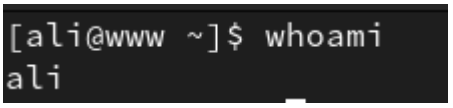
1. Hosting a website on Apache

- In this project . I will install, configure the `apache web server` .A static website will be hosted on the `apche server` on my machine.

Step 1:

First of all i would like to check and confirm the user information. The user who logged in in the system. . I will run the command

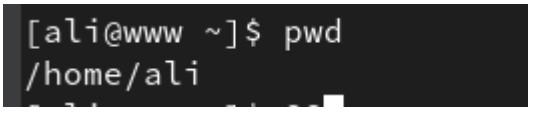
```
whoami
```



On this system the current user logged in has a user account with the name `ali`

It is a good practice to check the `current working directory` in the start . So run the command

```
pwd
```



- The current working directory is `/home/ali`

Moving on let me check the `hostname` of the system. The command is :

```
hostname
```

- My current hostname is `www.talha.com`. I want to change the hostname to `www.myproject.com`. For that i have to run the following command:

```
hostnamectl set-hostname www.myproject.com
```

- Check the information about your system
  1. Operating System Information
     - To check the `Linux Release version` of the OS and it's name we run the command

```
cat /etc/redhat-release
```

```
[ali@www ~]$ cat /etc/redhat-release
Rocky Linux release 9.3 (Blue Onyx)
```

We are using the OS `Rocky` .

- To check more details about the `OS` and its different `attributes` run the command :

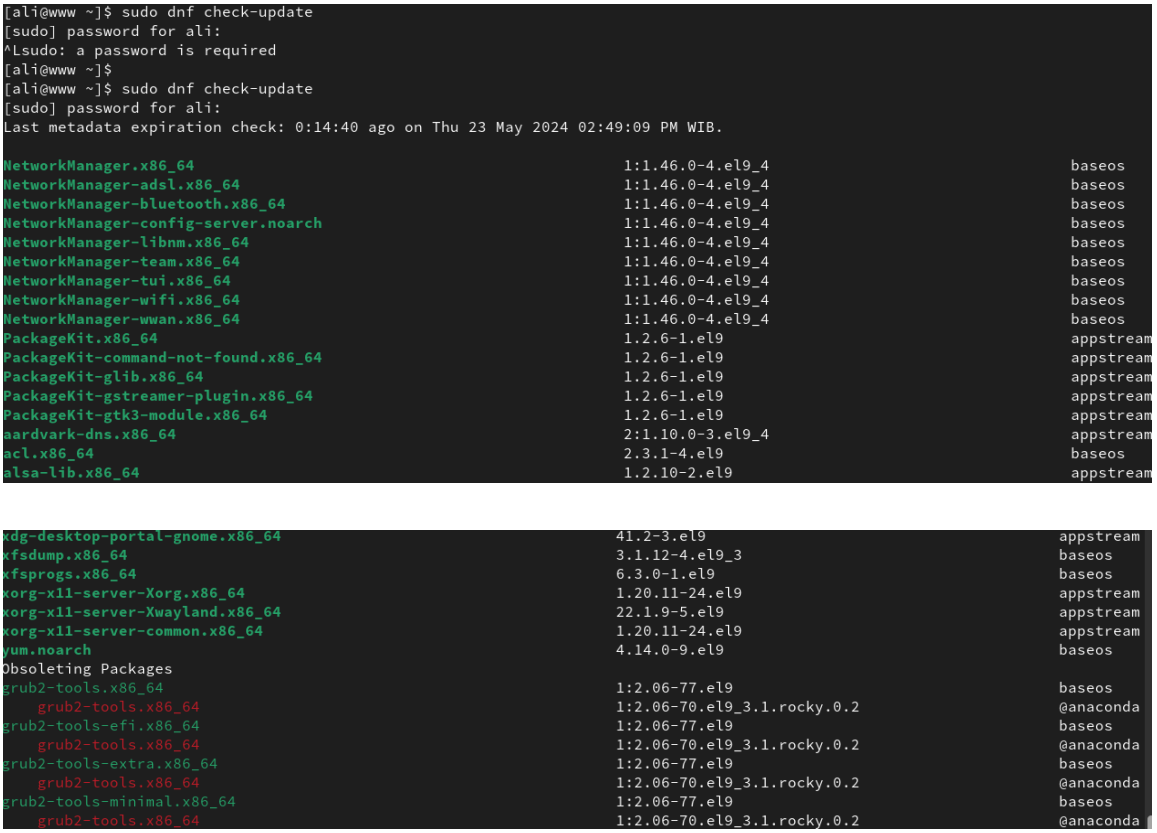
```
cat /etc/os-relesae
```

```
[ali@www ~]$ cat /etc/os-release
NAME="Rocky Linux"
VERSION="9.3 (Blue Onyx)"
ID="rocky"
ID_LIKE="rhel centos fedora"
VERSION_ID="9.3"
PLATFORM_ID="platform:el9"
PRETTY_NAME="Rocky Linux 9.3 (Blue Onyx)"
ANSI_COLOR="0;32"
LOGO="fedora-logo-icon"
CPE_NAME="cpe:/o:rocky:rocky:9::baseos"
HOME_URL="https://rockylinux.org/"
BUG_REPORT_URL="https://bugs.rockylinux.org/"
SUPPORT_END="2032-05-31"
ROCKY_SUPPORT_PRODUCT="Rocky-Linux-9"
ROCKY_SUPPORT_PRODUCT_VERSION="9.3"
REDHAT_SUPPORT_PRODUCT="Rocky Linux"
REDHAT_SUPPORT_PRODUCT_VERSION="9.3"
```

