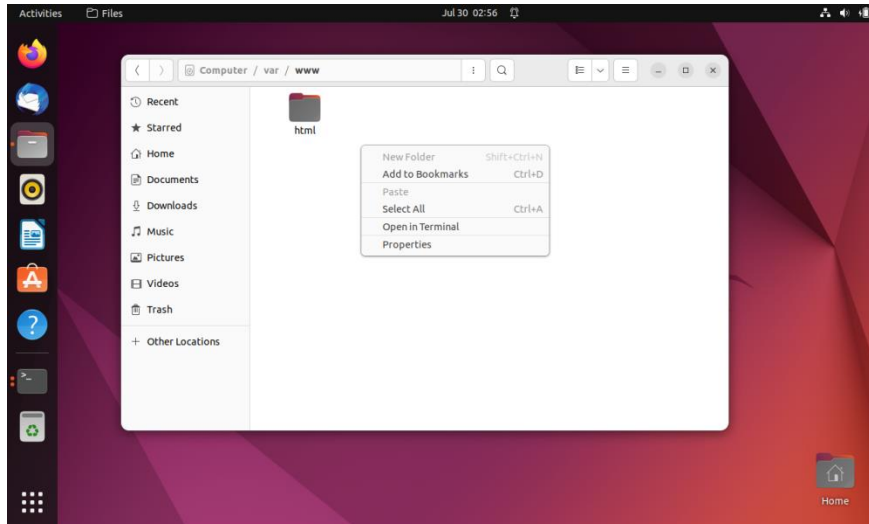
Step 5: Verify Nginx is Running

Open your web browser and type **localhost** or **http://localhost** into the address bar. If Nginx is working correctly, you should see the default Nginx welcome page.

