



Professional IT Business and Digitalization

IT Security Project Report

From

Yashwanth Pindi

Matriculation Number: s0590681

Supervisor: Prof. Dr. Nils Siebel

1 Table of Contents

1 Table of Contents	2
2 Acknowledgement	3
3 Introduction	4
4 General Information	5
5 Network Cards IP Information	10
6 Configuring the LAMP Server	19
6.1 Installing Apache	19
6.2 Installing Mariadb Database	21
6.3 Installing PHP	25
7 Wordpress	27
7.1 Wordpress Installation	27
7.2 MariaDB Database Creating for Wordpress	30
7.3 PHP Error	31
7.4 Wordpress Settings	33
8 Hardening the System	38
8.1 Changed the Port from 80 to 5975	38
8.2 ufw Firewall	42
8.3 firewall-cmd Firewall	46
8.4 Wordpress Hardening	48
8.5 fail2ban	48
8.6 Enabling all Services to start at startup	51
9 Easter Eggs	53
9.1 User John	53
10 References	56

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.

```

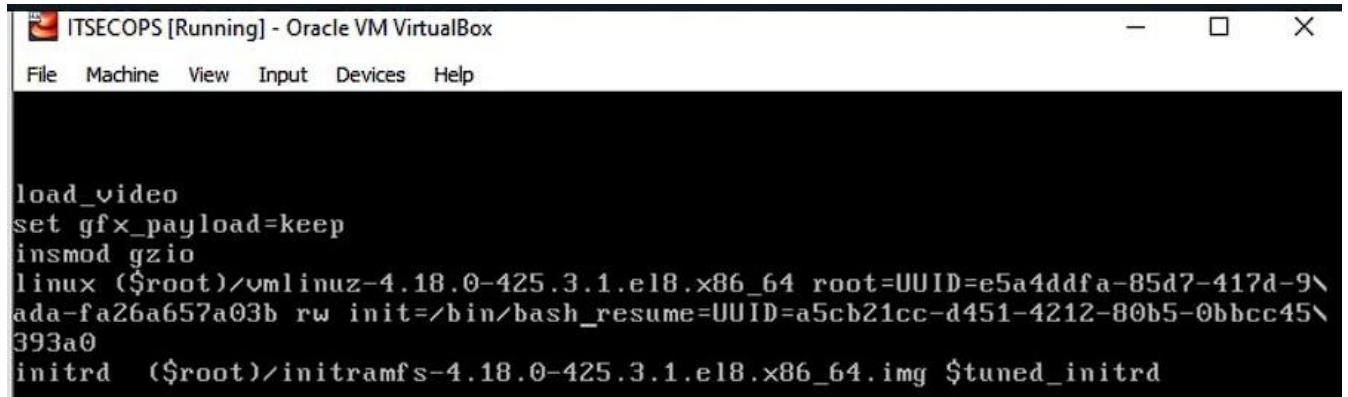
File Edit View Terminal Task Help
[72136_330898] usb 2-1.2: new SuperSpeed USB device number 8 using xhci_hcd
[72136_349142] usb 2-1.2: New USB device found, idVendor=1656, idProduct=0010, bcdDevice= 5.15
[72136_349147] usb 2-1.2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[72136_349148] usb 2-1.2: Product: Q-SYS Core Nano
[72136_349148] usb 2-1.2: Manufacturer: QSC, LLC
[72136_349149] usb 2-1.2: SerialNumber: CORENANO-PIEPIE-VB-1-1d4abf7d0a
[72136_365885] usb 2-1.2: Found UVC 1.00 device 0010 Q-SYS Core Nano (1656:0010)
[72136_394882] r8152 2-1.1:1.0 enp0s13f0u1u1: renamed from eth0
[72139_910781] usb 3-3.3: new high-speed USB device number 23 using xhci_hcd
[72140_056160] usb 3-3.3: New USB device found, idVendor=04b4, idProduct=6506, bcdDevice=50.00
[72140_056182] usb 3-3.3: New USB device strings: Mfr=0, Product=0, SerialNumber=0
[72140_058618] hub 3-3.3.1.0: USB hub found
[72140_058775] hub 3-3.3.1.0: 2 ports detected
[72142_267561] usb 3-3.5: new high-speed USB device number 24 using xhci_hcd
[72142_360127] usb 3-3.5: New USB device found, idVendor=17ef, idProduct=721c, bcdDevice= 0.01
[72142_360149] usb 3-3.5: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[72142_360153] usb 3-3.5: Manufacturer: Lenovo Powered Hub
[72142_360156] usb 3-3.5: SerialNumber: 00000000000000000000000000000000
[72144_454175] usb 3-3.3.2: new high-speed USB device number 25 using xhci_hcd
[72144_549794] usb 3-3.3.2: New USB device found, idVendor=2bd9, idProduct=0041, bcdDevice= 4.01
[72144_549805] usb 3-3.3.2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[72144_549808] usb 3-3.3.2: Product: Huddly Canvas
[72144_549810] usb 3-3.3.2: Manufacturer: Huddly
[72144_549812] usb 3-3.3.2: SerialNumber: G42K90116
[72144_554074] usb 3-3.3.2: Found UVC 1.1.0 device Huddly Canvas (2bd9:0041)
[72159_491616] asus_wmi: Unknown key code 0x77
[72170_267545] usb 2-2: new SuperSpeed Plus Gen 2x1 USB device number 9 using xhci_hcd
[72170_225614] usb 2-2: New USB device found, idVendor=0781, idProduct=558c, bcdDevice=10.12
[72170_225017] usb 2-2: New USB device strings: Mfr=2, Product=3, SerialNumber=1
[72170_225918] usb 2-2: Product: Extreme SSD
[72170_225819] usb 2-2: Manufacturer: SanDisk
[72170_225820] usb 2-2: SerialNumber: 313834363238343030313337
[72170_227181] scsi host0: uas
[72170_227843] scsi 0:0:0:0: Direct-Access     SanDisk Extreme SSD      1012 PQ: 0 ANSI: 6
[72170_230930] scsi 0:0:0:1: Enclosure      SanDisk SES Device      1012 PQ: 0 ANSI: 6
[72170_231753] scsi 0:0:0:0: Attached scsi generic sg0 type 0
[72170_231980] sd 0:0:0:0: [sda] Spinning up disk...
[72170_232870] scsi 0:0:0:1: Attached scsi generic sg1 type 13
[72171_250892] .read
[72171_263830] sd 0:0:0:0: [sda] 488396800 512-byte logical blocks: (250 GB/233 GiB)
[72171_263837] sd 0:0:0:0: [sda] 4096-byte physical blocks
[72171_263951] sd 0:0:0:0: [sda] Write Protect is off
[72171_263953] sd 0:0:0:0: [sda] Mode Sense: 67 00 10 08
[72171_264172] sd 0:0:0:0: [sda] Write cache: enabled, read cache: enabled, supports DPO and FUA
[72171_264351] sd 0:0:0:0: [sda] Preferred minimum I/O size 4096 bytes
[72171_264351] sd 0:0:0:0: [sda] Attached SCSI disk
[72171_264353] sd 0:0:0:0: [sda] Optimal transfer size 33553920 bytes not a multiple of preferred minimum block size (4096 bytes)
[72171_266110] sda: sdal sda2 sda3 sda4 < sda5 >
[72171_266685] sd 0:0:0:0: [sda] Attached SCSI disk

```

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.



```
load_video
set gfx_payload=keep
insmod gzio
linux ($root)/vmlinuz-4.18.0-425.3.1.e18.x86_64 root=UUID=e5a4ddfa-85d7-417d-9\ada-fa26a657a03b rw init=/bin/bash_resume=UUID=a5cb21cc-d451-4212-80b5-0bbcc45\393a0
initrd ($root)/initramfs-4.18.0-425.3.1.e18.x86_64.img $tuned_initrd
```

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.

Il /dev/sda - is not a file but a disk itself. 'Il' 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.

```
r lahf_lm cmp_legacy cr8_legacy abm sse4a misalignsse 3dnowprefetch ssbd vmmcall fsgsbase bmi1 bmi2  
rdseed clflushopt arat  
bugs : fxsave_leak sysret_ss_attrs null_seg spectre_v1 spectre_v2 retbleed  
bogomips : 4191.98  
TLB size : 3072 4K pages  
clflush size : 64  
cache_alignment : 64  
address sizes : 48 bits physical, 48 bits virtual  
power management:  
  
[root@it tmp]#  
[root@it tmp]#  
[root@it tmp]#  
[root@it tmp]#  
[root@it tmp]# ip 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 1000  
    link/ether 08:00:27:9b:4e:73 brd ff:ff:ff:ff:ff:ff  
[root@it tmp]#  
[root@it tmp]#  
[root@it tmp]# 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 1000  
    link/ether 08:00:27:9b:4e:73 brd ff:ff:ff:ff:ff:ff  
[root@it tmp]# _
```

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).

```
[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]# _
```

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:

```
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-108e36dabea6
DEVICE=enp0s3
ONBOOT=yes
~
~
```

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:

```
Rocky Linux 8.7 (Green Obsidian)
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 1000
    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 nopref ixroute enp0s3
            valid_lft 86372sec preferred_lft 86372sec
        inet6 fe80::a00:27ff:fe9b:4e73/64 scope link nopref ixroute
            valid_lft forever preferred_lft forever
[root@it ~]#
[root@it ~]#
```

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 1000
    link/ether 00:00:27:9b:4e:73 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic nopref ixroute enp0s3
        valid_lft 86372sec preferred_lft 86372sec
    inet6 fe80::a00:27ff:fe9b:4e73/64 scope link nopref ixroute
        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”

```

policycoreutils-python-utils-2.9-24.el8.noarch
python3-audit-3.0.7-5.el8.x86_64
python3-cairo-1.16.3-6.el8.x86_64
python3-html5lib-1.0.99999999-6.el8.noarch
python3-lxml-4.2.3-4.el8.x86_64
python3-pexpect-4.3.1-3.el8.noarch
python3-policycoreutils-2.9-24.el8.noarch
python3-ptyprocess-0.5.2-4.el8.noarch
python3-setools-4.3.0-5.el8.x86_64
python3-systemd-234-8.el8.x86_64
python3-unbound-1.16.2-5.el8_9.6.x86_64
setroubleshoot-plugins-3.3.14-1.el8.noarch
sscg-3.0.0-7.el8.x86_64
unbound-libs-1.16.2-5.el8_9.6.x86_64

protobuf-c-1.3.8-8.el8.x86_64
python3-bind-32:9.11.36-11.el8_9.1.noarch
python3-gobject-3.28.3-2.el8.x86_64
python3-libsemanage-2.9-9.el8_6.x86_64
python3-magic-5.33-25.el8.noarch
python3-ply-3.9-9.el8.noarch
python3-psutil-5.4.3-11.el8.x86_64
python3-pydbus-0.6.0-5.el8.noarch
python3-setuptools-39.2.0-7.el8.noarch
python3-tracer-0.7.5-2.el8.noarch
python3-webencodings-0.5.1-6.el8.noarch
setroubleshoot-server-3.3.26-5.el8.x86_64
tracer-common-0.7.5-2.el8.noarch
xkeyboard-config-2.28-1.el8.noarch

Complete!
[root@it ~]#
[root@it ~]#
[root@it ~]# sudo dnf install gpm
Last metadata expiration check: 8:42:26 ago on Sat 18 May 2024 11:07:51 AM CEST.
Dependencies resolved.
=====
 Package           Architecture   Version        Repository  Size
=====
Installing:
 gpm               x86_64         1.20.7-17.el8    appstream  199 k
Installing dependencies:
 SDL                x86_64         1.2.15-39.el8    appstream  217 k
 linuxconsoletools x86_64         1.6.0-4.el8     appstream  88 k

Transaction Summary
=====
Install 3 Packages

Total download size: 496 k
Installed size: 1.1 M
Is this ok [y/N]: -

```

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 lists 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”

```

root      8874  0.0  0.0  48652  3572 ?          S<   11:40  0:00 \_ /usr/sbin/sedispatch
avahi     8898  0.0  0.1  69584  4784 ?          Ss   11:40  0:00 avahi-daemon: running [it.local]
avahi     8899  0.0  0.0  69452  444 ?          S    11:40  0:00 \_ avahi-daemon: chroot helper
root      8904  0.0  0.1  50724  6208 ?          Ss   11:40  0:00 /usr/sbin/smard -n -q never
chrony    8919  0.0  0.0  121456  3824 ?          S    11:40  0:00 /usr/sbin/chronyd
root      8937  1.1  0.8  767552  32504 ?          Ssl  11:40  1:43 /usr/libexec/platform-python -Es
/usr/sbin/tuned -l -P
polkitd   9131  0.0  0.6  1622784  27376 ?          Ssl  11:40  0:00 /usr/lib/polkit-1/polkitd --no-de
bug
root      9139  0.0  0.4  502896  16500 ?          Ssl  11:40  0:00 /usr/sbin/ModemManager
root      9158  0.0  0.1  124912  4992 ?          Ssl  11:40  0:00 /usr/sbin/irqbalance --foreground
root      9201  0.0  1.4  670596  59324 ?          Ssl  11:40  0:00 /usr/libexec/platform-python -s /
usr/sbin/firewalld --nofork --nopid
root      17738  0.0  0.1  84000  7936 ?          Ss   11:41  0:00 /usr/lib/systemd/systemd-udevd
root      17759  0.0  0.1  76656  7392 ?          Ss   11:41  0:00 /usr/sbin/sshd -D -oCiphers=aes25
6-gcm@openssh.com,chacha20-poly1305@openssh.com,aes256-ctr,aes256-cbc,aes128-gcm@openssh.com,aes128-
ctr,aes128-cbc -oMACs=hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,umac-128-etm@openssh.c
om,hmac-sha2-512-etm@openssh.com,hmac-sha2-256,hmac-sha1,umac-128@openssh.com,hmac-sha2-512 -oGSSAPI
KexAlgorithms=gss-curve25519-sha256-,gss-nistp256-sha256-,gss-group14-sha256-,gss-group16-sha512-,gs
s-gex-sha1-,gss-group14-sha1- -oKexAlgorithms=curve25519-sha256,curve25519-sha256@libssh.org,ecdh-sh
a2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256,diffie-hellma
n-group14-sha256,diffie-hellman-group16-sha512,diffie-hellman-group18-sha512,diffie-hellman-group-ex
change-sha1,diffie-hellman-group14-sha1 -oHostKeyAlgorithms=ecdsa-sha2-nistp256,ecdsa-sha2-nistp256-
cert-v01@openssh.com,ecdsa-sha2-nistp384,ecdsa-sha2-nistp384-cert-v01@openssh.com,ecdsa-sha2-nistp52
1,ecdsa-sha2-nistp521-cert-v01@openssh.com,ssh-ed25519,ssh-ed25519-cert-v01@openssh.com,rsa-sha2-256
,rsa-sha2-256-cert-v01@openssh.com,rsa-sha2-512,rsa-sha2-512-cert-v01@openssh.com,ssh-rsa,ssh-rsa-ce
rt-v01@openssh.com -oPubkeyAcceptedKeyTypes=ecdsa-sha2-nistp256,ecdsa-sha2-nistp256-cert-v01@openssh
.com,ecdsa-sha2-nistp384,ecdsa-sha2-nistp384-cert-v01@openssh.com,ecdsa-sha2-nistp521,ecdsa-sha2-nis
tp521-cert-v01@openssh.com,ssh-ed25519,ssh-ed25519-cert-v01@openssh.com,rsa-sha2-256,rsa-sha2-256-ce
rt-v01@openssh.com,rsa-sha2-512,rsa-sha2-512-cert-v01@openssh.com,ssh-rsa,ssh-rsa-cert-v01@openssh.c
om -oCASignatureAlgorithms=ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-ed25519,r
sa-sha2-256,rsa-sha2-512,ssh-rsa
root      72076  0.0  0.0  8084   100 ?          Ss   11:52  0:00 /usr/sbin/gpm -m /dev/input/mice
-t exps2
[root@it ~]#
[root@it ~]#
[root@it ~]# ps aufxw | more_

```

This command lists 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_1]
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	_ [kaudittd]
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_biol]
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.

