Professional IT Business and Digitalization

IT Security Project Report

From

Yashwanth Pindi

Matriculation Number: s0590681

Supervisor: Prof. Dr. Nils Siebel

1 Table of Contents

2 Acknowledgement

I would like to express my sincere gratitude to my supervisor, Dr. Prof. Nils Siebel, for guiding me throughout this semester in IT Security and providing precious feedback to improve the quality of the project.

Through this course, I have gained extensive knowledge in the field of IT Security, particularly in the areas of LAMP servers and the hardening of servers and Linux systems. The comprehensive curriculum and hands-on projects have provided me with practical skills and a deeper understanding of these critical topics. This newfound knowledge has not only enhanced my technical abilities but also instilled in me a profound appreciation for the importance of securing IT infrastructures.

Furthermore, I am grateful to my peers and colleagues who have shared their knowledge, offered assistance, and provided valuable feedback throughout this journey. The collaborative spirit and sense of community have enriched my learning experience and have been crucial in overcoming the challenges encountered during this project.

I would also like to show our appreciation to everyone who remotely helped us with the project, who from our family and friends were also an important contribution to the completion of this research work.

3 Introduction

This IT Security Project aims to create a LAMP server out of a disk image provided by our Professor. The Professor, Prof. Dr. Nils Siebel, will give a configuration file for Virtual Machine (VM) box - native linux virtualization environment. We need to create a VM (with RAM, ROM) on our laptop / personal machine. Once that boots, the OS itself will have different RAM, CPU, and network Card. The network card (or the network interface) right now is not configured - so we will have to configure it. The network configuration is not very easy on linux. Different linux distributions have different ways to configure the network cards. It is not complicated but it is different for different linux systems (because all systems are different flavours of distributions).

4 General Information

RPM - redhat packet manager. Rocky linux is derived from the enterprise linux of redhat - so that’s why we have redhat packet manager and software running.
Why redhat linux? We are installing a server so we need an enterprise operating system. Redhat is specifically for servers and server maintenance. It is the best linux distribution for servers hence we are using it.

Dnf is the tool to install software packages in your system.
Yum is the older software manager but we can either use dnf or yum for installing packages into our system.

What is a raw file? Only a few programs can read raw files. It has raw data. The file format is very specific. Not formatted in a proprietary specific way. What does the hard disk data look like in its true original format? OS puts a structure. Raw hardware that it runs on doesn't have a structure.

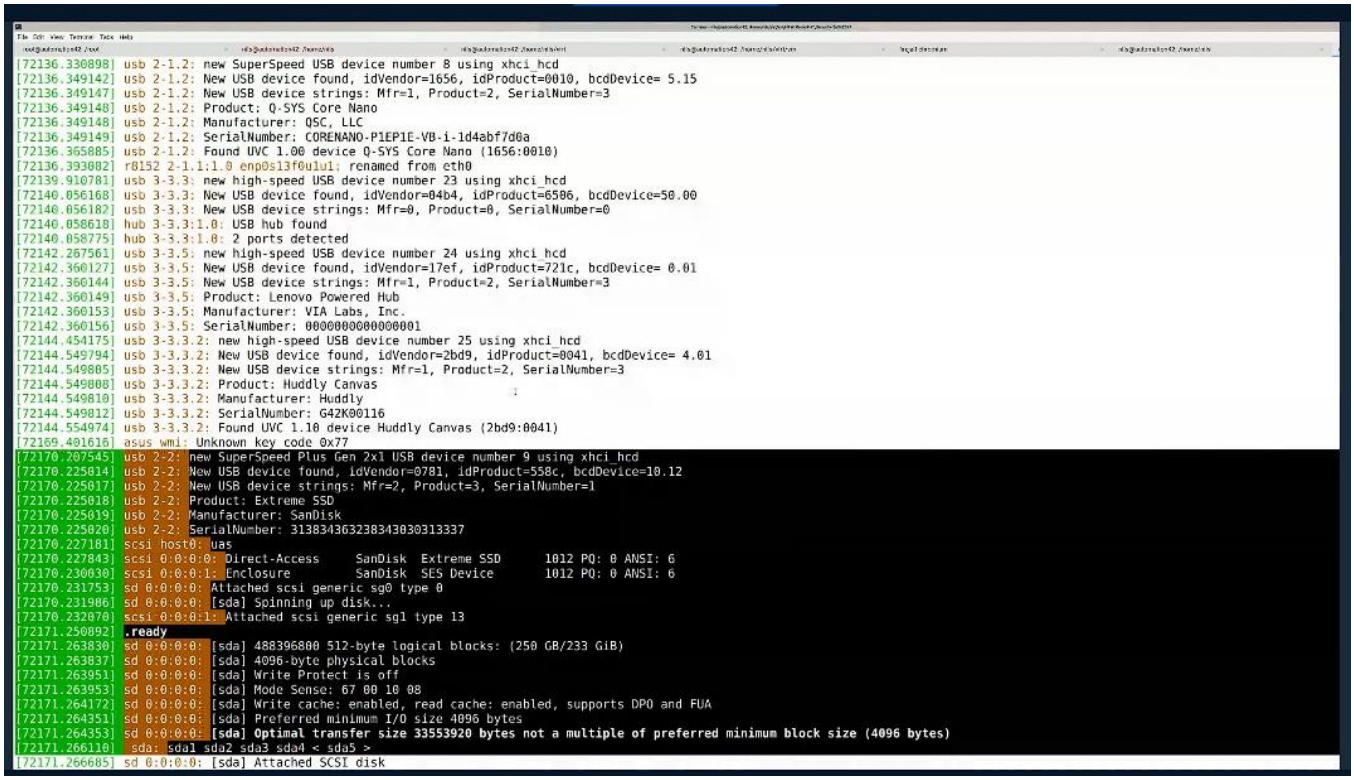


Figure 1: Boot up information on a file

How do you specify data if you put it into a file?

It has only 0s and 1s. It has many blocks. The data looks like blocks in the hard disk. A number of 512 bytes in one block and so on. The same data in the same order in a physical device and not a file. It is called raw data because it doesn't have a specific format but only blocks, blocks, blocks in a file (each 512 bytes long). Because it is not proprietary format but a raw format, any system can read it. But sadly, VM cannot understand these files. VM has its own proprietary file format which only it can use, so if you use a VM box, you have to convert the data from the raw format to the VM box format. Many people use the VM box and we can also convert it and use it. So 1 person can convert it and others can use it.

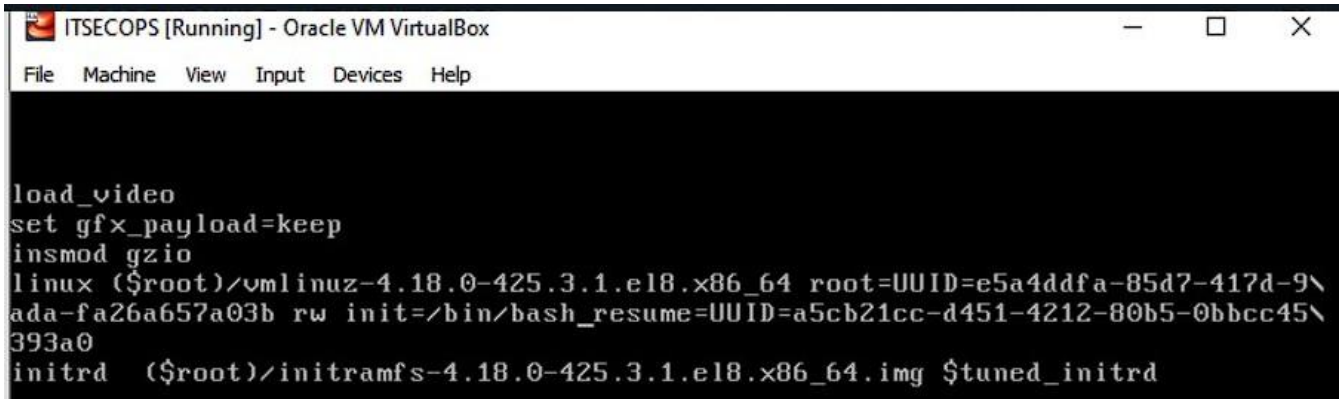


Figure 2: Setting up the VDI Image

Once we get the initial Virtual Disk Image (VDI) file, we need to configure it, and for that we make it rw (read write) in order to set the password and make ourselves the super user of our VDI image.

After this, we use the df command:

```
[root@it /]#
[root@it /]#
[root@it /]#
[root@it /]#
[root@it /]#
[root@it /]# passwd
Changing password for user root.
New password:
Retype new password:
passwd: Authentication token manipulation error
[root@it /]# df
Filesystem     1K-blocks    Used Available Use% Mounted on
devtmpfs          991932       0    991932   0% /dev
tmpfs            1009928       0   1009928   0% /dev/shm
tmpfs            1009928    8208   1001720   1% /run
tmpfs            1009928       0   1009928   0% /sys/fs/cgroup
/dev/sda2        5812320 2141200   3354960  39% /
[root@it /]# mount -o remount,rw /dev/sda2
[  905.569506] EXT4-fs (sda2): re-mounted. Opts: (null)
[root@it /]# passwd
Changing password for user root.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@it /]#
```

Df shows the file systems in our VDI. Note that temp file systems are lost when you close the computer.

Dev - stands for devices. Graphics, USB, or a hard disk is a device in Computer systems. In linux, devices are made very easy to access.

/dev is actually not a folder (although it looks like). It is a dev temp file system. Inside the folder you have entries that look like files but are not. A hard disk is basically a list of blocks. Hard disks are always partitions (of large spaces) (in windows it is 4 partitions - one of them is C:). Partitions are parts of the whole disk. USB has a partition but only 1 (the whole disk). Linux once it is booted has a virtual dev folder that has entries that are not files but correspond to physical hardware.

Dev/sda2 is the first hard disk of the system. Sda2 / had / vda - are one way or another. Sda2 is the first hard disk of the system. Sdb is the 2nd hard disk of the system.

ll /dev/sda - is not a file but a disk itself. 'll' at start gives the permission to view the partition disk space. Shows the time of the last modification date. This is the file that we made 'rw' while booting up. If you were to open this file (like fopen), then you will get the raw data from the disk. Linux has a very simple interface to devices. If you can read and write to a file - you can read and write whole disks in linux which makes it more

Ll /dev/sda* - shows all the partitions. There are 3 partitions (sda is broken into sda 1,2,3). One is boot and other is the sda main partition.

Df shows which partition is used for what. Sda1 is used.

Ll command works for that particular directory. * means it shows whatever starts with sda and on…

9 characters are in 3 groups. crw-rw-rw- . - is no. r is read w is write. These are permissions. Who is allowed to access it, write and read data to this particular device. These permissions define who is allowed what in your systems. One of the tasks of an OS is to keep people away from tempering from your system. If you have wrong permissions you might allow someone to do something or keep someone from doing something. First group refers to the owner of the file. Can be used to check the disk usage of each person. Ttys0 belongs to root. All of them in /dev belong to root because we are in the systemd folder. 2nd thing is some files should be readable only by a few groups and not others. All OS's have the concept of groups. Users are grouped into groups (eg. ProITD, EEE dept.). They all have a single user accounts. The ttys0 group is called dialout.

Modem was connected to serial port 0 . If you have a linux or unix - will be connected to s0. How can we govern who can use the modem (because there is a cost per minute to connect to the internet)? So they created a group called dialout. If you are a part of this group then you are allowed the permissions that belong to you (rw-). For ttys0 you can read and write. The 3rd group is anyone else other than the first 2 groups of people. These 3 groups are called: USER, GROUP, AND OTHER.

Rtc0 - is the real time clock. When you shut down the computer and boot up again it has

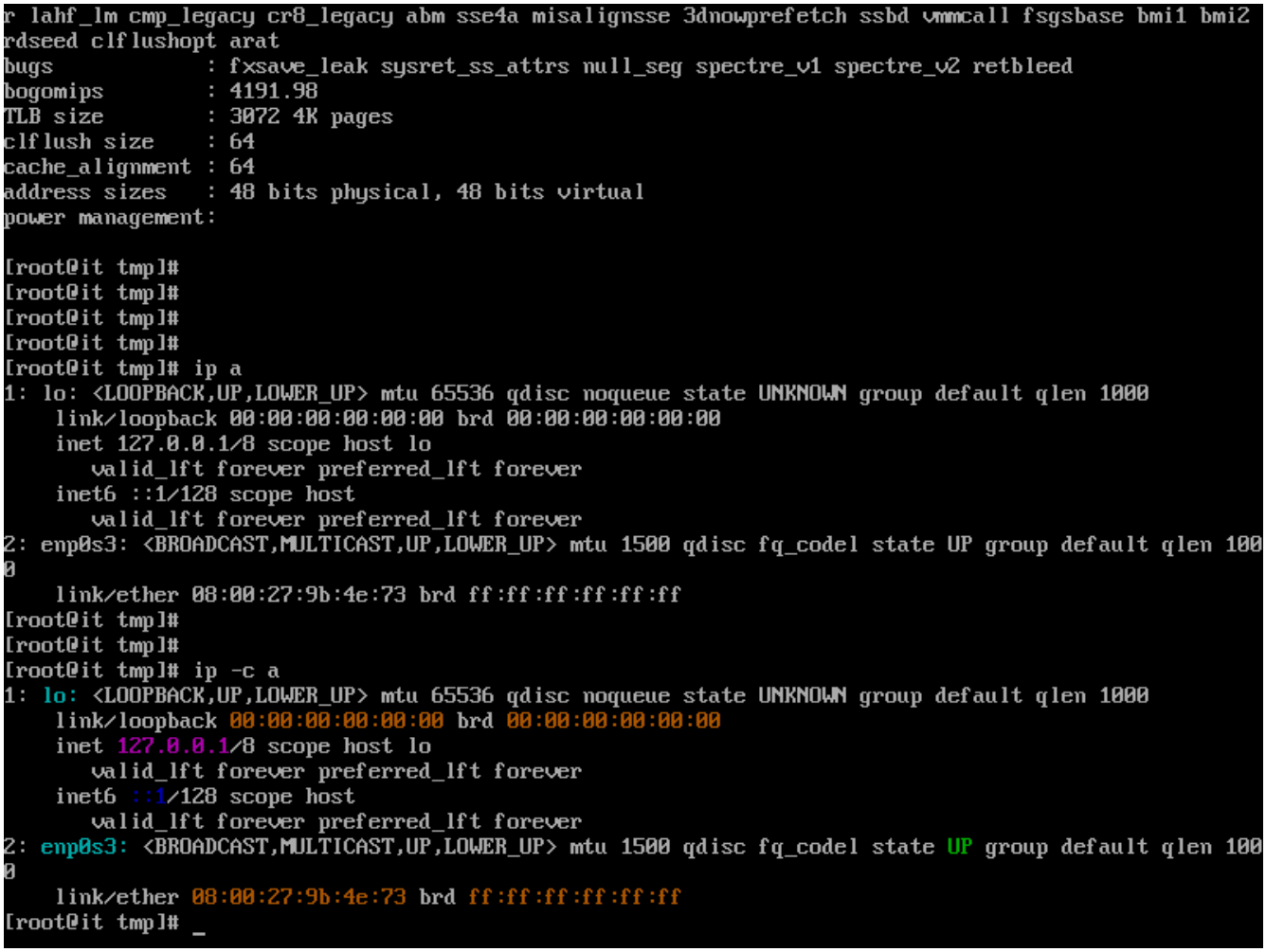
the date and time. The chip has date and time and it is powered by a small battery. Once the system starts the chip is not used anymore and the actual OS will check the time.

For eg. Yum if you call, it actually calls dnf-3. Its size is actually just the size of the NAME of the file it is pointing to. It is a link to the actual file that it is pointing to. These files will still be there on reboot.
If you have a large program, you don't want to put on desktop because it is too much space - so just create a link.

What does the **“sudo”** command do?
sudo is a program for Unix-like computer operating systems that enables users to run programs with the security privileges of another user, by default the superuser. You can also login to the system as a superuser and work on your server.

5 Network Cards IP Information

Network Card needs to be configured in order to access networks. They are also called as Network Interface Cards (NICs), are hardware components that enable computers and other devices to connect to a network. They provide the necessary interface for communication over a network, whether it be a local area network (LAN), a wide area network (WAN), or the internet.



Right now, we can see that the network card is not configured so we can't ping labor42.de (IP Address: 5.9.77.55).

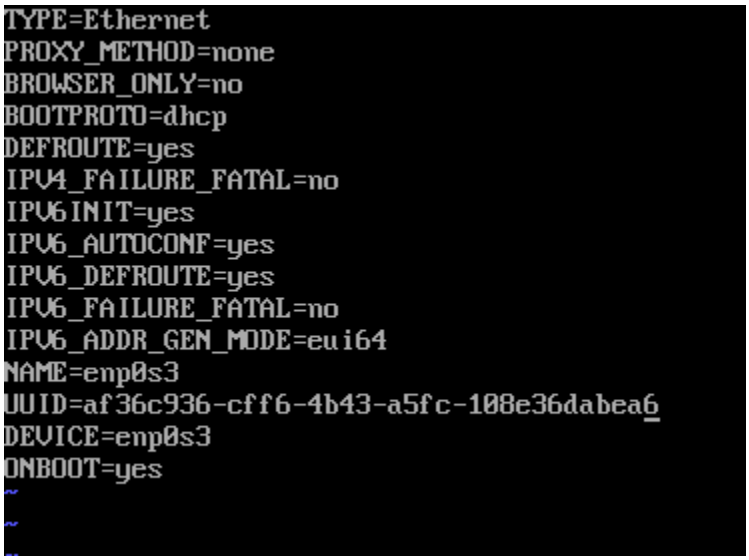
So once we enter the system configuration and create a copy of the enp1s0 network card (the copy is called enp1s3). The initial network card was given by the professor. We will be using enp0s3 for our server.

```
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 100
0
    link/ether 08:00:27:9b:4e:73 brd ff:ff:ff:ff:ff:ff
[root@it tmp]#
[root@it tmp]#
[root@it tmp]#
[root@it tmp]#
[root@it tmp]#
[root@it tmp]# ping labor42.de
ping: labor42.de: Name or service not known
[root@it tmp]#
[root@it tmp]#
[root@it tmp]#
[root@it tmp]# ll /etc/sysconfig/network
network          network-scripts/
[root@it tmp]# ll /etc/sysconfig/network-scripts/
total 4
-rw-r--r--. 1 root root 273 Dec  4  2022 ifcfg-enp1s0
[root@it tmp]#
[root@it tmp]#
[root@it tmp]# cd /etc/sysconfig/network-scripts/
[root@it network-scripts]#
[root@it network-scripts]# pwd
/etc/sysconfig/network-scripts
[root@it network-scripts]#
[root@it network-scripts]#
[root@it network-scripts]# ll
total 4
-rw-r--r--. 1 root root 273 Dec  4  2022 ifcfg-enp1s0
[root@it network-scripts]#
[root@it network-scripts]#
[root@it network-scripts]# cp ifcfg-enp1s0 ifcfg-enp1s3
[root@it network-scripts]#
```

After doing this, we configure the card so that our server can communicate with other networks. Now we open the contents of the newly created network card to see its details:

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=dhcp
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=eui64
NAME=enp0s3
UUID=af36c936-cff6-4b43-a5fc-108e36dabea9
DEVICE=enp0s3
ONBOOT=yes
~
~
```

Right now it is in an unconfigured state. We need to configure it to make it work. The contents of the configured file is:

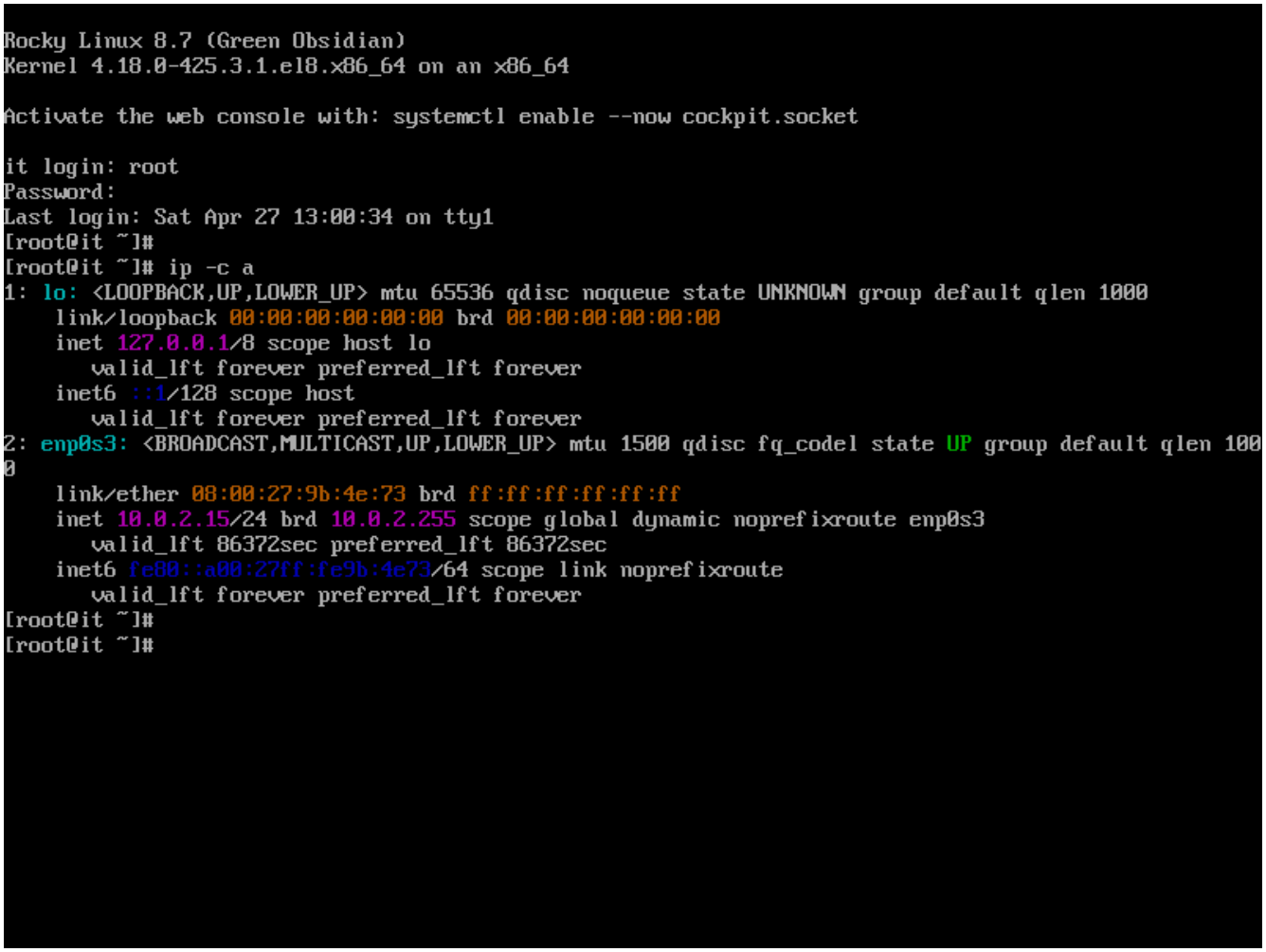


We now save the file by doing
“:wq”
since we are using a VI editor.

After this, we need to reboot the system so we do:
“Systemctl reboot”

Then re-check the network using
“ip -c a”

Now it will show IP addresses:



Now we try and ping labor42.de (IP Address: 5.9.77.55).

