By MK MAHWETE

LAMP Stack Installation & Configuration

The following documentation will take you step by step on how to setup and install LAMP stack open-source software used to host and run web based applications. I will try to make this straight forward and clear for ease of use to follow through.

Why LAMP?

✅ Open-source & Free – No licensing fees.
✅ Stable & Reliable – Used by many websites worldwide.
✅ Highly Customizable
✅ Cross-platform Alternatives – Similar stacks exist

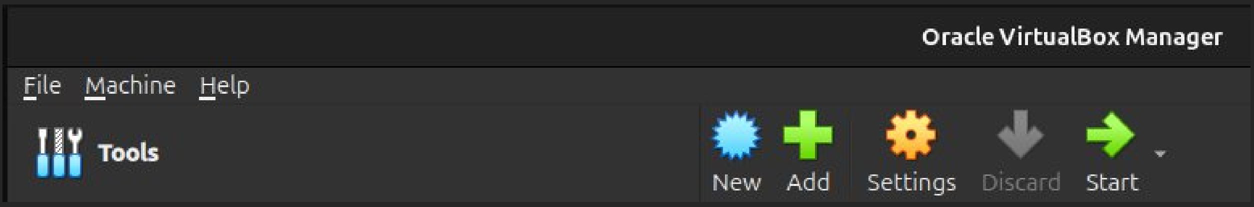
Step 1: Make sure that you have a Virtual Machine installed on your system

It's always advisable to run such applications on a virtual machine to avoid breaking down the main Operating System. VM enable us to run other OS for learning and experimenting with new tools and technologies.

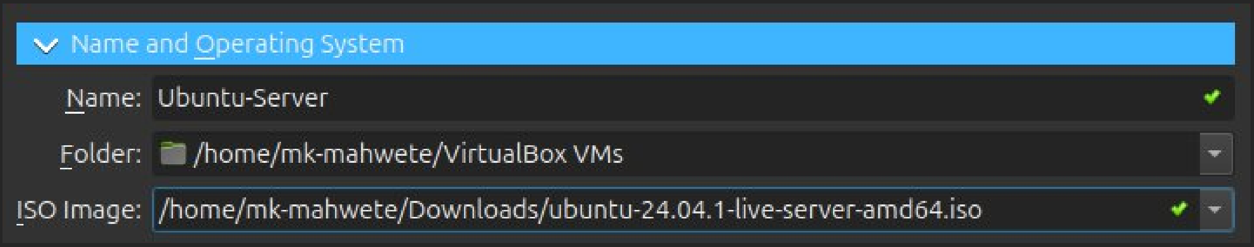
- I have a VirtualBox already installed on my machine. Get yours installed: https://www.virtualbox.org/wiki/Downloads
- Have Ubuntu Server.iso image disk downloaded: https://ubuntu.com/download/server

---

Step 2: Open the VM and create a new Virtual Machine



- Select the **blue icon** that looks like a start to create a new VM



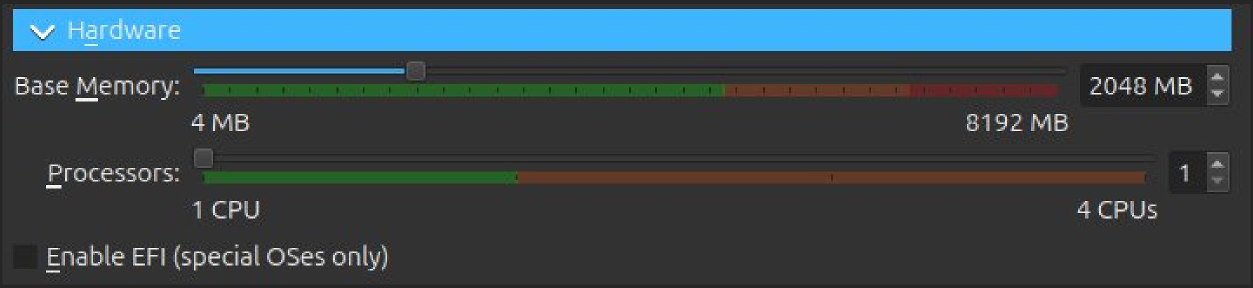
Name: Ubuntu-Server

Folder: /home/mk-mahwete/VirtualBox VMs

ISO Image: /home/mk-mahwete/Downloads/ubuntu-24.04.1-live-server-amd64.iso

- Give your new VM a name – in my case it's Ubuntu-Server to remember which VM is which

By MK MAHWETE

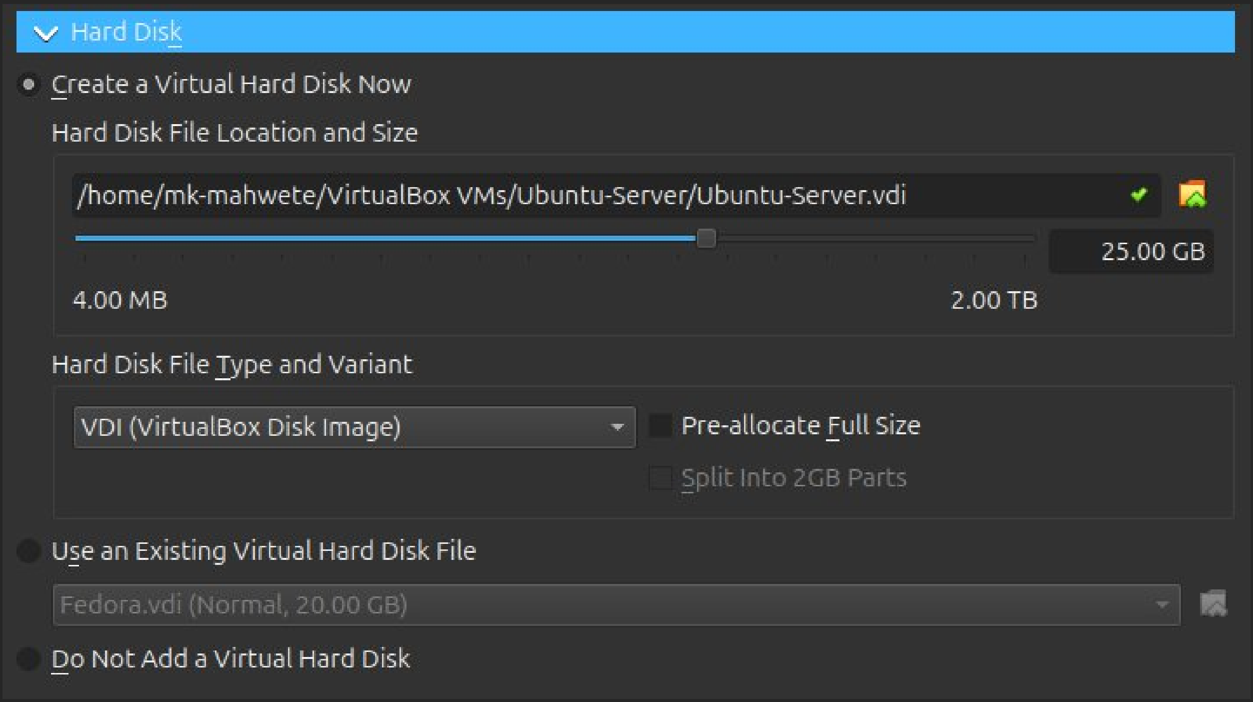

Hardware

Base Memory: 2048 MB (4 MB – 8192 MB)

Processors: 1 (1 CPU – 4 CPUs)

□ Enable EFI (special OSes only)

- Under Hardware, give your VM some Memory (RAM) of at least 2GB and 1 CPU


Hard Disk

● Create a Virtual Hard Disk Now

Hard Disk File Location and Size

/home/mk-mahwete/VirtualBox VMs/Ubuntu-Server/Ubuntu-Server.vdi

25.00 GB (4.00 MB – 2.00 TB)

Hard Disk File Type and Variant

VDI (VirtualBox Disk Image)