6 Configuring the LAMP Server

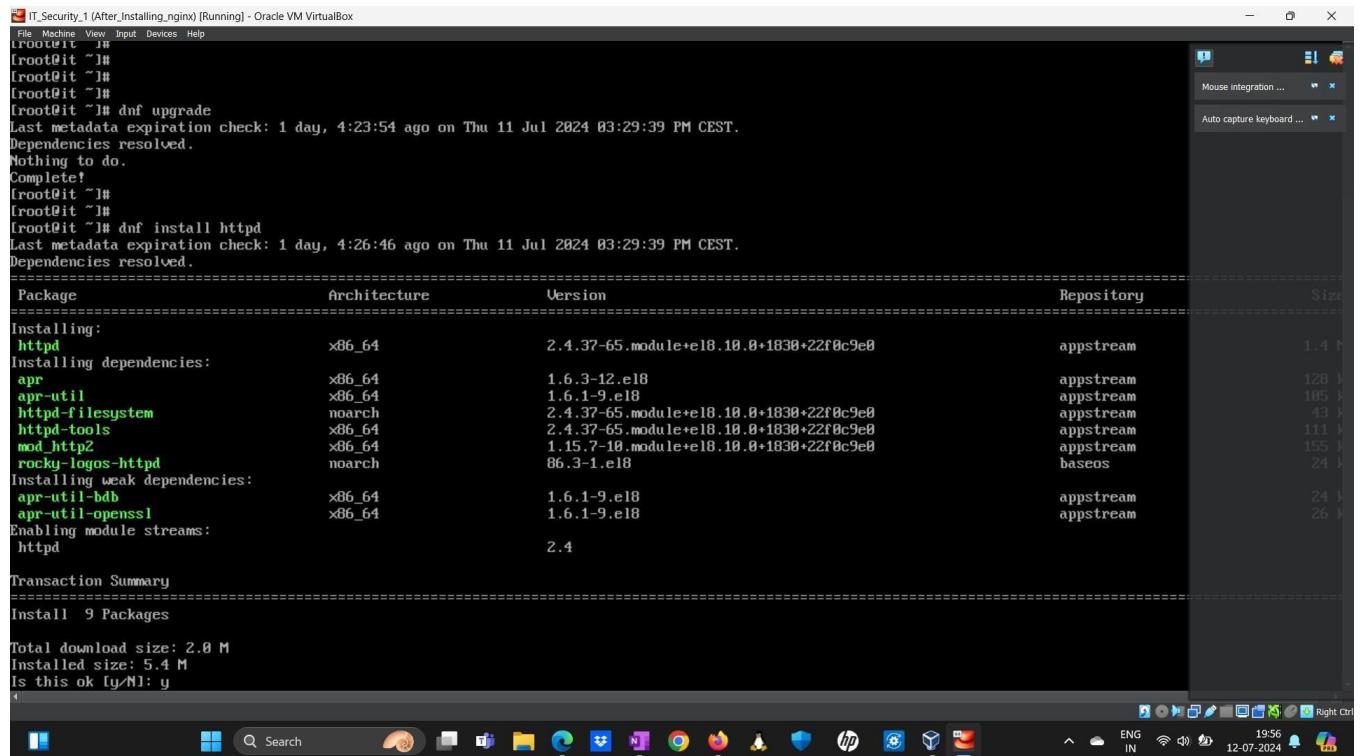
LAMP stands for “Linux Apache MariaDB PHP”. Here we have an OS , a web server software, a database (MySQL), and PHP for serverside scripting.

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



```
IT_Security_1 (After Installing nginx) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Iroot@it ~]#
Iroot@it ~]#
Iroot@it ~]#
Iroot@it ~]#
Iroot@it ~]# dnf upgrade
Last metadata expiration check: 1 day, 4:23:54 ago on Thu 11 Jul 2024 03:29:39 PM CEST.
Dependencies resolved.
Nothing to do.
Complete!
Iroot@it ~]#
Iroot@it ~]#
Iroot@it ~]# dnf install httpd
Last metadata expiration check: 1 day, 4:26:46 ago on Thu 11 Jul 2024 03:29:39 PM CEST.
Dependencies resolved.

=====
Package           Architecture      Version            Repository      Size
=====
Installing:
httpd             x86_64          2.4.37-65.module+e18.10.0+1830+22f0c9e0   appstream    1.4 M
Installing dependencies:
apr               x86_64          1.6.3-12.e18
apr-util          x86_64          1.6.1-9.e18
httpd-filesystem  noarch         2.4.37-65.module+e18.10.0+1830+22f0c9e0   appstream    43 K
httpd-tools        x86_64          2.4.37-65.module+e18.10.0+1830+22f0c9e0   appstream    111 K
mod_http2          x86_64          1.15.7-10.module+e18.10.0+1830+22f0c9e0   appstream    155 K
rocky-logos-httpd  noarch         86.3-1.e18
Enabling module streams:
httpd              x86_64          2.4

Transaction Summary
=====
Install 9 Packages

Total download size: 2.8 M
Installed size: 5.4 M
Is this ok [y/N]: y

```

We installed apache using the command:

“dnf install httpd”

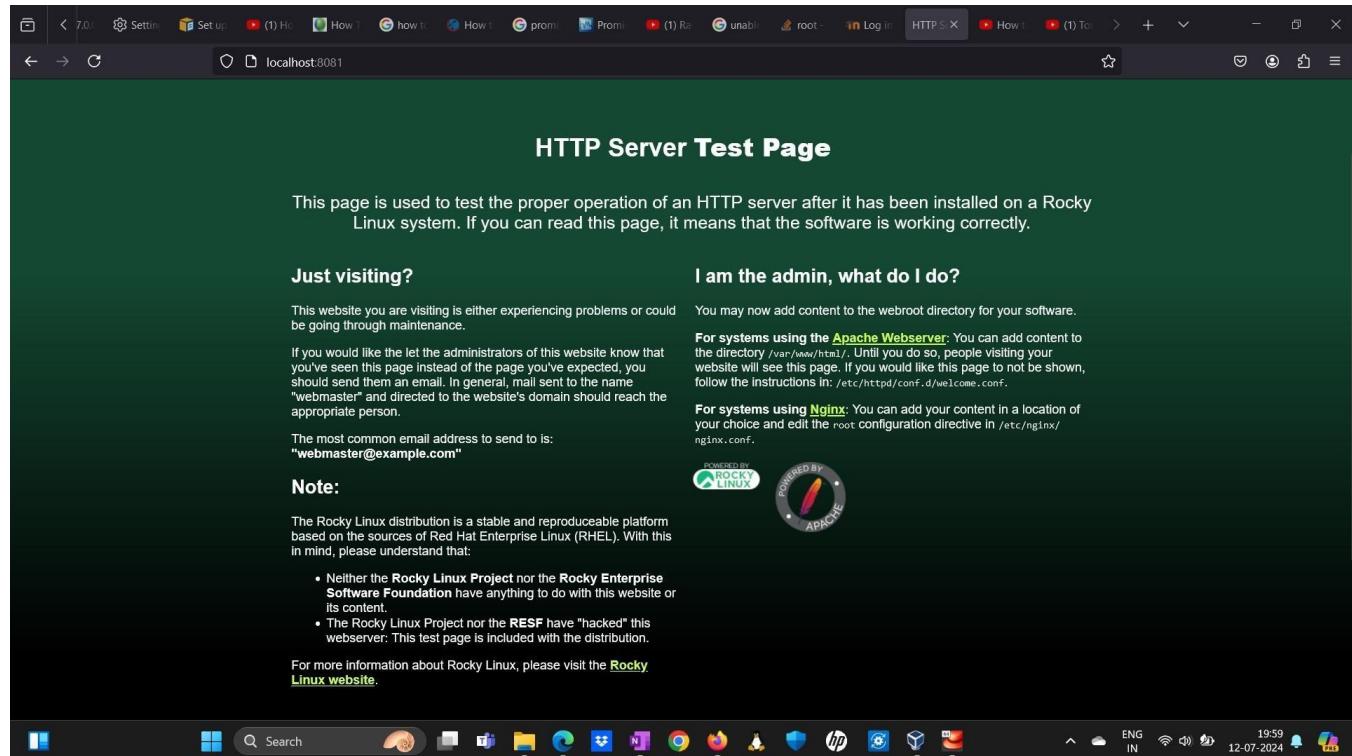
```

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.

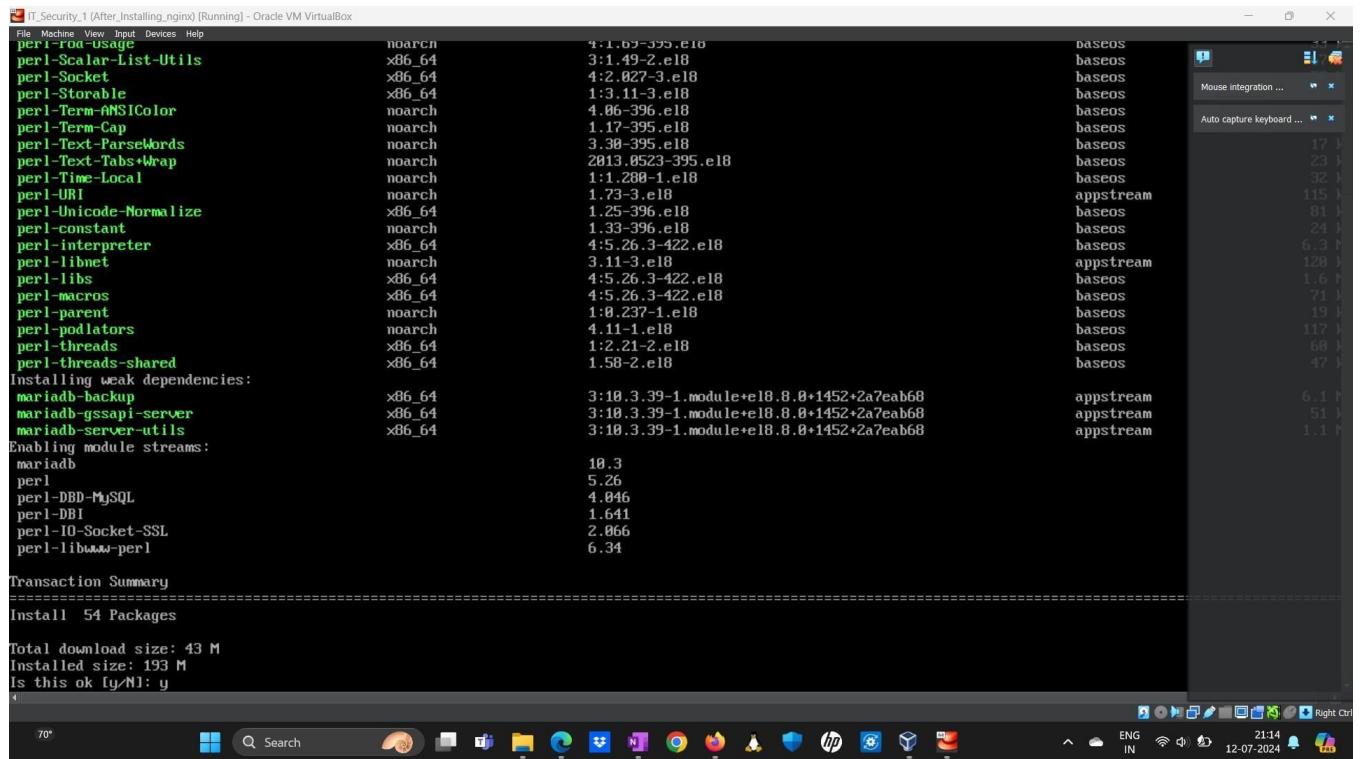


We can see the web page running successfully with the installed Apache server.

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”



```
IT:Security_1 (After_Installing.nginx) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
perl-rod-usage noarch 4:1.02-395.e18
perl-Scalar-List-Utils x86_64 3:1.49-2.e18
perl-Socket x86_64 4:2.027-3.e18
perl-Storable x86_64 1:3.11-3.e18
perl-Term-ANSIColor noarch 4.06-396.e18
perl-Term-Cap noarch 1.17-395.e18
perl-Text-ParseWords noarch 3.30-395.e18
perl-Text-TabsWrap noarch 2013.0523-395.e18
perl-Time-Local noarch 1:1.280-1.e18
perl-URI noarch 1.73-3.e18
perl-Unicode-Normalize x86_64 1.25-396.e18
perl-constant noarch 1.33-396.e18
perl-Interpreter x86_64 4:5.26.3-422.e18
perl-libnet noarch 3.11-3.e18
perl-libs x86_64 4:5.26.3-422.e18
perl-macros x86_64 4:5.26.3-422.e18
perl-parent noarch 1:0.237-1.e18
perl-podlators noarch 4.11-1.e18
perl-threads x86_64 1:2.21-2.e18
perl-threads-shared x86_64 1.58-2.e18
Installing weak dependencies:
mariadb-backup x86_64 3:10.3.39-1.module+el8.8.0+1452+2a7eab68 appstream
mariadb-gssapi-server x86_64 3:10.3.39-1.module+el8.8.0+1452+2a7eab68 appstream
mariadb-server-utils x86_64 3:10.3.39-1.module+el8.8.0+1452+2a7eab68 appstream
Enabling module streams:
mariadb 10.3
perl 5.26
perl-DBD-MySQL 4.046
perl-DBI 1.641
perl-IO-Socket-SSL 2.066
perl-libwww-perl 6.34
Transaction Summary
=====
Install 54 Packages
Total download size: 43 M
Installed size: 193 M
Is this ok [y/N]: y
[...]
70% 21:14 12-07-2024 ENG IN Right Ctrl
```

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

```
IT_Security_1 (After_Installing.nginx) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
JUL 12 21:17:19 IT.SICHERHEIT SYSTEM[1]: Started mariadb 10.3 database server.
[root@it conf.d]#
[root@it conf.d]#
[root@it conf.d]#
[root@it conf.d]# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
Enter current password for root (enter for none):
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

Set root password? [Y/n] Y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] _
```

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

```
By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.
```

```
Remove anonymous users? [Y/n] Y
... Success!
```

```
Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.
```

```
Disallow root login remotely? [Y/n] Y
```

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

```
By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.
```

```
Remove test database and access to it? [Y/n] n
```

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

```
MariaDB [mysql]> use mysql;
Database changed
MariaDB [mysql]> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
+-----+
3 rows in set (0.001 sec)