```
Kernel 4.18.0-425.3.1.el8.x86_64 on an x86_64

Activate the web console with: systemctl enable --now cockpit.socket

it login: root
Password:
Last login: Sat Apr 27 13:00:34 on tty1
[root@it ~]#
[root@it ~]# ip -c a
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
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 100
0
    link/ether 08:00:27:9b:4e:73 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 86372sec preferred_lft 86372sec
    inet6 fe80::a00:27ff:fe9b:4e73/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
[root@it ~]#
[root@it ~]# ping labor42.de
PING labor42.de (5.9.77.55) 56(84) bytes of data.
64 bytes from server3.siebel-hosting.com (5.9.77.55): icmp_seq=1 ttl=45 time=25.7 ms
64 bytes from server3.siebel-hosting.com (5.9.77.55): icmp_seq=2 ttl=45 time=19.1 ms
64 bytes from server3.siebel-hosting.com (5.9.77.55): icmp_seq=3 ttl=45 time=18.5 ms
64 bytes from server3.siebel-hosting.com (5.9.77.55): icmp_seq=4 ttl=45 time=18.9 ms
64 bytes from server3.siebel-hosting.com (5.9.77.55): icmp_seq=5 ttl=45 time=19.2 ms
64 bytes from server3.siebel-hosting.com (5.9.77.55): icmp_seq=6 ttl=45 time=19.7 ms
^C
--- labor42.de ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5013ms
rtt min/avg/max/mdev = 18.516/20.180/25.663/2.480 ms
[root@it ~]#
[root@it ~]# _
```

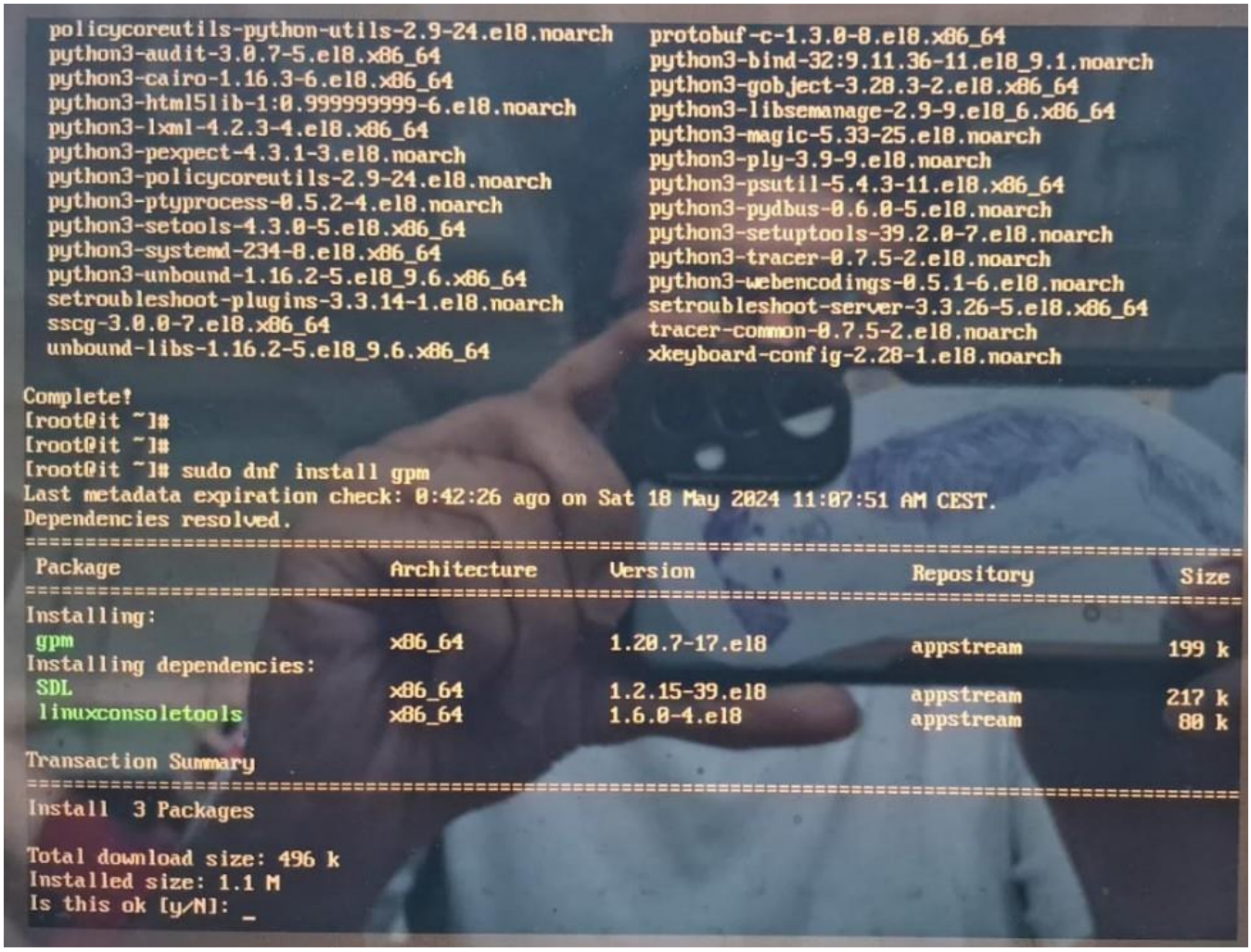
The ping was successful!

Hence, we have successfully configured our network card to communicate with other networks.

Alternatively, we can also do:
“systemctl restart NetworkManager”
This only restarts the network manager so you don’t need to reboot your system.

Why are we creating a new network card?
The initial slot was a PCI card and it had its own ports. We took it out and put in a different virtual card. In which slot and which port depends on the VDI image (and not the configuration file). So because we used the VDI image for creating our server - we need to do this. While Virtualbox created a new virtual machine, it also created a new network interface card. We configured the PCI card with our network card. The old name was not correct anymore. So we changed it.

In order to use the keyboard, run the following command:
“install gpm”



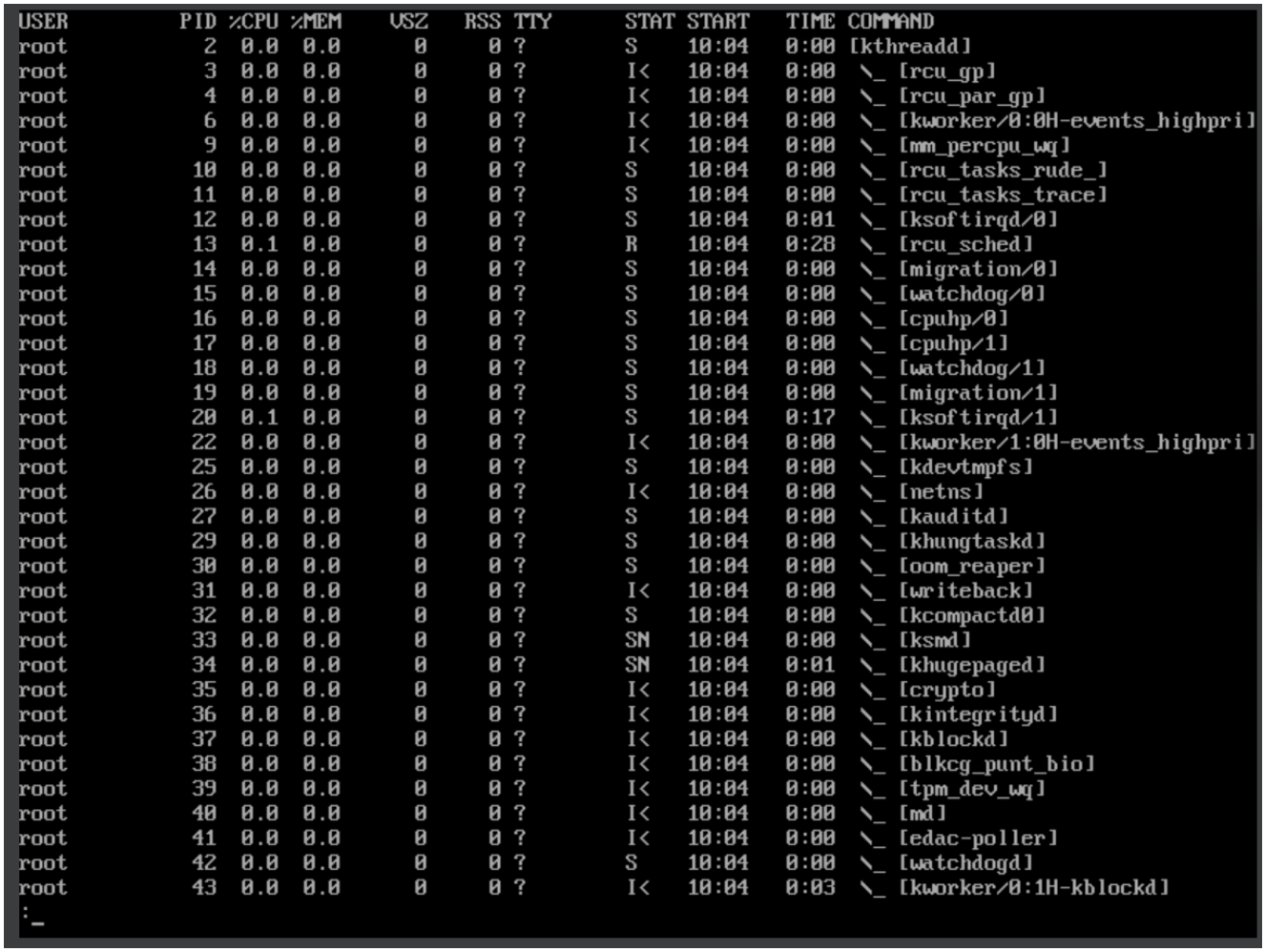
If you want any software package to automatically run at startup, you can use the:
“enable [package_name]”
Tip: In order to properly shut down the system, run the following command:
“systemctl halt“

What is the **“lsof”** command?
Lsof list’s all the packages that are running right now like daemons and other programs. So we check lsof and then these were the ones that were running and because of doing **“dnf up”**.

Pro Tip:
When we update something, only reboot those packages (like apache, network manager) . We can't reboot the whole system because servers are often remote and there are 100s of users present using it at all times.

“ps auxfw”

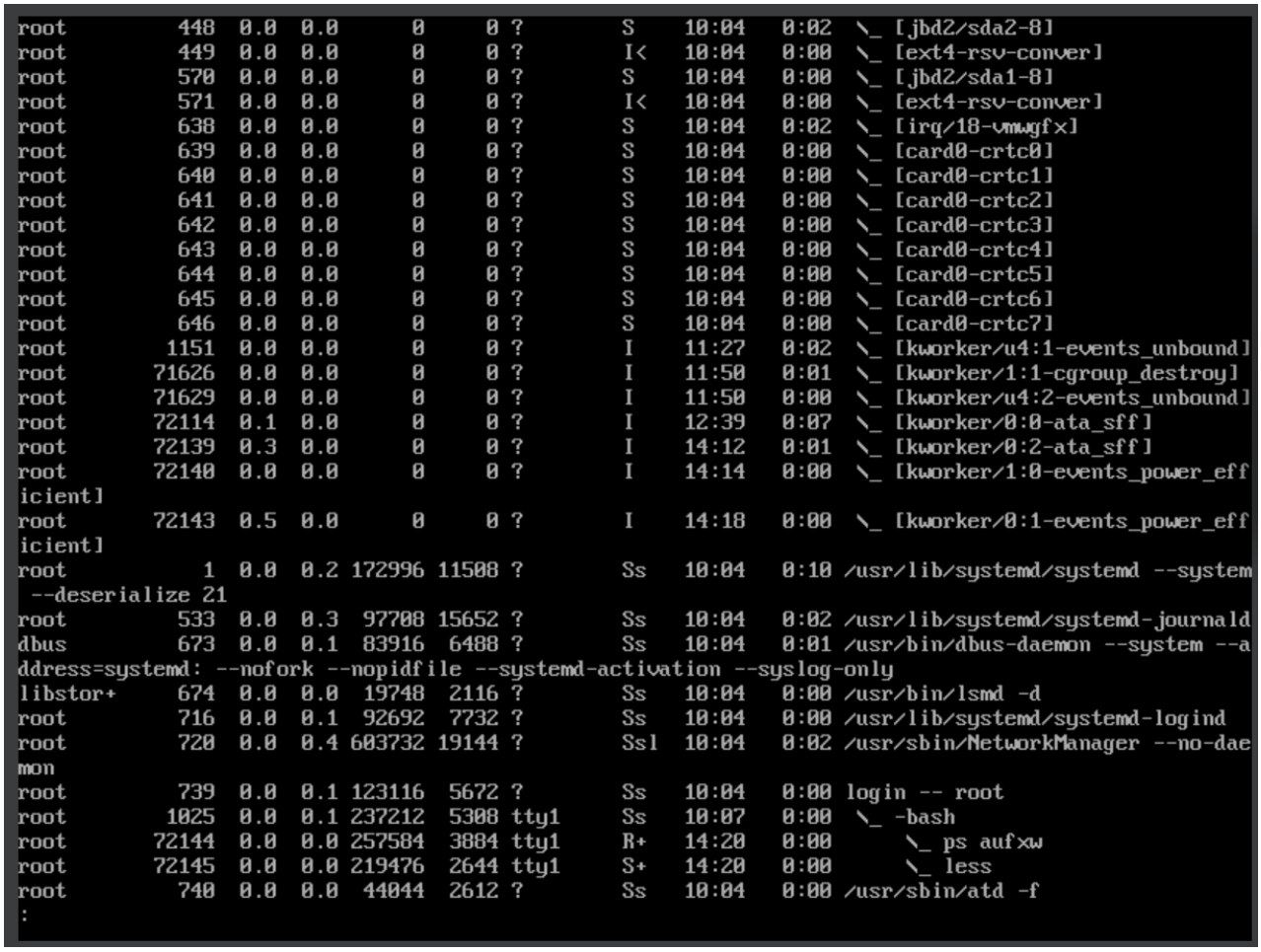
This command list down all the processes that are currently running in the system.



```
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           2  0.0  0.0      0     0 ?        S    10:04   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   10:04   0:00  \_ [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   10:04   0:00  \_ [rcu_par_gp]
root           6  0.0  0.0      0     0 ?        I<   10:04   0:00  \_ [kworker/0:0H-events_highpri]
root           9  0.0  0.0      0     0 ?        I<   10:04   0:00  \_ [mm_percpu_wq]
root          10  0.0  0.0      0     0 ?        S    10:04   0:00  \_ [rcu_tasks_rude_]
root          11  0.0  0.0      0     0 ?        S    10:04   0:00  \_ [rcu_tasks_trace]
root          12  0.0  0.0      0     0 ?        S    10:04   0:01  \_ [ksoftirqd/0]
root          13  0.1  0.0      0     0 ?        R    10:04   0:28  \_ [rcu_sched]
root          14  0.0  0.0      0     0 ?        S    10:04   0:00  \_ [migration/0]
root          15  0.0  0.0      0     0 ?        S    10:04   0:00  \_ [watchdog/0]
root          16  0.0  0.0      0     0 ?        S    10:04   0:00  \_ [cpuhp/0]
root          17  0.0  0.0      0     0 ?        S    10:04   0:00  \_ [cpuhp/1]
root          18  0.0  0.0      0     0 ?        S    10:04   0:00  \_ [watchdog/1]
root          19  0.0  0.0      0     0 ?        S    10:04   0:00  \_ [migration/1]
root          20  0.1  0.0      0     0 ?        S    10:04   0:17  \_ [ksoftirqd/1]
root          22  0.0  0.0      0     0 ?        I<   10:04   0:00  \_ [kworker/1:0H-events_highpri]
root          25  0.0  0.0      0     0 ?        S    10:04   0:00  \_ [kdevtmpfs]
root          26  0.0  0.0      0     0 ?        I<   10:04   0:00  \_ [netns]
root          27  0.0  0.0      0     0 ?        S    10:04   0:00  \_ [kauditd]
root          29  0.0  0.0      0     0 ?        S    10:04   0:00  \_ [khungtaskd]
root          30  0.0  0.0      0     0 ?        S    10:04   0:00  \_ [oom_reaper]
root          31  0.0  0.0      0     0 ?        I<   10:04   0:00  \_ [writeback]
root          32  0.0  0.0      0     0 ?        S    10:04   0:00  \_ [kcompactd0]
root          33  0.0  0.0      0     0 ?        SN   10:04   0:00  \_ [ksmd]
root          34  0.0  0.0      0     0 ?        SN   10:04   0:01  \_ [khugepaged]
root          35  0.0  0.0      0     0 ?        I<   10:04   0:00  \_ [crypto]
root          36  0.0  0.0      0     0 ?        I<   10:04   0:00  \_ [kintegrityd]
root          37  0.0  0.0      0     0 ?        I<   10:04   0:00  \_ [kblockd]
root          38  0.0  0.0      0     0 ?        I<   10:04   0:00  \_ [blkcg_punt_bio]
root          39  0.0  0.0      0     0 ?        I<   10:04   0:00  \_ [tpm_dev_wq]
root          40  0.0  0.0      0     0 ?        I<   10:04   0:00  \_ [md]
root          41  0.0  0.0      0     0 ?        I<   10:04   0:00  \_ [edac-poller]
root          42  0.0  0.0      0     0 ?        S    10:04   0:00  \_ [watchdogd]
root          43  0.0  0.0      0     0 ?        I<   10:04   0:03  \_ [kworker/0:1H-kblockd]
:_
```

Threads are program instances of the same program.
“_” - these are the kernel threads running in the system.

In the below picture, you can see the 2nd column which contains the process numbers. Process 1 is the first process that boots the system up. It is also the last system that runs when we shut down.



```
root         448  0.0  0.0      0     0 ?        S    10:04   0:02  \_ [jbd2/sda2-8]
root         449  0.0  0.0      0     0 ?        I<   10:04   0:00  \_ [ext4-rsv-conver]
root         570  0.0  0.0      0     0 ?        S    10:04   0:00  \_ [jbd2/sda1-8]
root         571  0.0  0.0      0     0 ?        I<   10:04   0:00  \_ [ext4-rsv-conver]
root         638  0.0  0.0      0     0 ?        S    10:04   0:02  \_ [irq/18-vmwgfx]
root         639  0.0  0.0      0     0 ?        S    10:04   0:00  \_ [card0-crtc0]
root         640  0.0  0.0      0     0 ?        S    10:04   0:00  \_ [card0-crtc1]
root         641  0.0  0.0      0     0 ?        S    10:04   0:00  \_ [card0-crtc2]
root         642  0.0  0.0      0     0 ?        S    10:04   0:00  \_ [card0-crtc3]
root         643  0.0  0.0      0     0 ?        S    10:04   0:00  \_ [card0-crtc4]
root         644  0.0  0.0      0     0 ?        S    10:04   0:00  \_ [card0-crtc5]
root         645  0.0  0.0      0     0 ?        S    10:04   0:00  \_ [card0-crtc6]
root         646  0.0  0.0      0     0 ?        S    10:04   0:00  \_ [card0-crtc7]
root        1151  0.0  0.0      0     0 ?        I    11:27   0:02  \_ [kworker/u4:1-events_unbound]
root       71626  0.0  0.0      0     0 ?        I    11:50   0:01  \_ [kworker/1:1-cgroup_destroy]
root       71629  0.0  0.0      0     0 ?        I    11:50   0:00  \_ [kworker/u4:2-events_unbound]
root       72114  0.1  0.0      0     0 ?        I    12:39   0:07  \_ [kworker/0:0-ata_sff]
root       72139  0.3  0.0      0     0 ?        I    14:12   0:01  \_ [kworker/0:2-ata_sff]
root       72140  0.0  0.0      0     0 ?        I    14:14   0:00  \_ [kworker/1:0-events_power_eff
icient]
root       72143  0.5  0.0      0     0 ?        I    14:18   0:00  \_ [kworker/0:1-events_power_eff
icient]
root           1  0.0  0.2 172996 11508 ?        Ss   10:04   0:10 /usr/lib/systemd/systemd --system
 --deserialize 21
root         533  0.0  0.3  97708 15652 ?        Ss   10:04   0:02 /usr/lib/systemd/systemd-journald
dbus         673  0.0  0.1  83916  6488 ?        Ss   10:04   0:01 /usr/bin/dbus-daemon --system --a
ddress=systemd: --nofork --nopidfile --systemd-activation --syslog-only
libstor+     674  0.0  0.0  19748  2116 ?        Ss   10:04   0:00 /usr/bin/lsmd -d
root         716  0.0  0.1  92692  7732 ?        Ss   10:04   0:00 /usr/lib/systemd/systemd-logind
root         720  0.0  0.4 603732 19144 ?        Ssl  10:04   0:02 /usr/sbin/NetworkManager --no-dae
mon
root         739  0.0  0.1 123116  5672 ?        Ss   10:04   0:00 login -- root
root        1025  0.0  0.1 237212  5308 tty1     Ss   10:07   0:00  \_ -bash
root       72144  0.0  0.0 257584  3884 tty1     R+   14:20   0:00      \_ ps auf×w
root       72145  0.0  0.0 219476  2644 tty1     S+   14:20   0:00      \_ less
root         740  0.0  0.0  44044  2612 ?        Ss   10:04   0:00 /usr/sbin/atd -f
:
```

6 Configuring the LAMP Server

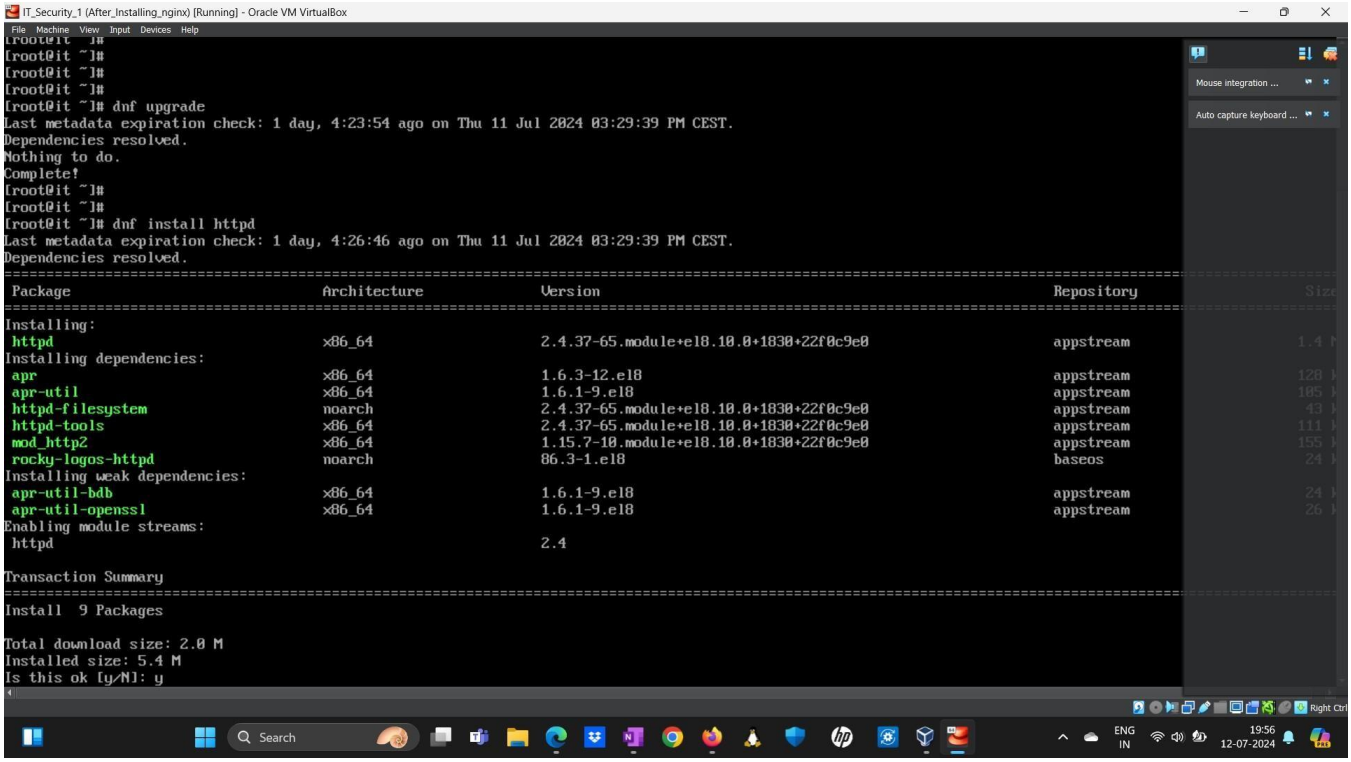
LAMP stands for “Linux Apache MariaDB PHP”. Here we have an OS , a web server software, a database (MySQL), and PHP for serverside scripting.

What do we need to run a server with a website? We need an OS (Linux), web server - software runs and waits for inputs like apache/nginx. Many web pages have a front-end and a back-end, when the name “LAMP” was created, Mariafdb was one to use. Maria DB is basically the same thing (can use PostGRESQL or noSQL whatever you want).

We are focusing now only on a server which runs moodle or facebook or (in our case, Wordpress). LAMP is the basis of backend server management. If you buy a web service (wordpress instance) - you can configure and create your own website.

Where can I rent a server to run my website? Have to have a LAMP for this. The LAMP system is the number 1 example of setting up a server. We don’t need a GUI for our server. Since we want to harden - we don't want that since we have to install 50-100 packages which means more tools for your attacker to find bugs in the packages that you have, and hence higher possibility of attack. So we must always remove unused programs.

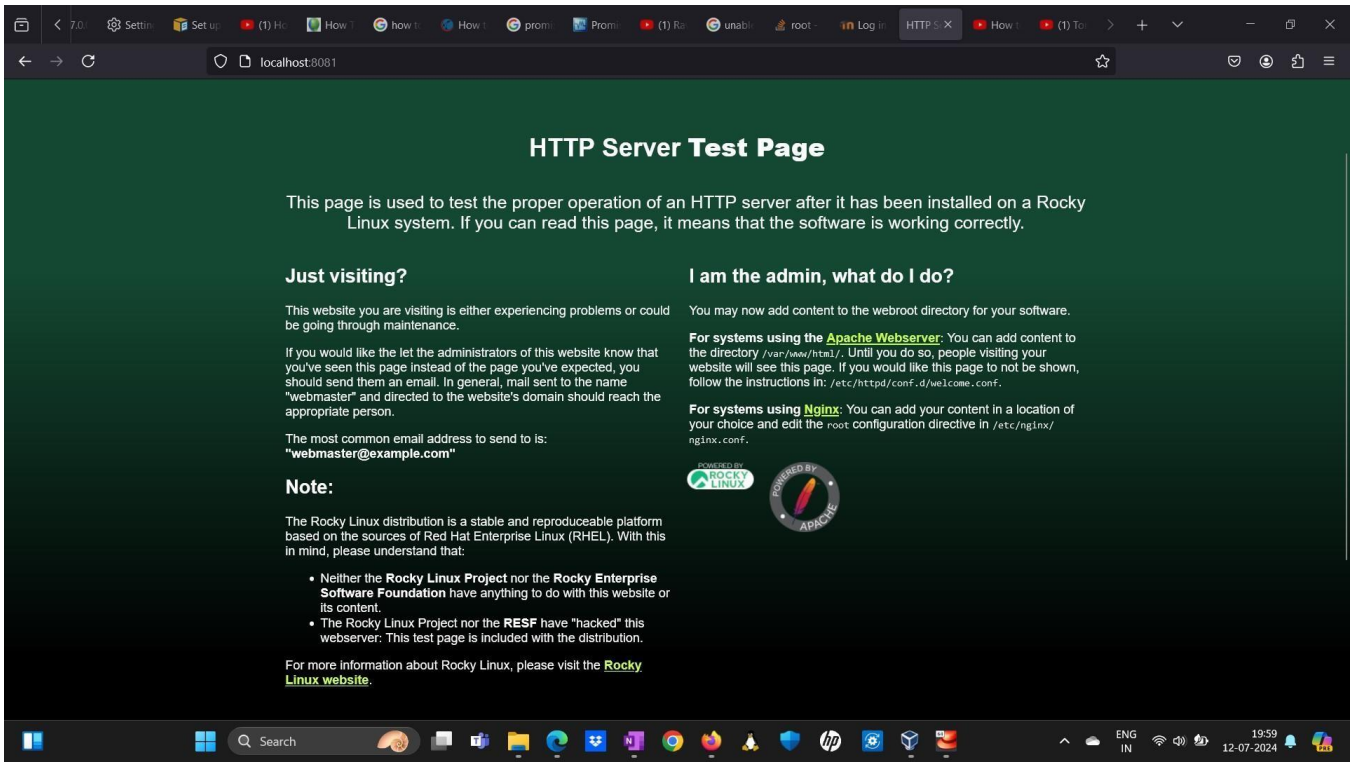
6.1 Installing Apache



We installed apache using the command:
“dnf install httpd”