□ Pre-allocate Full Size

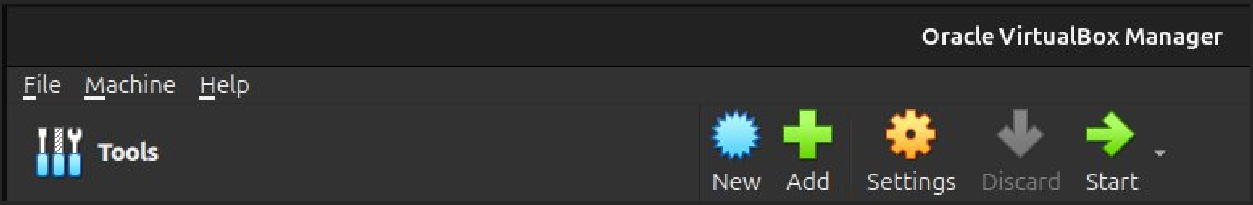
□ Split Into 2GB Parts

○ Use an Existing Virtual Hard Disk File

Fedora.vdi (Normal, 20.00 GB)

○ Do Not Add a Virtual Hard Disk

- Under Hard Disk, make sure to give you VM some HHD space for installation. Allocate at least 25GB upwards
- Then click **Finish** once you are done



- Now what left is so click the **green** arrow to start/launch your VM for installation

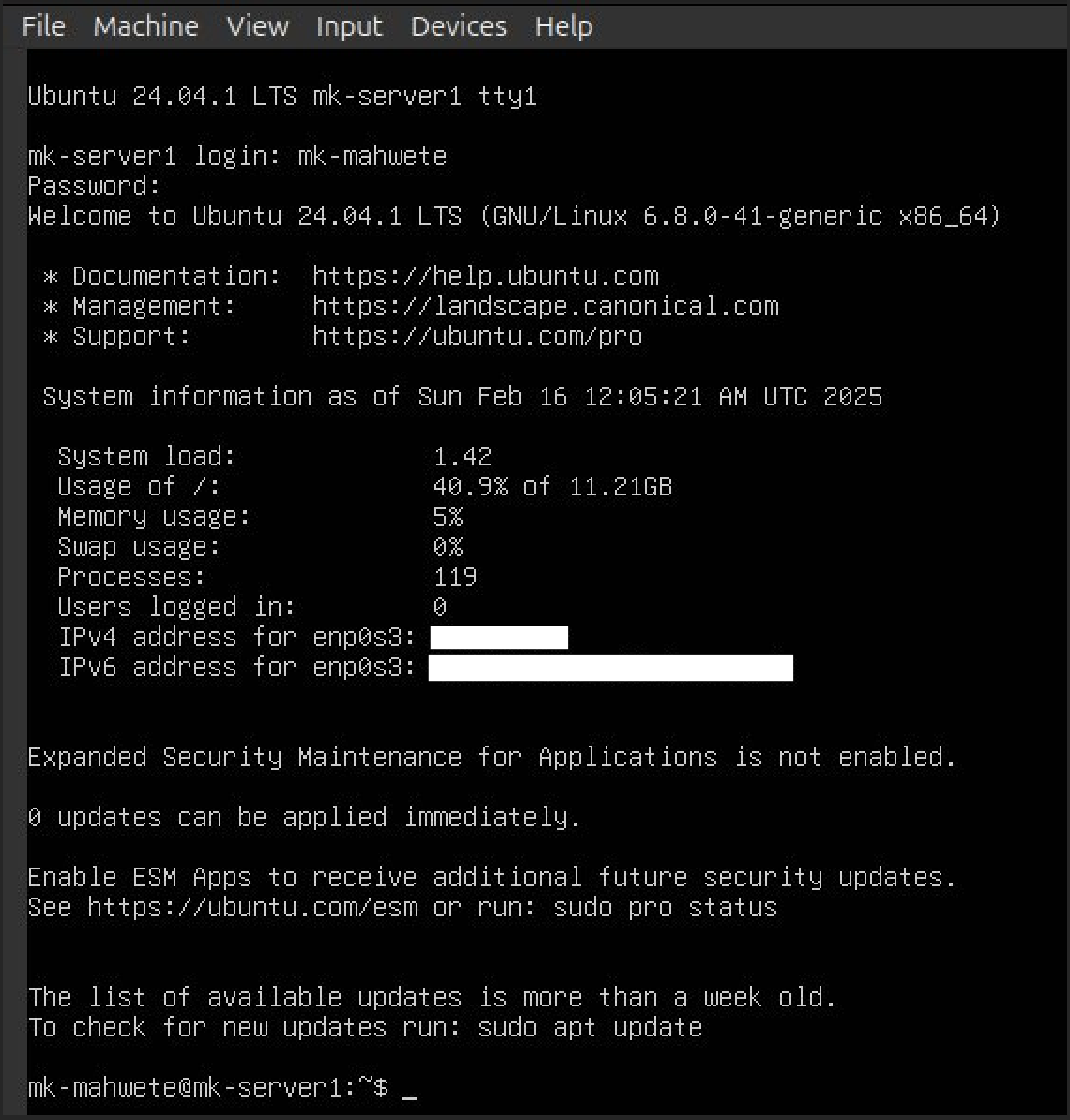
Now, after starting your VM. it’s time to go through all necessary step to setup your user name and password to login you machine

---

Step 3: login and updating your system before installing any software

By MK MAHWETE

After the installation, the first thing you will see is a CLI (Command Line Interface) – servers do not have Graphical Interfaces for better performance



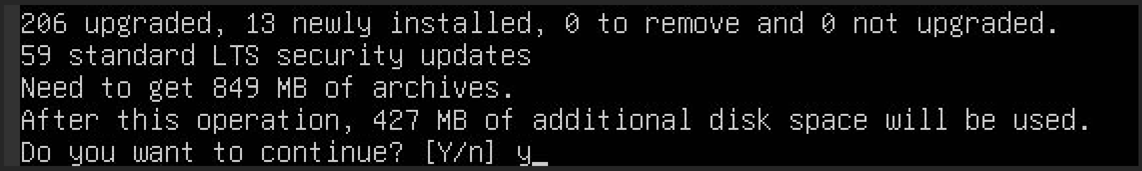
```
Ubuntu 24.04.1 LTS mk-server1 tty1

mk-server1 login: mk-mahwete
Password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-41-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Sun Feb 16 12:05:21 AM UTC 2025

  System load:             1.42
  Usage of /:              40.9% of 11.21GB
  Memory usage:            5%
  Swap usage:              0%
  Processes:               119
  Users logged in:         0
  IPv4 address for enp0s3: [redacted]
  IPv6 address for enp0s3: [redacted]


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

mk-mahwete@mk-server1:~$ _
```

Login to the machine by using your username and password you set during the installation process.

The first thing to do and always advisable is to keep your server up to date with the latest security patches: Run this command to update and upgrade

```
sudo apt update && sudo apt upgrade
```

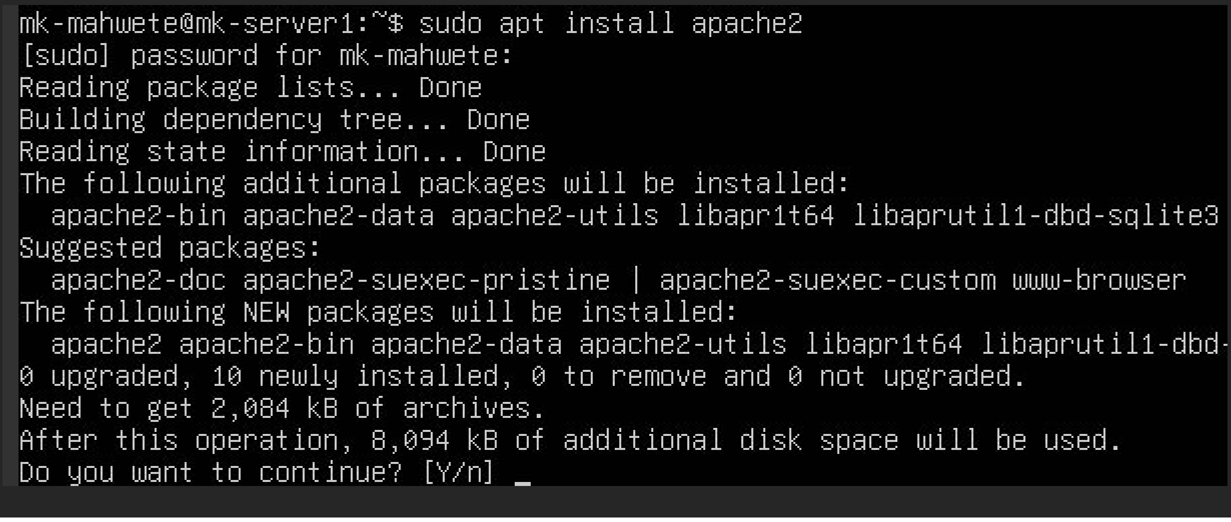
This command will update your system



```
206 upgraded, 13 newly installed, 0 to remove and 0 not upgraded.
59 standard LTS security updates
Need to get 849 MB of archives.
After this operation, 427 MB of additional disk space will be used.
Do you want to continue? [Y/n] y_
```