MariaDB [mysql]>
```

Now, we can check the default tables in mariadb:

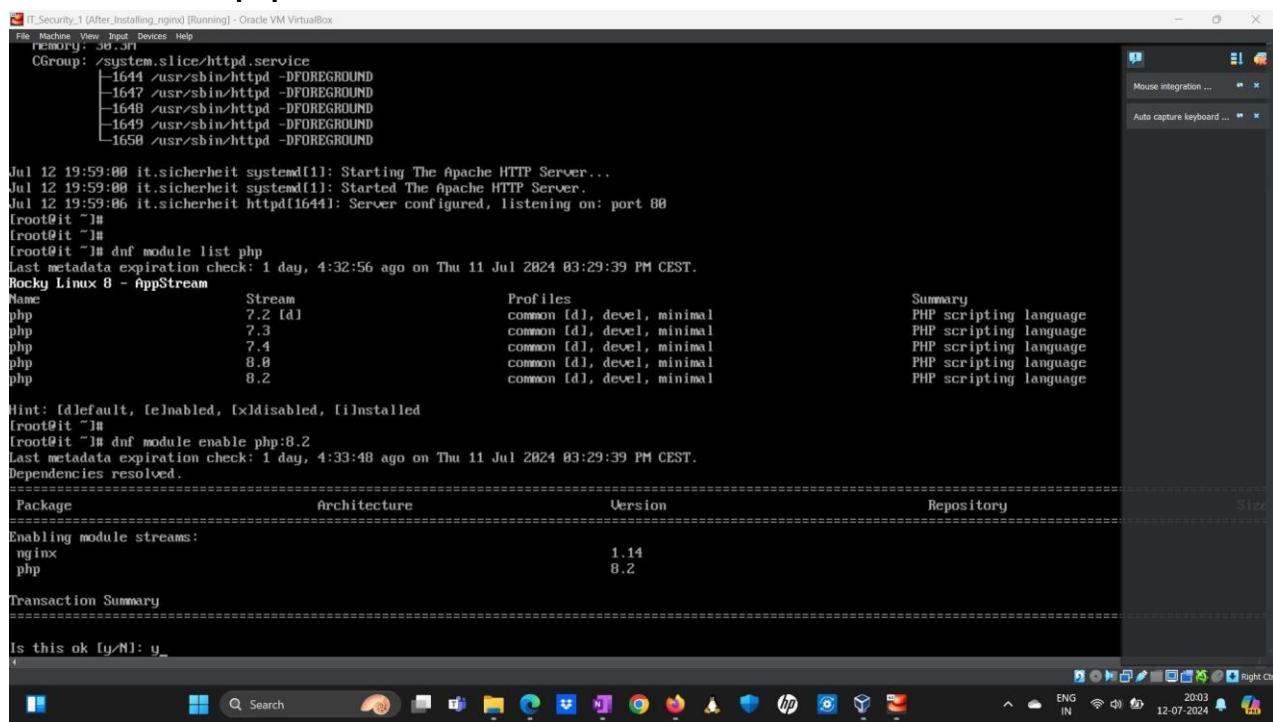
```
MariaDB [mysql]> show tables;
+-----+
| Tables_in_mysql |
+-----+
| column_stats      |
| columns_priv      |
| db                |
| event              |
| func              |
| general_log        |
| gtid_slave_pos    |
| help_category      |
| help_keyword       |
| help_relation      |
| help_topic         |
| 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 you 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”



```
IT_Security_1 (After Installing nginx) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Memory: 30.3M
CGroup: /system.slice/httd.service
└─1644 /usr/sbin/httd -DFOREGROUND
   ├─1647 /usr/sbin/httd -DFOREGROUND
   ├─1648 /usr/sbin/httd -DFOREGROUND
   ├─1649 /usr/sbin/httd -DFOREGROUND
   └─1650 /usr/sbin/httd -DFOREGROUND

Jul 12 19:59:00 it.sicherheit systemd[1]: Starting The Apache HTTP Server...
Jul 12 19:59:00 it.sicherheit systemd[1]: Started The Apache HTTP Server.
Jul 12 19:59:00 it.sicherheit httpd[1644]: Server configured, listening on: port 80
[root@it ~]#
[root@it ~]#
[root@it ~]# dnf module list php
Last metadata expiration check: 1 day, 4:32:56 ago on Thu 11 Jul 2024 03:29:39 PM CEST.
Rocky Linux 8 - AppStream
Name Stream Profiles Summary
php 7.2 [d] common [d], devel, minimal PHP scripting language
php 7.3 common [d], devel, minimal PHP scripting language
php 7.4 common [d], devel, minimal PHP scripting language
php 8.0 common [d], devel, minimal PHP scripting language
php 8.2 common [d], devel, minimal PHP scripting language

Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
[root@it ~]#
[root@it ~]# dnf module enable php:8.2
Last metadata expiration check: 1 day, 4:33:48 ago on Thu 11 Jul 2024 03:29:39 PM CEST.
Dependencies resolved.
=====
 Package Architecture Version Repository Size
=====
Enabling module streams:
  ngInx
  php
  1.14
  8.2

Transaction Summary
=====

Is this ok [y/N]: y_

```

Installation of PHP was successful:

```

IT_Security_1 (After_Installing_nginx) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
transaction test succeeded.
Running transaction
Preparing : 
Installing  : php-common-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
Installing  : php-cli-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
Installing  : php-opcache-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
Installing  : php-pdo-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
Installing  : php-xml-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
Installing  : oniguruma-6.8.2-3.el8.x86_64
Running scriptlet: oniguruma-6.8.2-3.el8.x86_64
Installing  : php-mbstring-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
Running scriptlet: nginx-filesystem-1:1.14.1-9.module+el8.4.0+542+81547229.noarch
Installing  : nginx-filesystem-1:1.14.1-9.module+el8.4.0+542+81547229.noarch
Installing  : php-fpm-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
Running scriptlet: php-fpm-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
Installing  : php-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
Running scriptlet: php-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
Running scriptlet: php-fpm-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
Verifying   : nginx-filesystem-1:1.14.1-9.module+el8.4.0+542+81547229.noarch
Verifying   : oniguruma-6.8.2-3.el8.x86_64
Verifying   : php-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
Verifying   : php-cli-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
Verifying   : php-common-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
Verifying   : php-fpm-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
Verifying   : php-mbstring-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
Verifying   : php-opcache-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
Verifying   : php-pdo-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
Verifying   : php-xml-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64

Installed:
nginx-filesystem-1:1.14.1-9.module+el8.4.0+542+81547229.noarch
php-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
php-common-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
php-mbstring-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
php-pdo-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64

Complete!
[root@it ~]#
[root@it ~]#

```

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.

```

IT_Security_1 (After_Installing_nginx) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
php-common-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
Verifying   : php-fpm-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
Verifying   : php-mbstring-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
Verifying   : php-opcache-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
Verifying   : php-pdo-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
Verifying   : php-xml-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64

Installed:
nginx-filesystem-1:1.14.1-9.module+el8.4.0+542+81547229.noarch
php-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
php-common-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
php-mbstring-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64
php-pdo-8.2.13-1.module+el8.18.0+1596+477f83f8.x86_64

Complete!
[root@it ~]#
[root@it ~]#
[root@it ~]#
[root@it ~]# systemctl start php-fpm
[root@it ~]# systemctl status php-fpm
● php-fpm.service - The PHP FastCGI Process Manager
   Loaded: loaded (/usr/lib/systemd/system/php-fpm.service; disabled; vendor preset: disabled)
     Active: active (running) since Fri 2024-07-12 20:06:47 CEST; 5s ago
       Main PID: 2429 (php-fpm)
          Status: "Ready to handle connections"
             Tasks: 6 (limit: 24884)
            Memory: 14.3M
           CGroup: /system.slice/php-fpm.service
                   ├─2429 php-fpm: master process (/etc/php-fpm.conf)
                   ├─2430 php-fpm: pool www
                   ├─2431 php-fpm: pool www
                   ├─2432 php-fpm: pool www
                   ├─2433 php-fpm: pool www
                   └─2434 php-fpm: pool www

Jul 12 20:06:47 it.sicherheit systemd[1]: Starting The PHP FastCGI Process Manager...
Jul 12 20:06:47 it.sicherheit systemd[1]: Started The PHP FastCGI Process Manager.
[root@it ~]#

```

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.

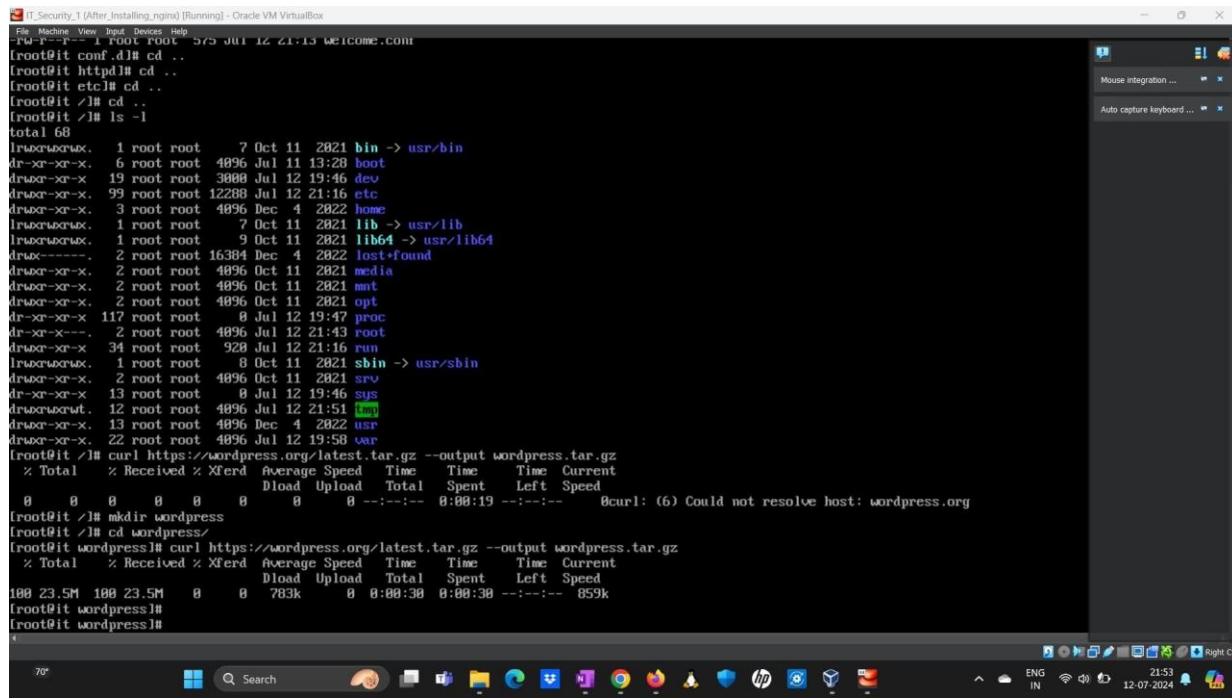
First, we created a folder to contain everything related to wordpress and then entered that directory using:

“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”



The screenshot shows a terminal window titled "IT_Security_1 (After_Installing.nginx) [Running] - Oracle VM VirtualBox". The terminal output is as follows:

```
File Machine View Input Devices Help
-rw-r--r-- 1 root root 375 Jul 12 21:13 welcome.com
[root@it ~]# cd conf.d
[root@it conf.d]# cd ..
[root@it etc]# cd ..
[root@it ~]# ls -l
total 68
lrwxrwxrwx. 1 root root 7 Oct 11 2021 bin -> usr/bin
dr-xr-xr-x. 6 root root 4096 Jul 11 13:28 boot
drwxr-xr-x. 19 root root 3088 Jul 12 19:46 dev
drwxr-xr-x. 99 root root 12288 Jul 12 21:16 etc
drwxr-xr-x. 3 root root 4096 Dec 4 2022 home
lrwxrwxrwx. 1 root root 7 Oct 11 2021 lib -> usr/lib
lrwxrwxrwx. 1 root root 9 Oct 11 2021 lib64 -> usr/lib64
drwx----- 2 root root 16384 Dec 4 2022 lost+found
drwxr-xr-x. 2 root root 4096 Oct 11 2021 media
drwxr-xr-x. 2 root root 4096 Oct 11 2021 mnt
drwxr-xr-x. 2 root root 4096 Oct 11 2021 opt
dr-xr-xr-x. 117 root root 8 Jul 12 19:47 proc
dr-xr-x---. 2 root root 4096 Jul 12 21:43 root
drwxr-xr-x. 34 root root 928 Jul 12 21:16 run
lrwxrwxrwx. 1 root root 8 Oct 11 2021 sbin -> usr/sbin
drwxr-xr-x. 2 root root 4096 Oct 11 2021 srv
drwxr-xr-x. 13 root root 8 Jul 12 19:46 sys
drwxrwxrwt. 12 root root 4096 Jul 12 21:51 tmp
drwxr-xr-x. 13 root root 4096 Dec 4 2022 usr
drwxr-xr-x. 22 root root 4096 Jul 12 19:58 var
[root@it ~]# curl https://wordpress.org/latest.tar.gz --output wordpress.tar.gz
  % Total    % Received % Xferd  Average Speed   Time   Time  Current
     0     0     0     0   0     0  --:--:--  0:00:19  --:--:-- 8curl: (6) Could not resolve host: wordpress.org
[root@it ~]# mkdir wordpress
[root@it ~]# cd wordpress/
[root@it wordpress]# curl https://wordpress.org/latest.tar.gz --output wordpress.tar.gz
  % Total    % Received % Xferd  Average Speed   Time   Time  Current
  100  23.5M  100 23.5M    0     0  783k      0  0:00:30  0:00:30  --:--:-- 859k
[root@it wordpress]#
[root@it wordpress]#
```

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 {} \;”

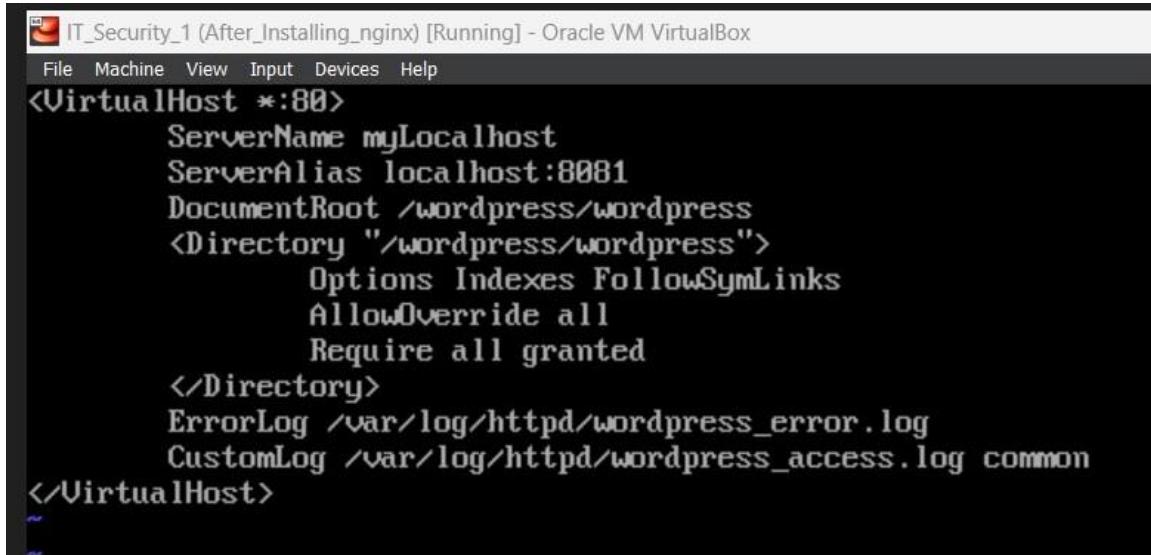
```

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]#

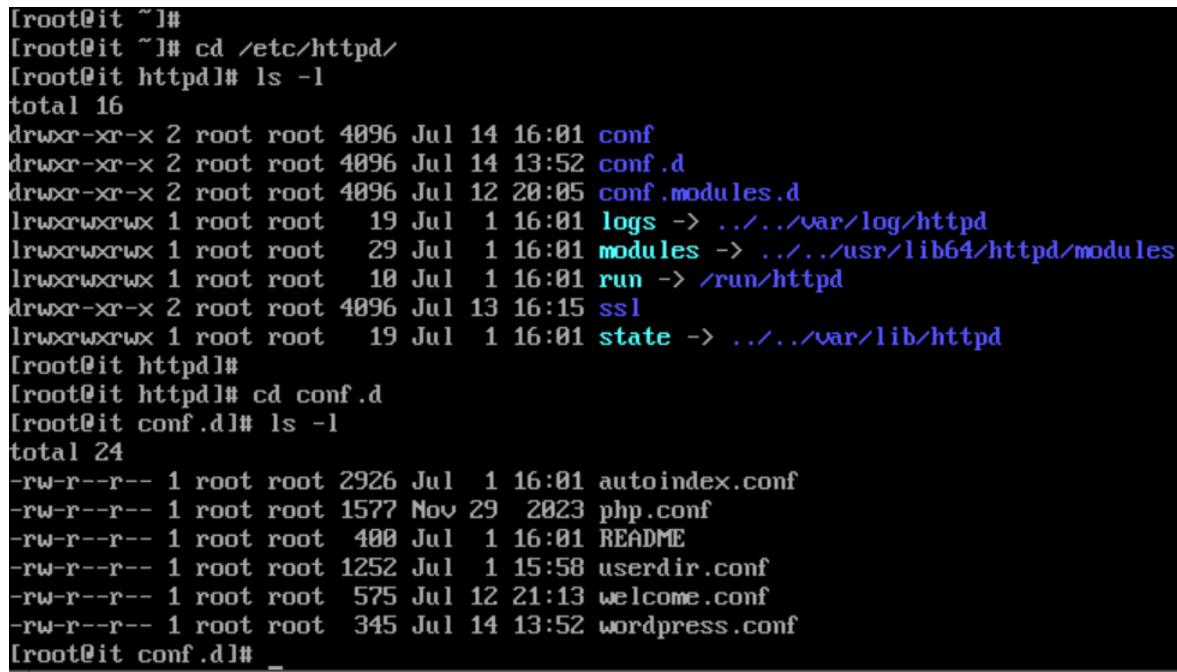
```