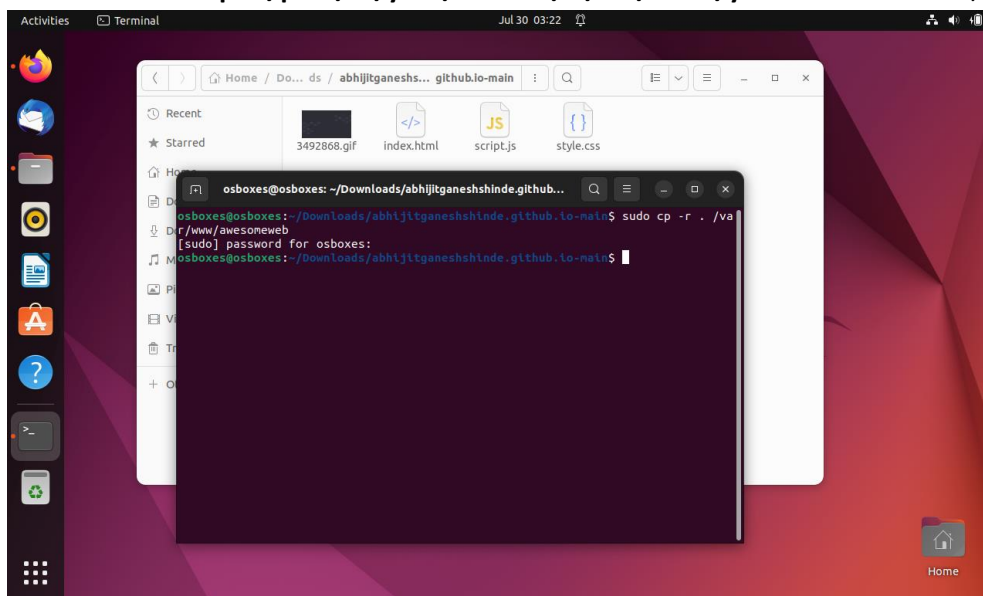


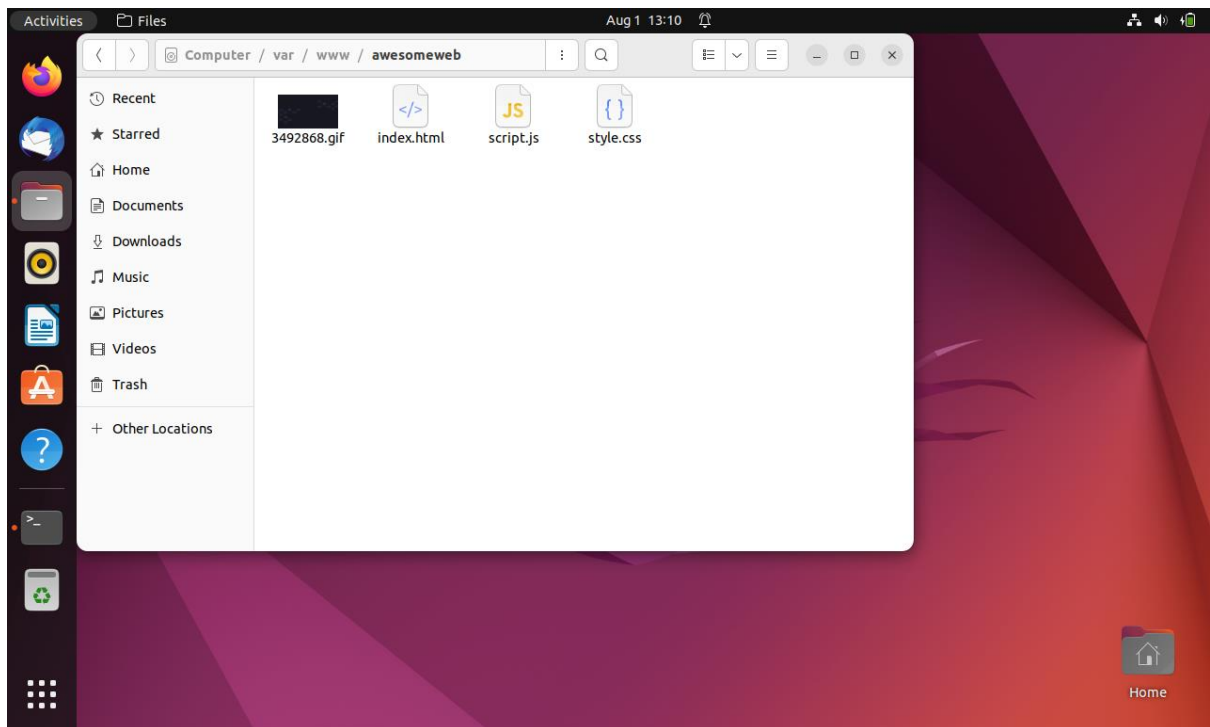
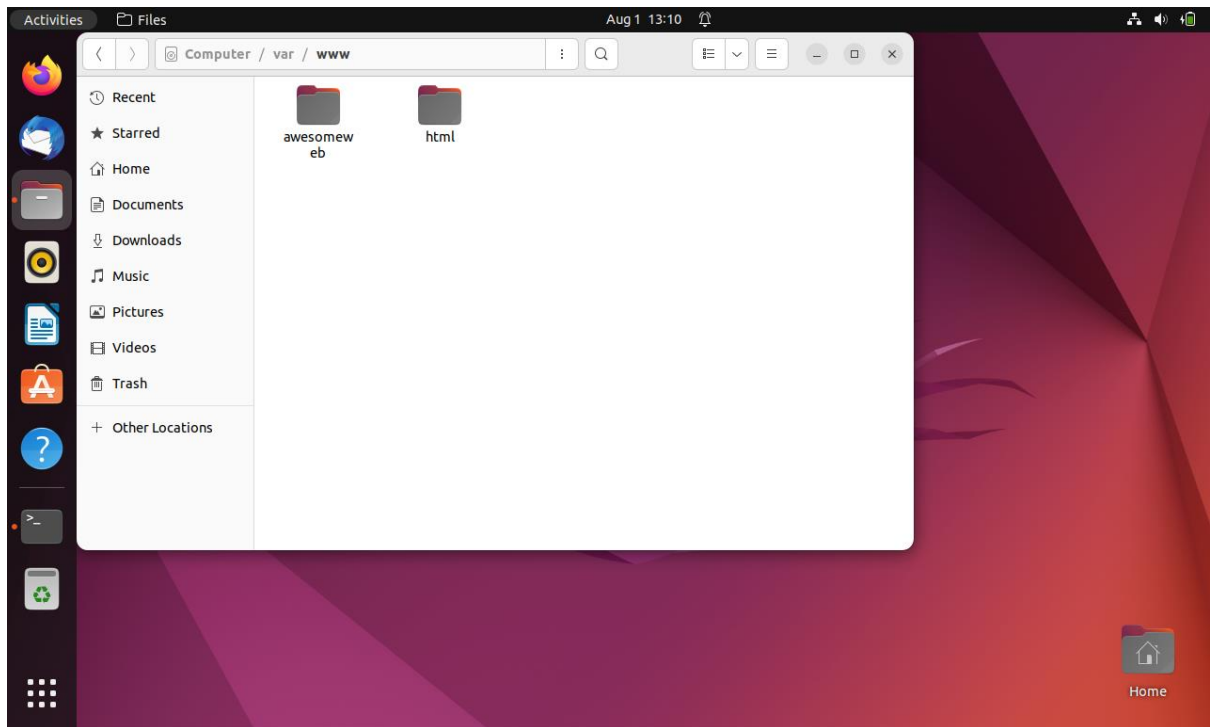
Step 6: Deploy Your Own Website