You shout see something like thing after running the command

Step 4: Installing Apache web server to handle incoming requests and serves web pages.



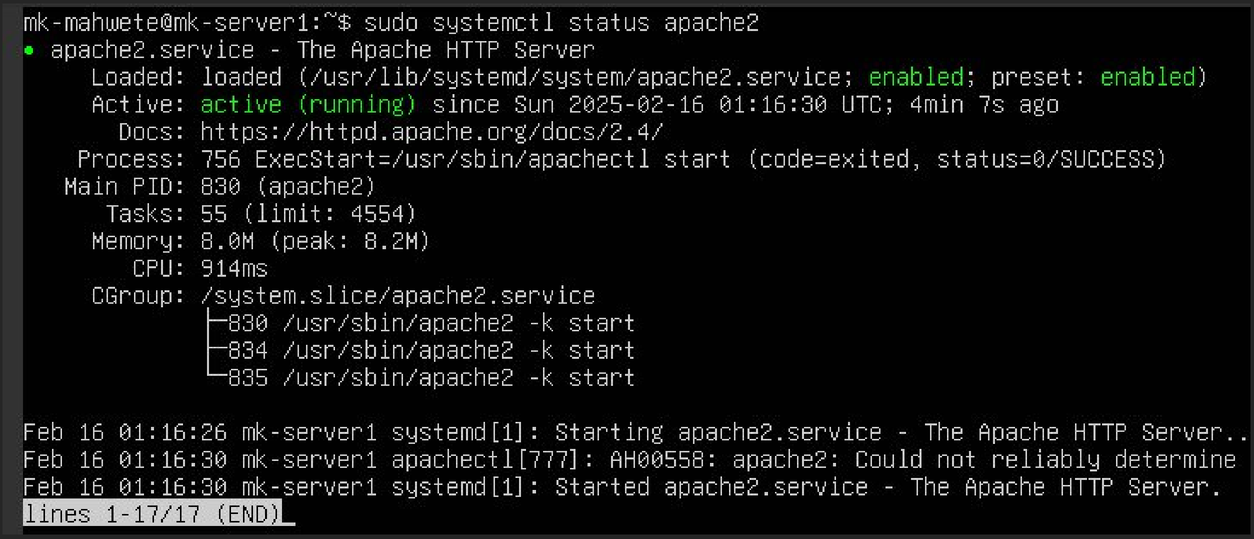
```
mk-mahwete@mk-server1:~$ sudo apt install apache2
[sudo] password for mk-mahwete:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 2,084 kB of archives.
After this operation, 8,094 kB of additional disk space will be used.
Do you want to continue? [Y/n] _
```

sudo apt install apache2

This command will install apache packages in your system for web server pages. Then run the following command to check the status of apache (either its enabled or disabled):

sudo systemctl status apache2

And you should see something like this:



```
mk-mahwete@mk-server1:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Sun 2025-02-16 01:16:30 UTC; 4min 7s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 756 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 830 (apache2)
      Tasks: 55 (limit: 4554)
     Memory: 8.0M (peak: 8.2M)
        CPU: 914ms
     CGroup: /system.slice/apache2.service
             ├─830 /usr/sbin/apache2 -k start
             ├─834 /usr/sbin/apache2 -k start
             └─835 /usr/sbin/apache2 -k start

Feb 16 01:16:26 mk-server1 systemd[1]: Starting apache2.service - The Apache HTTP Server..
Feb 16 01:16:30 mk-server1 apachectl[777]: AH00558: apache2: Could not reliably determine
Feb 16 01:16:30 mk-server1 systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-17/17 (END)
```

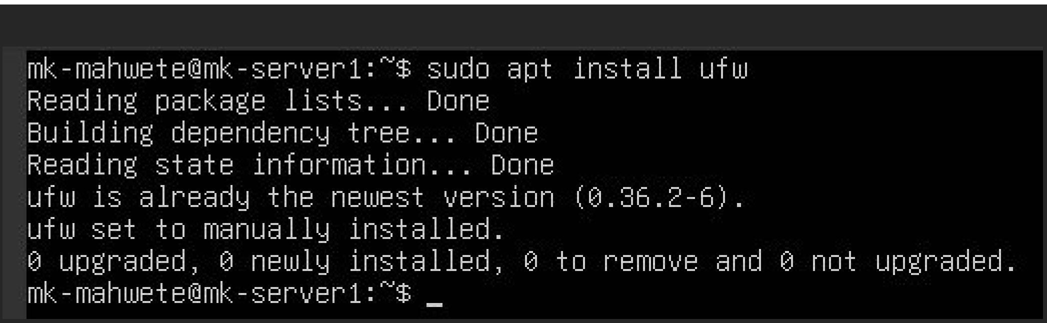
In my case it looks like my **Apache** is enabled

Step 5: Installing a **Firewall** (ufw) if your system doesn’t have one already

By MK MAHWETE

ufw is a firewall configuration tool that simplifies the management of netfilter, the standard firewall included in the linux kernal.

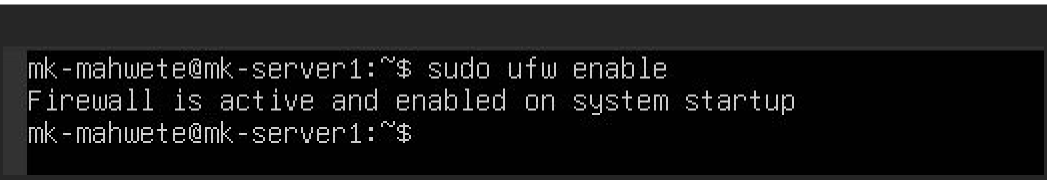
sudo apt install ufw



```
mk-mahwete@mk-server1:~$ sudo apt install ufw
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ufw is already the newest version (0.36.2-6).
ufw set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
mk-mahwete@mk-server1:~$ _
```

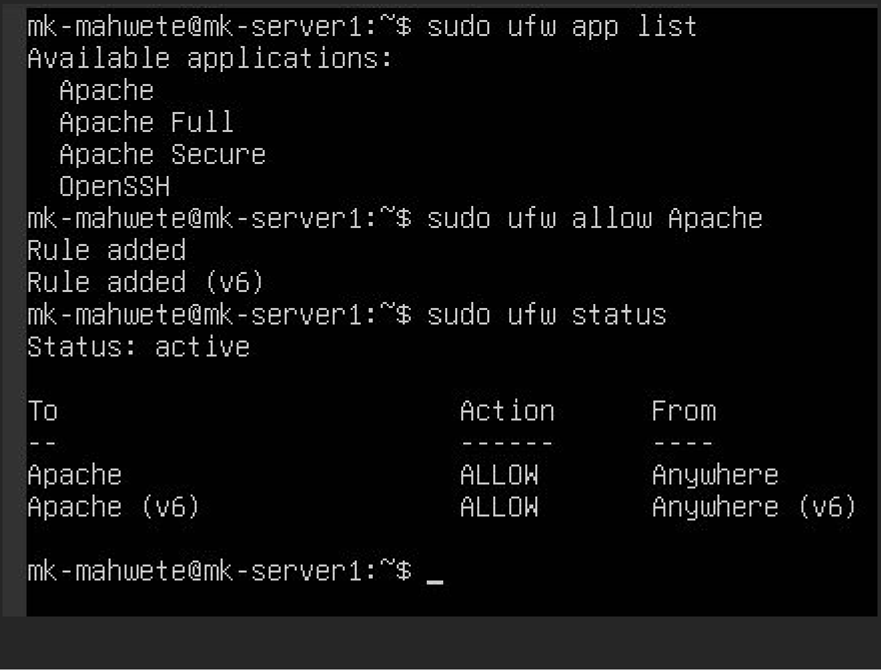
Enable is by typing the following command:

ufw enable



```
mk-mahwete@mk-server1:~$ sudo ufw enable
Firewall is active and enabled on system startup
mk-mahwete@mk-server1:~$
```