After this, we go to 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:



```
<VirtualHost *:80>
    ServerName mylocalhost
    ServerAlias localhost:8081
    DocumentRoot /wordpress/wordpress
    <Directory "/wordpress/wordpress">
        Options Indexes FollowSymLinks
        AllowOverride all
        Require all granted
    </Directory>
    ErrorLog /var/log/httpd/wordpress_error.log
    CustomLog /var/log/httpd/wordpress_access.log common
</VirtualHost>
```



```
[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:

```
[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 12 22:22 wordpress.conf
[root@it conf.d]# systemctl restart httpd
[root@it conf.d]# 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 Fri 2024-07-12 22:25:13 CEST; 2s ago
       Docs: man:httpd.service(8)
   Main PID: 9888 (httpd)
     Status: "Started, listening on: port 80"
        Tasks: 213 (limit: 24884)
      Memory: 29.2M
     CGroup: /system.slice/httpd.service
             ├─9888 /usr/sbin/httpd -DFOREGROUND
             ├─9889 /usr/sbin/httpd -DFOREGROUND
             ├─9890 /usr/sbin/httpd -DFOREGROUND
             ├─9891 /usr/sbin/httpd -DFOREGROUND
             ├─9892 /usr/sbin/httpd -DFOREGROUND

Jul 12 22:25:13 it.sicherheit systemd[1]: Starting The Apache HTTP Server...
Jul 12 22:25:13 it.sicherheit systemd[1]: Started The Apache HTTP Server.
Jul 12 22:25:13 it.sicherheit httpd[9888]: Server configured, listening on: port 80
[root@it conf.d]#
```

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)

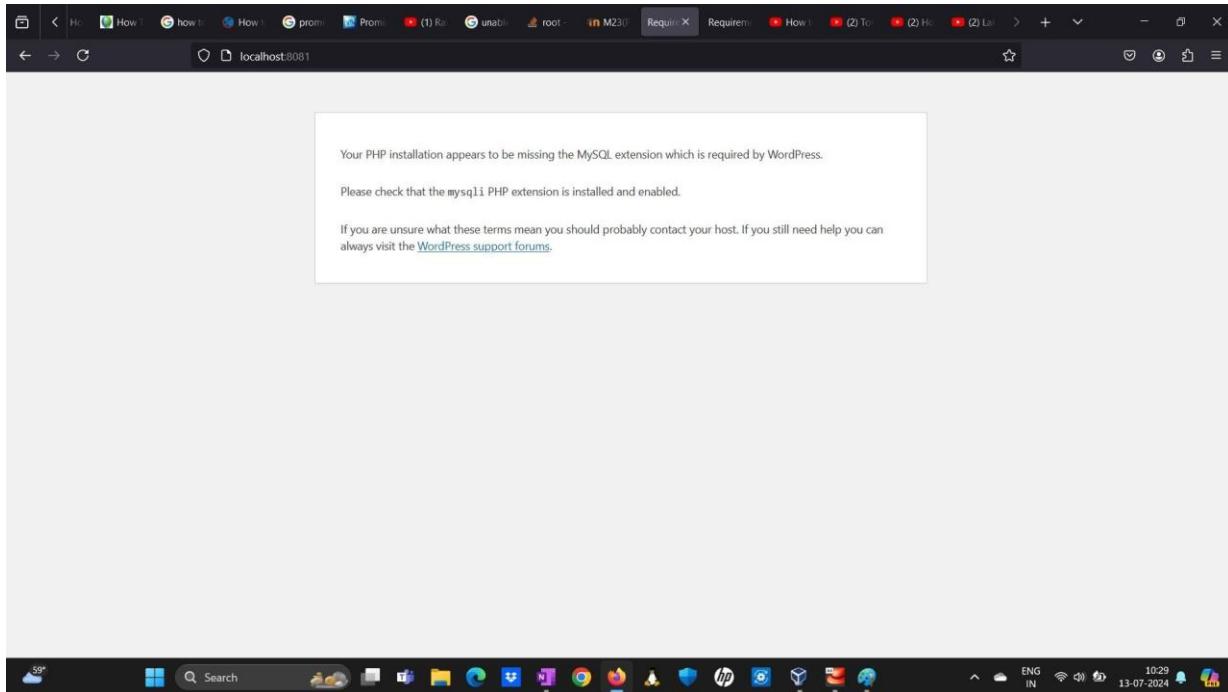
MariaDB [(none)]> _
```

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”

```
IT_Security_1 (After_Installing_nginx) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
[Root@it ~]# dnf install php-mysqlnd php-gd php-json php-mbstring php-pecl-zip php-xml php-imagick
Last metadata expiration check: 1 day, 19:04:38 ago on Thu 11 Jul 2024 03:29:39 PM CEST.
Package php-common-8.2.13-1.module+el8.10.0+1596+477f03f8.x86_64 is already installed.
Package php-mbstring-8.2.13-1.module+el8.10.0+1596+477f03f8.x86_64 is already installed.
Package php-xml-8.2.13-1.module+el8.10.0+1596+477f03f8.x86_64 is already installed.
No match for argument: php-imagick
Error: Unable to find a match: php-imagick
[Root@it ~]# dnf install php-mysqlnd php-gd php-json php-mbstring php-pecl-zip php-xml
Last metadata expiration check: 1 day, 19:08:04 ago on Thu 11 Jul 2024 03:29:39 PM CEST.
Package php-common-8.2.13-1.module+el8.10.0+1596+477f03f8.x86_64 is already installed.
Package php-mbstring-8.2.13-1.module+el8.10.0+1596+477f03f8.x86_64 is already installed.
Package php-xml-8.2.13-1.module+el8.10.0+1596+477f03f8.x86_64 is already installed.
Dependencies resolved.
=====
 Package           Architecture     Version            Repository      Size
=====
Installing:
 php-gd           x86_64          8.2.13-1.module+el8.10.0+1596+477f03f8   appstream      86 
 php-mysqlnd      x86_64          8.2.13-1.module+el8.10.0+1596+477f03f8   appstream      187 
 php-pecl-zip     x86_64          1.22.3-1.module+el8.10.0+1596+477f03f8   appstream      68 
Installing dependencies:
 gd               x86_64          2.2.5-7.el8
 libXpm            x86_64          2.1-14.el8
 libXpm            x86_64          3.5.12-11.el8
 libjpeg-turbo    x86_64          1.5.3-12.el8
 libtiff           x86_64          4.0.9-31.el8
 libwebp           x86_64          1.0.8-10.el8
 libzipp           x86_64          1.7.3-1.module+el8.10.0+1596+477f03f8   appstream      143 
Transaction Summary
=====
Install 10 Packages
Total download size: 1.2 M
Installed size: 3.2 M
Is this ok [y/N]: _
```

Install all the packages, and then restart everything using:

“systemctl restart httpd”

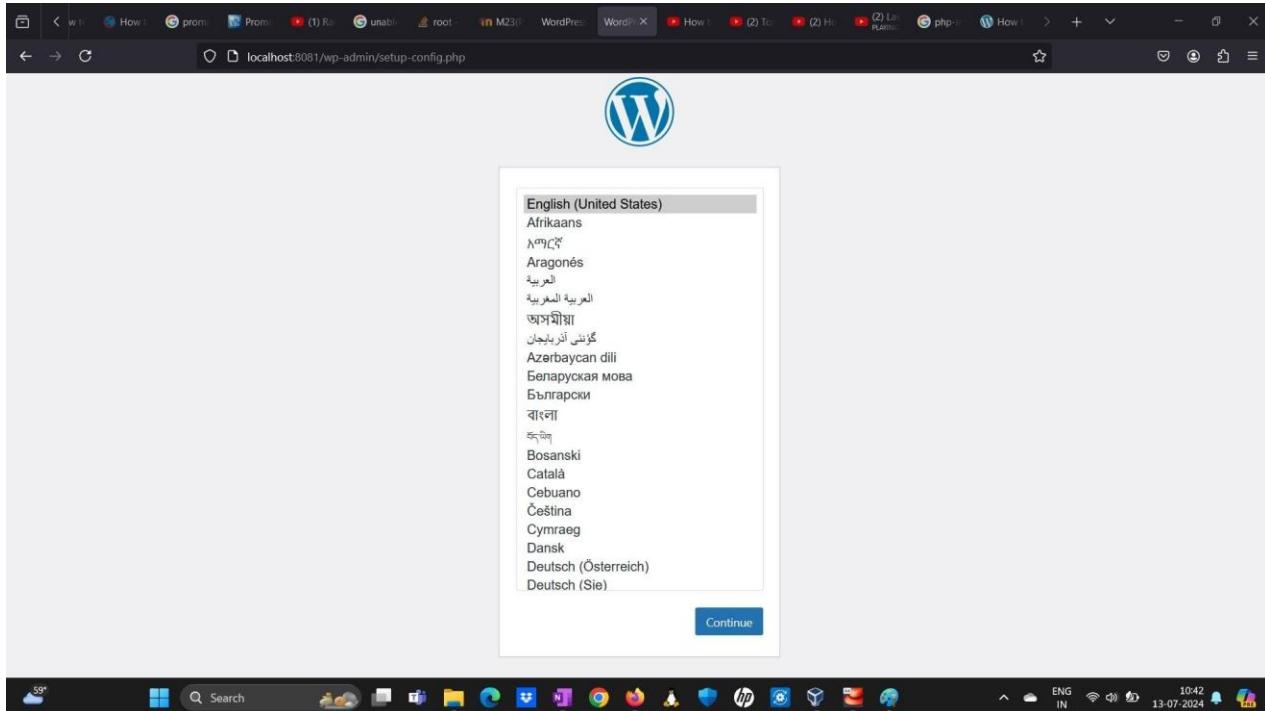
“systemctl restart php-fpm”

“systemctl restart mariadb”
“systemctl restart snapd”

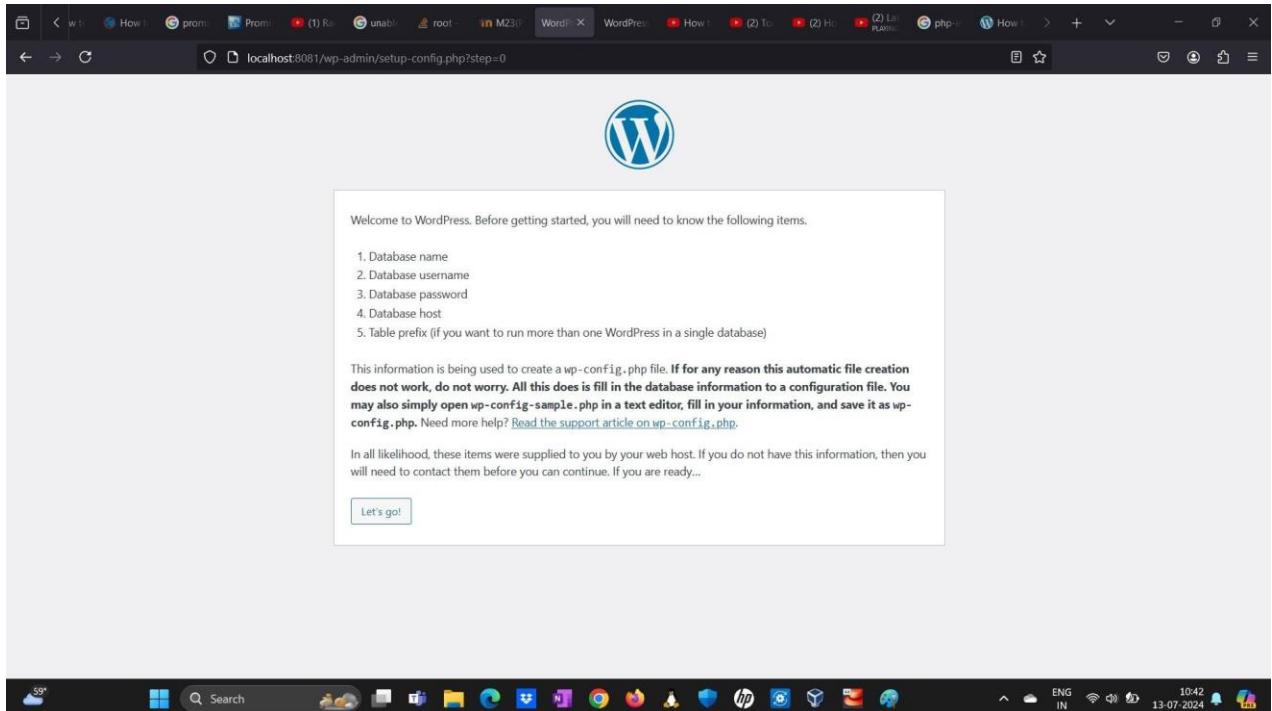
7.4 Wordpress Settings

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

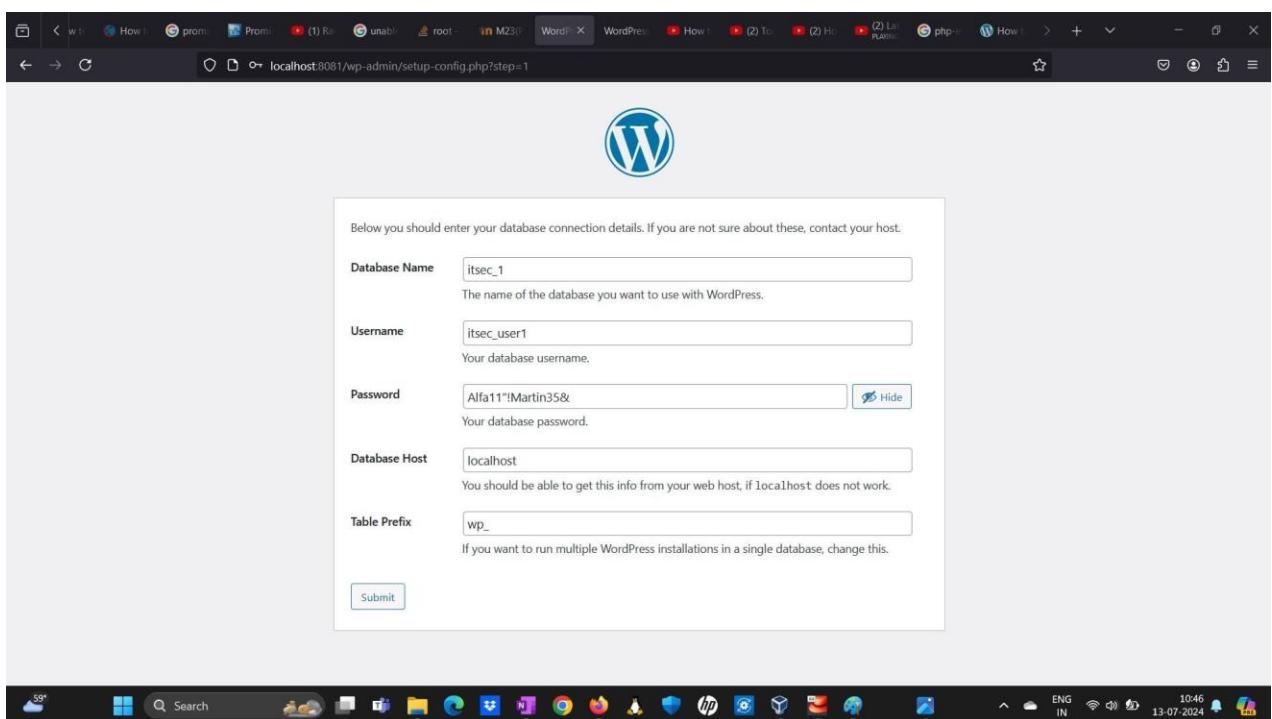
Select the language.