```
it login: root
Password:
Last login: Mon Jul 15 10:55:59 on tty1
[root@it ~]#
[root@it ~]# systemctl start httpd
[root@it ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
  Drop-In: /etc/systemd/system/httpd.service.d
           └─php-fpm.conf
   Active: active (running) since Tue 2024-07-16 10:04:51 CEST; 6s ago
     Docs: man:httpd.service(8)
 Main PID: 1054 (httpd)
   Status: "Started, listening on: port 80"
    Tasks: 213 (limit: 24884)
   Memory: 40.4M
   CGroup: /system.slice/httpd.service
           ├─1054 /usr/sbin/httpd -DFOREGROUND
           ├─1060 /usr/sbin/httpd -DFOREGROUND
           ├─1061 /usr/sbin/httpd -DFOREGROUND
           ├─1062 /usr/sbin/httpd -DFOREGROUND
           └─1064 /usr/sbin/httpd -DFOREGROUND

Jul 16 10:04:50 it.sicherheit systemd[1]: Starting The Apache HTTP Server...
Jul 16 10:04:51 it.sicherheit systemd[1]: Started The Apache HTTP Server.
Jul 16 10:04:51 it.sicherheit httpd[1054]: Server configured, listening on: port 80
[root@it ~]#
[root@it ~]#
```

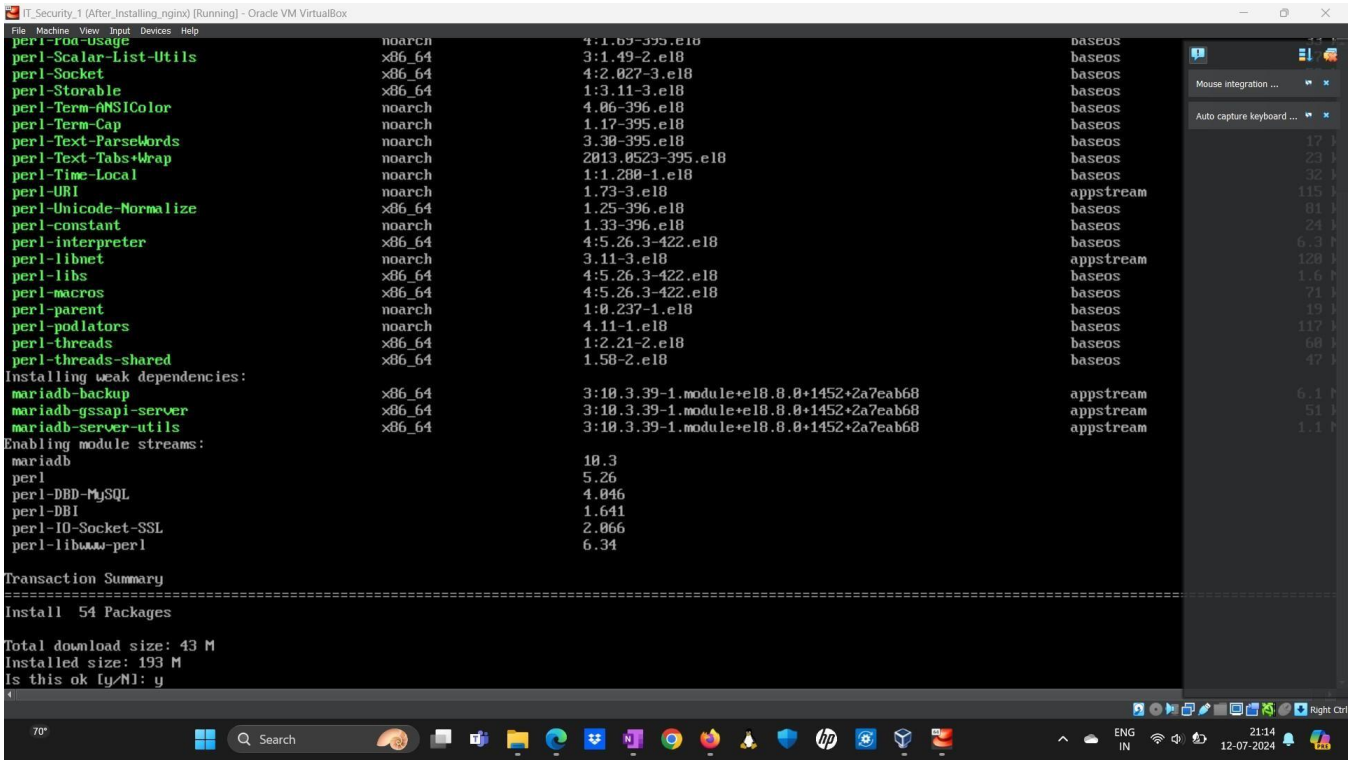
We have installed and run the Apache server successfully.



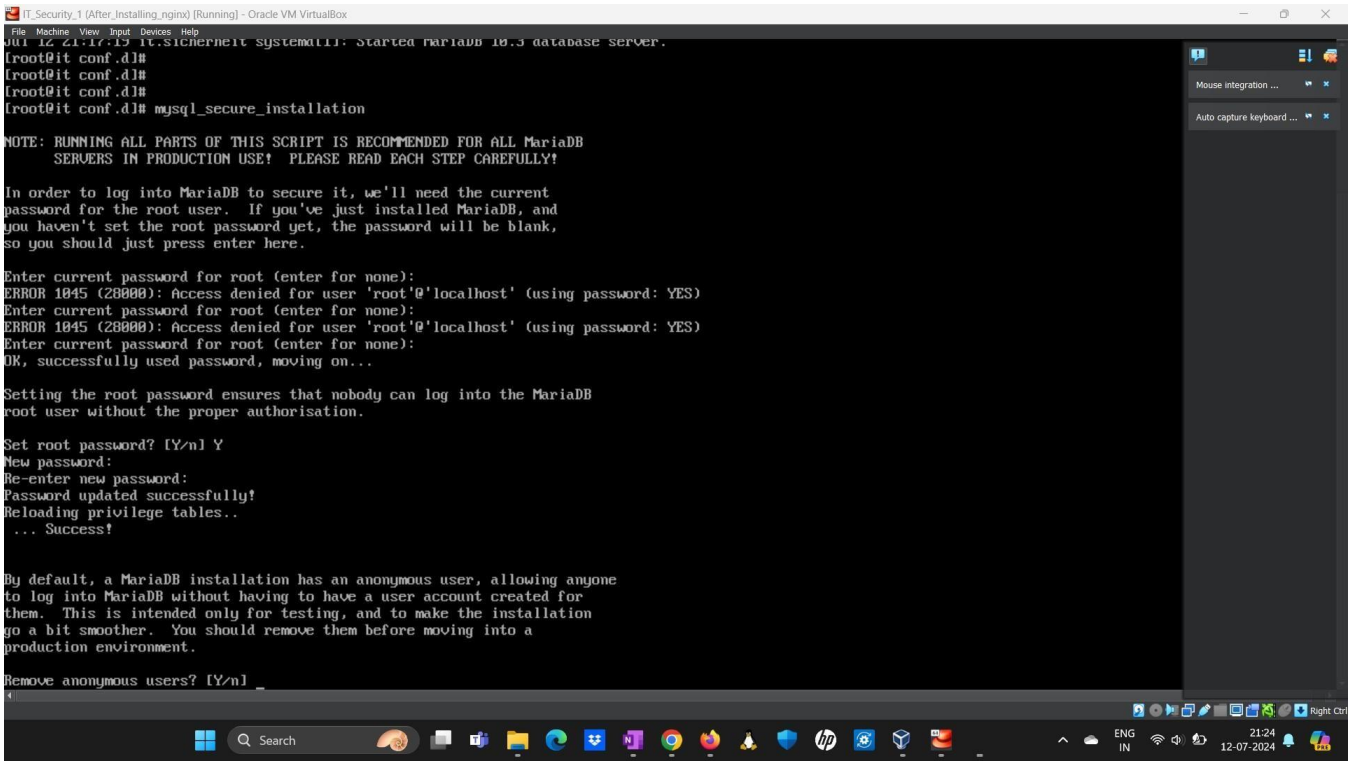
We can see the web page running successfully with the installed Apache server.

6.2 Installing Mariadb Database

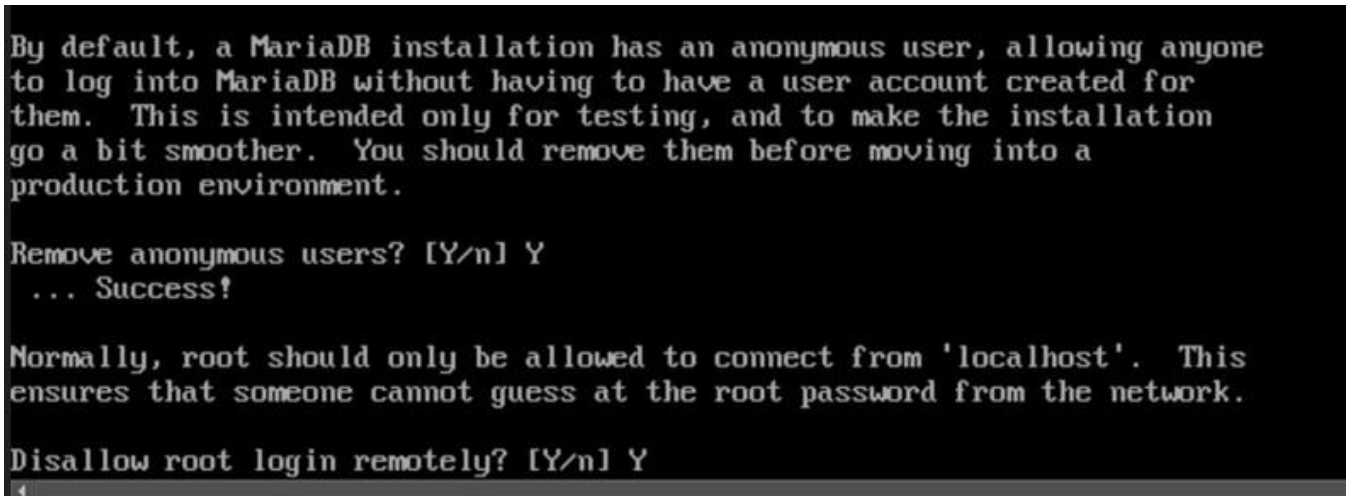
Now, we will install a mariadb database in order to store our database and the associated tables that we will use with wordpress. We install it using:
“dnf install mariadb-server”



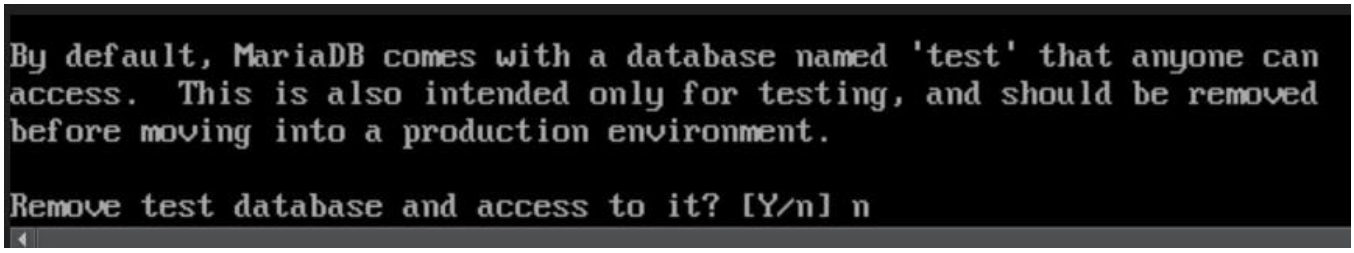
On prompting to add a password (optional), we added a password to make our database more secure:



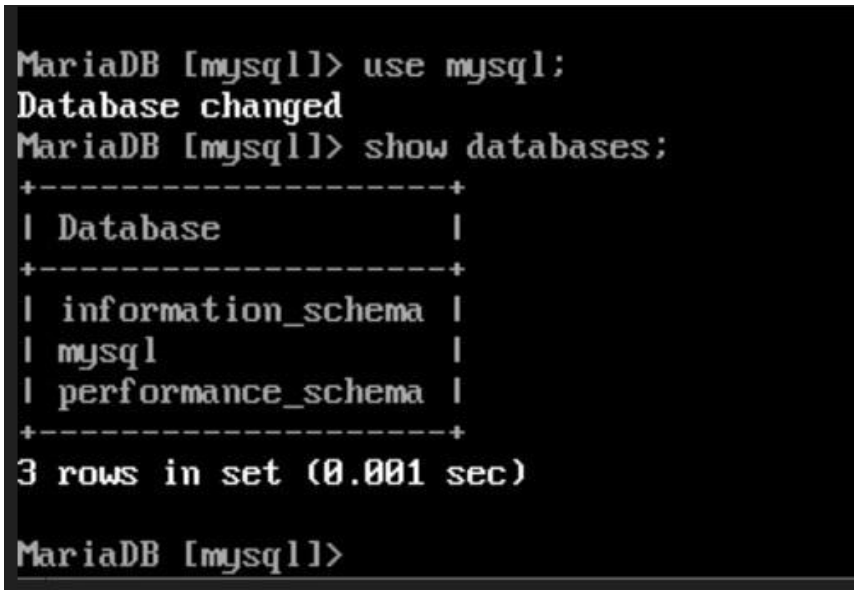
Removing anonymous users because only I want to access my database:



Removing the test database because it can be accessed by anyone.



After the above steps, mariadb installation was successful. Now, we will check the initial databases that are present in mariadb:



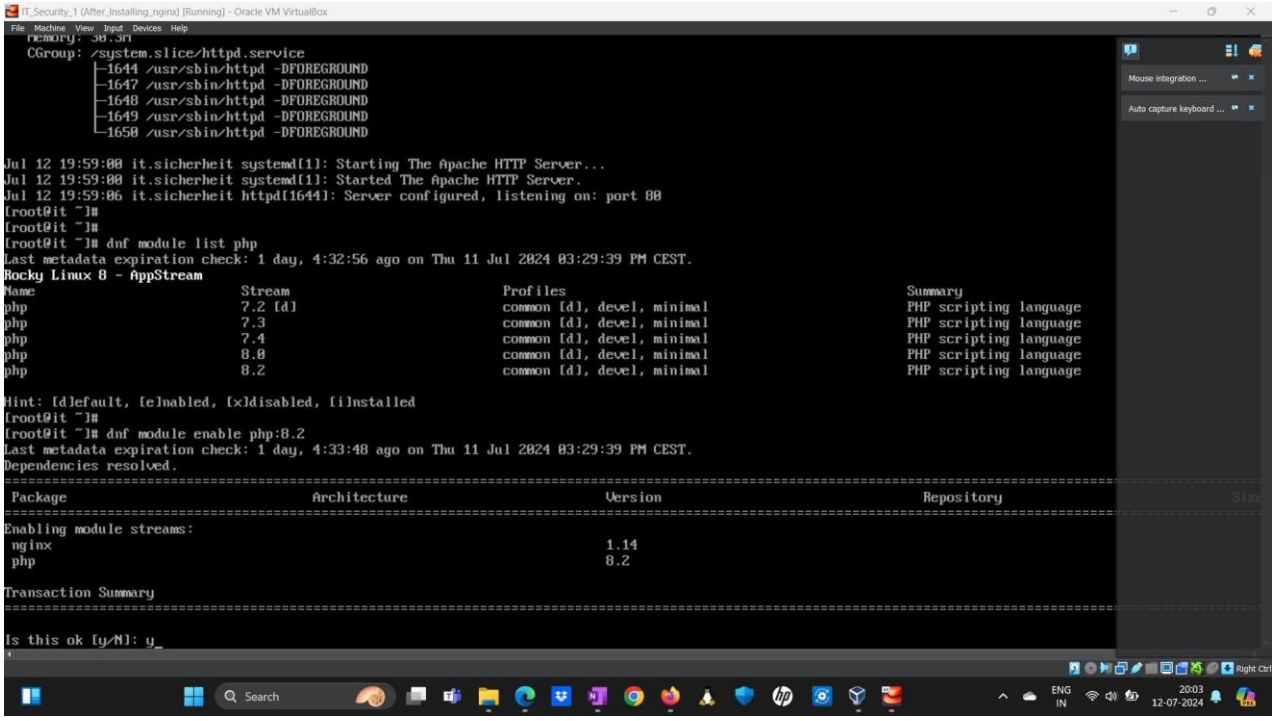
Now, we can check the default tables in mariadb:

```
MariaDB [mysql]> show tables;
+---------------------------+
| Tables_in_mysql           |
+---------------------------+
| column_stats              |
| columns_priv              |
| db                        |
| event                     |
| func                      |
| general_log               |
| gtid_slave_pos            |
| help_category             |
| help_keyword              |
| help_relation             |
| help_topic                |
| host                      |
| index_stats               |
| innodb_index_stats        |
| innodb_table_stats        |
| plugin                    |
| proc                      |
| procs_priv                |
| proxies_priv              |
| roles_mapping             |
| servers                   |
| slow_log                  |
| table_stats               |
| tables_priv               |
| time_zone                 |
| time_zone_leap_second     |
| time_zone_name            |
| time_zone_transition      |
| time_zone_transition_type |
| transaction_registry      |
| user                      |
+---------------------------+
31 rows in set (0.004 sec)

MariaDB [mysql]>
```

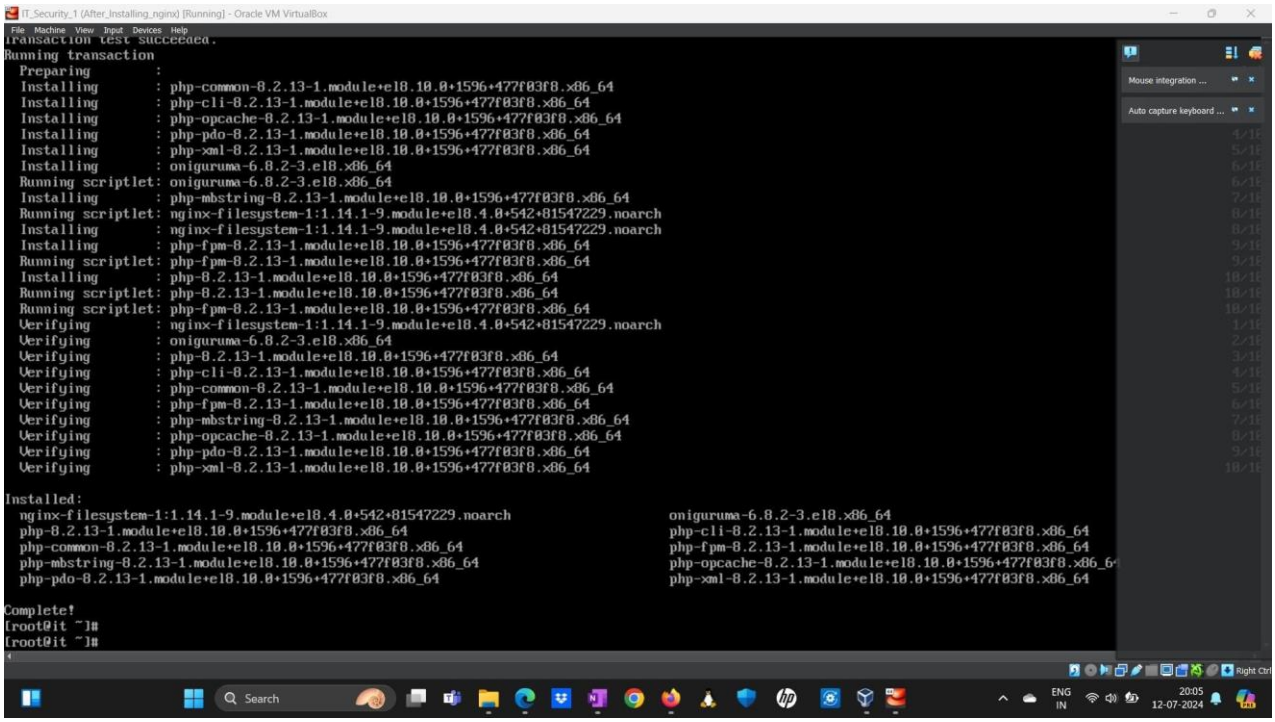
6.3 Installing PHP

PHP is a scripting language for serverside scripting. Serverside scripting is different from client side scripting. For example, when you open a moodle website as a user, you connect to the server. You have a firefox browser and apache web server. These both communicate. You have a javascript on the web browser (client side scripting), and this will communicate with apache (serverside scripting). Moodle will send some code and that will execute on your server (apache). Database lookup is done on server side - will query your database and when your send a request, it will query the database and sends back the data.

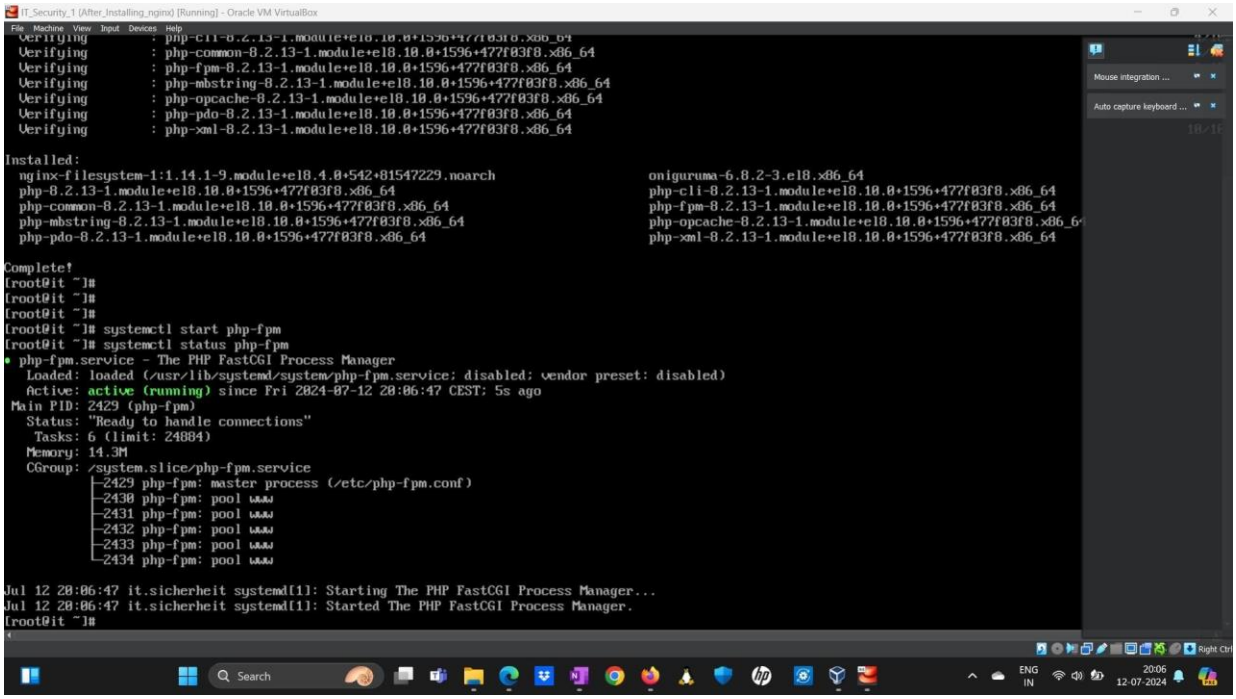
Now, we are checking the different versions of PHP and installing the latest version using:
“dnf module list php”



Installation of PHP was successful:



Now, we will verify if PHP is running successfully or not using the:
“systemctl start php-fpm” - to start php
“systemctl status php-fpm” - to check the current status of php.



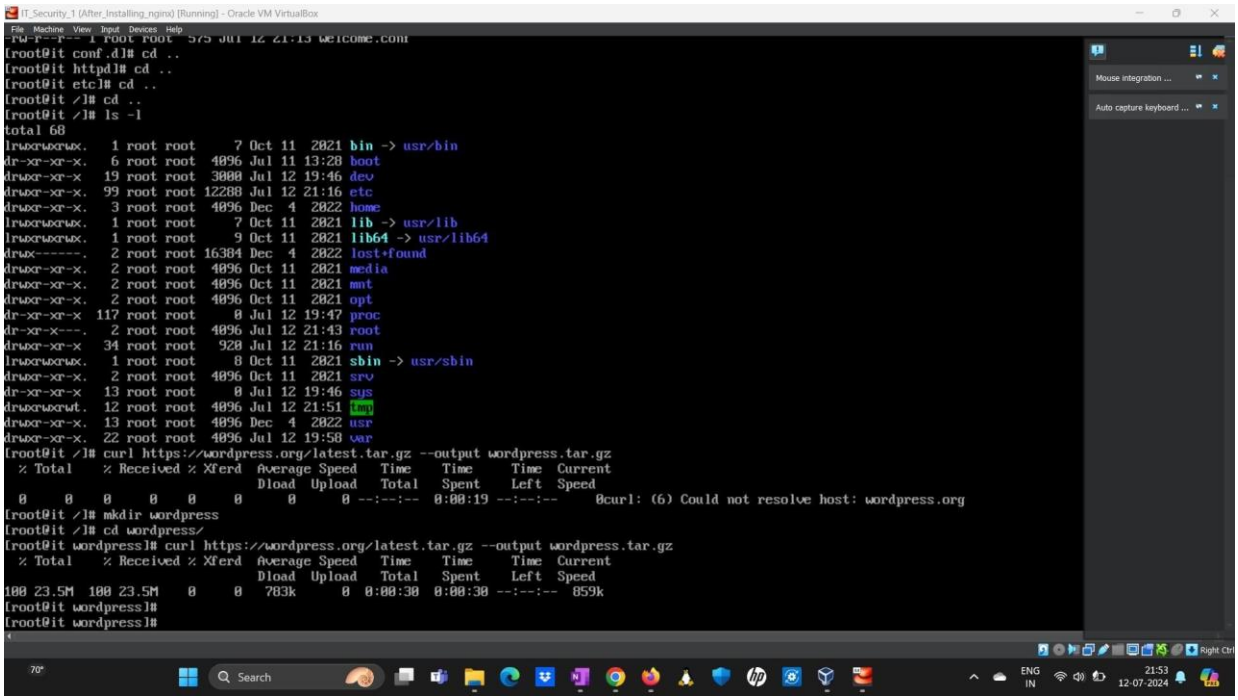
7 Wordpress

7.1 Wordpress Installation

A Content Management System (CMS) is a software application that enables users to create, edit, manage, and publish digital content without needing extensive technical knowledge. CMS platforms are often used to build and manage websites, blogs, and other online content.

For testing a LAMP (Linux, Apache, MySQL/MariaDB, PHP) server, several popular CMS options are available. One of the most popular CMS's is Wordpress. Hence, we will be installing Wordpress into our LAMP server.

First, we created a folder to contain everything related to wordpress and then entered that directory using:
"mkdir wordpress"
"cd wordpress"
After that, we installed the latest gz file (similar to zip files) of the latest wordpress version from the wordpress website using the curl command:
"curl https://wordpress.org/latest.tar.gz –output wordpress.tar.gz"



To extract the files just downloaded:
"tar xf wordpress.tar.gz"
After that, we change the ownership of wordpress to apache so that we can use it on our web server. We do this using the command:
"chown -R apache:apache wordpress/"

