## Table of Contents

Author....1
Audience....1
Prerequisites....1
Method....2
Activate Button....2
Apache Start Button....4
Apache Stop Button....4
MySql Start Button....5
MySql Stop Button....5
Installation....5
Appendix....6
Bypassing sudo password prompt. Ubuntu 18.04....6
Recovery with a boot USB....7
The Method to bypass sudo password prompt....7
Installing ImageMagick-7 on Ubuntu 18.04....7
Disclaimer....7
Credits....7

# Author

Zephyrinus Evarist Rodrigues Date: 27/11/2020

# Audience

This document is for those who run Ubuntu or other debian linux operating systems and want a simple GUI application to monitor and control the Apache and MySQL servers running on the local machine (localhost). It uses bash shell, ImageMagick-6 and HTML with javascript. ImageMagick does it’s magic and acts as the middleware between shell script and HTML.

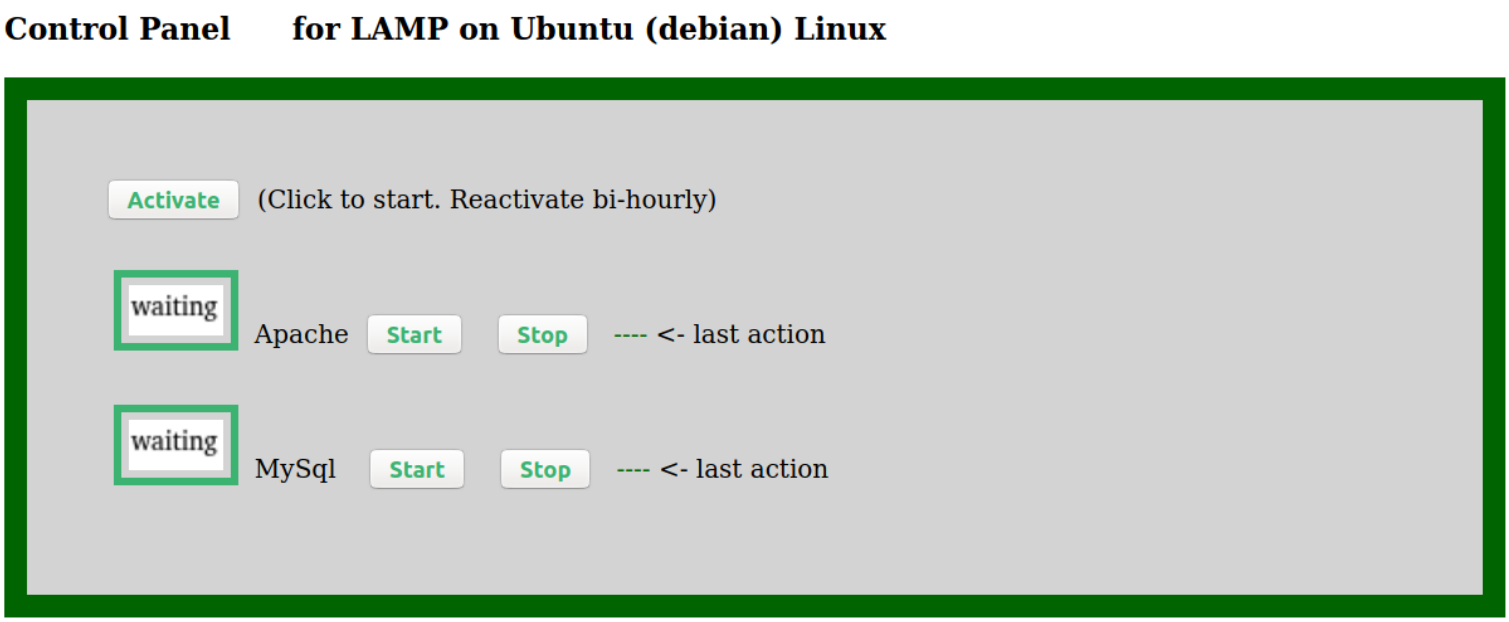
# Prerequisites

- Ubuntu or other debian based linux operating system.
- Firefox web browser (tested) or any other suitable javascript enabled web browser.
- ImageMagick-6 application . This ships as default with Ubuntu 18.04 . It would probably work with ImageMagick-7 but has not been tested with this version. See more in the Appendix here.
- Secure ImageMagick-6 by editing the policy.xml
- Installation of LAMP on your Linux machine. Here is how
- Granting the logged in user sudo permissions without password prompt (optional)

# Method

To bring up the the application, click on the file app_zj1.html.