Now I want you to allow Apache for request HTTP: use the following commands

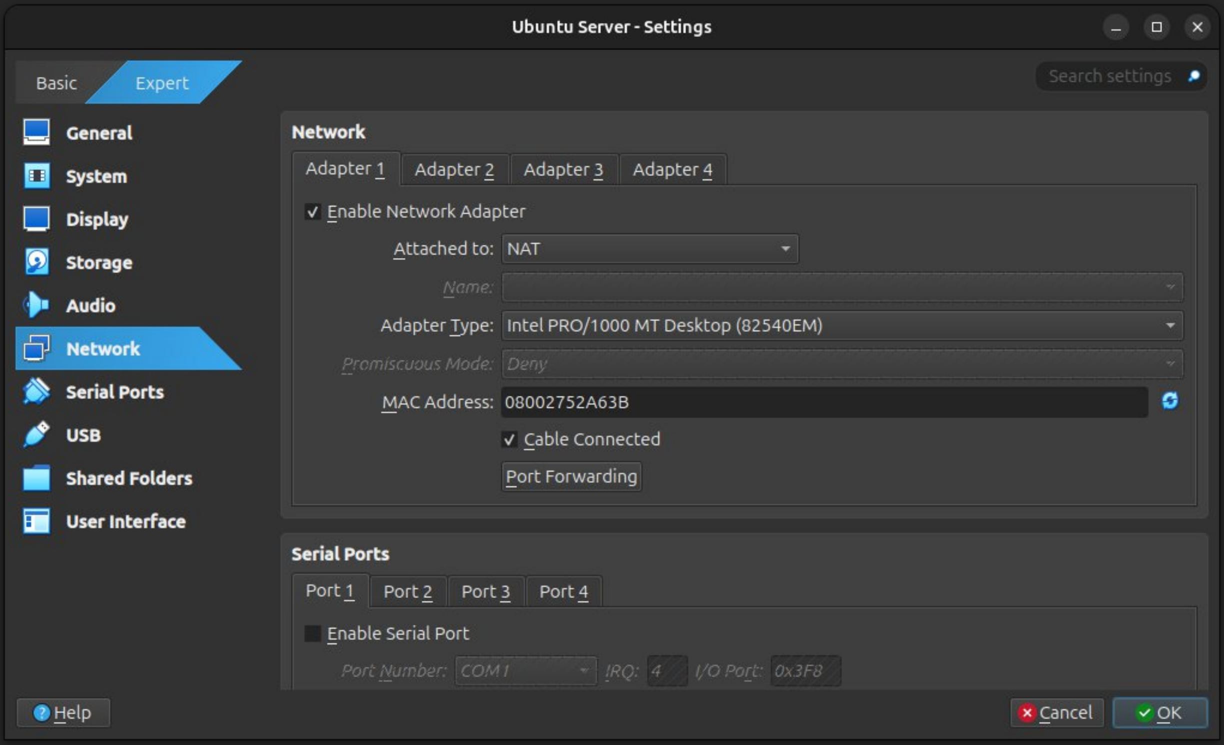


```
mk-mahwete@mk-server1:~$ sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  OpenSSH
mk-mahwete@mk-server1:~$ sudo ufw allow Apache
Rule added
Rule added (v6)
mk-mahwete@mk-server1:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
Apache                     ALLOW       Anywhere
Apache (v6)                ALLOW       Anywhere (v6)

mk-mahwete@mk-server1:~$ _
```

Now We have run into a simple problem that it’s easy to solve. Networking

Our server doesn’t have an IP address to communicate with other computers and we need one. If you haven’t done this before.Power Off your VM and Press the Orange setting icon to make a small change to out network adapter.

By MK MAHWETE



Network

Adapter 1 | Adapter 2 | Adapter 3 | Adapter 4

☑ Enable Network Adapter

Attached to: NAT

Name:

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Deny

MAC Address: 08002752A63B

☑ Cable Connected

Port Forwarding

Serial Ports

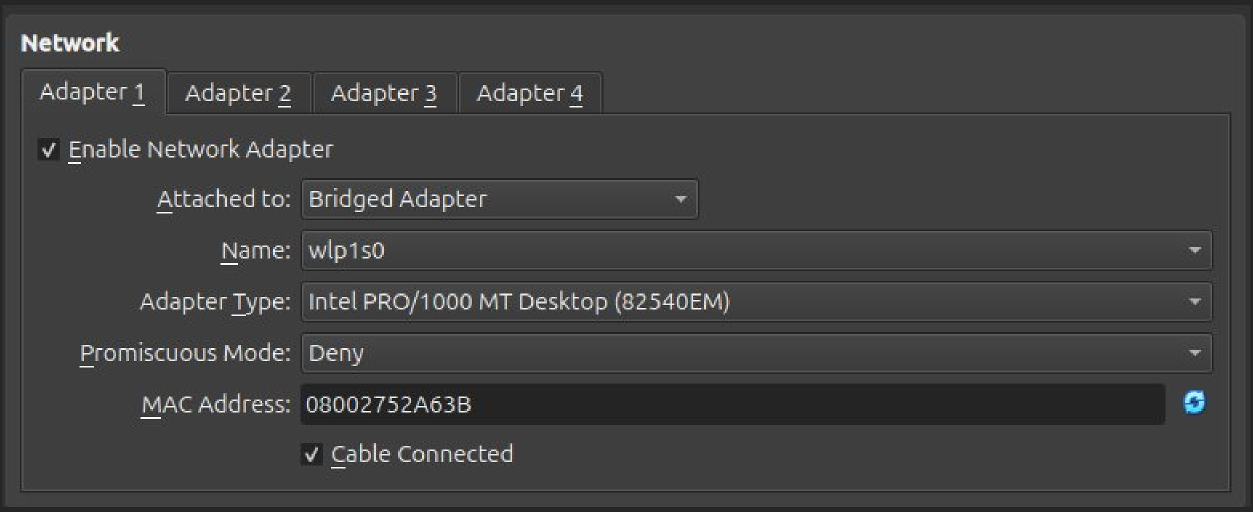
Port 1 | Port 2 | Port 3 | Port 4

□ Enable Serial Port

Port Number: COM1 IRQ: 4 I/O Port: 0x3F8

By default network called NAT (Network Address Translation). It’s a virtual router created by VBox that sits between the host and VM and incoming traffic is blocked by default (unless you set up port forwarding in the VirtualBox)

In my case I will go and change the network adapter to Bridged Adapter because i want the VM to get its own IP address from the same DHCP server as my host (it also gives me the ability to configure my own Static IP address on my LAN)



Network

Adapter 1 | Adapter 2 | Adapter 3 | Adapter 4

☑ Enable Network Adapter

Attached to: Bridged Adapter

Name: wlp1s0

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Deny

MAC Address: 08002752A63B

☑ Cable Connected

Now that the issue of Networking is out of the way, now it’s time to start our VM to test if Apache is working correctly