```
[root@it wordpress]#
[root@it wordpress]#
[root@it wordpress]# tar xf wordpress.tar.gz
[root@it wordpress]# ls -l
total 24124
drwxr-xr-x 5 nobody nobody     4096 Jun 24 19:16 wordpress
-rw-r--r-- 1 root   root   24696391 Jul 12 21:53 wordpress.tar.gz
[root@it wordpress]#
[root@it wordpress]# ls -la
total 24132
drwxr-xr-x   3 root   root       4096 Jul 12 21:55 .
dr-xr-xr-x. 19 root   root       4096 Jul 12 21:52 ..
drwxr-xr-x   5 nobody nobody     4096 Jun 24 19:16 wordpress
-rw-r--r--   1 root   root   24696391 Jul 12 21:53 wordpress.tar.gz
[root@it wordpress]#
[root@it wordpress]# chown -R apache:apache wordpress/
[root@it wordpress]# ls -la
total 24132
drwxr-xr-x   3 root   root       4096 Jul 12 21:55 .
dr-xr-xr-x. 19 root   root       4096 Jul 12 21:52 ..
drwxr-xr-x   5 apache apache     4096 Jun 24 19:16 wordpress
-rw-r--r--   1 root   root   24696391 Jul 12 21:53 wordpress.tar.gz
[root@it wordpress]#
```

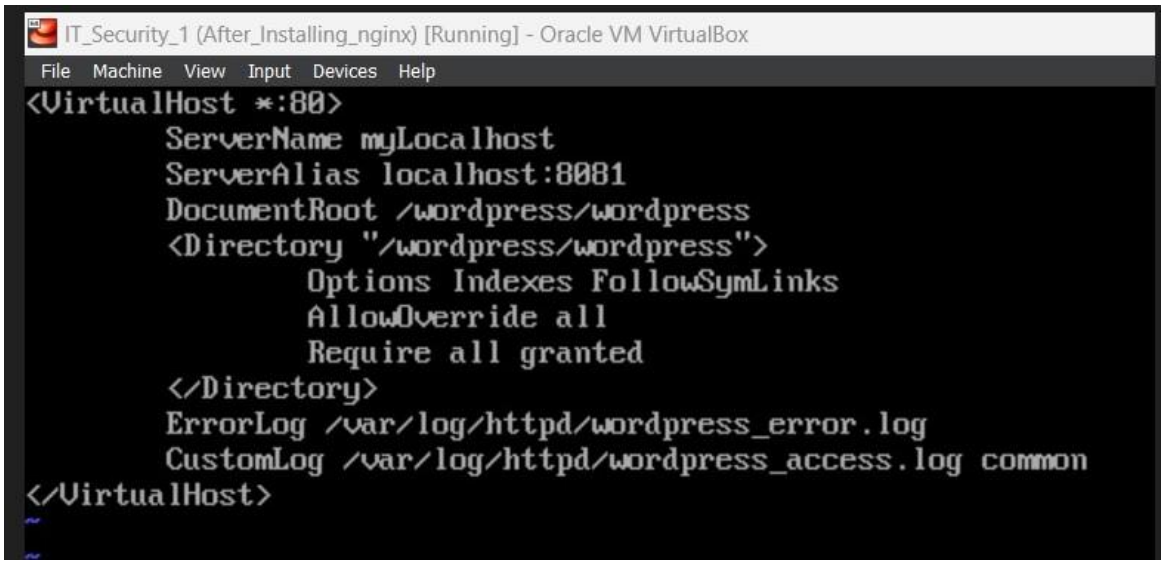
Now, we are changing the wordpress files' and directories permissions to according to what is recommended by wordpress. This is:

1. **rwxr-xr-x** / chmod 755 for all the directories: This stands for 7 for all the owners (rwx or read, write, and execute), 5 for the groups (r-x or read and execute only), and 5 for others (r-x or read and execute only). Owner has full access to the directories, but groups and others can only read and execute the directories but not change them.
2. **rw-r--r--** / chmod 644 for all the files: This stands for 6 for all the owners (rw- or read, write, but no execute), 4 for the groups (read only), and 4 for others (read only). Owner can read and write but not execute the files, and groups and others can only read the files.

We are doing the above process using the following commands:
"find wordpress/wordpress -type d -exec chmod 755 {} \;"
"find wordpress/wordpress -type f -exec chmod 644 {} \;"

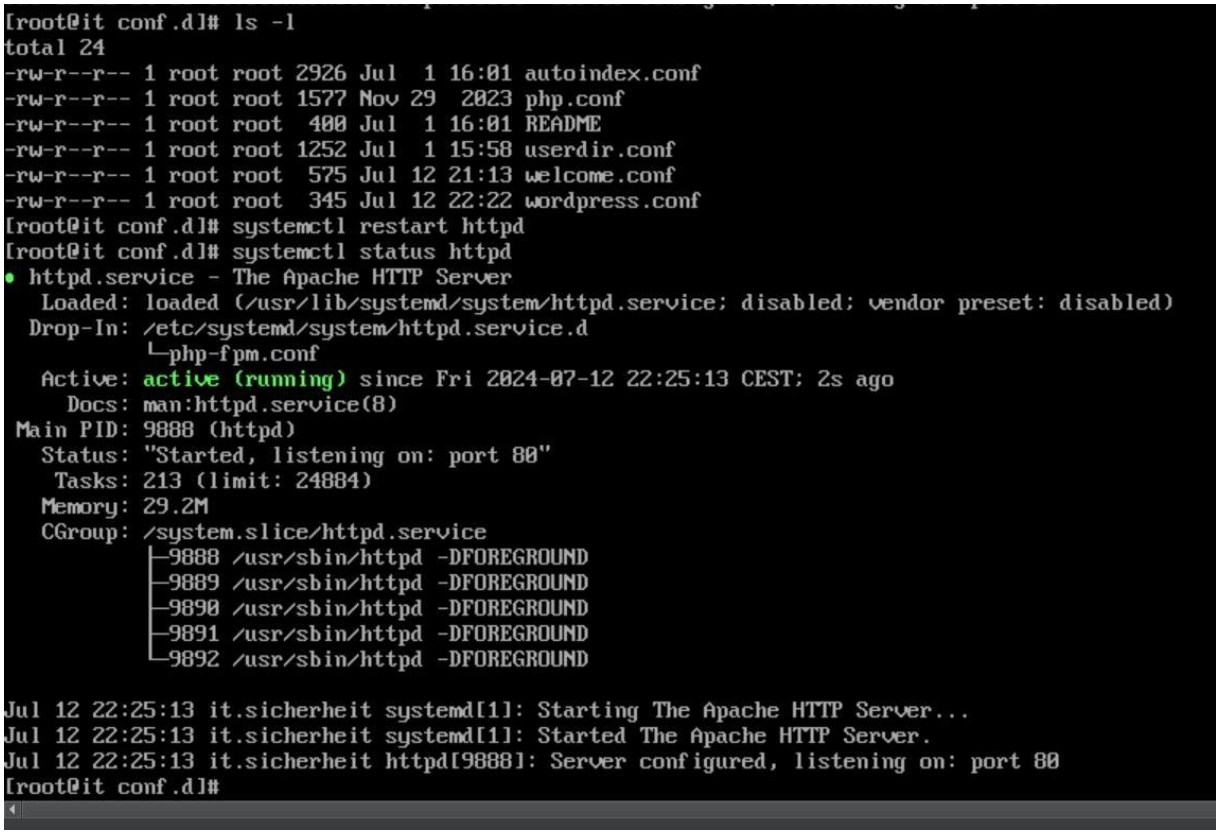
```
total 24132
drwxr-xr-x   3 root   root       4096 Jul 12 21:55 .
dr-xr-xr-x. 19 root   root       4096 Jul 12 21:52 ..
drwxr-xr-x   5 apache apache     4096 Jun 24 19:16 wordpress
-rw-r--r--   1 root   root   24696391 Jul 12 21:53 wordpress.tar.gz
[root@it wordpress]#
[root@it wordpress]#
[root@it wordpress]# find /wordpress/wordpress -type d -exec chmod 755 {} \;
[root@it wordpress]# find /wordpress/wordpress -type f -exec chmod 644 {} \;
[root@it wordpress]# _
```

After this, we go the the apache configuration directory. Here we create a folder called as **“wordpress.conf”**
This file will contain information about how to configure and use the wordpress package on the apache server. The contents of the wordpress.conf file are listed below:




```
[root@it ~]#
[root@it ~]# cd /etc/httpd/
[root@it httpd]# ls -l
total 16
drwxr-xr-x 2 root root 4096 Jul 14 16:01 conf
drwxr-xr-x 2 root root 4096 Jul 14 13:52 conf.d
drwxr-xr-x 2 root root 4096 Jul 12 20:05 conf.modules.d
lrwxrwxrwx 1 root root   19 Jul  1 16:01 logs -> ../../var/log/httpd
lrwxrwxrwx 1 root root   29 Jul  1 16:01 modules -> ../../usr/lib64/httpd/modules
lrwxrwxrwx 1 root root   10 Jul  1 16:01 run -> /run/httpd
drwxr-xr-x 2 root root 4096 Jul 13 16:15 ssl
lrwxrwxrwx 1 root root   19 Jul  1 16:01 state -> ../../var/lib/httpd
[root@it httpd]#
[root@it httpd]# cd conf.d
[root@it conf.d]# ls -l
total 24
-rw-r--r-- 1 root root 2926 Jul  1 16:01 autoindex.conf
-rw-r--r-- 1 root root 1577 Nov 29  2023 php.conf
-rw-r--r-- 1 root root  400 Jul  1 16:01 README
-rw-r--r-- 1 root root 1252 Jul  1 15:58 userdir.conf
-rw-r--r-- 1 root root  575 Jul 12 21:13 welcome.conf
-rw-r--r-- 1 root root  345 Jul 14 13:52 wordpress.conf
[root@it conf.d]# _
```

After this, we have to restart the apache server:



After successfully restarting the apache server, we can confirm that the wordpress has been configured successfully with apache.

7.2 MariaDB Database Creating for Wordpress

Login into the mariadb database that we just created. After that, we create a database called as:
Database: **“itsec_1”**
After that, we create a user who can use the database using:
“create user ‘itsec_user1’@’localhost’ identified by ‘Alfa11”!Martin35&’;”
Note that the end of the above command is the password (you can choose any password you want for the database that you just created.

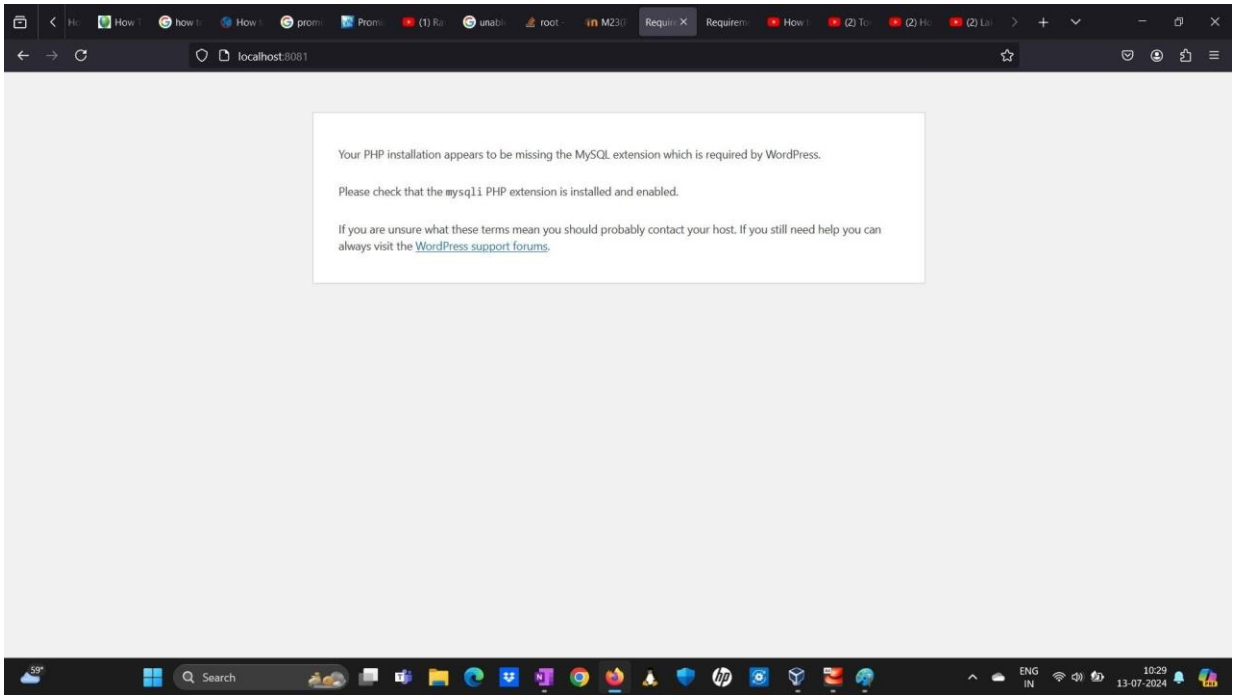
```
[root@it ~]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.3.39-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]>
MariaDB [(none)]> create database itsec_1;
Query OK, 1 row affected (0.015 sec)

MariaDB [(none)]> create user 'itsec_user1'@'localhost' identified by 'Alfa11"!Martin35&';
Query OK, 0 rows affected (0.005 sec)

MariaDB [(none)]>
```

Now, we are granting all privileges of the database we created to the user we created. After that, we do **“flush privileges”** to refresh all the privileges.

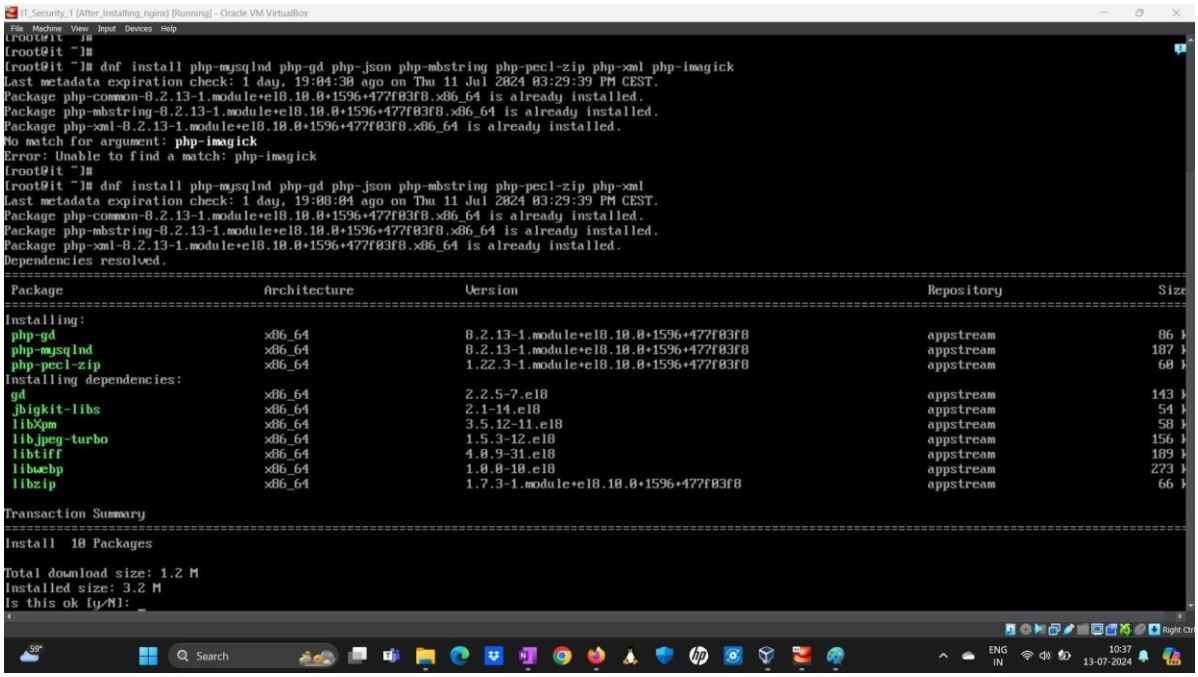
```
MariaDB [(none)]>
MariaDB [(none)]> create database itsec_1;
Query OK, 1 row affected (0.015 sec)

MariaDB [(none)]> create user 'itsec_user1'@'localhost' identified by 'Alfa11"!Martin35&';
Query OK, 0 rows affected (0.005 sec)

MariaDB [(none)]> grant all privileges on itsec_1.* TO 'itsec_user1'@'localhost';
Query OK, 0 rows affected (0.005 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.006 sec)