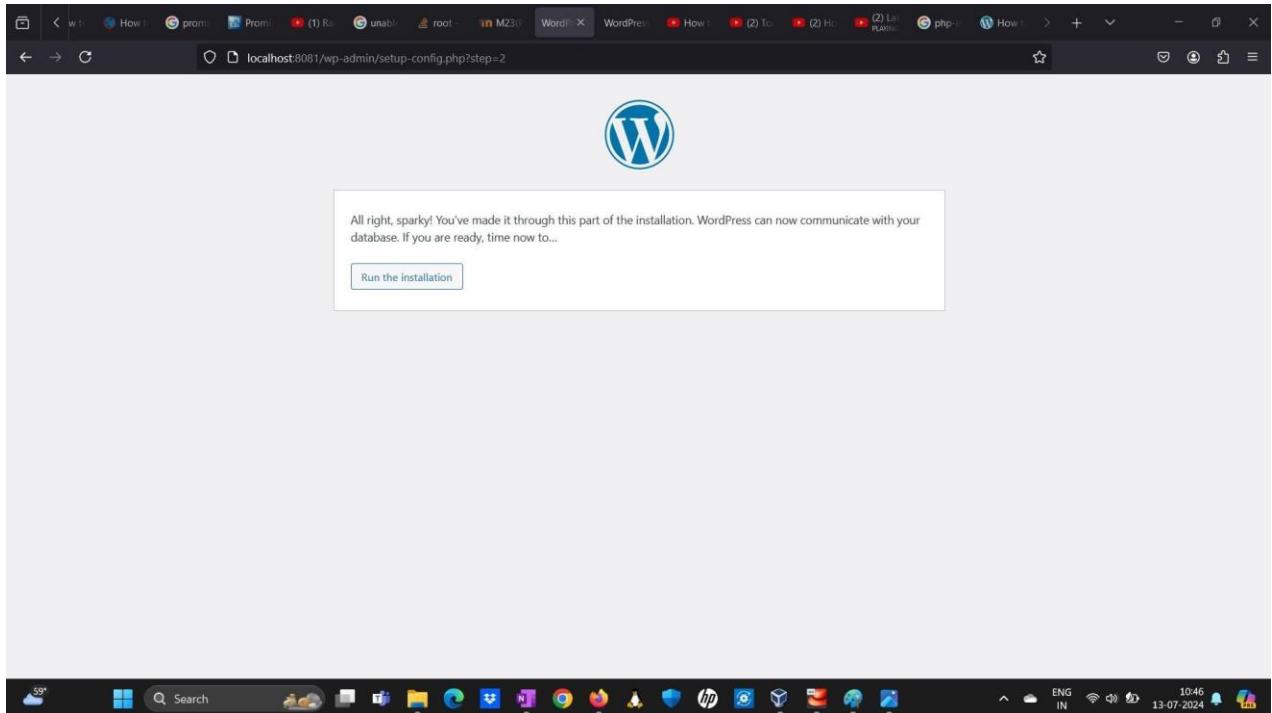
It will then give you a prompt about the database information that we need:



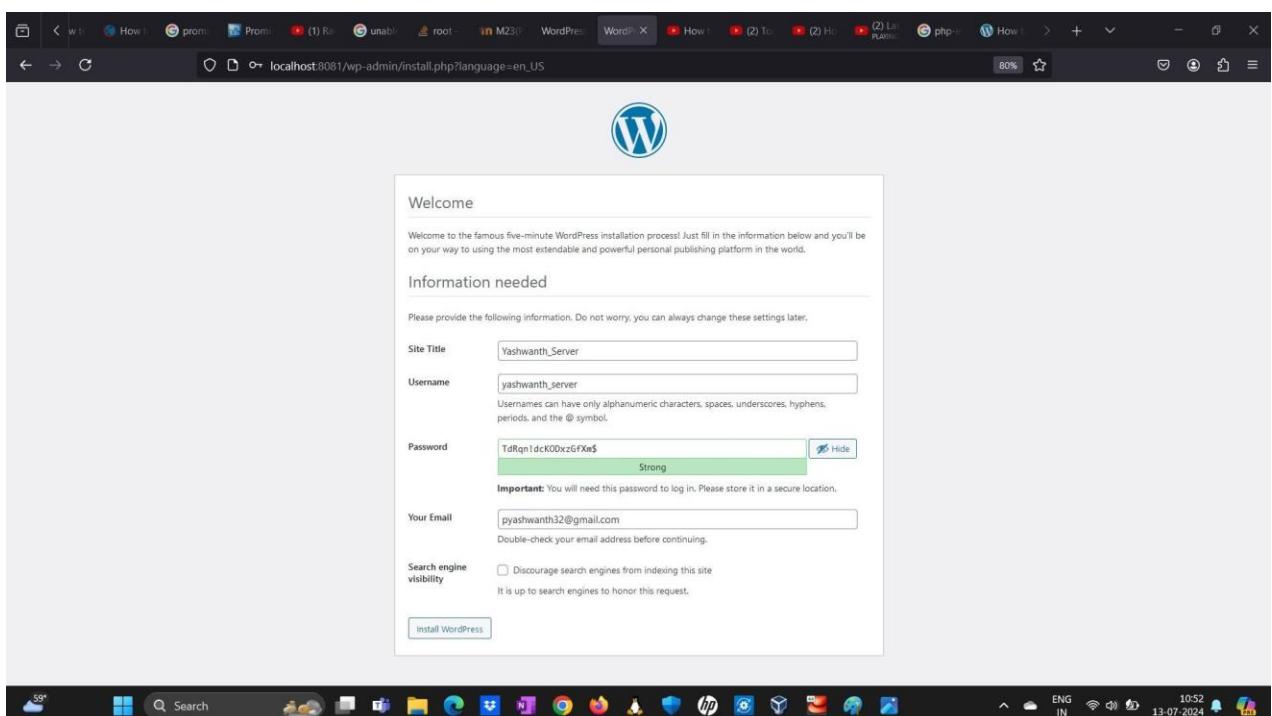
Here, we specify the details of the database that we just created on mariadb.



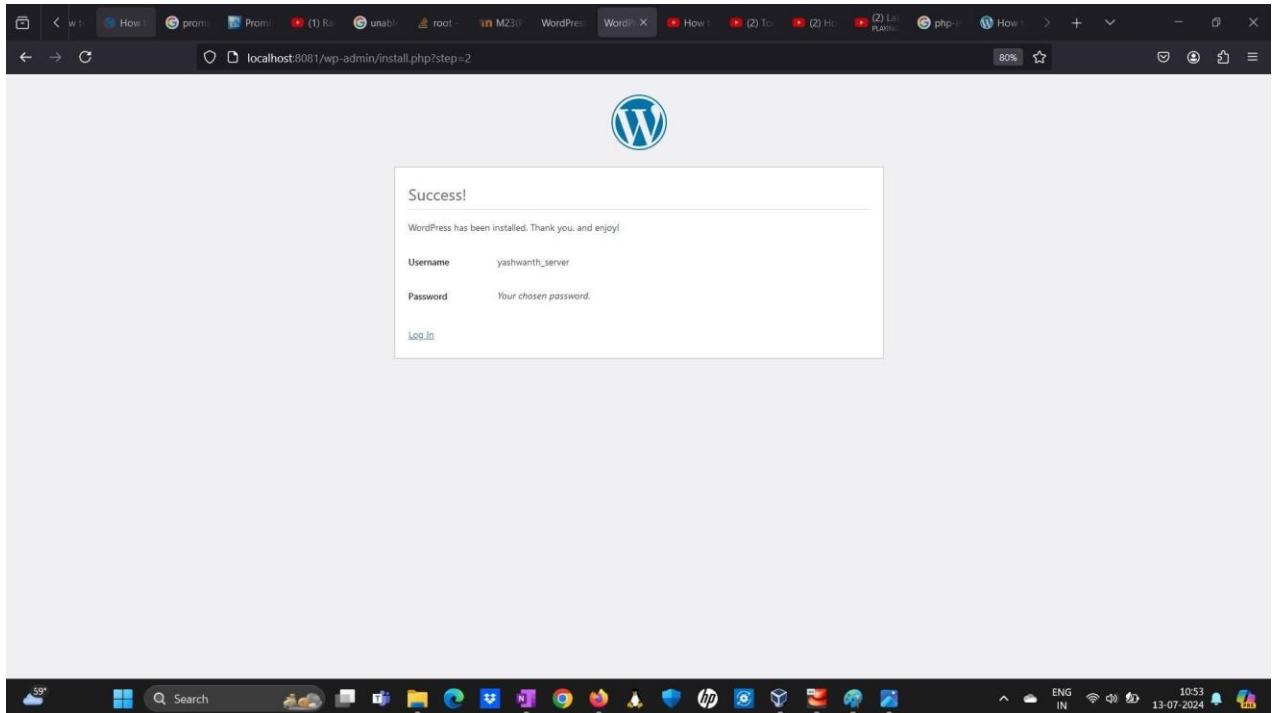
After this installation, we are ready to use Wordpress!



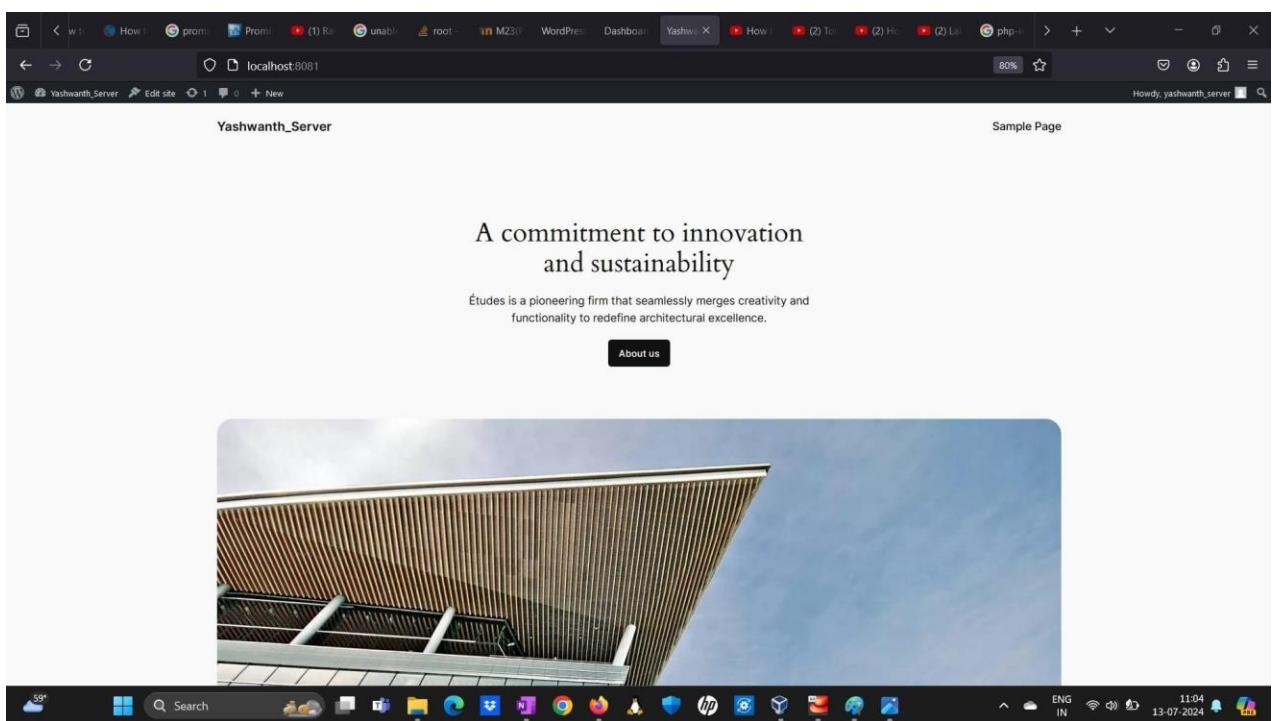
Now, we do the wordpress installation process by providing the details mentioned below:



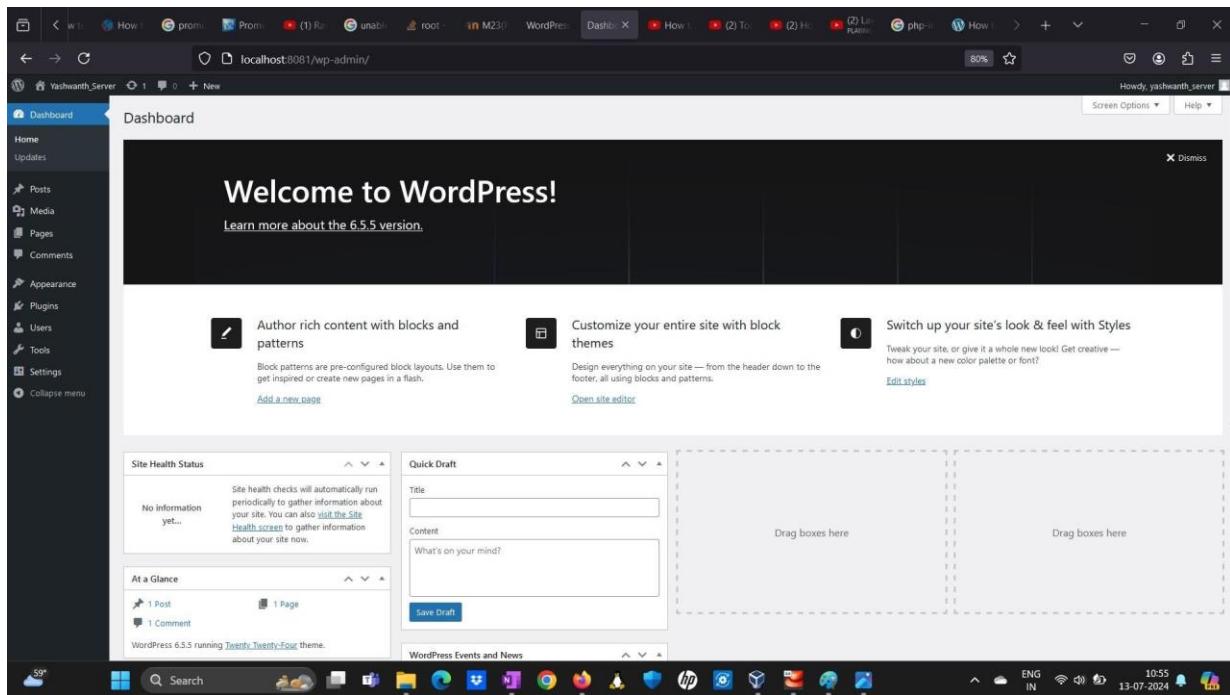
Wordpress will finally be installed:



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/”

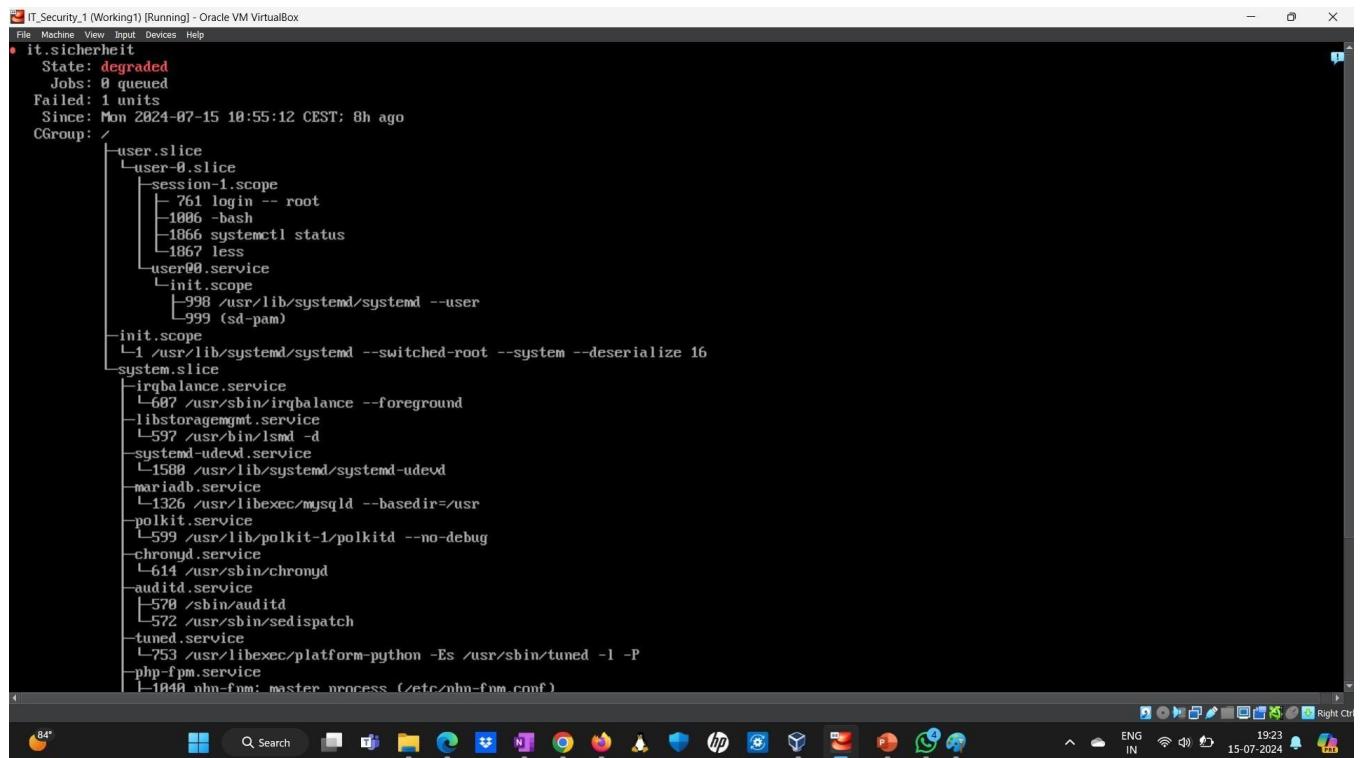


8 Hardening the System

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.



```
IT_Security_1 (Working1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
● it.sicherheit
  State: degrada...
  Jobs: 0 queued
  Failed: 1 units
  Since: Mon 2024-07-15 10:55:12 CEST; 8h ago
  CGrou...
CGroup: /
  └─user.slice
    └─user-0.slice
      ├─session-1.scope
        ├─761 login -- root
        ├─1086 -bash
        ├─1866 systemctl status
        ├─1867 less
      └─user@0.service
        └─init.scope
          ├─998 /usr/lib/systemd/systemd --user
          └─999 (sd-pam)
    └─init.scope
      └─1 /usr/lib/systemd/systemd --switched-root --system --deserializ...
  └─system.slice
    └─irqbalance.service
      └─687 /usr/sbin/irqbalance --foreground
    └─libstoragemgmt.service
      └─597 /usr/bin/lsmmd -d
    └─systemd-udevd.service
      └─1580 /usr/lib/systemd/systemd-udevd
    └─mariadb.service
      └─1326 /usr/libexec/mysqld --basedir=/usr
    └─polkit.service
      └─599 /usr/lib/polkit-1/polkitd --no-debug
    └─chronyd.service
      └─614 /usr/sbin/chronyd
    └─auditd.service
      └─570 /sbin/auditd
    └─572 /usr/sbin/sedispatch
    └─tuned.service
      └─753 /usr/libexec/platform-python -Es /usr/sbin/tuned -l -P
    └─php-fpm.service
      └─1040 php-fpm: master process (/etc/php-fpm.conf)
```

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.

```

IT_Security_1 (Working1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
# consult the online docs. You have been warned.
#
# Configuration and logfile names: If the filenames you specify for many
# of the server's control files begin with "/" (or "drive:" for Win32), the
# server will use that explicit path. If the filenames do *not* begin
# with "/", the value of ServerRoot is prepended -- so 'log/access_log'
# with ServerRoot set to '/www' will be interpreted by the
# server as '/www/log/access_log', whereas '/log/access_log' will be
# interpreted as '/log/access_log'.
#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# Do not add a slash at the end of the directory path. If you point
# ServerRoot at a non-local disk, be sure to specify a local disk on the
# Mutex directive, if file-based mutexes are used. If you wish to share the
# same ServerRoot for multiple httpd daemons, you will need to change at
# least PidFile.
#
ServerRoot "/etc/httpd"
#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 5975
#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
"httpd.conf" 356L, 11981C written
[root@bit conf]# vi httpd.conf_

```

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.

```

IT_Security_1 (Working1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
<VirtualHost *:5975>
    ServerName mylocalhost
    ServerAlias localhost:8081
    DocumentRoot /wordpress/wordpress
    <Directory "/wordpress/wordpress">
        Options Indexes FollowSymLinks
        AllowOverride all
        Require all granted
    </Directory>
    ErrorLog /var/log/httpd/wordpress_error.log
    CustomLog /var/log/httpd/wordpress_access.log common
</VirtualHost>
~_
~_

```

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]# 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: 24084)
     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 in a future release.

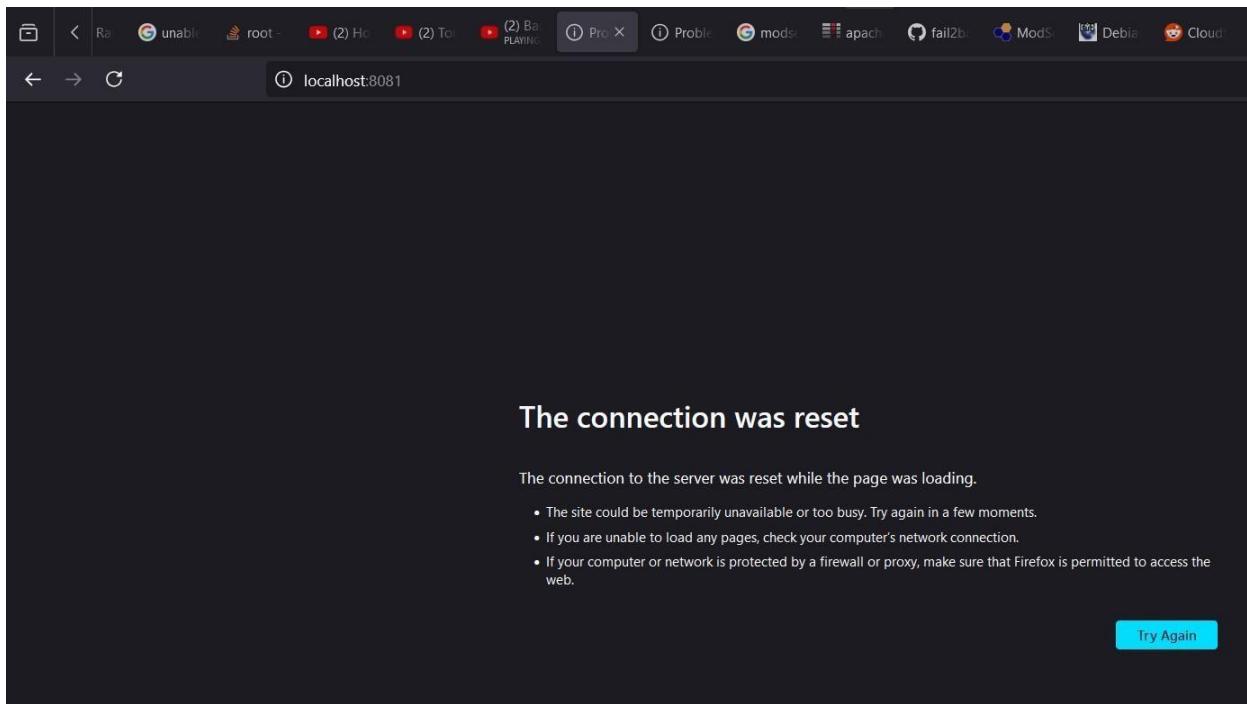
[root@it conf.d]# 
[root@it conf.d]# firewall-cmd --permanent --add-port=5975/tcp
success
[root@it conf.d]# 
[root@it conf.d]# firewall-cmd --reload
success
[root@it conf.d]# systemctl restart firewalld
[root@it conf.d]#
```

After this, change the apache listening ports so that apache can also listen to port 5975. This is done using:

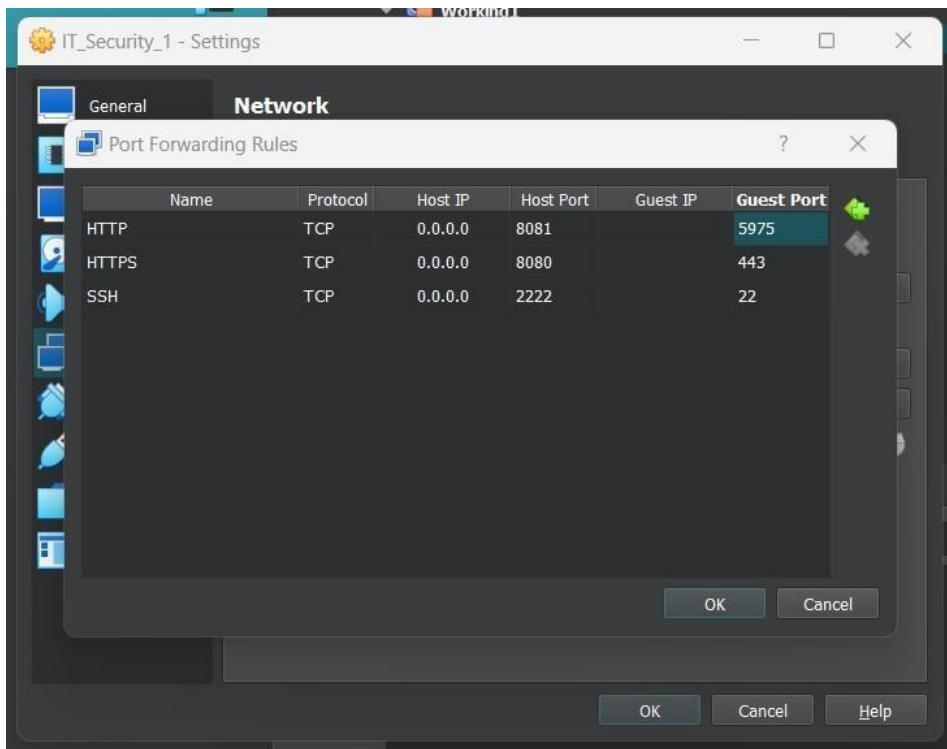
“ss -tuln | grep 5975”