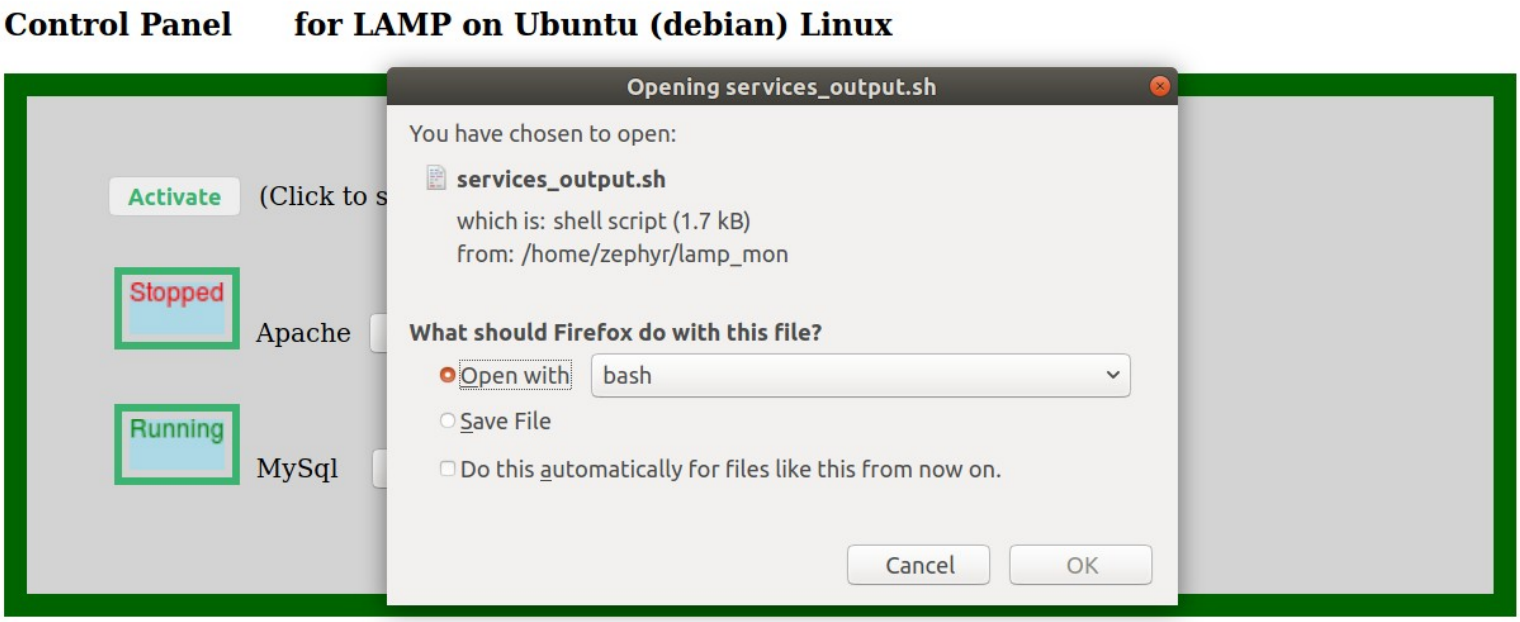
This should bring up the the following page in the web browser.




# Activate Button

FIRST

'Activate' on-click calls services_output.sh. This outputs to ps1.txt. Javascript deactivates the button on-click . For the bash script to run the user must click on the “ Run Bash” provided by the Firefox web browser. See Image below.




If you have not bypassed the sudo password prompt, you will have to enter your password each time you call the script.

NEXT

services_output.sh process script updates every two minutes and runs for 2 hours.

NEXT

Check if apache2 daemon is running. If running javascript to output Apache 'Running' else javascript to output 'Apache Stopped'

How is this done?

In services_output.sh you have

# checks that apache is running and returns 1 as count if running or 0 if it has stopped.

cat /home/zephyr/Desktop/batch/ps1.txt | grep apache2 | grep root | grep -c start > apache_count.txt;

If the apache daemon has been started apache_count.txt should have the value of 1 in it.

The count “1” from count.txt gets read into a variable. This variable gets assigned to a text variable containing “Running”

The “Running” text variable is then output to an image file using the ImageMagick-6 convert utility.

This gets displayed in the App window using html (and javascript).