MariaDB [(none)]> _
```

7.3 PHP Error

Now, we check our web server using:
“localhost:8081” or **“127.0.0.1:8081”**
We can see the following error.



There are a few PHP dependencies that wordpress needs that we still need to install and run. For this purpose, we do:
“dnf install php-mysqlnd php-gd php-json php-mbstring php-pecl-zip php-xml”



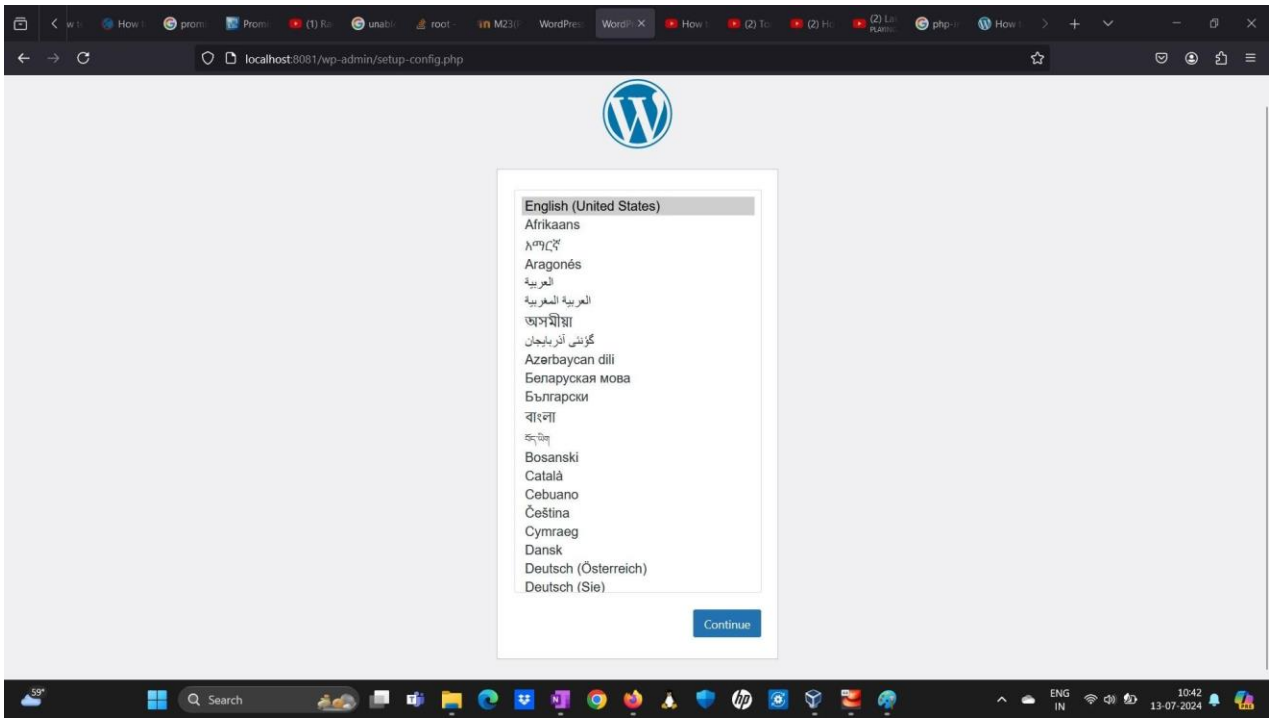
Install all the packages, and then restart everything using:
“systemctl restart httpd”
“systemctl restart php-fpm”

“systemctl restart mariadb”
“systemctl restart snapd”

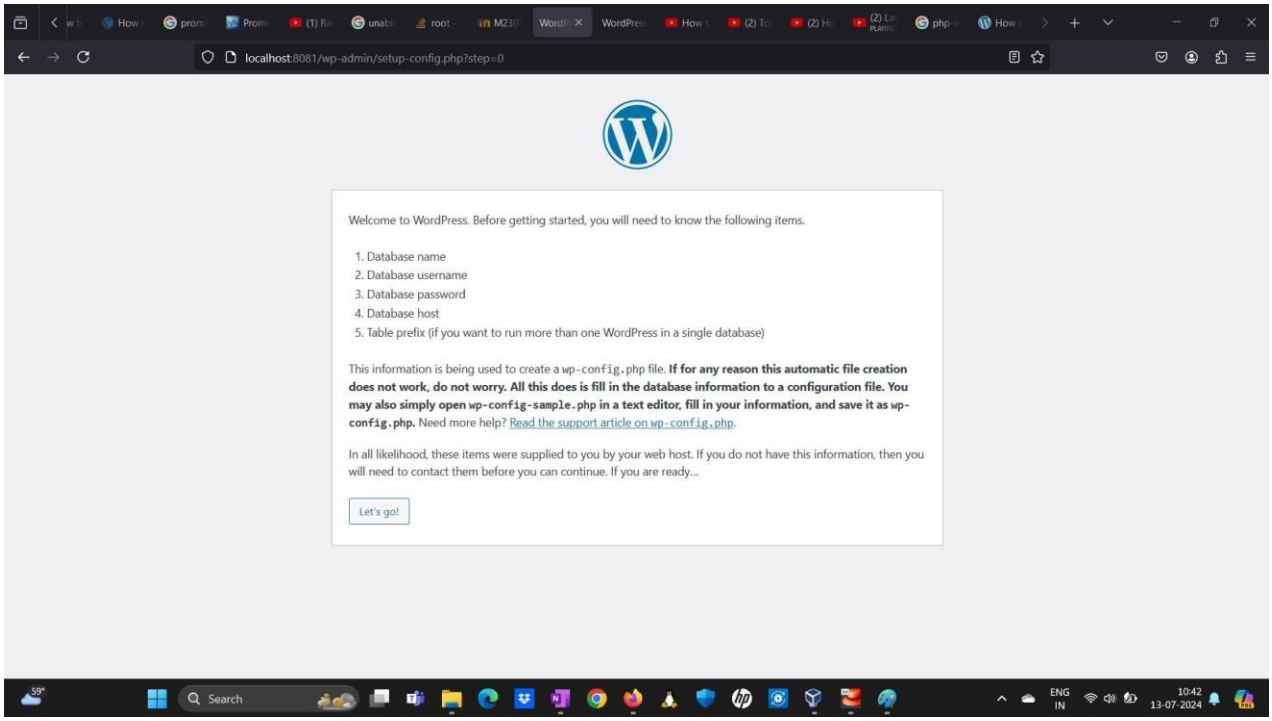
7.4 Wordpress Settings

After that, refresh the browser link again and you will be able to see wordpress:

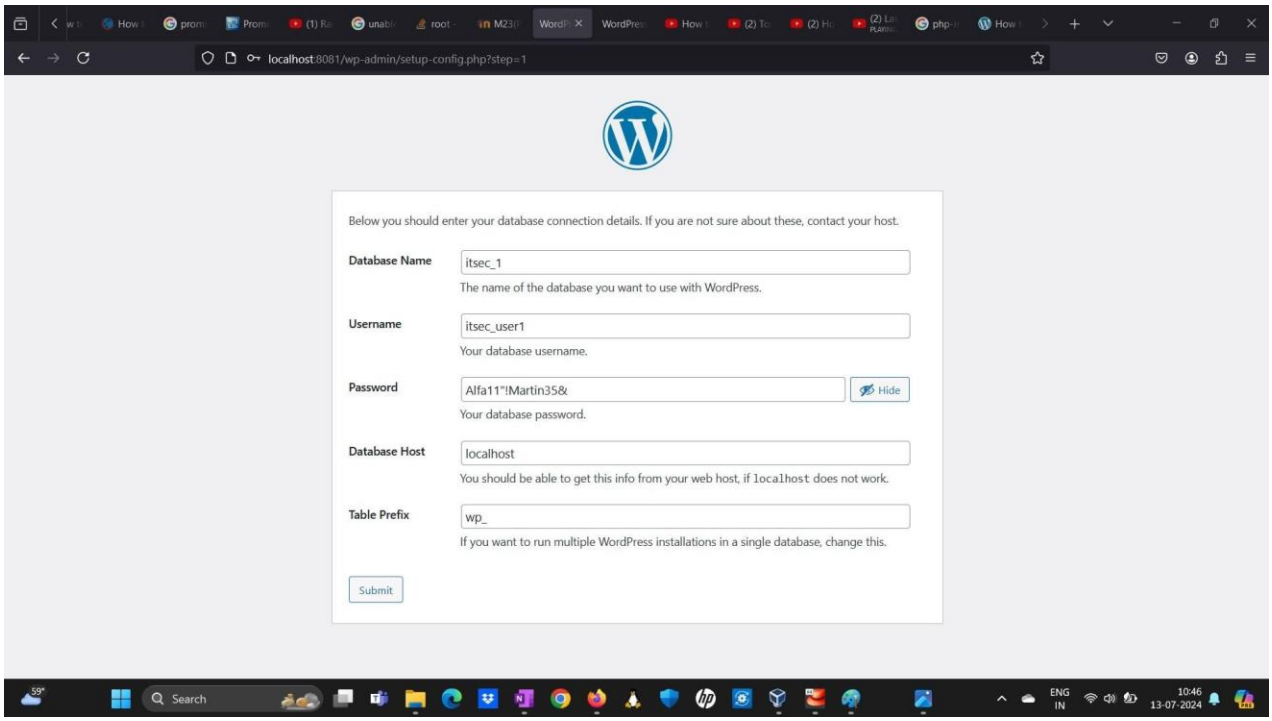
Select the language.



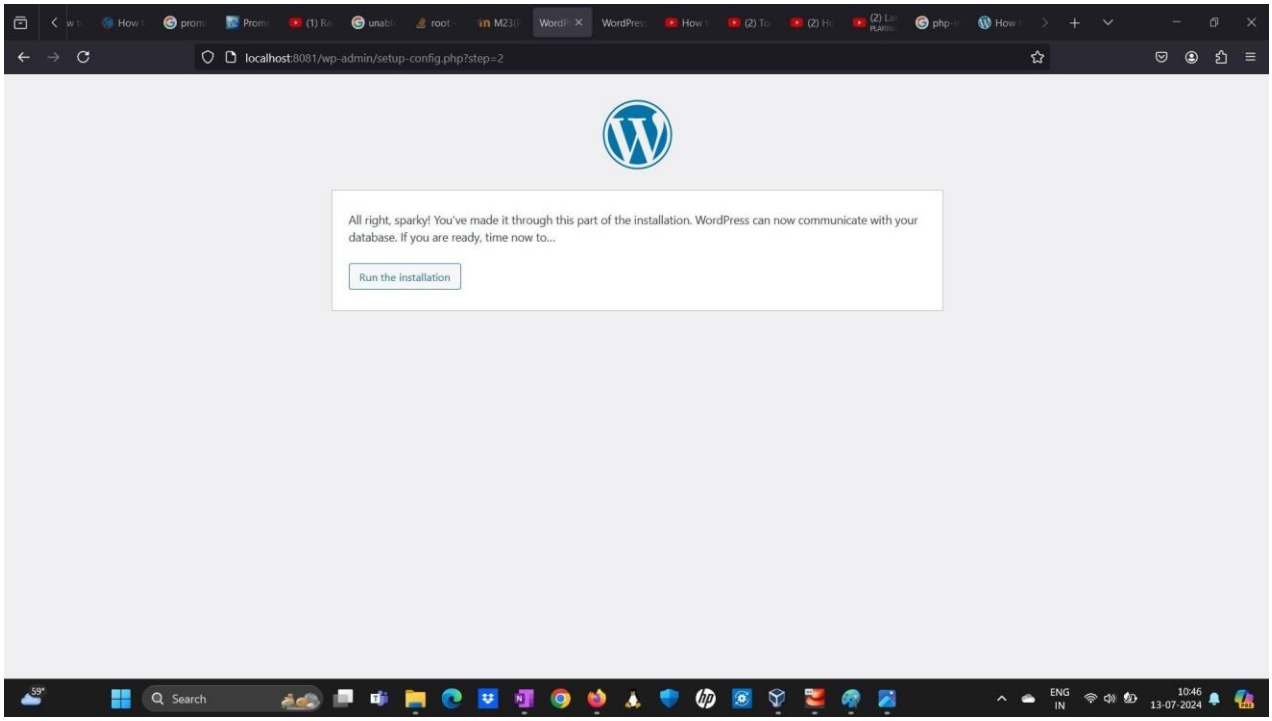
It will then give you a prompt about the database information that we need:



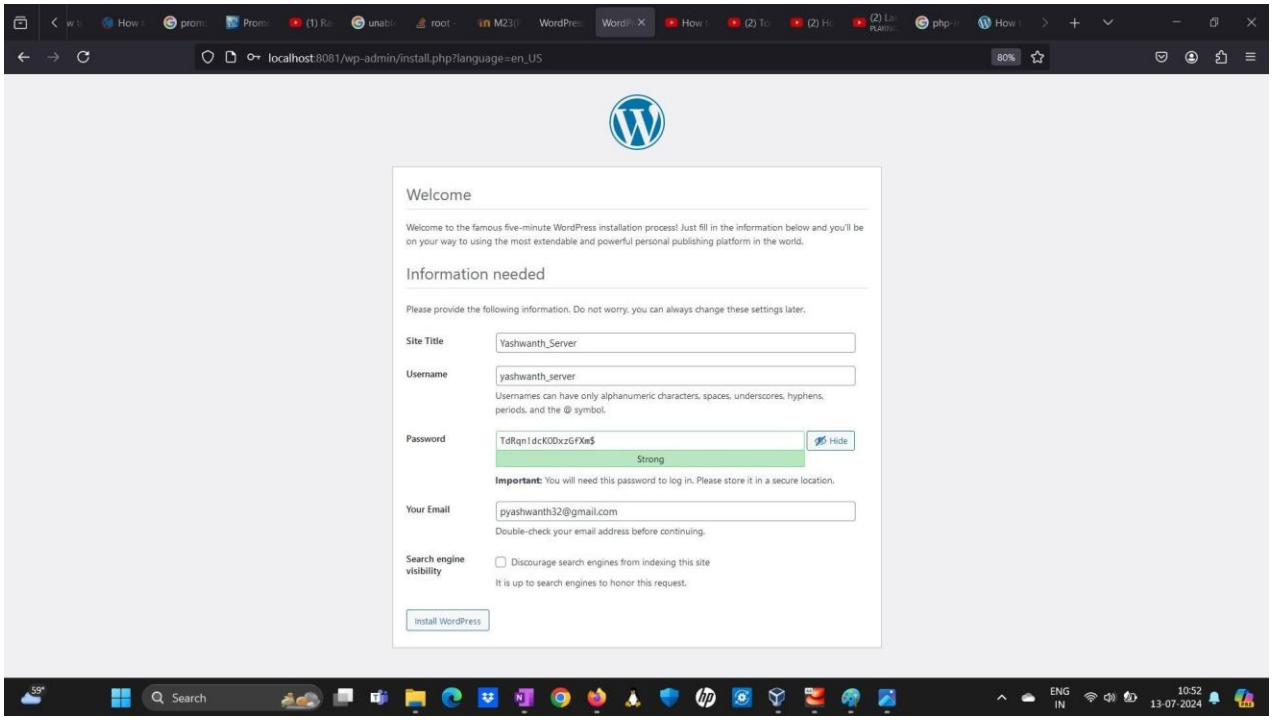
Here, we specify the details of the database that we just created on mariadb.



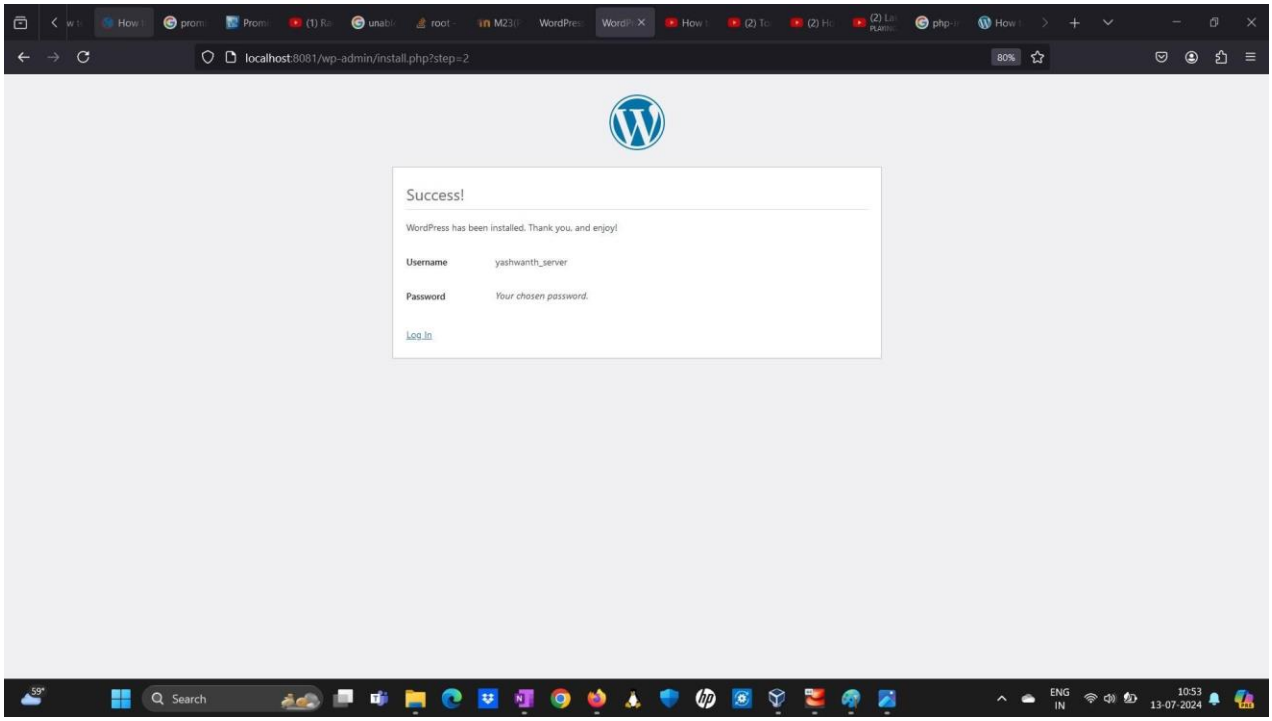
After this installation, we are ready to use Wordpress!



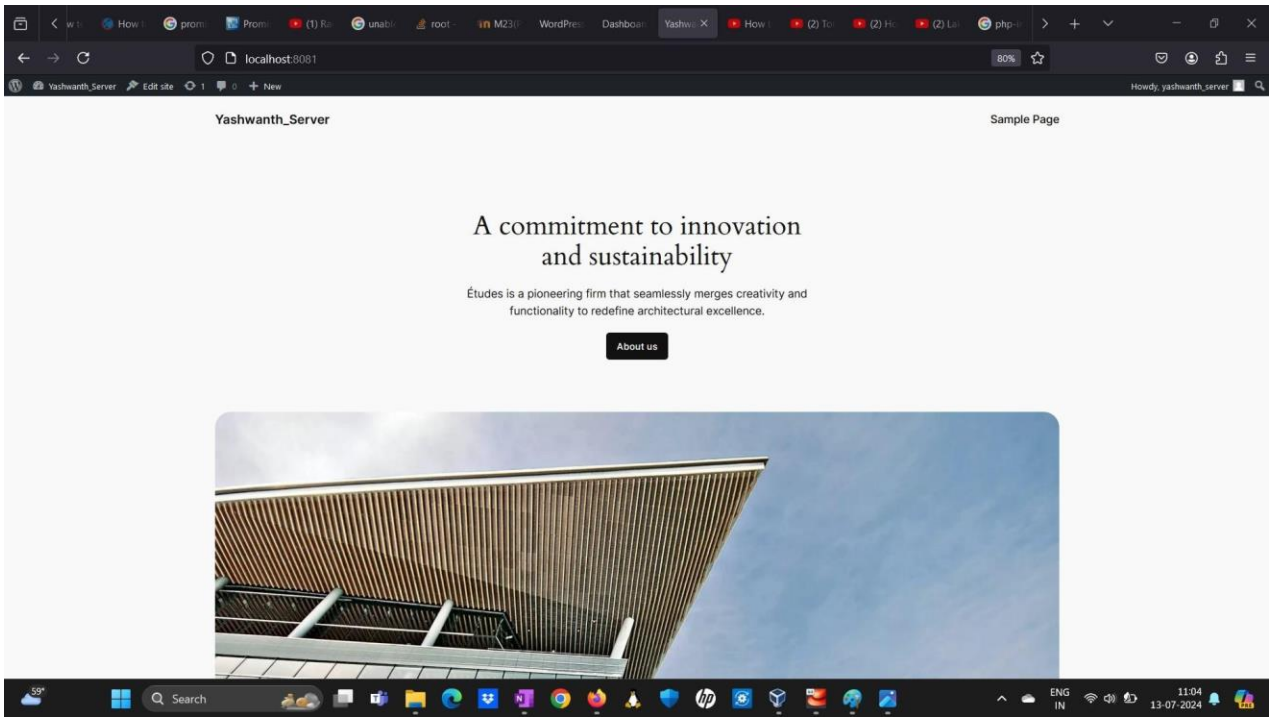
Now, we do the wordpress installation process by providing the details mentioned below:



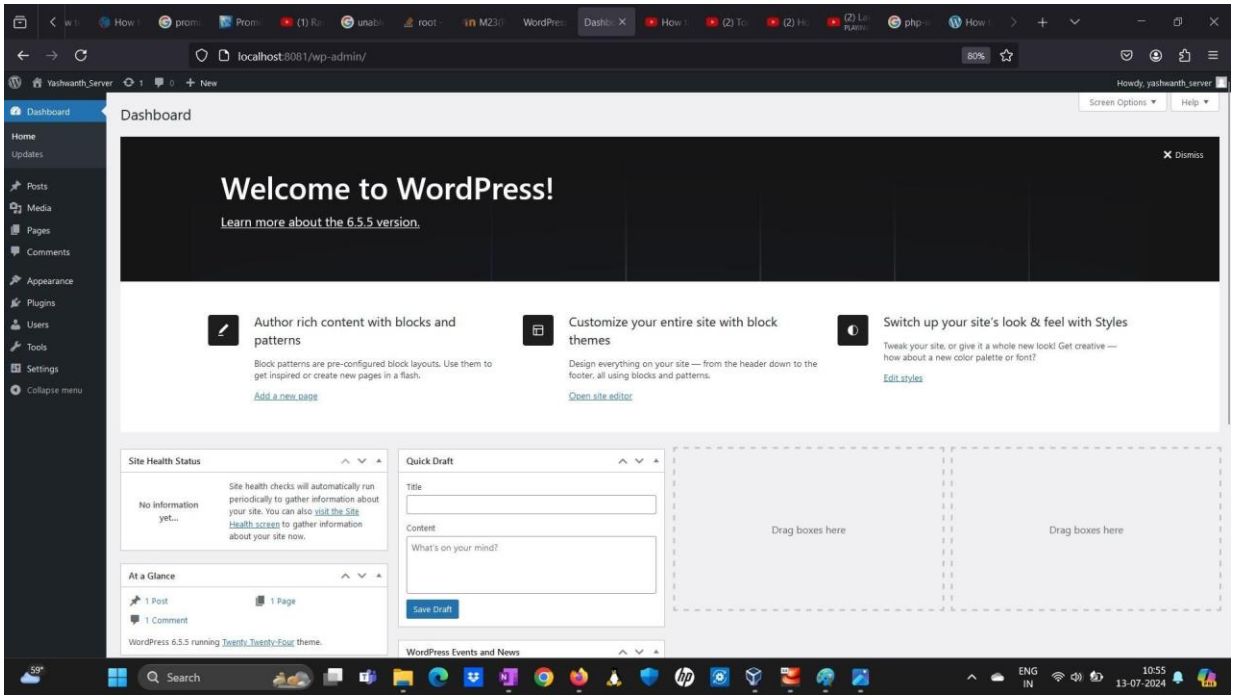
Wordpress will finally be installed:



On refreshing the page:
“localhost:8081” or “127.0.0.1:8081”



On the wordpress admin page, you have access to settings of wordpress:
“localhost:8081/wp-admin/” or “127.0.0.1:8081/wp-admin/”

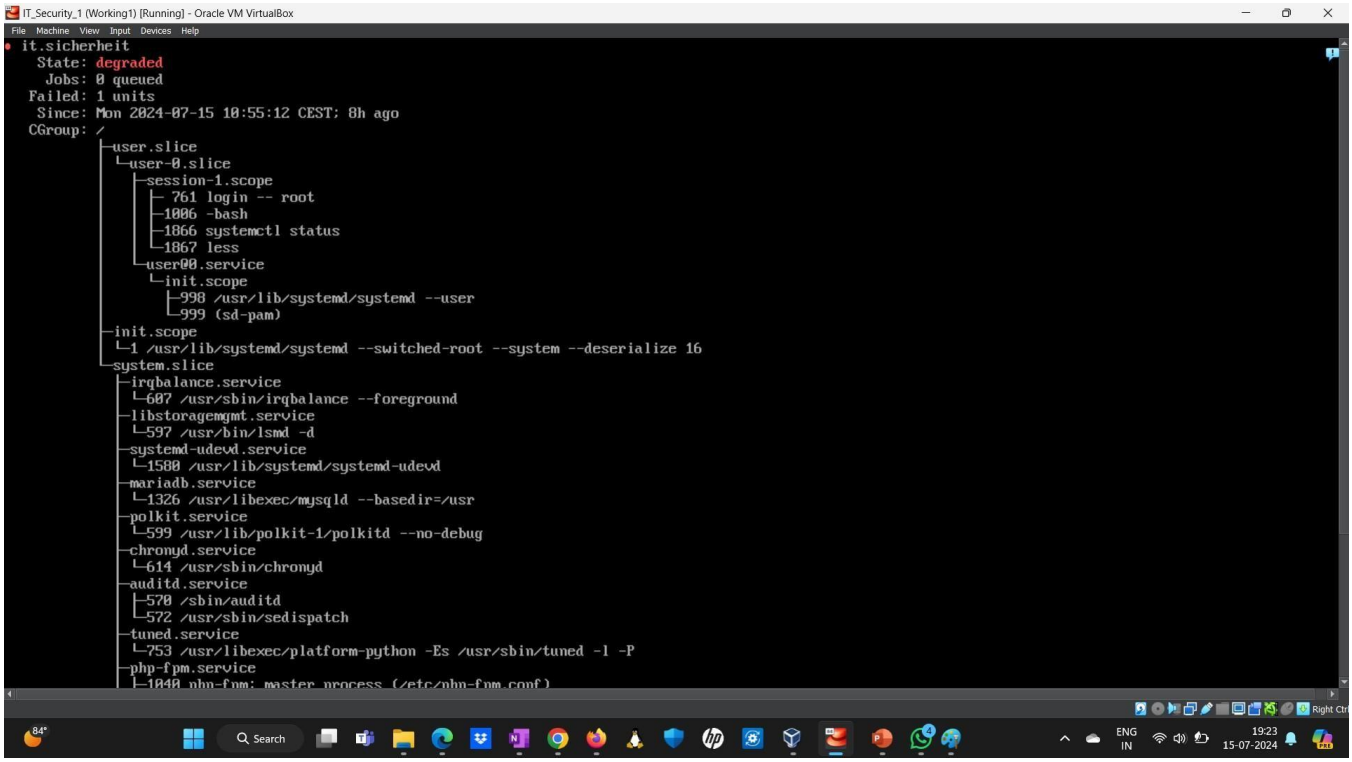

localhost:8081/wp-admin/
Yashwanth_Server
New
Howdy, yashwanth_server
Dashboard
Home
Updates
Posts
Media
Pages
Comments
Appearance
Plugins
Users
Tools
Settings
Collapse menu
Dashboard
Screen Options
Help
Dismiss
Welcome to WordPress!
Learn more about the 6.5.5 version.
Author rich content with blocks and patterns
Block patterns are pre-configured block layouts. Use them to get inspired or create new pages in a flash.
Add a new page
Customize your entire site with block themes
Design everything on your site — from the header down to the footer, all using blocks and patterns.
Open site editor
Switch up your site's look & feel with Styles
Tweak your site, or give it a whole new look! Get creative — how about a new color palette or font?
Edit styles
Site Health Status
No information yet...
Site health checks will automatically run periodically to gather information about your site. You can also visit the Site Health screen to gather information about your site now.
At a Glance
1 Post
1 Page
1 Comment
WordPress 6.5.5 running Twenty Twenty-Four theme.
Quick Draft
Title
Content
What's on your mind?
Save Draft
WordPress Events and News
Drag boxes here
Drag boxes here
Search
ENG
IN
10:55
13-07-2024

8 Hardening the System

Hardening is making the server as resilient as possible to outside attackers. One of the methods is to remove unused programs using dnf. One aspect of hardening a system in the IT world is removing systems we don't need because if an attacker enters a system, he needs to have as few resources as possible - more softwares, more possibilities he or she has to attack a system. This is more difficult and if you remove and reboot, then it is good. There are big packages that you can remove and identify all the unwanted programs.

“systemctl” - modern way to start, stop, enable, and disable services is using systemctl. Before they used init. But now it is systemctl.

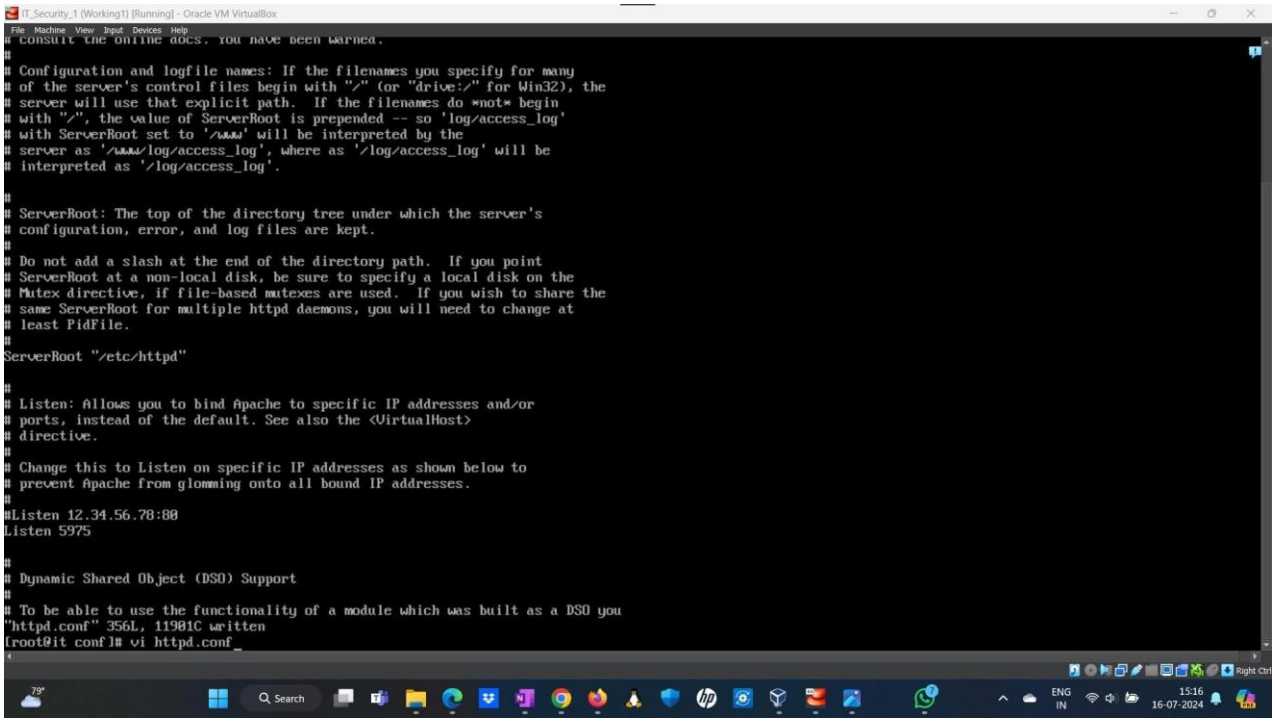
“systemctl status” lists down all the status of the entire system currently.



In the above diagram, at any given time, the lesser the number of lines the better. Each of these are running and waiting for something - and if an attacker sees these services, he/she can try to make use of the vulnerabilities in the system on one of these and it will run.

8.1 Changed the Port from 80 to 5975

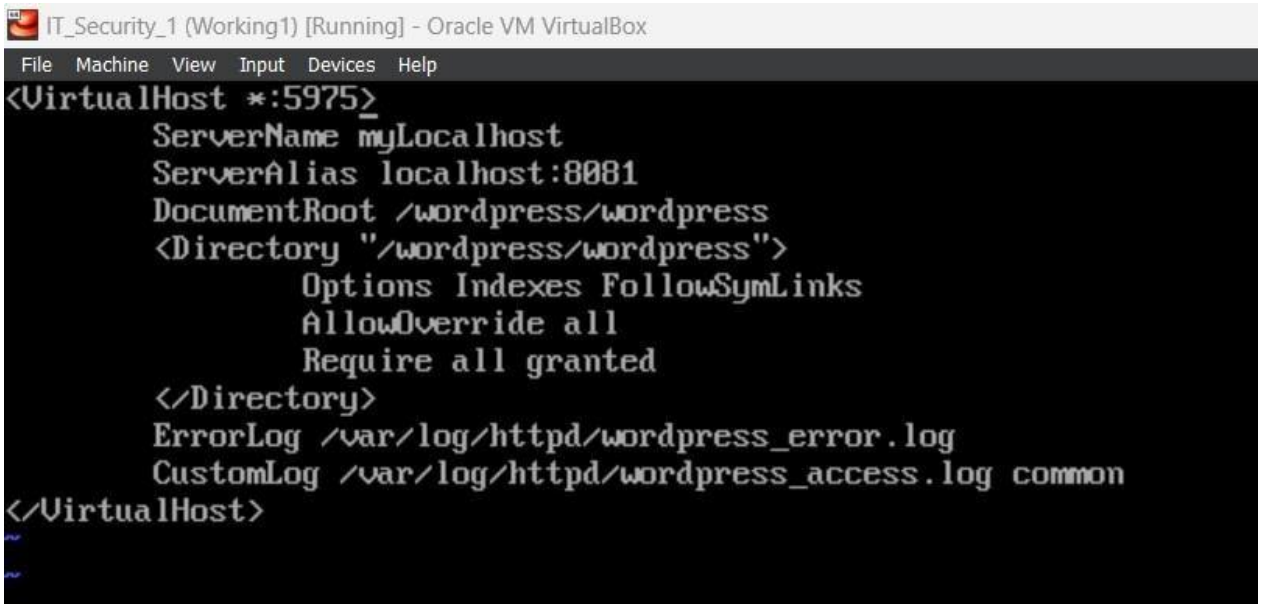
Since everyone always uses Port 80 or Port 443 for routing web traffic, I decided to harden the system by routing my web traffic from a different port. So I used the port 5975 for all the Apache web traffic routing.
In the httpd.conf file of Apache for configuring, I changed the listen port from 80 to 5975.



Since we are a virtual host server (located in my Laptop), in the wordpress.conf file present in the folder

/etc/httpd/conf.d/wordpress.conf

We change the port from 80 to port 5975.



Now, since I have installed firewalld firewall, adding the rule for allowing traffic through port 5975 too.

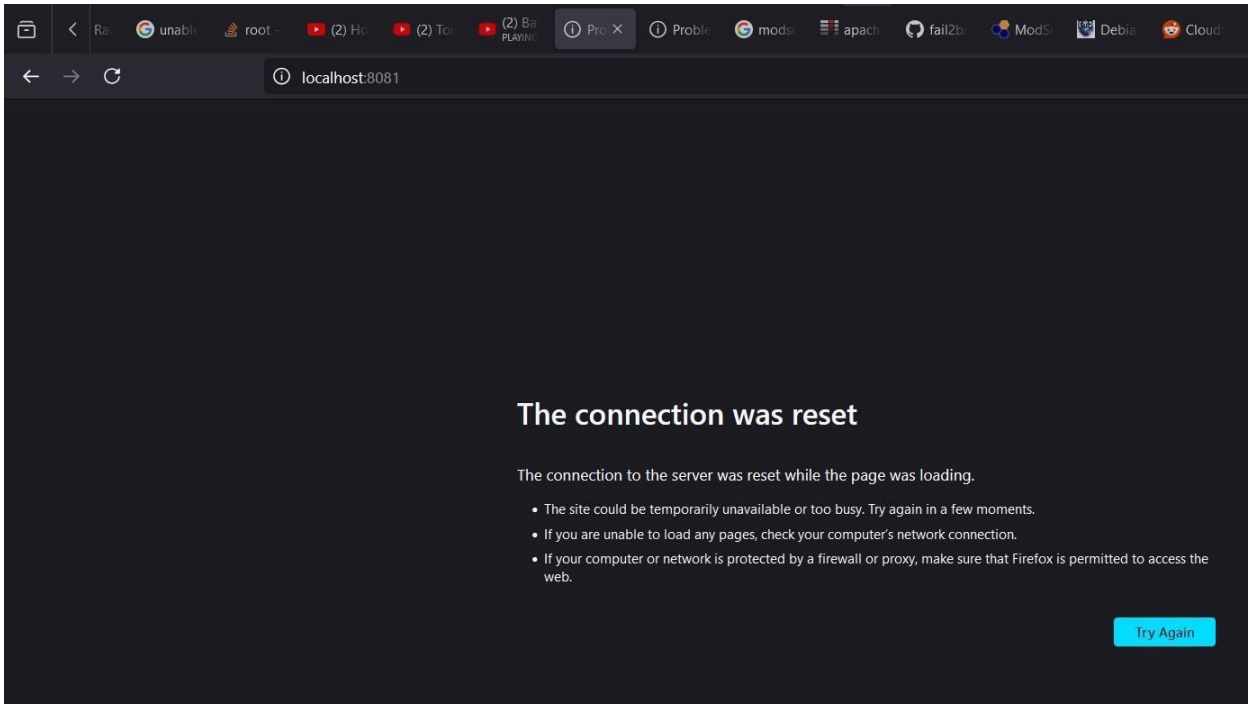
“firewall-cmd –permanent –add-port=5975/tcp”