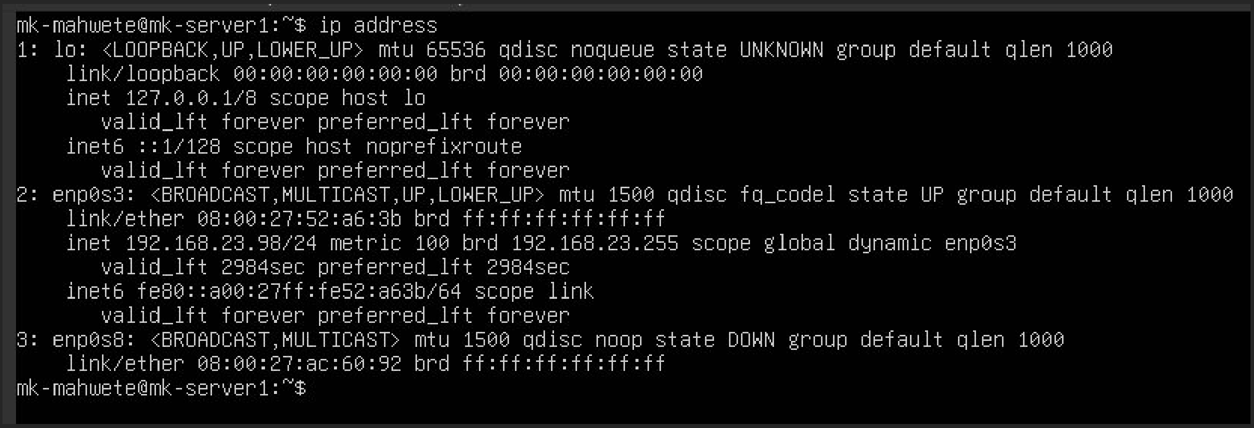
NOTE: Bridged Adapter doesn’t work well with Internal Wireless cards so its best to use Ethernet (RJ45)

By MK MAHWETE

Step 6: Obtaining Virtual Machine’s IP address and testing of Apache

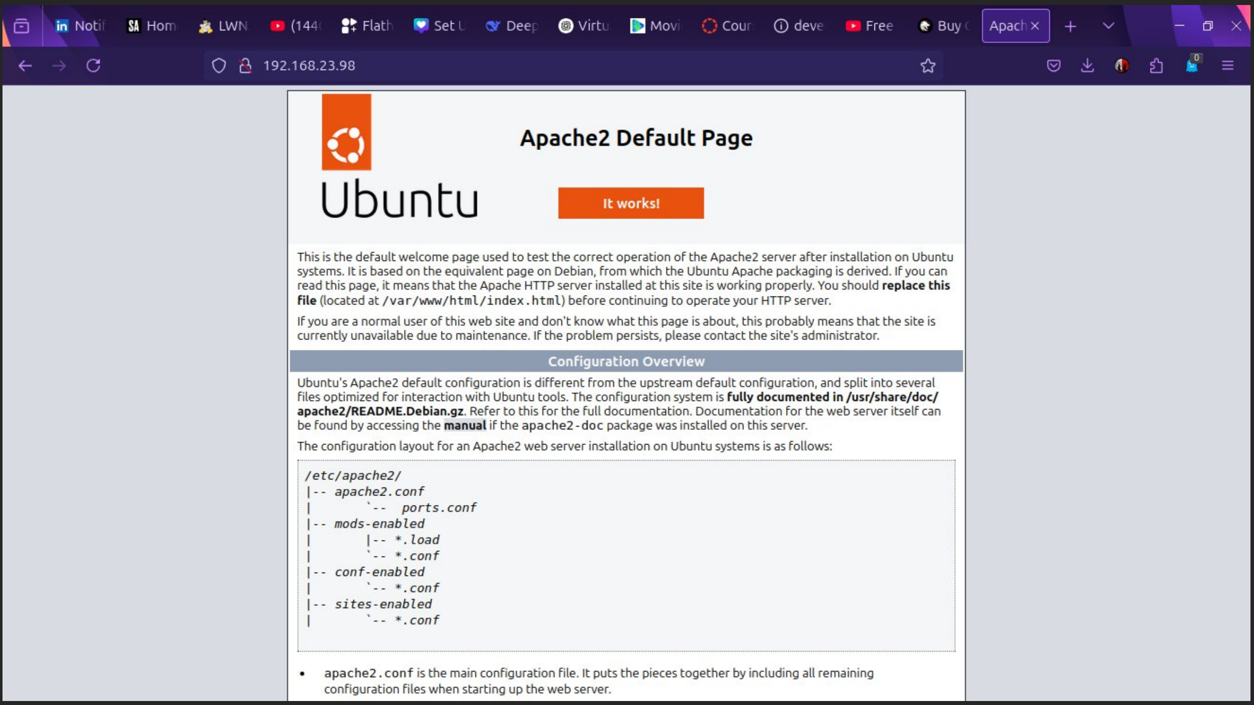
Now what’s left before moving on to the next step is getting out VM IP address and launch the default web page that came with apache. To get IP, type the following command:

```
ip address
```



```
mk-mahwete@mk-server1:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:52:a6:3b brd ff:ff:ff:ff:ff:ff
    inet 192.168.23.98/24 metric 100 brd 192.168.23.255 scope global dynamic enp0s3
       valid_lft 2984sec preferred_lft 2984sec
    inet6 fe80::a00:27ff:fe52:a63b/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 08:00:27:ac:60:92 brd ff:ff:ff:ff:ff:ff
mk-mahwete@mk-server1:~$
```

Look for the second interface 2: in my case my IP is 192.168.23.98 So i will copy this IP to my search engine (Google)



Apache2 Default Page

Ubuntu

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
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.

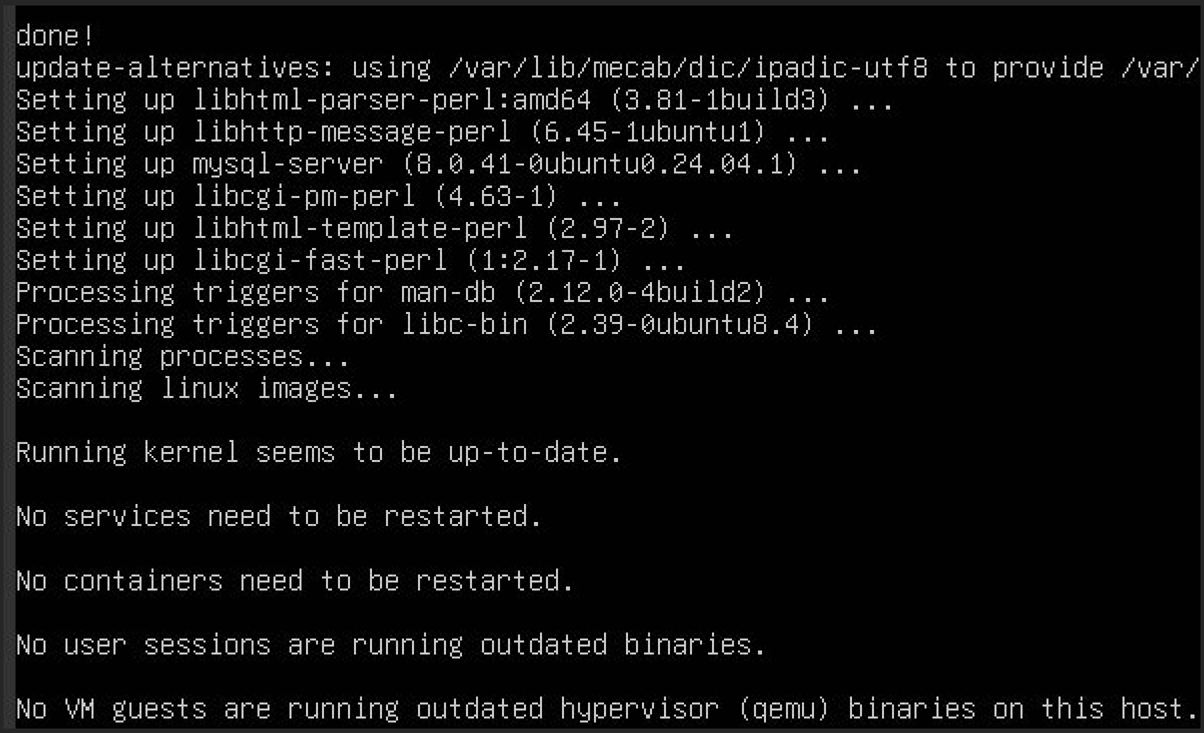
It works!