```
[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]#
```

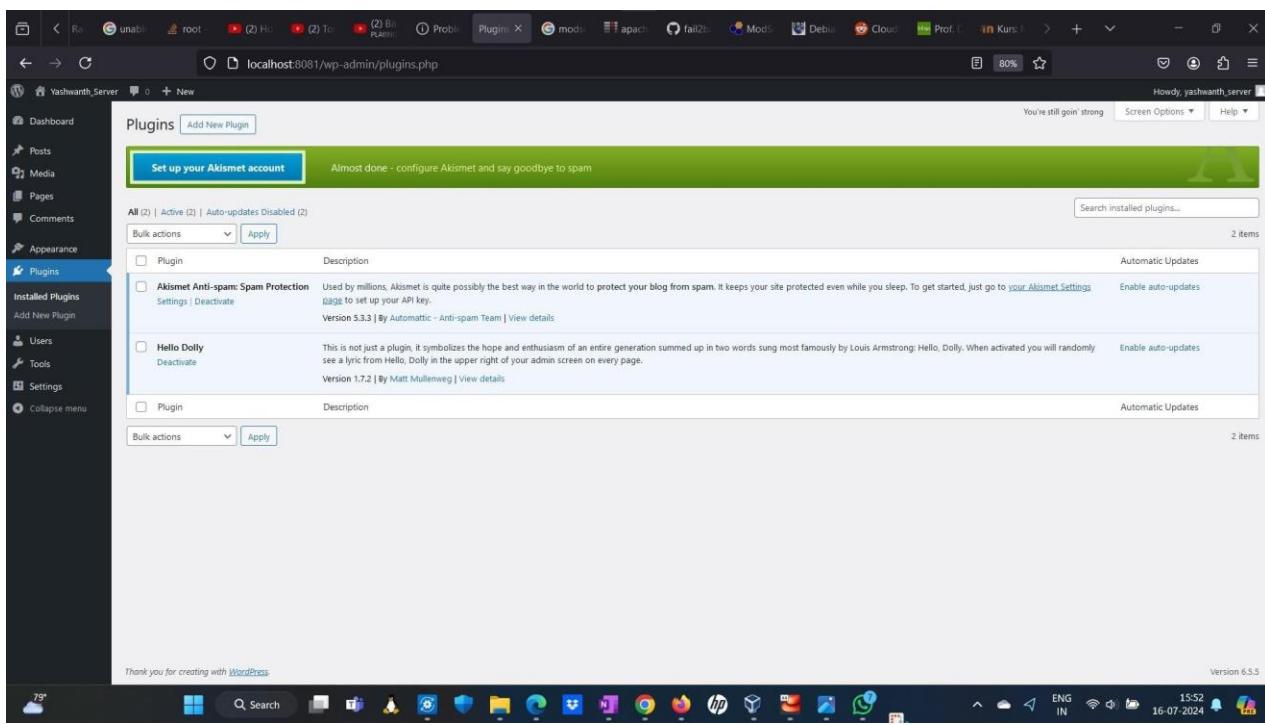
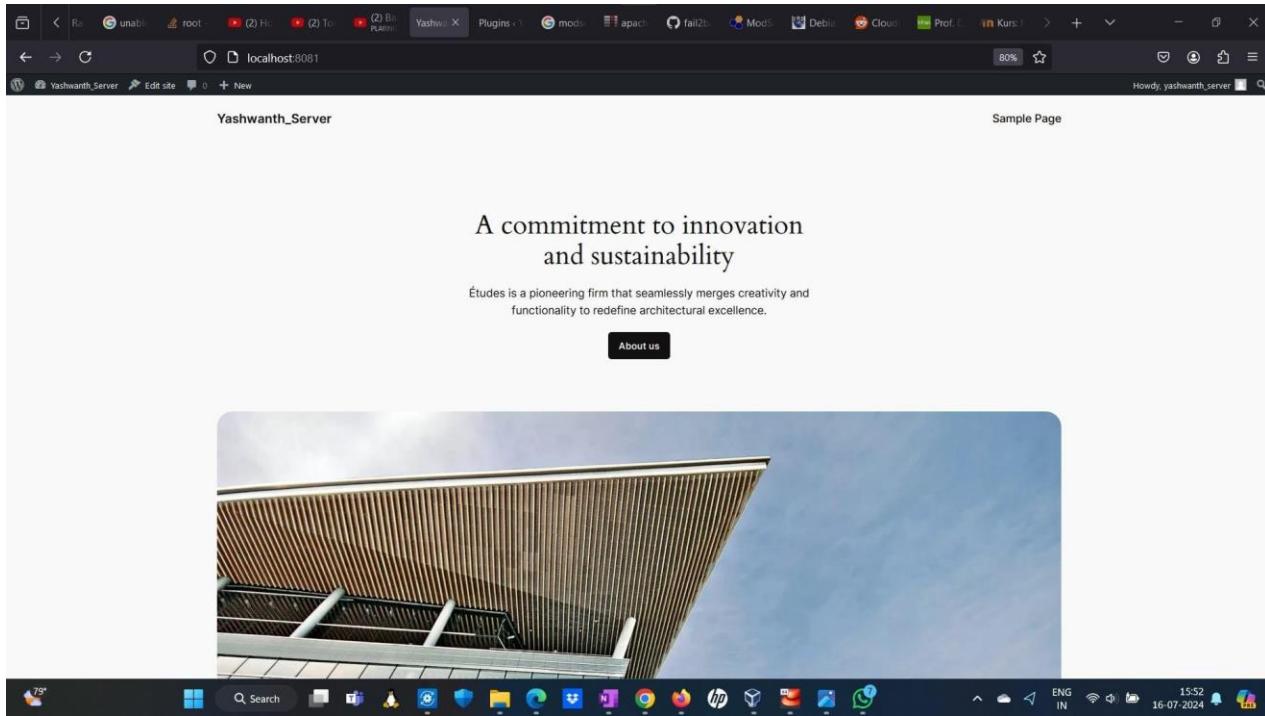
Now, on reloading our localhost:8081



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.



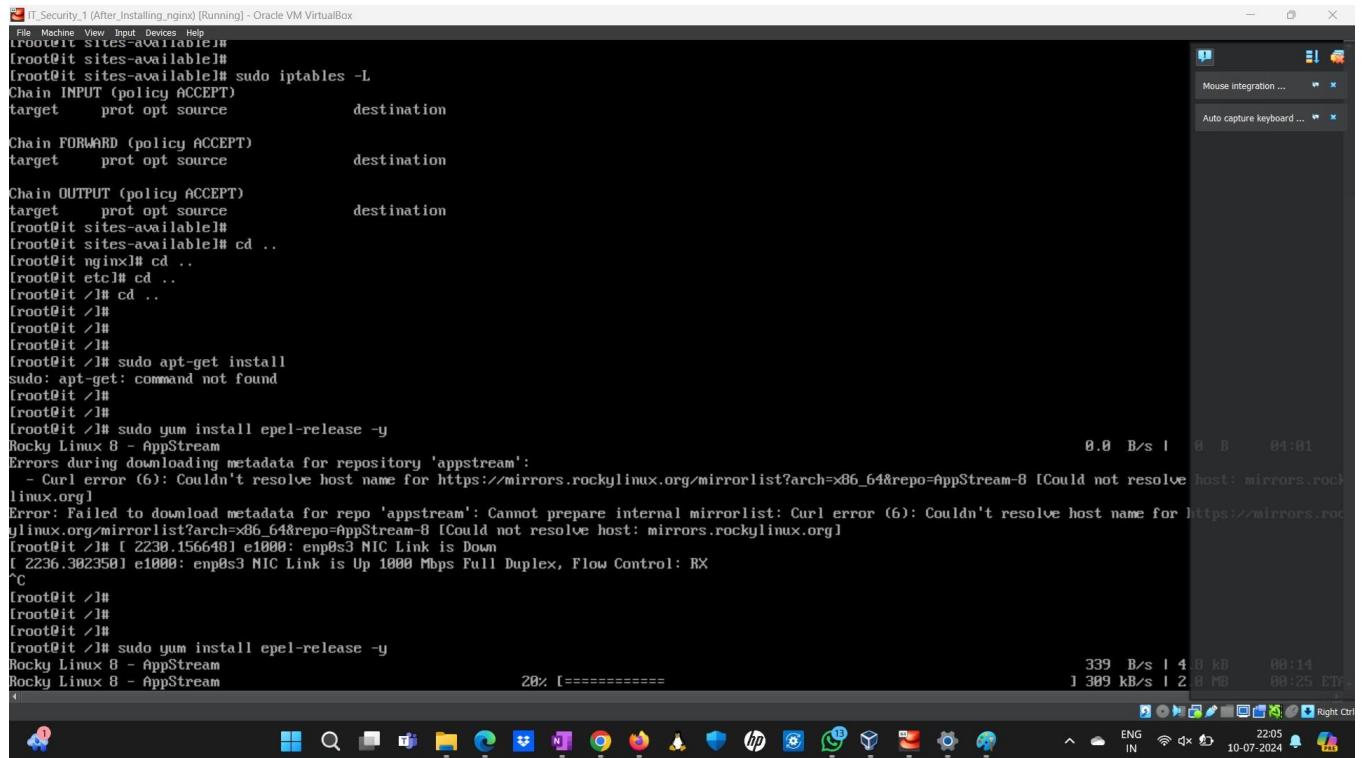
After this, I checked and everything is working perfectly.



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.



```
[root@it sites-available]# sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@it sites-available]# cd ..
[root@it nginx]# cd ..
[root@it etc]# cd ..
[root@it /]# cd ..
[root@it /]#
[root@it /]#
[root@it /]# sudo apt-get install
sudo: apt-get: command not found
[root@it /]#
[root@it /]#
[root@it /]# sudo yum install epel-release -y
Rocky Linux 8 - AppStream
Errors during downloading metadata for repository 'appstream':
 - Curl error (6): Couldn't resolve host name for https://mirrors.rockylinux.org/mirrorlist?arch=x86_64&repo=AppStream-8 [Could not resolve host: mirrors.rockylinux.org]
Error: Failed to download metadata for repo 'appstream': Cannot prepare internal mirrorlist: Curl error (6): Couldn't resolve host name for https://mirrors.rockylinux.org/mirrorlist?arch=x86_64&repo=AppStream-8 [Could not resolve host: mirrors.rockylinux.org]
[root@it /] [ 2230.156648] e1000: enp0s3 NIC Link is Down
[root@it /] [ 2236.382350] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
^C
[root@it /]#
[root@it /]#
[root@it /]#
[root@it /]# sudo yum install epel-release -y
Rocky Linux 8 - AppStream
Rocky Linux 8 - AppStream
20% [=====]
339 B/s | 4.8 KB 08:14
1 309 kB/s | 2.8 MB 08:25 EDT

```

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”

```

IT_Security_1 (After_Installing.nginx) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
ROCKY LINUX 8 - Extras
nginx stable repo
Dependencies resolved.
=====
  Package           Architecture      Version       Repository
=====
Installing:
  epel-release     noarch          8-18.el8      extras
=====
Transaction Summary
=====
Install 1 Package
Total download size: 24 k
Installed size: 35 k
Downloading Packages:
epel-release-8-18.el8.noarch.rpm
=====
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing : 
  Installing : epel-release-8-18.el8.noarch
    Running scriptlet: epel-release-8-18.el8.noarch
Many EPEL packages require the CodeReady Builder (CRB) repository.
It is recommended that you run /usr/bin/crb enable to enable the CRB repository.
  Verifying   : epel-release-8-18.el8.noarch
=====
Installed:
  epel-release-8-18.el8.noarch
=====
Complete!
[root@it ~]#
[root@it ~]# sudo yum install ufw -y
Extra Packages for Enterprise Linux 8 - x86_64
48% [=====]
100 kB/s | 6.7 MB 00:19 ETX
4
[1/1] epel-release-8-18.el8.noarch
=====
[root@it ~]#

```

Now, we are enabling the package to start up automatically after booting up.
“systemctl enable ufw”

```

IT_Security_1 (After_Installing.nginx) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
ufw-0.35-14.el8.noarch.rpm
=====
Total
Extra Packages for Enterprise Linux 8 - x86_64
Importing GPG key 0x2FB6D6A1:
  Userid : "Fedora EPEL (8) <epel@fedoraproject.org>"
  Fingerprint: 94E2 79EB 8D8F 25B2 181B ADF1 21EA 45AB 2F86 D6A1
  From   : /etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-8
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing : 
  Installing : ufw-0.35-14.el8.noarch
    Running scriptlet: ufw-0.35-14.el8.noarch
  Verifying   : ufw-0.35-14.el8.noarch
=====
Installed:
  ufw-0.35-14.el8.noarch
=====
Complete!
[root@it ~]#
[root@it ~]# systemctl enable ufw
Created symlink /etc/systemd/system/basic.target.wants/ufw.service → /usr/lib/systemd/system/ufw.service.
[root@it ~]#
[root@it ~]# systemctl start ufw
[root@it ~]# ufw status
Status: active
To           Action      From
--           --         --
SSH          ALLOW      Anywhere
224.0.0.251 mDNS      ALLOW      Anywhere
SSH (v6)      ALLOW      Anywhere (v6)
ff02::fb mDNS      ALLOW      Anywhere (v6)

```

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

```
[root@it ~]#  
[root@it ~]# ufw allow 80/tcp  
Skipping adding existing rule  
Skipping adding existing rule (v6)  
[root@it ~]#  
[root@it ~]#  
[root@it ~]# ufw status numbered  
Status: active  
  
 To           Action    From  
 --           -----  
 [ 1] SSH          ALLOW IN  Anywhere  
 [ 2] 224.0.0.251 mDNS ALLOW IN  Anywhere  
 [ 3] 80/tcp        ALLOW IN  Anywhere  
 [ 4] 22            DENY IN   Anywhere  
 [ 5] 443           DENY IN   Anywhere  
 [ 6] 80            ALLOW IN  Anywhere  
 [ 7] SSH (v6)      ALLOW IN  Anywhere (v6)  
 [ 8] ff02::fb mDNS ALLOW IN  Anywhere (v6)  
 [ 9] 80/tcp (v6)   ALLOW IN  Anywhere (v6)  
[10] 22 (v6)       DENY IN   Anywhere (v6)  
[11] 443 (v6)      DENY IN   Anywhere (v6)  
[12] 80 (v6)       ALLOW IN  Anywhere (v6)
```

```
[root@it ~]#  
[root@it ~]# ufw logging on  
Logging enabled  
[root@it ~]# _
```

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.

```
[root@it conf.d]# ufw deny ssh
Rule updated
Rule updated (v6)
[root@it conf.d]#
[root@it conf.d]#
[root@it conf.d]# ufw deny https
Rule updated
Rule updated (v6)
[root@it conf.d]#
[root@it conf.d]#
```

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”

```
IT_Security_1 (After_Installing_nginx) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Tasks: 3 (limit: 24000)
Memory: 4.7M
CGroup: /system.slice/nginx.service
    └─1013 nginx: master process /usr/sbin/nginx -c /etc/nginx/nginx.conf
        ├─1014 nginx: worker process
        ├─1015 nginx: worker process

Jul 11 11:35:01 it.sicherheit systemd[1]: Starting nginx - high performance web server...
Jul 11 11:35:02 it.sicherheit systemd[1]: Started nginx - high performance web server.
[root@it ~]#
[root@it ~]#
[root@it ~]#
[root@it ~]#
[root@it ~]#
[root@it ~]# firewall-cmd --add-port=443/tcp --permanent
FirewallD is not running
[root@it ~]#
[root@it ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
    Active: inactive (dead)
      Docs: man:firewalld(1)
[root@it ~]#
[root@it ~]# systemctl start firewalld
[root@it ~]# 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 Thu 2024-07-11 12:07:19 CEST; 4s ago
    Docs: man:firewalld(1)
  Main PID: 1047 (firewalld)
    Tasks: 2 (limit: 24084)
   Memory: 37.2M
  CGroup: /system.slice/firewalld.service
          └─1047 /usr/libexec/platform-python -s /usr/sbin/firewalld --nofork --nopid

Jul 11 12:07:19 it.sicherheit systemd[1]: Starting firewalld - dynamic firewall daemon...
Jul 11 12:07:19 it.sicherheit systemd[1]: Started firewalld - dynamic firewall daemon.
Jul 11 12:07:19 it.sicherheit firewalld[1047]: WARNING: AllowZoneDrifting is enabled. This is considered an insecure configuration option. It will be removed in lines 1-13/13 (END)
[root@it ~]#

```

```
IT_Security_1 (After_Installing_nginx) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Tasks: 11 (limit: 24000)
JUL 11 11:35:02 IT.SICHERHEIT systemd[1]: Started nginx - high performance web server.
[root@it ~]#
[root@it ~]#
[root@it ~]#
[root@it ~]#
[root@it ~]#
[root@it ~]# firewall-cmd --add-port=443/tcp --permanent
FirewallD is not running
[root@it ~]#
[root@it ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
    Active: inactive (dead)
      Docs: man:firewalld(1)
[root@it ~]#
[root@it ~]# systemctl start firewalld
[root@it ~]# 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 Thu 2024-07-11 12:07:19 CEST; 4s ago
    Docs: man:firewalld(1)
  Main PID: 1047 (firewalld)
    Tasks: 2 (limit: 24084)
   Memory: 37.2M
  CGroup: /system.slice/firewalld.service
          └─1047 /usr/libexec/platform-python -s /usr/sbin/firewalld --nofork --nopid

Jul 11 12:07:19 it.sicherheit systemd[1]: Starting firewalld - dynamic firewall daemon...
Jul 11 12:07:19 it.sicherheit systemd[1]: Started firewalld - dynamic firewall daemon.
Jul 11 12:07:19 it.sicherheit firewalld[1047]: WARNING: AllowZoneDrifting is enabled. This is considered an insecure configuration option. It will be removed in lines 1-13/13 (END)

[root@it ~]#
[root@it ~]# firewall-cmd --add-port=443/tcp --permanent
success
[root@it ~]# firewall-cmd --add-port=80/tcp --permanent
success
[root@it ~]# firewall-cmd --add-port=22/tcp --permanent
success
[root@it ~]#