```
[root@it conf.d]#
[root@it conf.d]#
[root@it conf.d]# firewall-cmd --permanent --add-port=5975/tcp
FirewallD is not running
[root@it conf.d]# systemctl start firewalld
[root@it conf.d]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2024-07-16 15:24:11 CEST; 4s ago
     Docs: man:firewalld(1)
 Main PID: 2478 (firewalld)
    Tasks: 2 (limit: 24804)
   Memory: 37.0M
   CGroup: /system.slice/firewalld.service
           └─2478 /usr/libexec/platform-python -s /usr/sbin/firewalld --nofork --nopid

Jul 16 15:24:10 it.sicherheit systemd[1]: Starting firewalld - dynamic firewall daemon...
Jul 16 15:24:11 it.sicherheit systemd[1]: Started firewalld - dynamic firewall daemon.
Jul 16 15:24:11 it.sicherheit firewalld[2478]: WARNING: AllowZoneDrifting is enabled. This is considered an insecure configuration option. It will be removed i

[root@it conf.d]#
[root@it conf.d]# firewall-cmd --permanent --add-port=5975/tcp
success
[root@it conf.d]#
[root@it conf.d]# firewall-cmd --reload
success
[root@it conf.d]# systemctl restart firewalld
[root@it conf.d]#
```

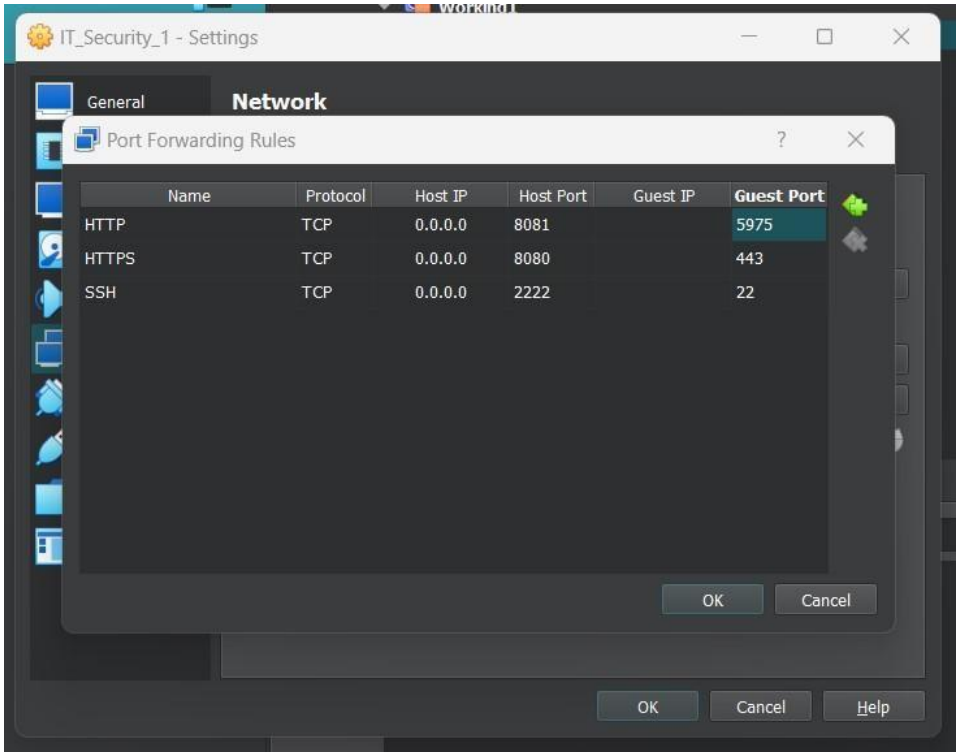
After this, change the apache listening ports so that apache can also listen to port 5975. This is done using:
“ss -tuln | grep 5975”