Step 7: Installing MySQL to databases

By MK MAHWETE

Now for our third part of installing LAMP stack. We need to install MySQL to manage, design, and create our databases. Type the following command to install it

```
sudo apt install mysql-server
```

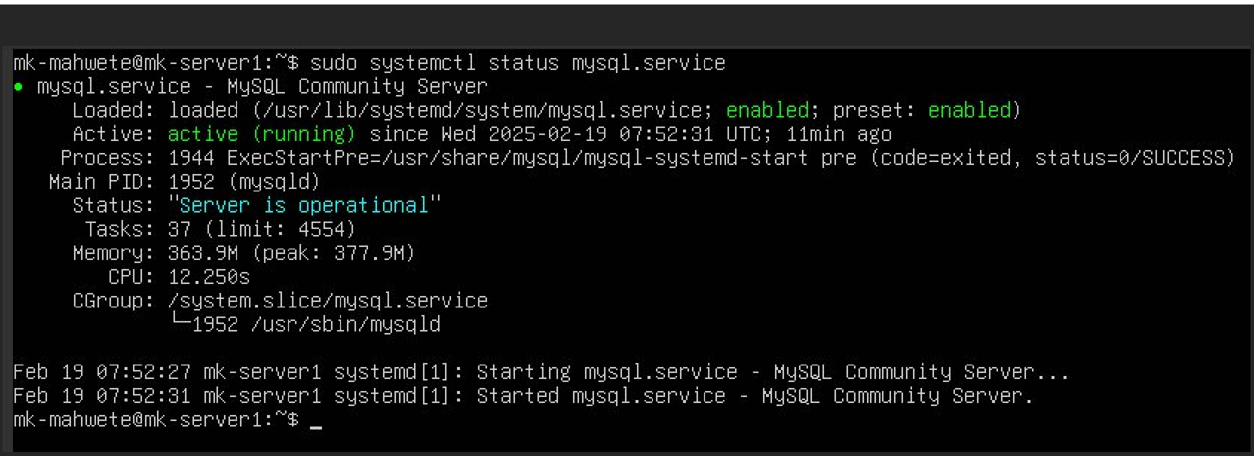


```
done!
update-alternatives: using /var/lib/mecab/dic/ipadic-utf8 to provide /var/
Setting up libhtml-parser-perl:amd64 (3.81-1build3) ...
Setting up libhttp-message-perl (6.45-1ubuntu1) ...
Setting up mysql-server (8.0.41-0ubuntu0.24.04.1) ...
Setting up libcgi-pm-perl (4.63-1) ...
Setting up libhtml-template-perl (2.97-2) ...
Setting up libcgi-fast-perl (1:2.17-1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.4) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

MySQL server installed without any issues, now we just have to set up the password and root account for the root manager. But first we need to confirm that MySQL services are enabled on our machine, just like we did with Apache

Use the following command to check:

```
sudo systemctl status mysql.service
```



```
mk-mahwete@mk-server1:~$ sudo systemctl status mysql.service
● mysql.service - MySQL Community Server
     Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
     Active: active (running) since Wed 2025-02-19 07:52:31 UTC; 11min ago
    Process: 1944 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
   Main PID: 1952 (mysqld)
     Status: "Server is operational"
      Tasks: 37 (limit: 4554)
     Memory: 363.9M (peak: 377.9M)
        CPU: 12.250s
     CGroup: /system.slice/mysql.service
             └─1952 /usr/sbin/mysqld

Feb 19 07:52:27 mk-server1 systemd[1]: Starting mysql.service - MySQL Community Server...
Feb 19 07:52:31 mk-server1 systemd[1]: Started mysql.service - MySQL Community Server.
mk-mahwete@mk-server1:~$ _
```

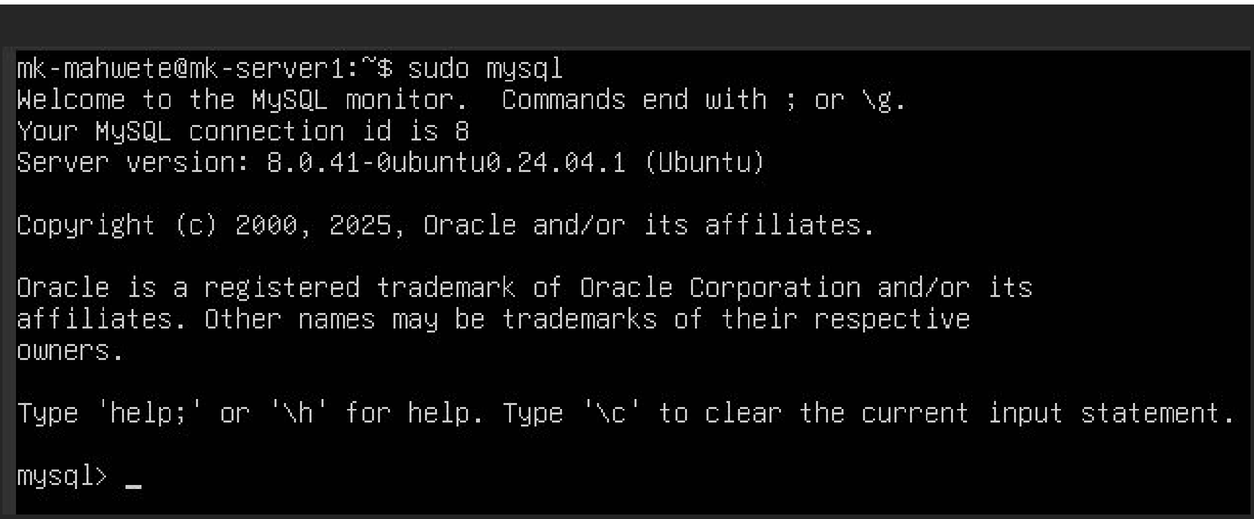
Our service are enabled, if not use this command to enable them:

```
sudo systemctl start mysql.service
```

By MK MAHWETE

Now for our next step, because we are not done with, lets log into our database (MySQL) by typing the following command:

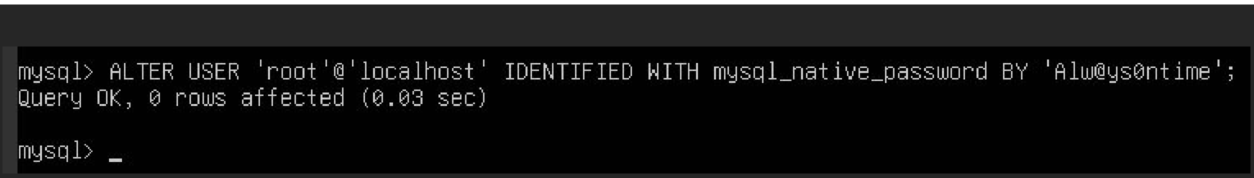
sudo mysql