Else

count.txt has a value of 0 which gets assigned to a text variable containing “Stopped”

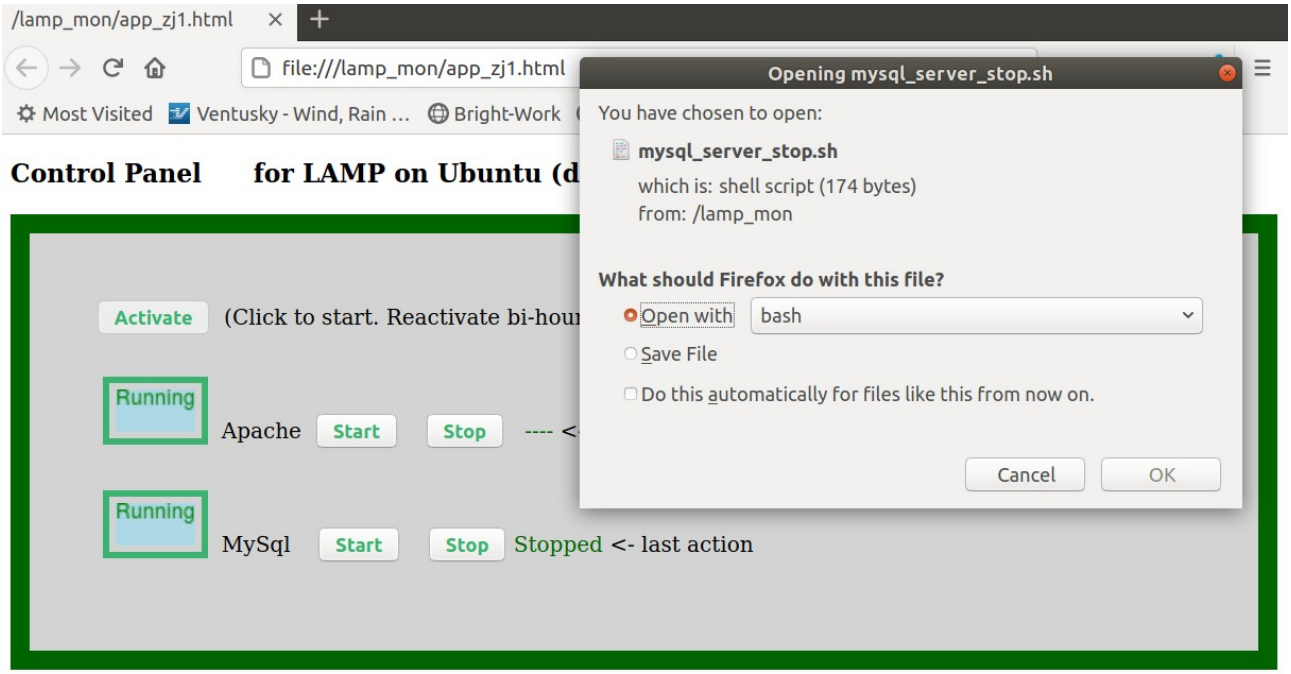
In the HTML page which is used to display status of the Apache service, javascript is used to update the image every 30 secs.

AND

Check if mysqld daemon is running. If running javascript to output  MySql 'Running' else JavaScript to output 'MySql Stopped'  The script is similar to that for Apache.

## Apache Start Button

Clicking on the Apache Start button bring up the Firefox dialogue box saying “You have chosen to open apache_server.sh. What should Firefox do  with this file?” The  “Open with bash” option should be selected by default. If not select this option.  See Image below:




On clicking “Okay” The script will run. If you haven’t bypassed the sudo password prompt (see section “ Bypassing sudo password prompt”) , you will have to enter this password.

In the Control Panel application window the Apache Button  status “<-last action” will have changed from “----” or “Stopped”to “Started”. The Apache status image should change to “Running “ if it was displaying “Stopped” previously.

## Apache Stop Button

Clicking on the Apache Stop button bring up the Firefox dialogue box saying “You have chosen to open apache_server.sh. What should Firefox do  with this file?” The  “Open with bash” option should be selected by default. If not select this option.

On clicking “Okay” The script will run. If you haven’t bypassed the sudo password prompt (see section “ Bypassing sudo password prompt”) , you will have to enter this password.

In the Control Panel application window the Apache Button  status “<-last action” will have changed from “----” or “Running” to “Stopped”. The Apache status image should change to “Stopped“ if it was displaying “Running” previously.

## MySql Start Button

This works the same way as the as the Apache Start Button.

## MySql Stop Button

This works the same way as the as the Apache Stop Button.