```
[root@it conf.d]#
[root@it conf.d]# firewall-cmd --permanent --add-port=5975/tcp
success
[root@it conf.d]#
[root@it conf.d]# firewall-cmd --reload
success
[root@it conf.d]# systemctl restart firewalld
[root@it conf.d]#
[root@it conf.d]# ss -tuln | grep 5975
[root@it conf.d]#
[root@it conf.d]#
```

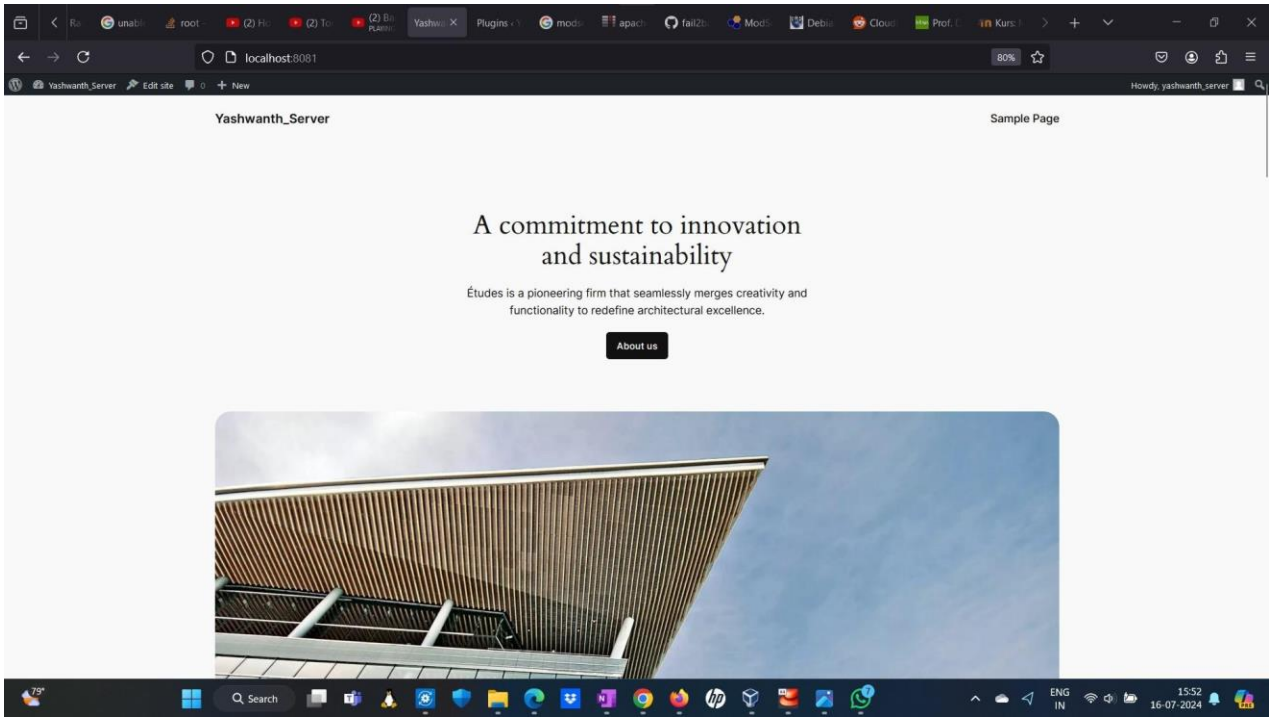
Now, on reloading our localhost:8081

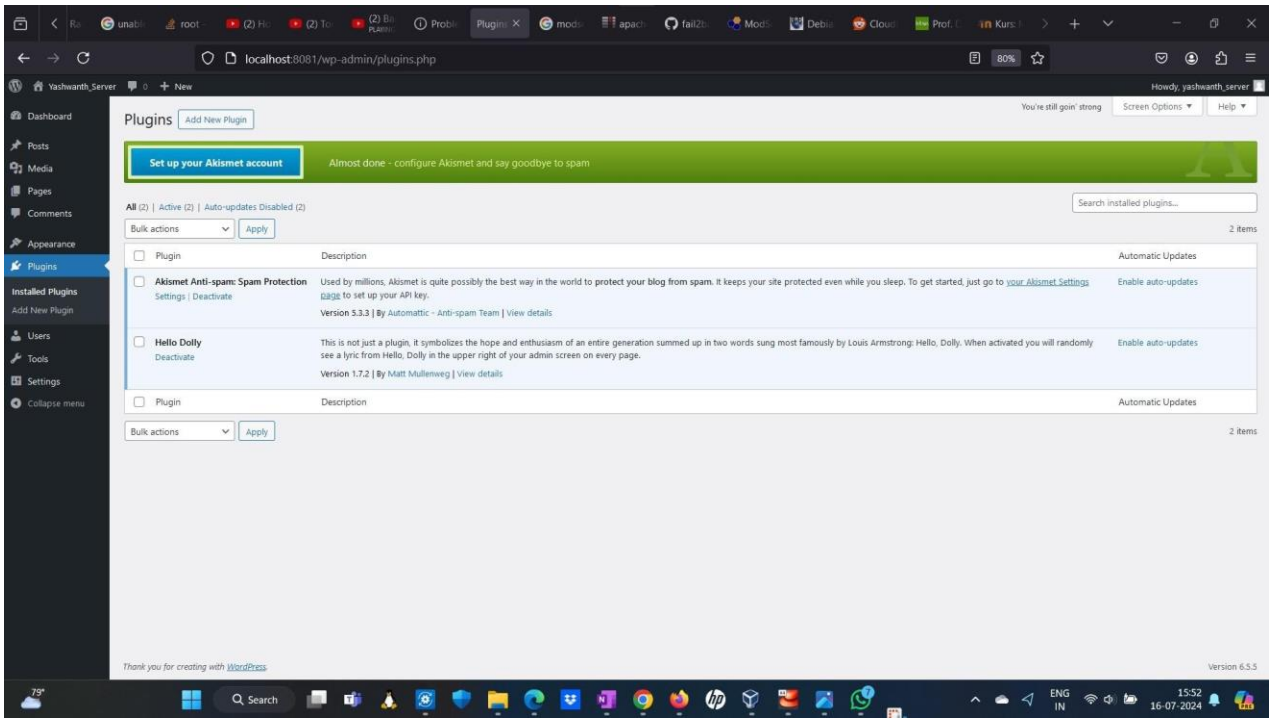


There seems to be an error. Then I changed the port forwarding settings of the VirtualBox. I changed the Guest Port from 80 to 5975.



After this, I checked and everything is working perfectly.

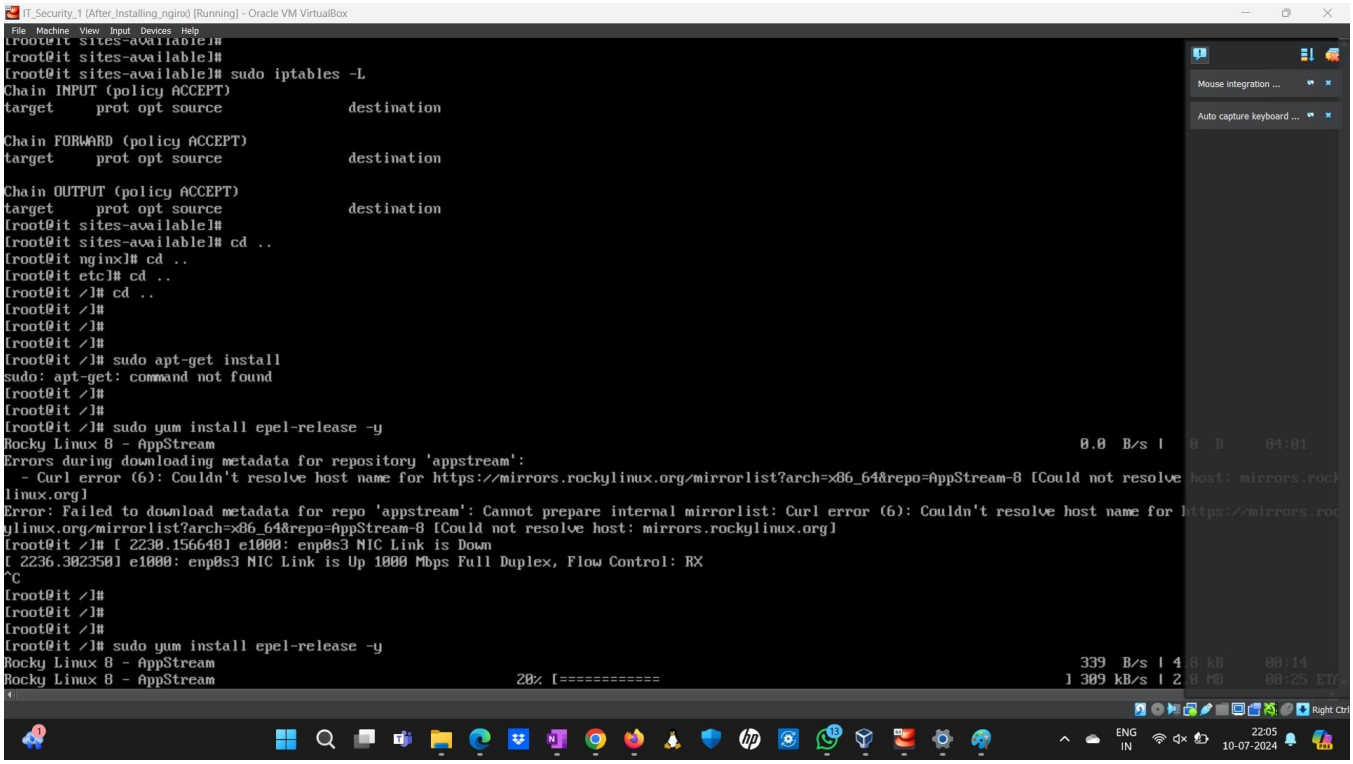




8.2 ufw Firewall

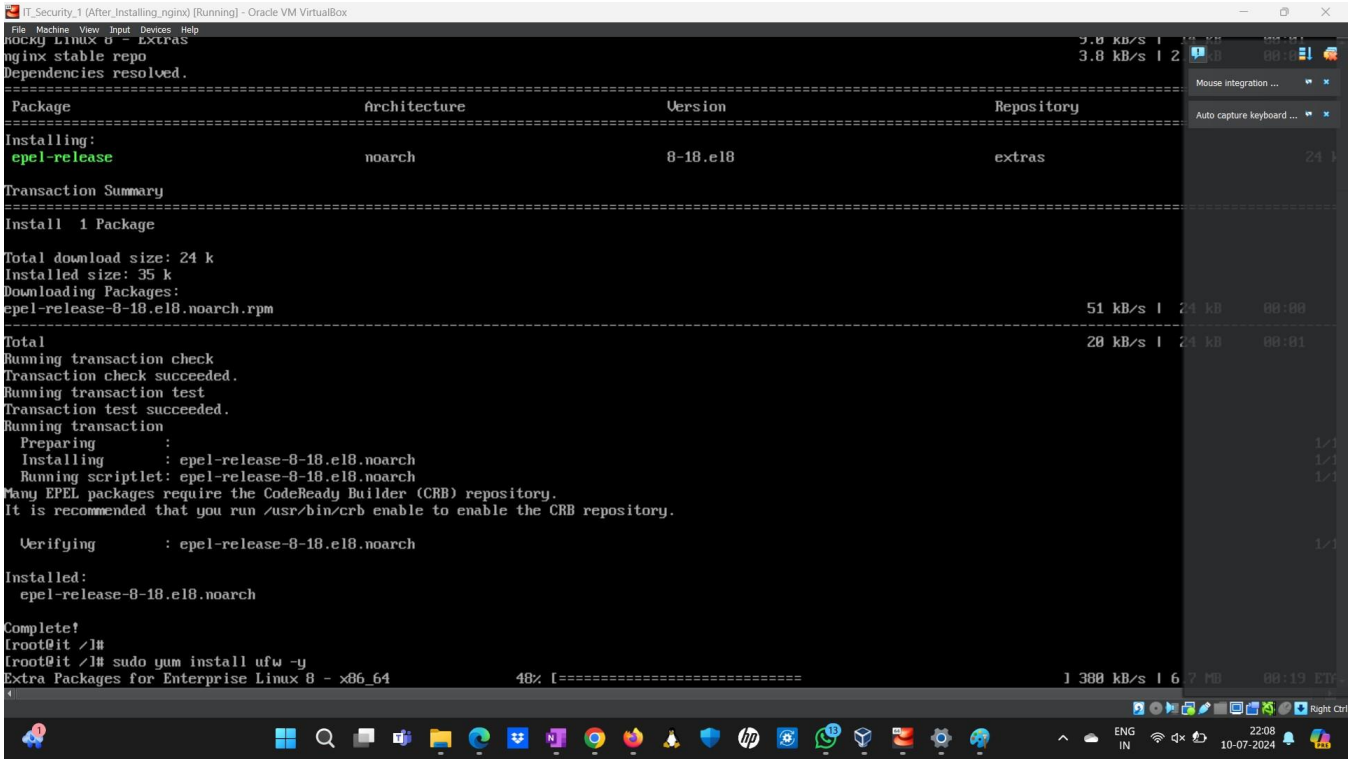
Installing the ufw firewall using the
“yum install epel-release”

Command. “Ufw” stands for Uncomplicated Firewall. It is a similar interface to iptables for managing a Linux firewall.



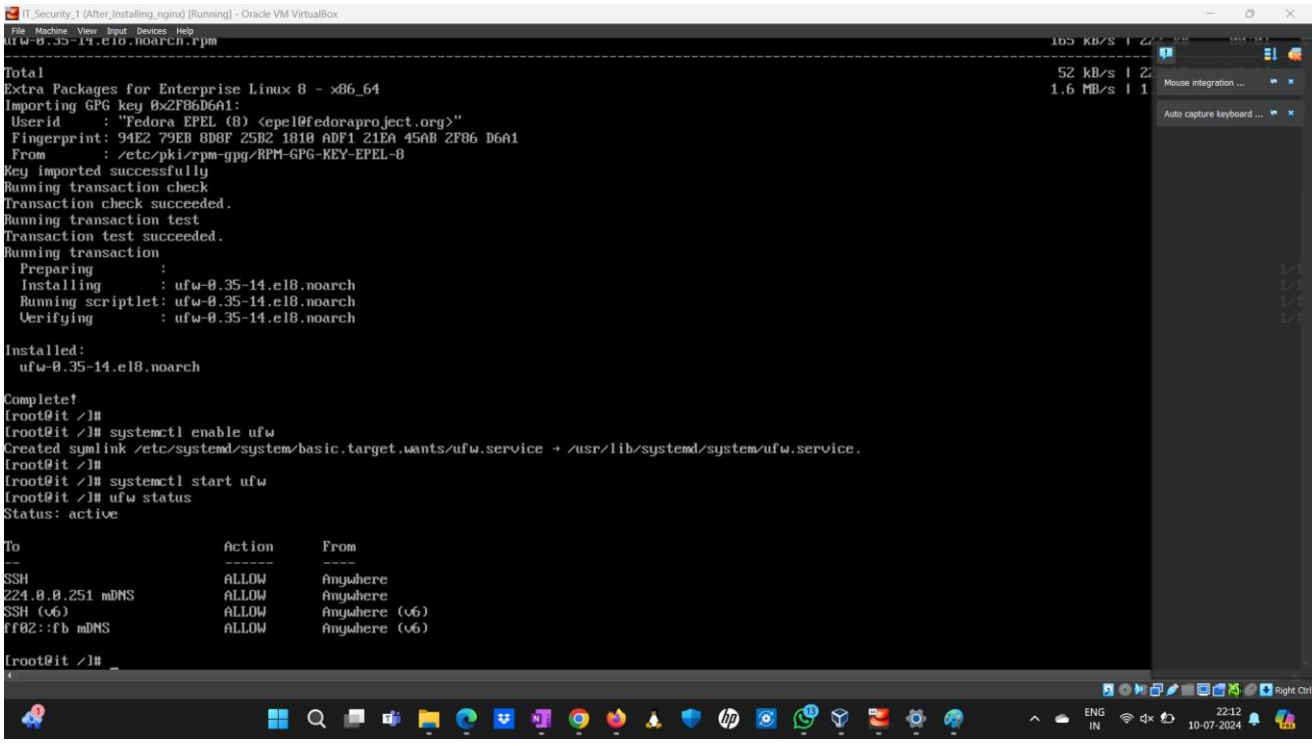
We are first installing epel-release - which are Extra Packages for Enterprise Linux. They are a large collection of software packages that are not available in the default repositories.

After this, we do
“yum install ufw”

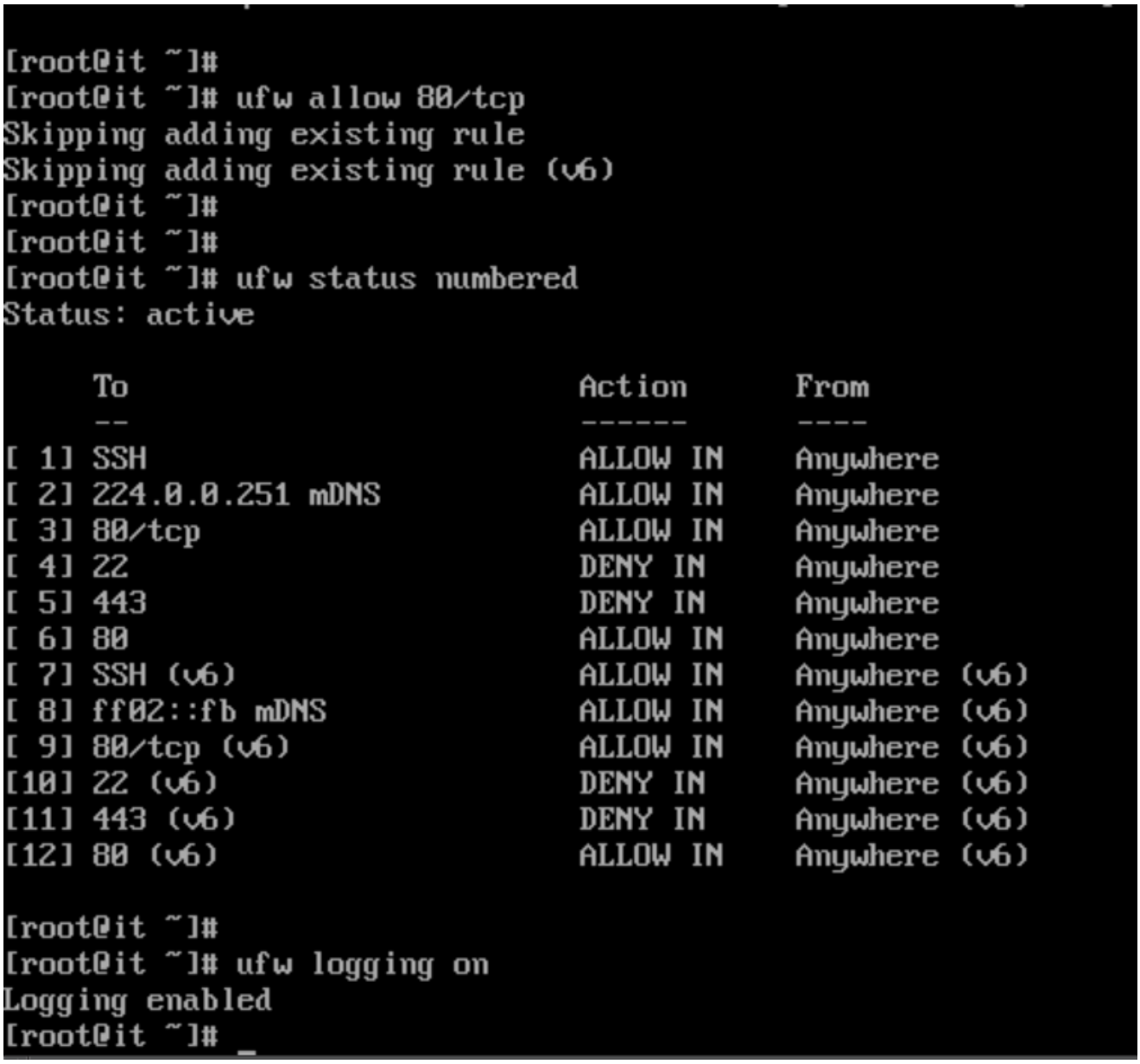


Now, we are enabling the package to start up automatically after booting up.

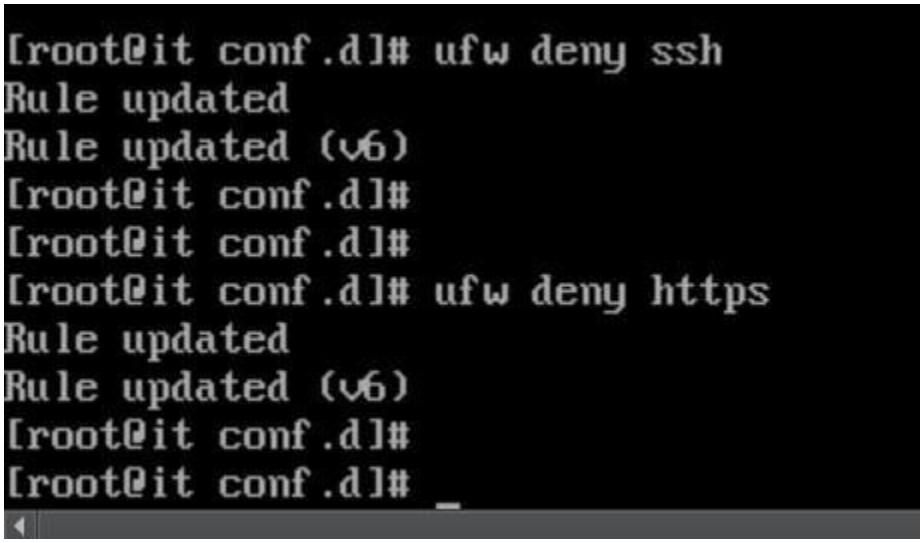
“systemctl enable ufw”



I have added a list of rules for ufw. One can check the list of all the rules using:
“ufw status numbered”



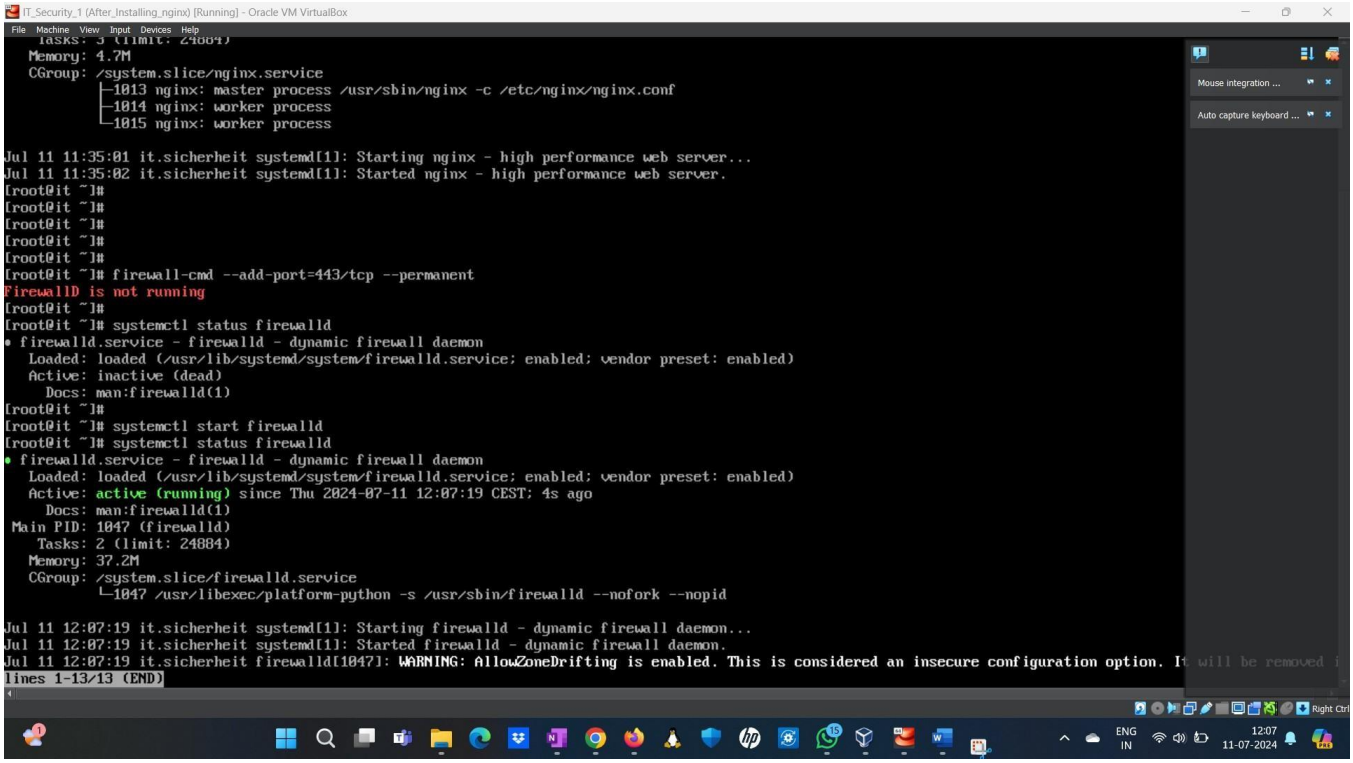
After this, I realized that we don’t need any kind of HTTPS and SSH traffic since we won’t be logging in using a secure shell or using the HTTPS port. Hence I disabled both these ports.

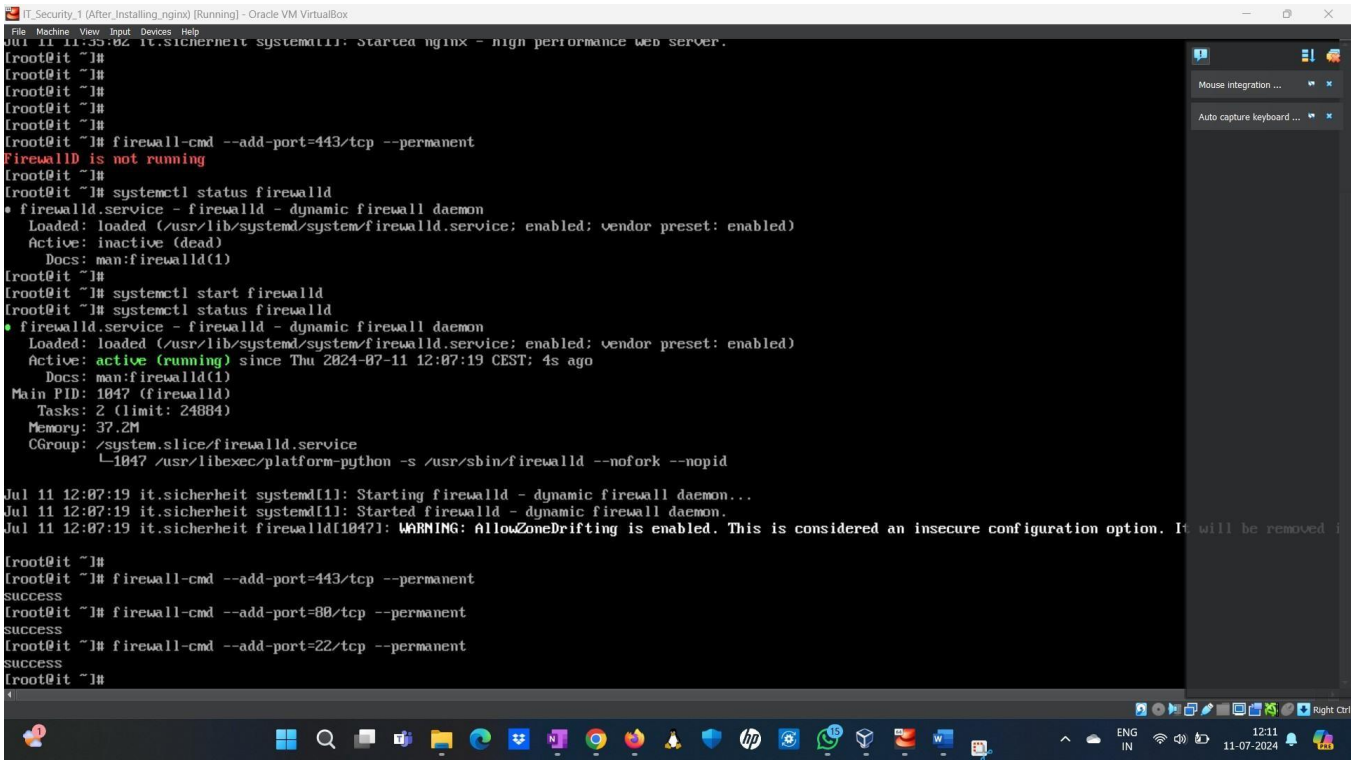


Ufw logging is also enabled to check the list of all the packets that are being allowed and denied by the firewall.

8.3 firewall-cmd Firewall

Firewall-cmd is a much more advanced package for firewall security. It has many more features such as allowing for more granular control and configuration, including defining different zones with varying levels of trust and applying rules to specific zones. We will be configuring the firewall using the following commands:
“systemctl start firewall”
“systemctl status firewall”
“firewall-cmd –add-port=443/tcp –permanent”
“firewall-cmd –add-port=80/tcp –permanent”
“firewall-cmd –add-port=22/tcp –permanent”



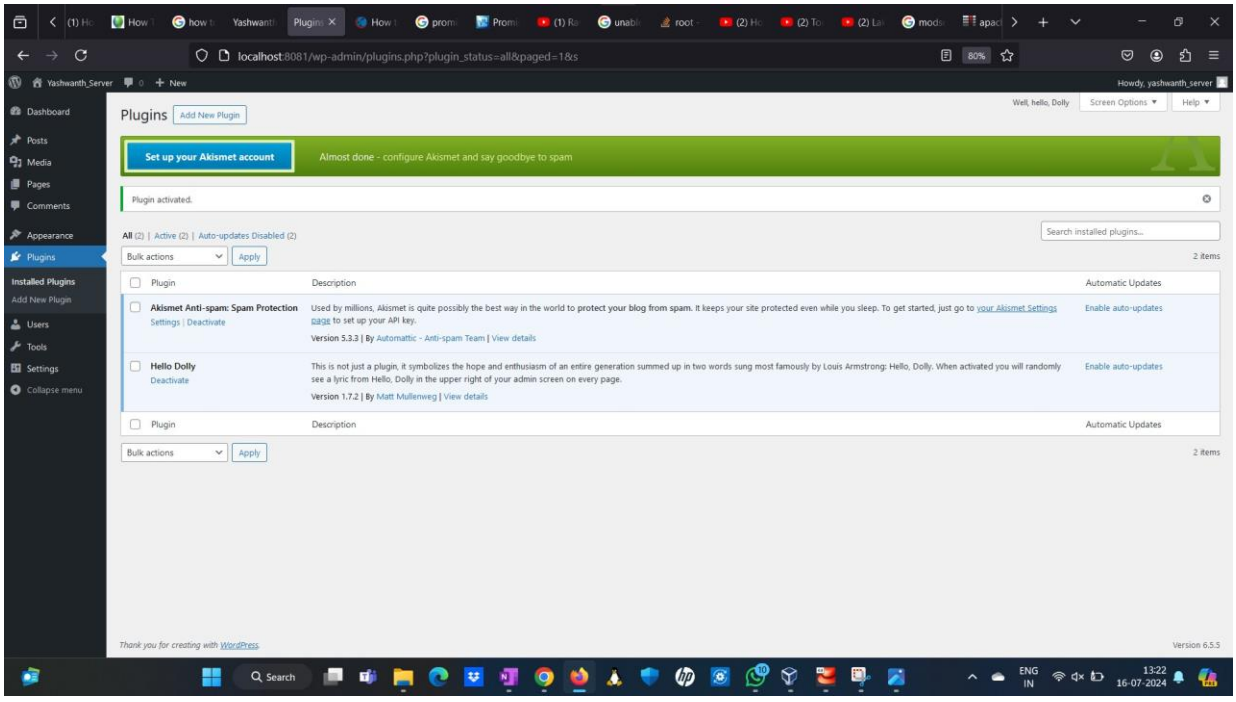


These 3 ports are used for traffic to be shared with the host computer (to see the wordpress LAMP server state and monitoring).

8.4 Wordpress Hardening

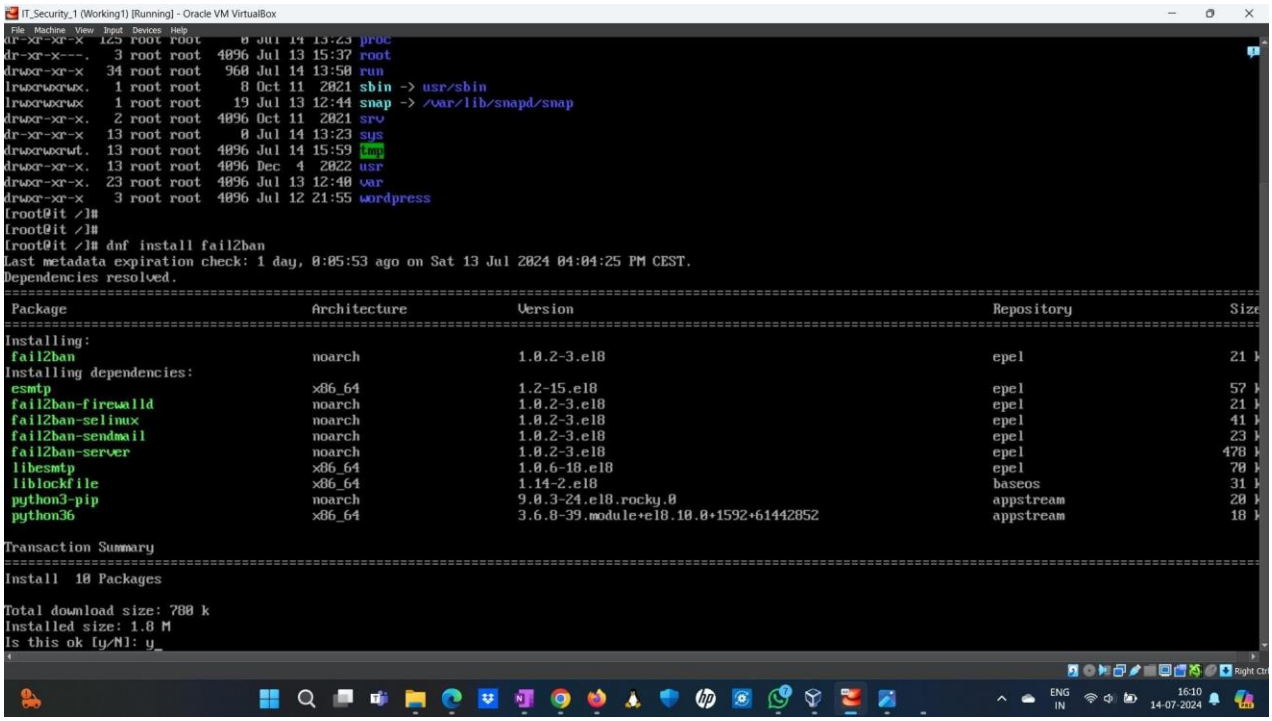
There are a few hardenings that can be done from the Wordpress recommendations. These 2 are:

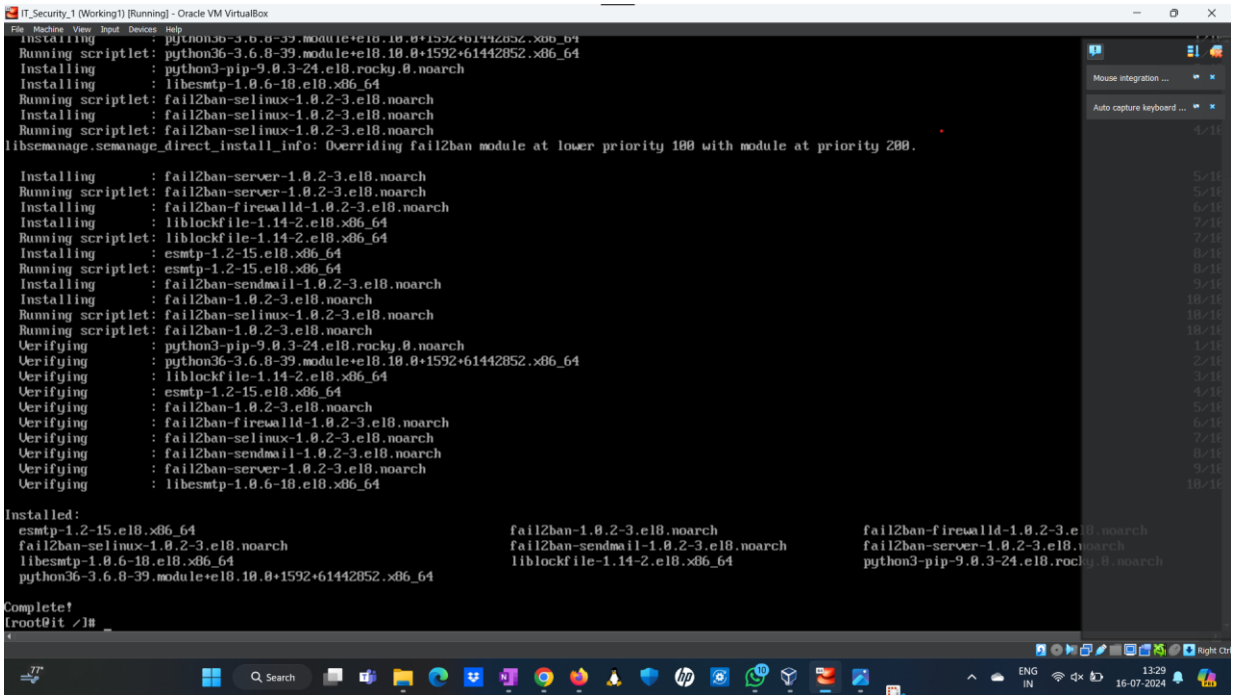
1. Akismet Anti-spam Spam Protection: Used by millions, Akismet is quite possibly the best way in the world to **protect your blog from spam**. It keeps your site protected even while you sleep. To get started, just go to your Akismet Settings page to set up your API key.
2. Hello Dolly: This is not just a plugin, it symbolizes the hope and enthusiasm of an entire generation summed up in two words sung most famously by Louis Armstrong: Hello, Dolly. When activated you will randomly see a lyric from Hello, Dolly in the upper right of your admin screen on every page.



8.5 fail2ban

fail2ban is a service


IT_Security_1 (Working1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
dr-xr-xr-x 125 root root 0 Jul 14 13:23 proc
dr-xr-x---. 3 root root 4096 Jul 13 15:37 root
drwxr-xr-x 34 root root 960 Jul 14 13:50 run
lrwxrwxrwx. 1 root root 8 Oct 11 2021 sbin -> usr/sbin
lrwxrwxrwx 1 root root 19 Jul 13 12:44 snap -> /var/lib/snapd/snap
drwxr-xr-x. 2 root root 4096 Oct 11 2021 srv
dr-xr-xr-x 13 root root 0 Jul 14 13:23 sys
drwxrwxrwt. 13 root root 4096 Jul 14 15:59 tmp
drwxr-xr-x. 13 root root 4096 Dec 4 2022 usr
drwxr-xr-x. 23 root root 4096 Jul 13 12:40 var
drwxr-xr-x 3 root root 4096 Jul 12 21:55 wordpress
[root@it /]#
[root@it /]#
[root@it /]# dnf install fail2ban
Last metadata expiration check: 1 day, 0:05:53 ago on Sat 13 Jul 2024 04:04:25 PM CEST.
Dependencies resolved.
Package Architecture Version Repository Size
Installing:
fail2ban noarch 1.0.2-3.el8 epel 21 k
Installing dependencies:
esmtp x86_64 1.2-15.el8 epel 57 k
fail2ban-firewalld noarch 1.0.2-3.el8 epel 21 k
fail2ban-selinux noarch 1.0.2-3.el8 epel 41 k
fail2ban-sendmail noarch 1.0.2-3.el8 epel 23 k
fail2ban-server noarch 1.0.2-3.el8 epel 478 k
libesmtp x86_64 1.0.6-18.el8 epel 70 k
liblockfile x86_64 1.14-2.el8 baseos 31 k
python3-pip noarch 9.0.3-24.el8.rocky.0 appstream 20 k
python36 x86_64 3.6.8-39.module+el8.10.0+1592+61442852 appstream 18 k
Transaction Summary
Install 10 Packages
Total download size: 780 k
Installed size: 1.8 M
Is this ok [y/N]: y_
ENG IN 16:10 14-07-2024


IT_Security_1 (Working1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Running scriptlet: python36-3.6.8-39.module+el8.10.0+1592+61442852.x86_64
Installing : python3-pip-9.0.3-24.el8.rocky.0.noarch
Installing : libesmtp-1.0.6-18.el8.x86_64
Running scriptlet: fail2ban-selinux-1.0.2-3.el8.noarch
Installing : fail2ban-selinux-1.0.2-3.el8.noarch
Running scriptlet: fail2ban-selinux-1.0.2-3.el8.noarch
libsemanage.semanage_direct_install_info: Overriding fail2ban module at lower priority 100 with module at priority 200.
Installing : fail2ban-server-1.0.2-3.el8.noarch
Running scriptlet: fail2ban-server-1.0.2-3.el8.noarch
Installing : fail2ban-firewalld-1.0.2-3.el8.noarch
Installing : liblockfile-1.14-2.el8.x86_64
Running scriptlet: liblockfile-1.14-2.el8.x86_64
Installing : esmtp-1.2-15.el8.x86_64
Running scriptlet: esmtp-1.2-15.el8.x86_64
Installing : fail2ban-sendmail-1.0.2-3.el8.noarch
Installing : fail2ban-1.0.2-3.el8.noarch
Running scriptlet: fail2ban-selinux-1.0.2-3.el8.noarch
Running scriptlet: fail2ban-1.0.2-3.el8.noarch
Verifying : python3-pip-9.0.3-24.el8.rocky.0.noarch
Verifying : python36-3.6.8-39.module+el8.10.0+1592+61442852.x86_64
Verifying : liblockfile-1.14-2.el8.x86_64
Verifying : esmtp-1.2-15.el8.x86_64
Verifying : fail2ban-1.0.2-3.el8.noarch
Verifying : fail2ban-firewalld-1.0.2-3.el8.noarch
Verifying : fail2ban-selinux-1.0.2-3.el8.noarch
Verifying : fail2ban-sendmail-1.0.2-3.el8.noarch
Verifying : fail2ban-server-1.0.2-3.el8.noarch
Verifying : libesmtp-1.0.6-18.el8.x86_64
Installed:
esmtp-1.2-15.el8.x86_64
fail2ban-1.0.2-3.el8.noarch
fail2ban-firewalld-1.0.2-3.el8.noarch
fail2ban-selinux-1.0.2-3.el8.noarch
fail2ban-sendmail-1.0.2-3.el8.noarch
fail2ban-server-1.0.2-3.el8.noarch
libesmtp-1.0.6-18.el8.x86_64
liblockfile-1.14-2.el8.x86_64
python3-pip-9.0.3-24.el8.rocky.0.noarch
python36-3.6.8-39.module+el8.10.0+1592+61442852.x86_64
Complete!
[root@it /]# _
Mouse integration ...
Auto capture keyboard ...
Search
ENG IN 13:29 16-07-2024