```
mk-mahwete@mk-server1:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.41-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> _
```

As you can see mysql did not ask us for our root user password. We need to set one. Use the following SQL command to create a root account and password:

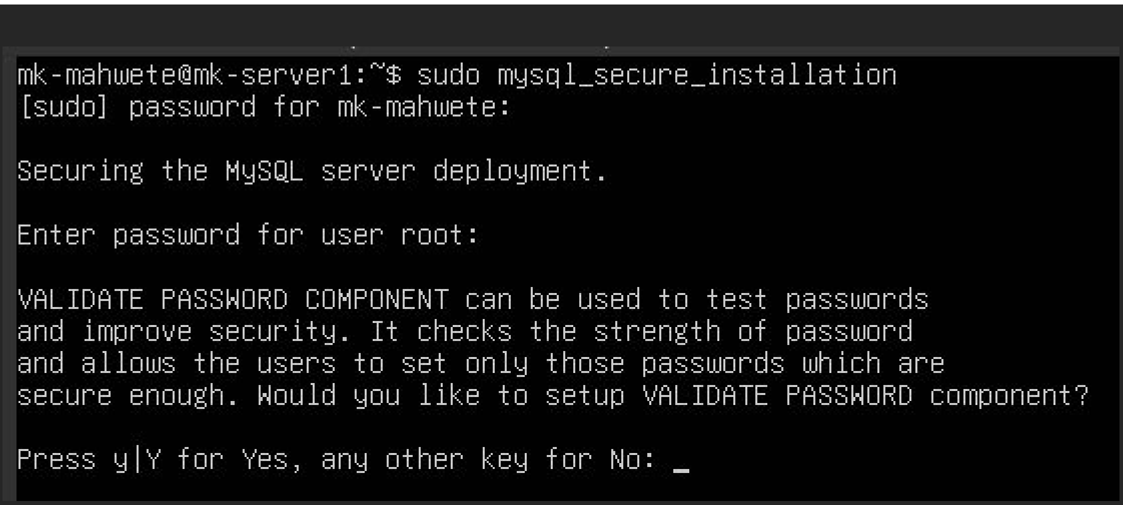
ALTER USER ‘root’@’localhost’ IDENTIFIED WITH mysql_native_password BY ‘your_password’



```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'Alw@ys0ntime';
Query OK, 0 rows affected (0.03 sec)

mysql> _
```

Now lets make out server more secure: type the following command to install all the required security installs:

sudo mysql_secure_installation



```
mk-mahwete@mk-server1:~$ sudo mysql_secure_installation
[sudo] password for mk-mahwete:

Securing the MySQL server deployment.

Enter password for user root:

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: _
```

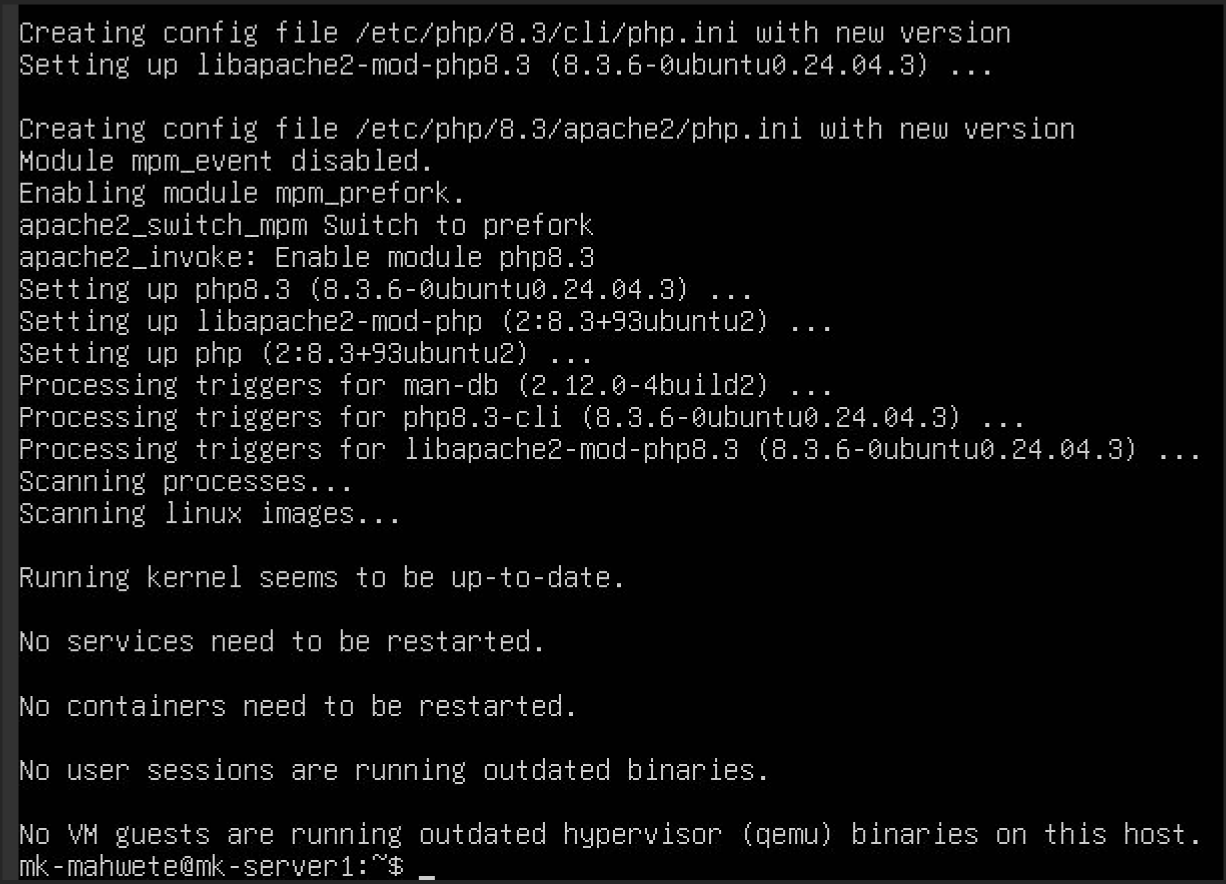
After running the script, it will ask you for our root password you set earlier on when setting up root user account and password. For validating password strength you can choose yes if you gonna use this for any production. Follow the on-screen and complete the setup.

Step 8: installing PHP

PHP (Hypertext Preprocessor) is a server-side scripting language used to build websites and web applications. It runs on the server and generates dynamic web pages before sending them to the user's browser.

We need to install the required packages, use this command to install them:

```
sudo apt install php libapache2-mod-php php-mysql
```



```
Creating config file /etc/php/8.3/cli/php.ini with new version
Setting up libapache2-mod-php8.3 (8.3.6-0ubuntu0.24.04.3) ...

Creating config file /etc/php/8.3/apache2/php.ini with new version
Module mpm_event disabled.
Enabling module mpm_prefork.
apache2_switch_mpm Switch to prefork
apache2_invoke: Enable module php8.3
Setting up php8.3 (8.3.6-0ubuntu0.24.04.3) ...
Setting up libapache2-mod-php (2:8.3+93ubuntu2) ...
Setting up php (2:8.3+93ubuntu2) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for php8.3-cli (8.3.6-0ubuntu0.24.04.3) ...
Processing triggers for libapache2-mod-php8.3 (8.3.6-0ubuntu0.24.04.3) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
mk-mahwete@mk-server1:~$ _
```

PHP is up and running

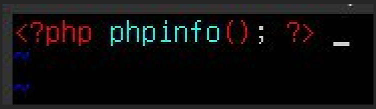
Now I want us to create a php file info under the following directory:

/var/www/html

You can use any CLI text editor of your choice, in my case I will be using Vim

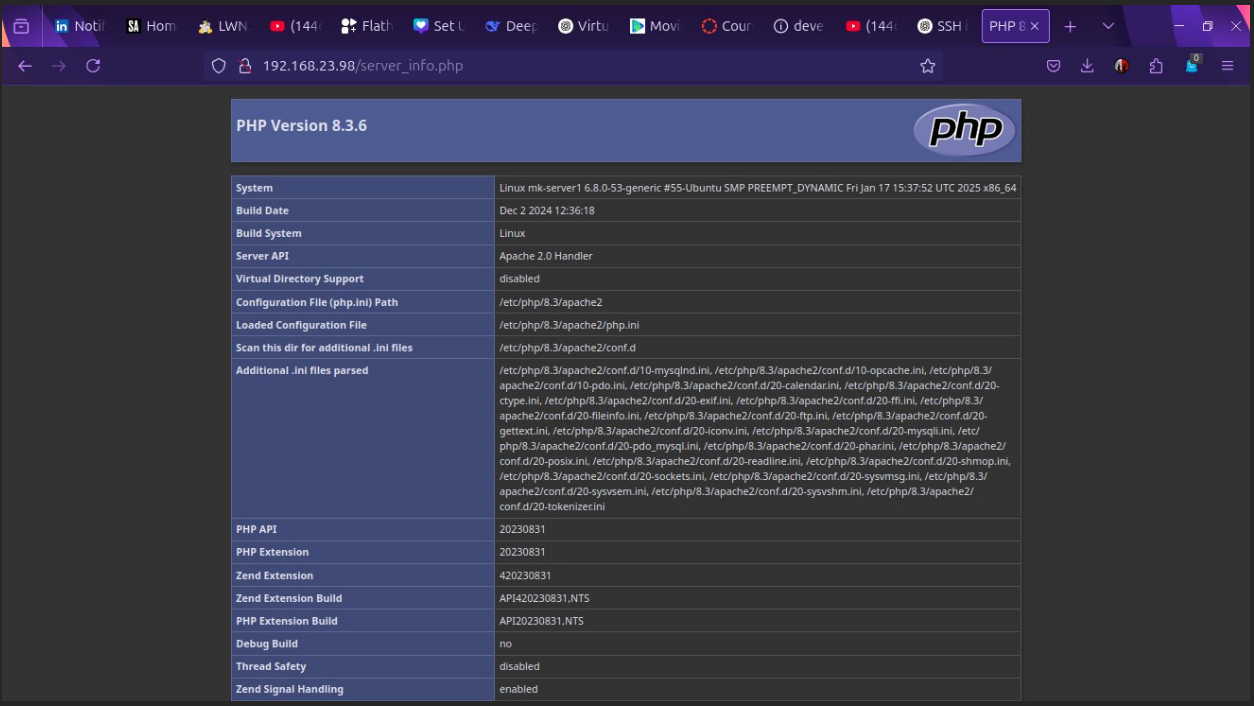
```
vim /var/www/html/server_info.php
```

By MK MAHWETE