To deploy your own website, go to "Files" => "Other Locations" => "Computer" => "var" => "www." Create a folder here and add your website's code inside it.



Alternatively, you can copy your website files to the web root directory using the following command: **sudo cp -r /path/to/your/website/* /var/www/your-website-folder/**





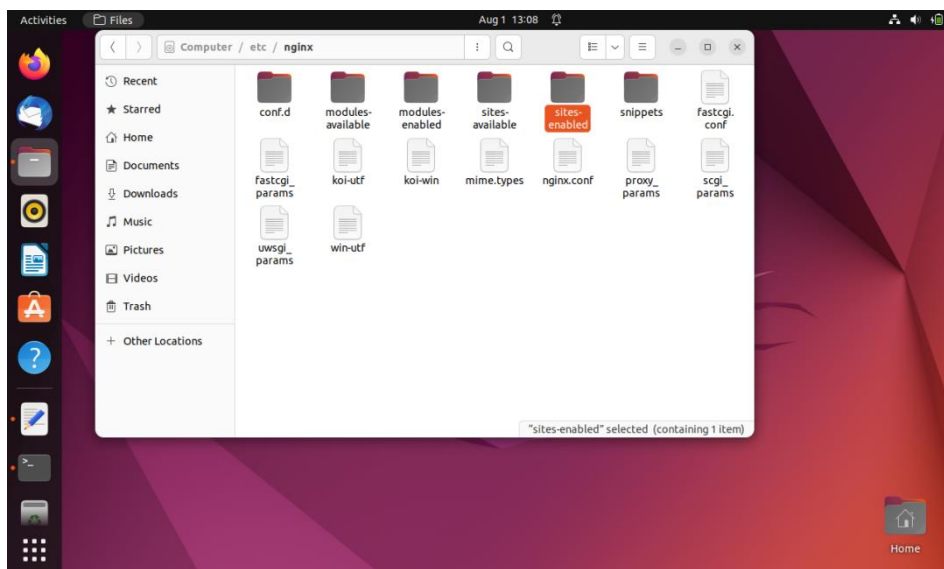
Step 7: Configure Nginx for Your Website