```
Complete!
[root@it /]#
[root@it /]#
[root@it /]# cp /etc/fail2ban/jail.conf /etc/fa
fail2ban/    favicon.png
[root@it /]# cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
[root@it /]#
[root@it /]#
[root@it /]#
```

I am creating a local copy of this file for your custom configurations to avoid overwriting changes during updates. Then you can change it using:
“vi jail.local”
You can see the configuration lines adding under [apache-auth].

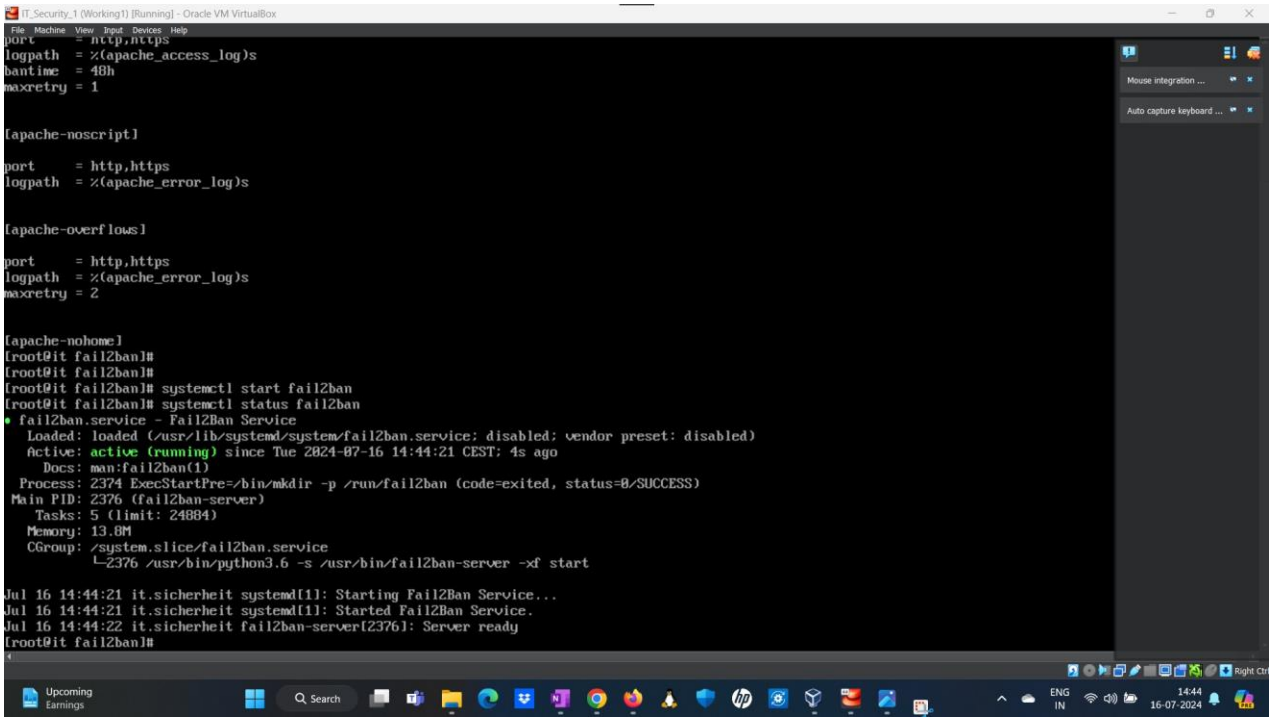
```
[apache-auth]
enabled  = true
port     = http,https
filter   = apache-auth
logpath  = %(apache_error_log)s
maxretry = 3
bantime  = 86400

[apache-badbots]
# Ban hosts which agent identifies spammer robots crawling the web
# for email addresses. The mail outputs are buffered.
port     = http,https
logpath  = %(apache_access_log)s
"jail.local" 982L, 25674C written
[root@it fail2ban]#
[root@it fail2ban]# _
```

The explanation for each of the sentences are:

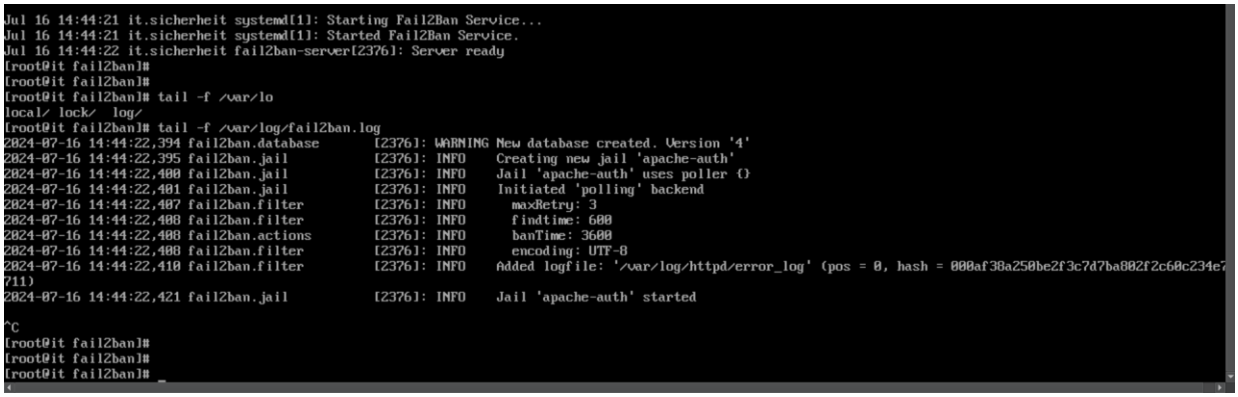
1. enabled: We are enabling the jail.
2. port: fail2ban should monitor http and https.
3. logpath: The path to your Apache error log.
4. maxretry: The number of allowed retries before we ban an IP.
5. bantime: In order to make it harder, I have increased the ban time to 86400 seconds (1 day).

Now, we are checking the status of fail2ban.
“systemctl start fail2ban”

“systemctl status fail2ban”

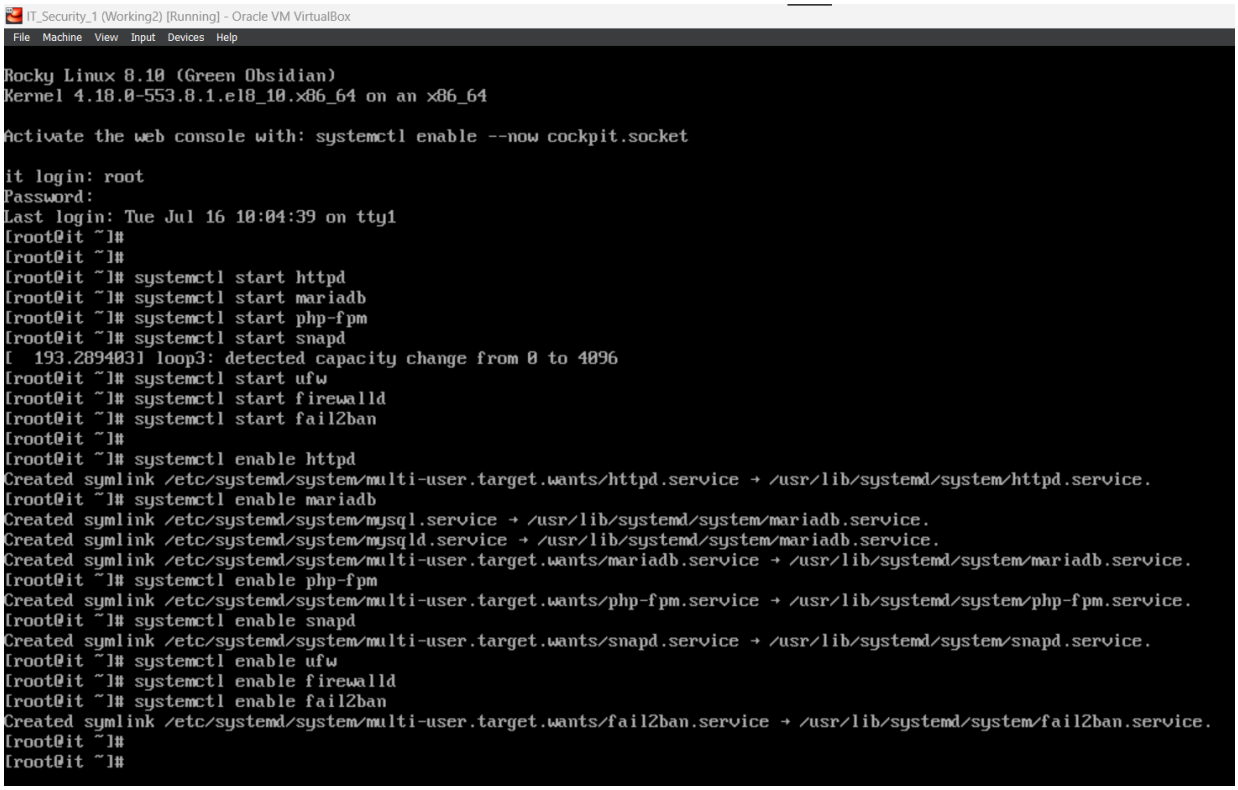


Now we are just checking the fail2ban software logs to see if everything is working fine.



8.6 Enabling all Services to start at startup

Enabling all the services necessary for wordpress to start automatically since it is generally good practice. They let critical services to be always running and also the services start in the right order when we startup.


IT_Security_1 (Working2) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Rocky Linux 8.10 (Green Obsidian)
Kernel 4.18.0-553.8.1.el8_10.x86_64 on an x86_64
Activate the web console with: systemctl enable --now cockpit.socket
it login: root
Password:
Last login: Tue Jul 16 10:04:39 on tty1
[root@it ~]#
[root@it ~]#
[root@it ~]# systemctl start httpd
[root@it ~]# systemctl start mariadb
[root@it ~]# systemctl start php-fpm
[root@it ~]# systemctl start snapd
[193.289403] loop3: detected capacity change from 0 to 4096
[root@it ~]# systemctl start ufw
[root@it ~]# systemctl start firewalld
[root@it ~]# systemctl start fail2ban
[root@it ~]#
[root@it ~]# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@it ~]# systemctl enable mariadb
Created symlink /etc/systemd/system/mysql.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/mysqld.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /usr/lib/systemd/system/mariadb.service.
[root@it ~]# systemctl enable php-fpm
Created symlink /etc/systemd/system/multi-user.target.wants/php-fpm.service → /usr/lib/systemd/system/php-fpm.service.
[root@it ~]# systemctl enable snapd
Created symlink /etc/systemd/system/multi-user.target.wants/snapd.service → /usr/lib/systemd/system/snapd.service.
[root@it ~]# systemctl enable ufw
[root@it ~]# systemctl enable firewalld
[root@it ~]# systemctl enable fail2ban
Created symlink /etc/systemd/system/multi-user.target.wants/fail2ban.service → /usr/lib/systemd/system/fail2ban.service.
[root@it ~]#
[root@it ~]#

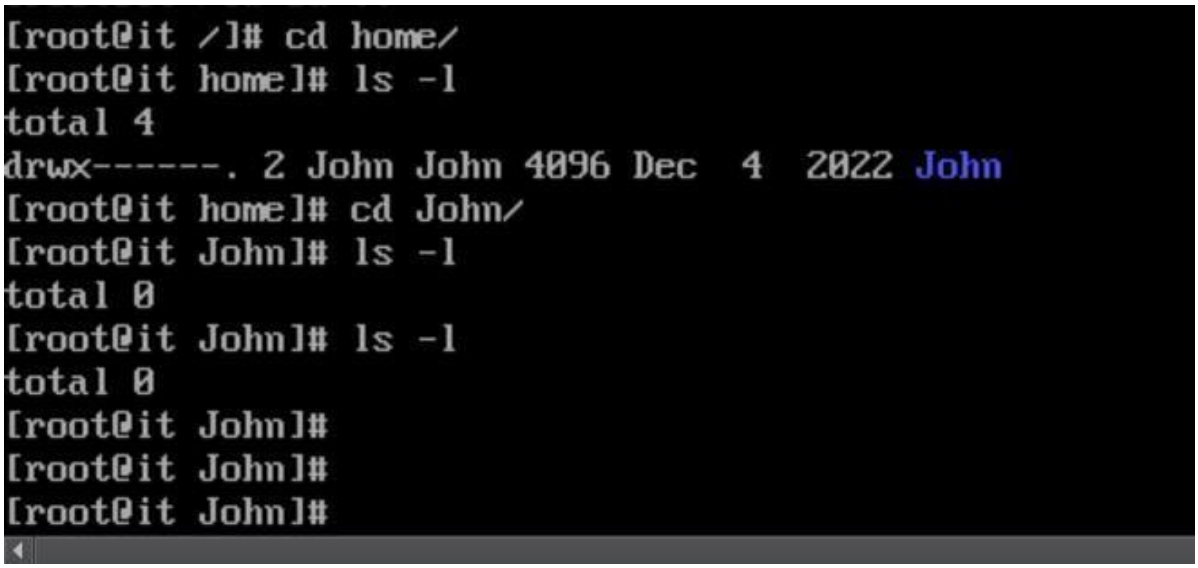
9 Easter Eggs

9.1 User John

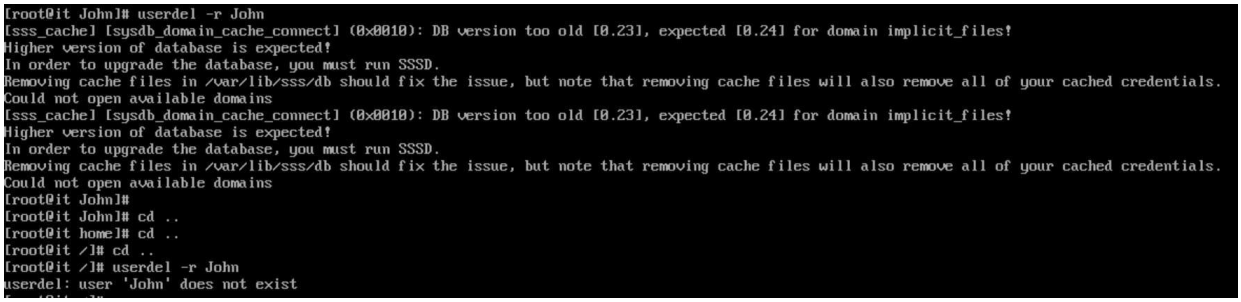
On browsing and looking through the system, I checked the list of users. This can be done through the:
“cat /etc/passwd”
It is used to display the contents of the /etc/passwd file, which contains user account information. There were a list of many users present in the system - most of them are system users such as apache, mysql. But then, I found a user called ‘John’ who is a regular user and not a system user. This is because system users generally have user IDs lower than 1000.
John’s data is:

1. Username: john
2. UID: 1001
3. GID: 1001
4. His/her home directory: /home/john

John is an intruder in our system.
So I checked his home directory, but John was clever not to leave any files in his home directory.



So to secure my system, I decided to delete this user.



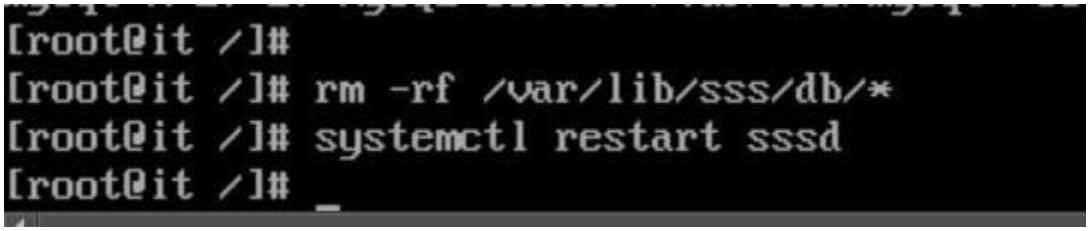
The output you're seeing is related to issues with the System Security Services Daemon (SSSD) cache. It says that the:

1. DB Version Too Old: It says that the database version for SSSD currently is 0.23 but a higher version (0.24) is required.
2. Removing Cache Files: The suggestion to remove cache files in /var/lib/sss/db to clean the system.
3. Could Not Open Available Domains: This implies that SSSD is unable to access the necessary domain information

In order to resolve these issues:

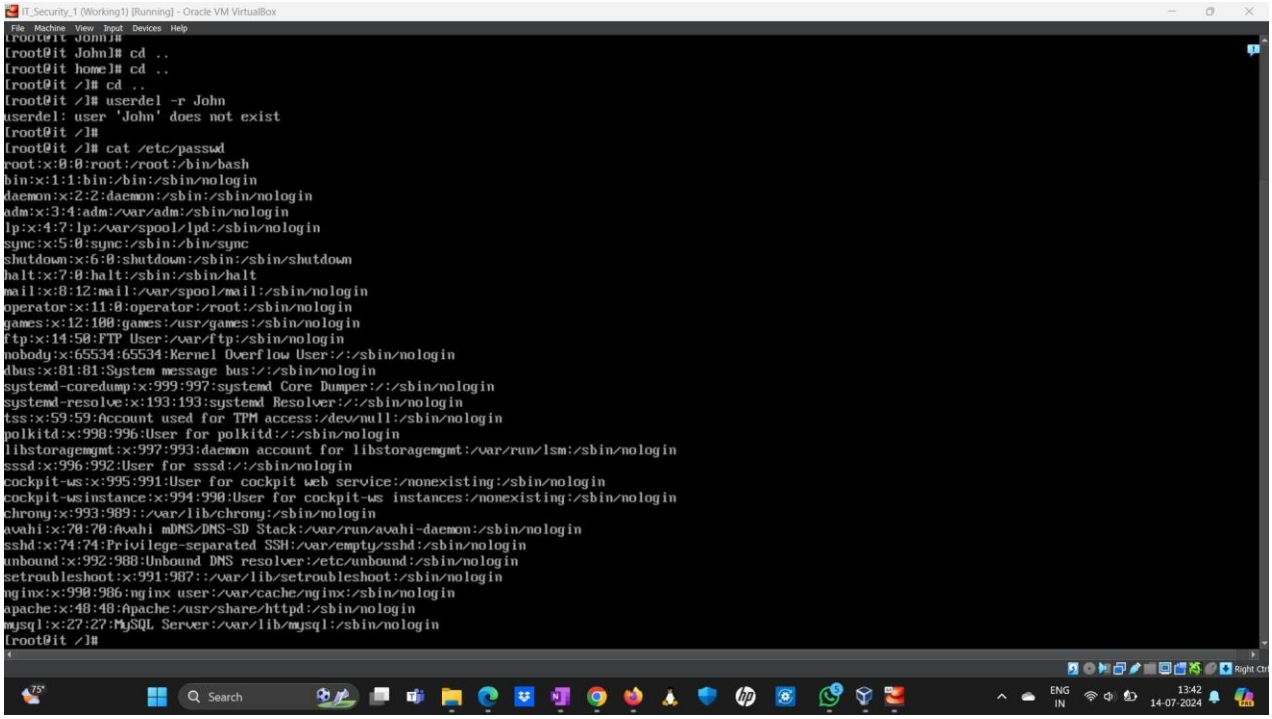
1. I tried to remove the SSSD cache:
   “sudo rm -rf /var/lib/sss/db/*”
   After removing the files using the above command, the old files will be deleted.

2. Restart the entire system:
   “sudo systemctl restart sssd”



Now that we have resolved this issue, we recheck if ‘John’ was deleted successfully.
“userdel -r John”.

We also recheck this using the:
cat /etc/passwd



It says that John doesn’t exist which shows we have successfully mitigated this Easter Egg.

10 References

[1] Ellingwood, J. (2021, June 30). *How To Set Up a Firewall Using FirewallD on CentOS 7*. DigitalOcean. https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-using-firewalld-on-centos-7

[2] GeeksforGeeks. (2024, April 17). *How To Install WordPress On Rocky Linux 9*. GeeksforGeeks. https://www.geeksforgeeks.org/how-to-install-wordpress-on-rocky-linux-9/

[3] Cezar, M., & Cezar, M. (2018, January 31). *How to Change Apache HTTP Port in Linux*. How to Change Apache HTTP Port in Linux. https://www.tecmint.com/change-apache-port-in-linux/

[4] *How to install LAMP on AlmaLinux 8.4 / Rocky Linux 8.4 - Tutorials and How To - CloudCone*. (2021, August 31). Tutorials and How To - CloudCone. https://cloudcone.com/docs/article/how-to-install-lamp-on-almalinux-8-4-rocky-linux-8-4/

[5] Gomez, J. (2023, March 17). *Install LAMP Stack On Rocky Linux 9 {Step By Step}* | *LinuxTeck*. LinuxTeck. https://www.linuxteck.com/install-lamp-stack-on-rocky-linux-9/

[6] *How To Install WordPress On RockyLinux 8*. (n.d.). https://wiki.crowncloud.net/?How_to_Install_WordPress_on_RockyLinux_8

[7] Garnett, A. (2022, July 4). *How To Set Up a Firewall Using firewalld on Rocky Linux 8*. DigitalOcean. https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-using-firewalld-on-rocky-linux-8

[8] Maker, E. R. (2024, January 22). How to Harden Debian/Ubuntu/Rocky Desktop - Ed Roof Maker - Medium. *Medium*. https://medium.com/@ed-roof-maker/how-to-harden-debian-ubuntu-rocky-desktop-63f81f7ecf78

[9] Kimes, N. (2024, June 13). *How to Secure Your LAMP Server: A Comprehensive Guide*. AskHandle. https://www.askhandle.com/blog/how-to-secure-your-lamp-server--a-comprehensive-guide

[10] Fail2ban. (n.d.). *GitHub - fail2ban/fail2ban: Daemon to ban hosts that cause multiple authentication errors*. GitHub. https://github.com/fail2ban/fail2ban

[11] *Chmod 755*. (2023, August 3). https://www.warp.dev/terminus/chmod-755

[12] Rietta, F. (2014, May 27). *ModSecurity and Fail2Ban as an Intrusion Prevention System*. https://rietta.com/blog/mod-security-and-fail2ban-as-an-intrusion-prevention-system/

[13] *Fail2Ban and modsecurity not working*. (n.d.). Server Fault. https://serverfault.com/questions/845270/fail2ban-and-modsecurity-not-working

[14] Ravutha Official. (2018, November 26). *How to Setting Up Promiscuous Mode (1st Method)* [Video]. YouTube. https://www.youtube.com/watch?v=YAvri_B4sK4

[15] Lame Creations. (2024, July 3). *ProxMox Tutorial* | *How to Enable Promiscuous Mode* [Video]. YouTube. https://www.youtube.com/watch?v=plKBAmYzobg

[16] Tony Teaches Tech. (2022b, March 10). *How to Install a LAMP Server on Rocky Linux* [Video]. YouTube. https://www.youtube.com/watch?v=V8auUbpETlg

[17] Tony Teaches Tech. (2022c, March 15). *How To Install WordPress on Rocky Linux (Apache LAMP server)* [Video]. YouTube. https://www.youtube.com/watch?v=iCfAuPPrr04

[18] Tony Teaches Tech. (2022b, February 17). *How to Install an SSL Certificate on Rocky Linux (from Let's Encrypt)* [Video]. YouTube. https://www.youtube.com/watch?v=yQr28f4KGu8

[19] Verma, S. (2023, August 21). *How to Setup Port Forwarding on Linux? - Scaler Topics*. Scaler Topics. https://www.scaler.com/topics/linux-port-forwarding/

[20] *localhost*. (n.d.). https://locallhost.com/

[21] *Sudo*. (2024, July 5). https://en.wikipedia.org/wiki/Sudo

[22] *Promiscuous Mode - Allow VMs vs Allow All? - virtualbox.org*. (2011, August 30). https://forums.virtualbox.org/viewtopic.php?t=44258

[23] *How to fix the error "E: Unable to locate package wireless-tools" in termux*. (n.d.). Stack Overflow.

https://stackoverflow.com/questions/65933209/how-to-fix-the-error-e-unable-to-locate-package-wireless-tools-in-termux

[24] [24] https://www.reddit.com/r/linuxadmin/comments/lwbrh0/what_are_some_better_alternatives_to_fail2ban/?rdt=54888&onetap_auto=true&one_tap=true

[25] [25] https://www.reddit.com/r/selfhosted/comments/aln2d9/cloudflare_free_fail2ban_other_security_hardening/

[26] https://linuxconfig.org/redhat-8-stop-start-firewall