# Installation

- Create a directory /lamp_mon at the top most (root) directory level with Terminal and command line.

**sudo mkdir /lamp_mon**

- Grant ownership of the directory to the currently logged in user , (account that the you usually login with).

**sudo chown $USER -R /lamp_mon**

- Copy the following files into this directory after umzipping. There seven files in total.

apache_server_start.sh

apache_server_stop.sh

mysql_server_start.sh

mysql_server_stop.sh

services_output.sh

app_zj1.html

waitng.png

- Secure ImageMagick-6

By default ImageMagick-6 on Ubuntu 18.04 is locked down for security by the policy.xml to prevent file to image reads.

Edit the policy.xml with sudo privileges using the nano editor.

To do this open Terminal and run

**sudo nano /etc/ImageMagick-6/policy.xml**

Change the path if policy.xml is at different location. The above is from a standard Ubuntu 18.04 install

Find

<policy domain="path" rights="none" pattern="@*"/>

and rem it out as follows

```
<!--  <policy domain="path" rights="none" pattern="@*"/> -->
```

Then add the following below to block root directories. Copy and paste into nano. If you have additional root directories you can add these.

<policy domain="path" rights="none" pattern="@/bin*"/>

<policy domain="path" rights="none" pattern="@/boot*"/>

<policy domain="path" rights="none" pattern="@/cdrom*"/>

<policy domain="path" rights="none" pattern="@/dev*"/>

<policy domain="path" rights="none" pattern="@/etc*"/>

<policy domain="path" rights="none" pattern="@/home*"/>

<policy domain="path" rights="none" pattern="@/lib*"/>

<policy domain="path" rights="none" pattern="@/lib64*"/>

<policy domain="path" rights="none" pattern="@/lost+found*"/>

<policy domain="path" rights="none" pattern="@/media*"/>

<policy domain="path" rights="none" pattern="@/mnt*"/>

<policy domain="path" rights="none" pattern="@/opt*"/>

<policy domain="path" rights="none" pattern="@/proc*"/>

<policy domain="path" rights="none" pattern="@/root*"/>

<policy domain="path" rights="none" pattern="@/run*"/>

<policy domain="path" rights="none" pattern="@/sbin*"/>

<policy domain="path" rights="none" pattern="@/snap*"/>

Press Ctrl +X to exit, then “Y” to save and confirm write to policy.xml file.

If you make a mistake just press Ctrl +X and “N” to discard changes and start again.

# Appendix

## Bypassing sudo password prompt. Ubuntu 18.04

This is the only method that works– but make sure you have Ubuntu Boot USB in case you make a typo because you will get locked out of sudo. A backup admin account does not help as for some reason, the $USER policy seems to apply to all accounts in the admin group.

### Recovery with a boot USB

Assuming competency with the nano text editor.

After booting up from the USB drive, click on the Harddisk in File manager to create a mount point. Open Terminal. Type **dh -f**

This gives you a list of devices and their mount points.

**cd** to mount point of the boot harddrive. Then **cd** /etc/sudoers.d

next, **sudo nano** $user where  $user is a the policy file with the logged in user's name.  Correct the typo, save the file and reboot.

### The Method to bypass sudo password prompt

The preferred way to grant individual (or group) permissions would be to add files under `/etc/sudoers.d`

This separates local changes from the default policy and saves time in case the distribution sudoers file changes.

To make the currently logged in user  a sudoer and to make `sudo` not prompt them for a password, use

```
echo "$USER ALL=(ALL:ALL) NOPASSWD: ALL" | sudo tee /etc/sudoers.d/$USER
```

this will create a file called `/etc/sudoers.d/$USER` (where `$USER` is the username of the user that you were logged in as when you ran that command), making it clear which user is granted permission. You can also replace $USER with a valid user name.

## Installing ImageMagick-7 on Ubuntu 18.04

Version  does appear to be in the debian repository as yet. If you want to have a go install it you can find information on the [ImageMagick website.]

# Disclaimer

While every effort has been made to test the software and provide accurate information and instruction, no warranty either explicit or implied is given. Use it at your own risk.

# Credits

It has take me two and a half months to get put this bit of software together. I am very gratefully to all the contributors of the Open Source Community. You are really superb people. A big thank you to all on Stack Exchange, Stack Overflow, Jesin's blog, WW3 Schools, devcurry, geeksforgeeks,

coderwall, AskUbuntu.com – you are so many that have helped and I am sure to have missed out a good few. Thank you all very much in helping me in my efforts to becoming a coder :).