```
<?php phpinfo(); ?>
```

Type the following inside your php file to display info about our server and save the file. Lets test out PHP file by typing our server IP on google search bar or 192.168.23.98/server_info.php

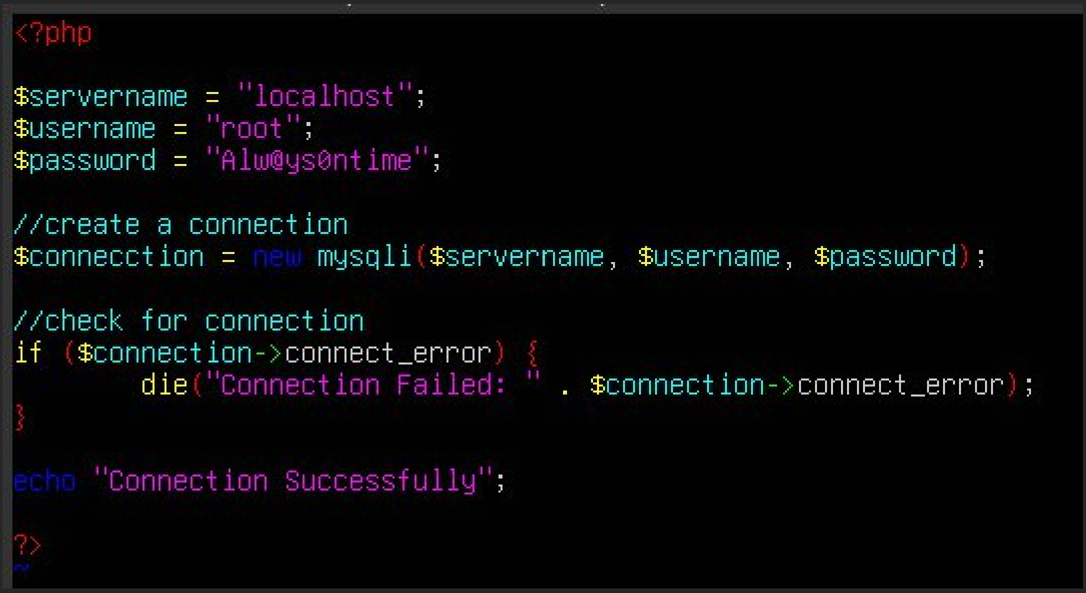


PHP Version 8.3.6

System	Linux mk-server1 6.8.0-53-generic #55-Ubuntu SMP PREEMPT_DYNAMIC Fri Jan 17 15:37:52 UTC 2025 x86_64
Build Date	Dec 2 2024 12:36:18
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.3/apache2
Loaded Configuration File	/etc/php/8.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.3/apache2/conf.d
Additional .ini files parsed	/etc/php/8.3/apache2/conf.d/10-mysqlnd.ini, /etc/php/8.3/apache2/conf.d/10-opcache.ini, /etc/php/8.3/apache2/conf.d/10-pdo.ini, /etc/php/8.3/apache2/conf.d/20-calendar.ini, /etc/php/8.3/apache2/conf.d/20-ctype.ini, /etc/php/8.3/apache2/conf.d/20-exif.ini, /etc/php/8.3/apache2/conf.d/20-ffi.ini, /etc/php/8.3/apache2/conf.d/20-fileinfo.ini, /etc/php/8.3/apache2/conf.d/20-ftp.ini, /etc/php/8.3/apache2/conf.d/20-gettext.ini, /etc/php/8.3/apache2/conf.d/20-iconv.ini, /etc/php/8.3/apache2/conf.d/20-mysqli.ini, /etc/php/8.3/apache2/conf.d/20-pdo_mysql.ini, /etc/php/8.3/apache2/conf.d/20-phar.ini, /etc/php/8.3/apache2/conf.d/20-posix.ini, /etc/php/8.3/apache2/conf.d/20-readline.ini, /etc/php/8.3/apache2/conf.d/20-shmop.ini, /etc/php/8.3/apache2/conf.d/20-sockets.ini, /etc/php/8.3/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.3/apache2/conf.d/20-sysvsem.ini, /etc/php/8.3/apache2/conf.d/20-sysvshm.ini, /etc/php/8.3/apache2/conf.d/20-tokenizer.ini
PHP API	20230831
PHP Extension	20230831
Zend Extension	420230831
Zend Extension Build	API420230831,NTS
PHP Extension Build	API20230831,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled

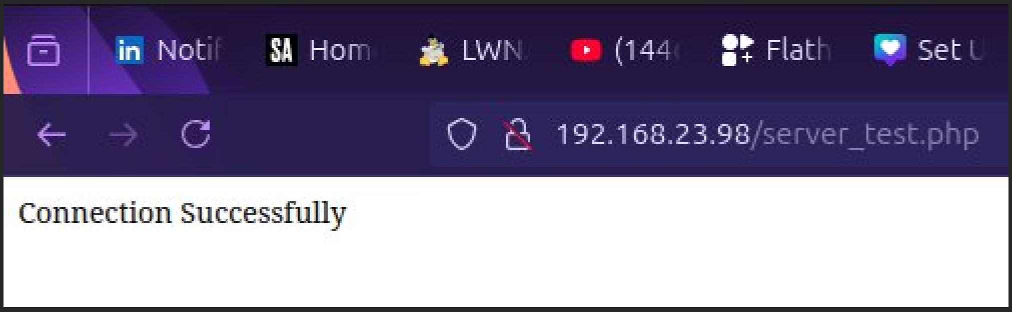
Our PHP is working just fine.

For our last step is testing if PHP can connect with our database. Lest’s create a file called server_test.php under the same directory /var/www/html



```
<?php

$servername = "localhost";
$username = "root";
$password = "Alw@ys0ntime";

//create a connection
$connecction = new mysqli($servername, $username, $password);

//check for connection
if ($connection->connect_error) {
        die("Connection Failed: " . $connection->connect_error);
}

echo "Connection Successfully";

?>
```

Lets test out PHP file by typing our server IP on google search bar or 192.168.23.98/server_test.php



192.168.23.98/server_test.php

Connection Successfully

The test was a success!

Now we have reached the END of our setup and installation. This is a very powerful setup for someone who want to setup a website on their hardware or locally.