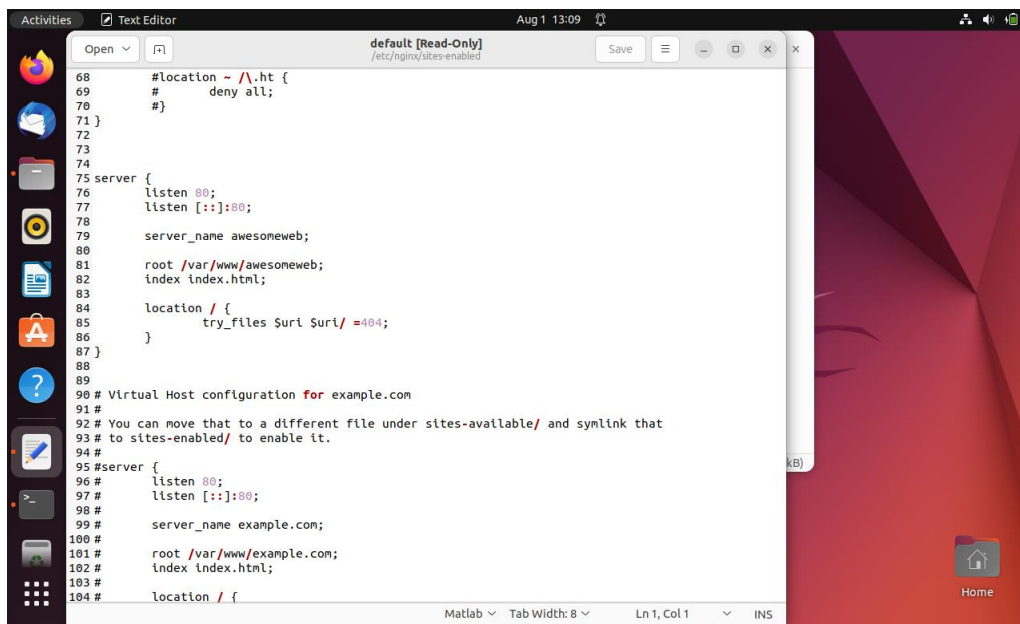
Go to the Nginx configuration directory by navigating to "Files" => "Other Locations" => "Computer" => "etc" => "nginx" => "sites-enabled."

Open the default file using a text editor, and add server details for your website. An example configuration might look like this:

```
server {  
  
    listen 80;  
  
    listen [::]:80;  
  
    server_name awesomeweb;  
  
  
    root /var/www/ awesomeweb;  
  
    index index.html;  
  
  
    location / {  
  
        try_files $uri $uri/ =404;  
  
    }  
  
}
```

Replace awesomeweb with your desired domain or server name and /var/www/awesomeweb with the actual path to your website files.

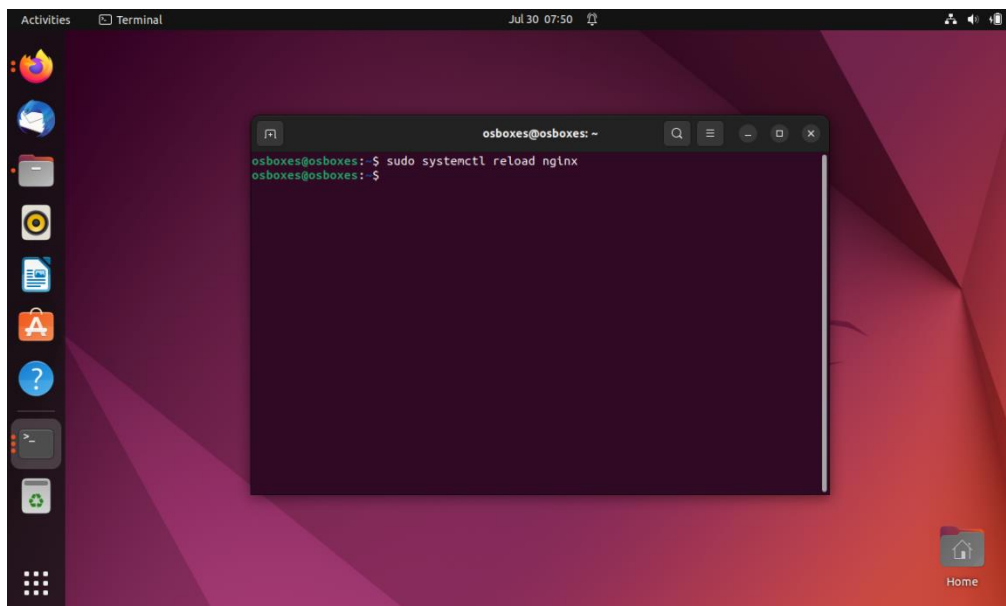




Step 8: Reload Nginx Configuration

After making changes to the Nginx configuration, reload Nginx to apply the changes:

sudo systemctl reload nginx



Step 9: View Your Website

Now, open your web browser and visit your website by entering the domain or server name you configured in the Nginx file (ex. `http://awesomeweb`). You should see your website content displayed in the browser.