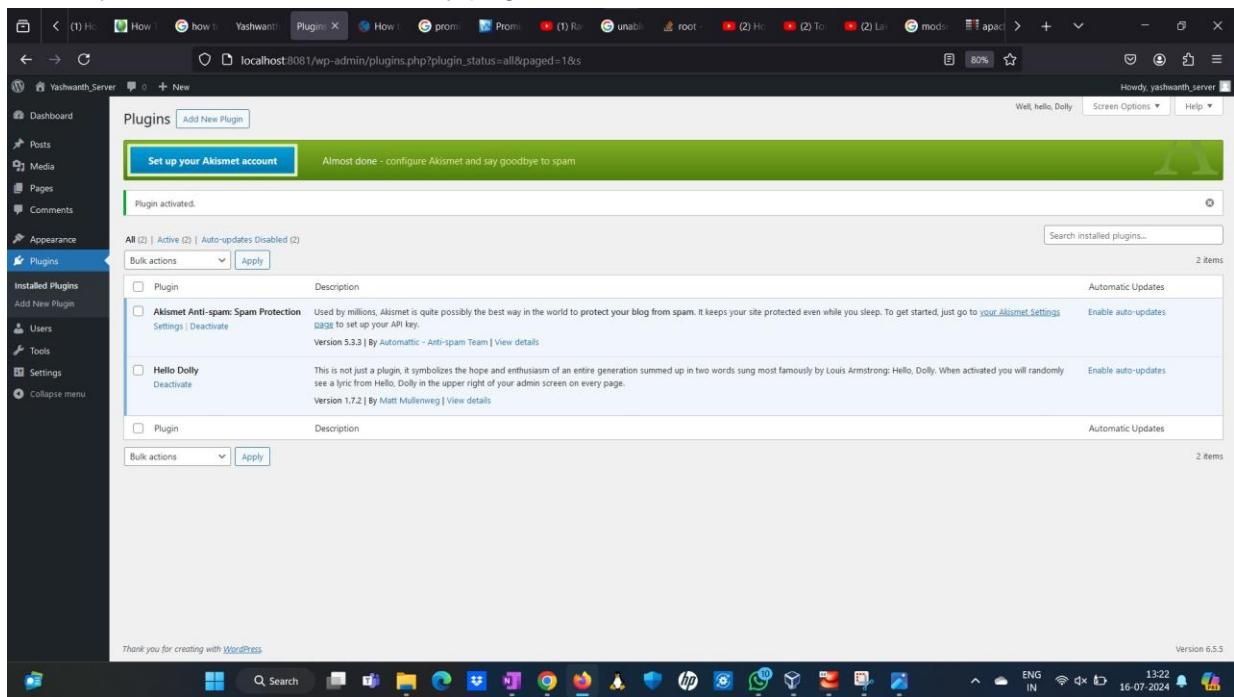
```

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  3 root root  4096 Jul 13 15:37 root
drwxr-xr-x  34 root root  960 Jul 14 13:58 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 12:48 var
drwxr-xr-x  3 root root  4096 Jul 12 21:55 wordpress
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  |
|--------------------------|--------------|----------------------------------------|------------|-------|
| <hr/>                    |              |                                        |            |       |
| Installing:              |              |                                        |            |       |
| fail2ban                 | noarch       | 1.0.2-3.el8                            | epel       | 21 kB |
| Installing dependencies: |              |                                        |            |       |
| esmtp                    | x86_64       | 1.2-15.el8                             | epel       | 57 kB |
| fail2ban-firewalld       | noarch       | 1.0.2-3.el8                            | epel       | 21 kB |
| fail2ban-selinux         | noarch       | 1.0.2-3.el8                            | epel       | 41 kB |
| fail2ban-sendmail        | noarch       | 1.0.2-3.el8                            | epel       | 23 kB |
| fail2ban-server          | noarch       | 1.0.2-3.el8                            | epel       | 79 kB |
| libesmtp                 | x86_64       | 1.0.6-18.el8                           | epel       | 79 kB |
| libblockfile             | x86_64       | 1.14-2.el8                             | baseos     | 31 kB |
| python3-pip              | noarch       | 9.0.3-24.el8.rocky.8                   | appstream  | 28 kB |
| python36                 | x86_64       | 3.6.8-39.module+e18.10.0+1592+61442852 | appstream  | 18 kB |



---



Transaction Summary



---



Install 10 Packages



Total download size: 700 k  
Installed size: 1.0 M  
Is this ok [y/N]: y



---



[IT_Security_1 (Working1) [Running] - Oracle VM VirtualBox]



```
File Machine View Input Devices Help
Running scriptlet: python36-3.6.8-39.module+e18.10.0+1592+61442852.x86_64
Installing : python3-pip-9.0.3-24.el8.rocky.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 : libblockfile-1.14-2.el8.x86_64
Running scriptlet: libblockfile-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.noarch
Verifying : python36-3.6.8-39.module+e18.10.0+1592+61442852.x86_64
Verifying : libblockfile-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-selinux-1.0.2-3.el8.noarch
libesmtp-1.0.6-18.el8.x86_64
python36-3.6.8-39.module+e18.10.0+1592+61442852.x86_64

Complete!
root@it ~%
```


```

```
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 /]#
[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]# _
```

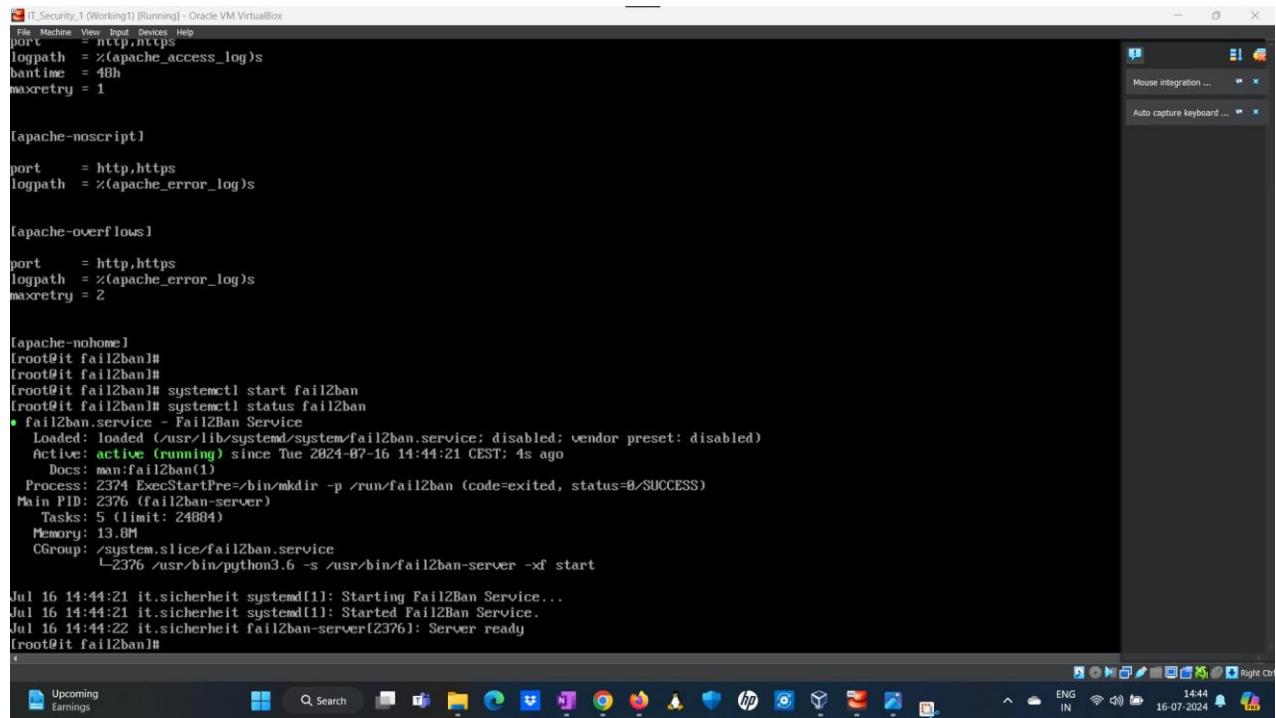
The explanation for each of the sentences are:

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”

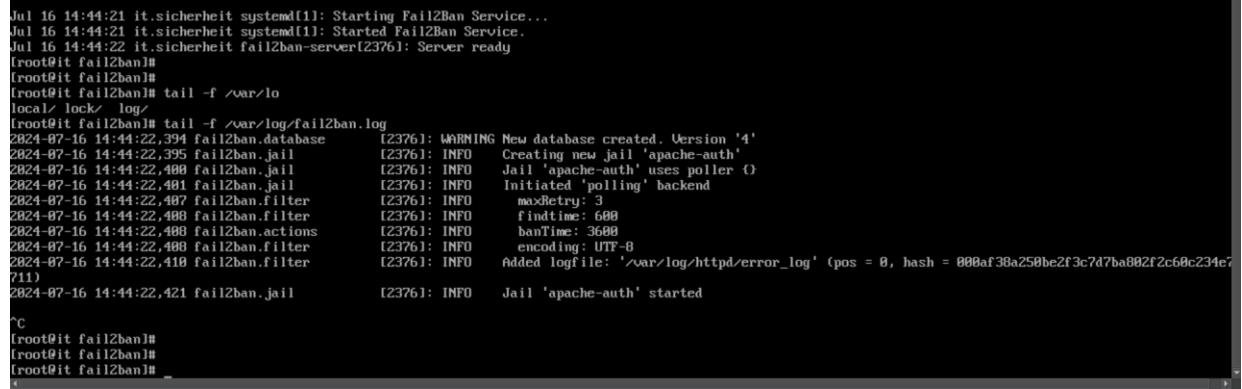
“systemctl status fail2ban”



```
[root@it_sicherheit ~]# systemctl status fail2ban
● fail2ban.service - Fail2Ban Service
   Loaded: loaded (/usr/lib/systemd/system/fail2ban.service; disabled; vendor preset: disabled)
   Active: active (running) since Tue 2024-07-16 14:44:21 CEST; 4s ago
     Docs: man:fail2ban(1)
   Process: 2374 ExecStartPre=/bin/mkdir -p /run/fail2ban (code=exited, status=0/SUCCESS)
   Main PID: 2376 (fail2ban-server)
      Tasks: 5 (limit: 24884)
     Memory: 13.0M
        CPU: 0.000 CPU(s) used
       CGroup: /system.slice/fail2ban.service
              └─2376 /usr/bin/python3.6 -s /usr/bin/fail2ban-server -x start

Jul 16 14:44:21 it.sicherheit systemd[1]: Starting Fail2Ban Service...
Jul 16 14:44:21 it.sicherheit systemd[1]: Started Fail2Ban Service.
Jul 16 14:44:22 it.sicherheit fail2ban-server[2376]: Server ready
[root@it_sicherheit ~]#
```

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



```
Jul 16 14:44:21 it.sicherheit systemd[1]: Starting Fail2Ban Service...
Jul 16 14:44:21 it.sicherheit systemd[1]: Started Fail2Ban Service.
Jul 16 14:44:22 it.sicherheit fail2ban-server[2376]: Server ready
[root@it_sicherheit ~]# tail -f /var/log/fail2ban.log
local/ lock/ log/
[root@it_sicherheit ~]# tail -f /var/log/fail2ban.log
2024-07-16 14:44:22,394 fail2ban.database          [2376]: WARNING New database created. Version '4'
2024-07-16 14:44:22,395 fail2ban.jail           [2376]: INFO  Creating new jail 'apache-auth'
2024-07-16 14:44:22,400 fail2ban.jail           [2376]: INFO  Jail 'apache-auth' uses poller O
2024-07-16 14:44:22,401 fail2ban.jail           [2376]: INFO  Initiated 'polling' backend
2024-07-16 14:44:22,407 fail2ban.filter         [2376]: INFO  maxRetry: 3
2024-07-16 14:44:22,408 fail2ban.filter         [2376]: INFO  findTime: 600
2024-07-16 14:44:22,408 fail2ban.actions        [2376]: INFO  banTime: 3600
2024-07-16 14:44:22,408 fail2ban.filter         [2376]: INFO  encoding: UTF-8
2024-07-16 14:44:22,410 fail2ban.filter         [2376]: INFO  Added logfile: '/var/log/httpd/error_log' (pos = 0, hash = 000af38a258be2f3c7d7ba802f2c680c234e7
711)
2024-07-16 14:44:22,421 fail2ban.jail          [2376]: INFO  Jail 'apache-auth' started
^C
[root@it_sicherheit ~]#
```

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.e18_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 ttym1
[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 4896
[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.

```
[root@it ~]# cd home/
[root@it home]# ls -l
total 4
drwx----- 2 John John 4096 Dec  4 2022 John
[root@it home]# cd John/
[root@it John]# ls -l
total 0
[root@it John]# ls -l
total 0
[root@it John]#
[root@it John]#
[root@it John]#
```

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

```
[root@it John]# userdel -r John
[sssd_cache1 sysdb_domain_cache_connect] (0x0010): DB version too old [0.23], expected [0.24] for domain implicit_files!
Higher version of database is expected!
In order to upgrade the database, you must run SSSD.
Removing cache files in /var/lib/sssd/db should fix the issue, but note that removing cache files will also remove all of your cached credentials.
Could not open available domains
[sssd_cache1 sysdb_domain_cache_connect] (0x0010): DB version too old [0.23], expected [0.24] for domain implicit_files!
Higher version of database is expected!
In order to upgrade the database, you must run SSSD.
Removing cache files in /var/lib/sssd/db should fix the issue, but note that removing cache files will also remove all of your cached credentials.
Could not open available domains
[root@it John]#
[root@it John]# cd ..
[root@it home]# cd ..
[root@it /]# cd ..
[root@it /]# userdel -r John
userdel: user 'John' does not exist
```

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/sssd/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/sssd/db/*”

After removing the files using the above command, the old files will be deleted.

2. Restart the entire system:

“sudo systemctl restart sssd”

```
[root@it /]#
[root@it /]# rm -rf /var/lib/sssd/db/*
[root@it /]# systemctl restart sssd
[root@it /]# _
```

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_Security_1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
[root@it John]# 
[root@it John]# cd ..
[root@it home]# cd ..
[root@it ~]# 
[root@it ~]# userdel -r John
userdel: user 'John' does not exist
[root@it ~]# 
[root@it ~]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:8:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:8:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:58:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/sbin/nologin
dbus:x:81:81:System message bus:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/sbin/nologin
tss:x:59:59:ccount used for TPM access:/dev/null:/sbin/nologin
polkitd:x:998:996:User for polkitd:/sbin/nologin
libstoragemgmt:x:997:993:daemon account for libstoragemgmt:/var/run/lsm:/sbin/nologin
sssd:x:996:992:User for sssd://sbin/nologin
cockpit-ws:x:995:991:User for cockpit web service:/nonexisting:/sbin/nologin
cockpit-wsinstance:x:994:990:User for cockpit-ws instances:/nonexisting:/sbin/nologin
chrony:x:993:989://var/lib/chrony:/sbin/nologin
avahi:x:70:70:avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
unbound:x:992:988:Unbound DNS resolver:/etc/unbound:/sbin/nologin
setroubleshoot:x:991:987:/var/lib/setroubleshoot:/sbin/nologin
nginx:x:998:986:nginx user:/var/cache/nginx:/sbin/nologin
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
mysql:x:27:27:MySQL Server:/var/lib/mysql:/sbin/nologin
[root@it ~]#
```

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-firewall-d-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-firewall-d-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=V8auUbpETIg>
- [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://localhost.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/alm2d9/cloudflare_free_fail2ban_other_security_hardening/

[26]

<https://linuxconfig.org/redhat-8-stop-start-firewall>