- Now I would like to check that my operating system is up to date or not. Is there any new updates available for it. To just check the `updates availabe` for the current `OS`

run the command. If your user do not have `root` privileges either add the user in `wheel` or use the `sudo` . Sudo stands for `super user do` :

```
sudo dnf check-update
```



```
[ali@www ~]$ sudo dnf check-update
[sudo] password for ali:
^Lsudo: a password is required
[ali@www ~]$
[ali@www ~]$ sudo dnf check-update
[sudo] password for ali:
Last metadata expiration check: 0:14:40 ago on Thu 23 May 2024 02:49:09 PM WIB.

NetworkManager.x86_64                                1:1.46.0-4.el9_4                    baseos
NetworkManager-adsl.x86_64                           1:1.46.0-4.el9_4                    baseos
NetworkManager-bluetooth.x86_64                      1:1.46.0-4.el9_4                    baseos
NetworkManager-config-server.noarch                  1:1.46.0-4.el9_4                    baseos
NetworkManager-libnm.x86_64                          1:1.46.0-4.el9_4                    baseos
NetworkManager-team.x86_64                           1:1.46.0-4.el9_4                    baseos
NetworkManager-tui.x86_64                            1:1.46.0-4.el9_4                    baseos
NetworkManager-wifi.x86_64                           1:1.46.0-4.el9_4                    baseos
NetworkManager-wwan.x86_64                           1:1.46.0-4.el9_4                    baseos
PackageKit.x86_64                                    1.2.6-1.el9                         appstream
PackageKit-command-not-found.x86_64                  1.2.6-1.el9                         appstream
PackageKit-glib.x86_64                               1.2.6-1.el9                         appstream
PackageKit-gstreamer-plugin.x86_64                   1.2.6-1.el9                         appstream
PackageKit-gtk3-module.x86_64                        1.2.6-1.el9                         appstream
aardvark-dns.x86_64                                  2:1.10.0-3.el9_4                    appstream
acl.x86_64                                           2.3.1-4.el9                         baseos
alsa-lib.x86_64                                      1.2.10-2.el9                        appstream
```



```
xdg-desktop-portal-gnome.x86_64                      41.2-3.el9                          appstream
xfsdump.x86_64                                       3.1.12-4.el9_3                      baseos
xfsprogs.x86_64                                      6.3.0-1.el9                         baseos
xorg-x11-server-Xorg.x86_64                          1.20.11-24.el9                      appstream
xorg-x11-server-Xwayland.x86_64                      22.1.9-5.el9                        appstream
xorg-x11-server-common.x86_64                        1.20.11-24.el9                      appstream
yum.noarch                                           4.14.0-9.el9                        baseos
Obsoleting Packages
grub2-tools.x86_64                                   1:2.06-77.el9                       baseos
    grub2-tools.x86_64                               1:2.06-70.el9_3.1.rocky.0.2         @anaconda
grub2-tools-efi.x86_64                               1:2.06-77.el9                       baseos
    grub2-tools.x86_64                               1:2.06-70.el9_3.1.rocky.0.2         @anaconda
grub2-tools-extra.x86_64                             1:2.06-77.el9                       baseos
    grub2-tools.x86_64                               1:2.06-70.el9_3.1.rocky.0.2         @anaconda
grub2-tools-minimal.x86_64                           1:2.06-77.el9                       baseos
    grub2-tools.x86_64                               1:2.06-70.el9_3.1.rocky.0.2         @anaconda
```

- I consider it important for my current project to `update` my `machine` before i deploy the `web server` . For that I will run the command

```
#To cehck the list of all available updates and size of to
sudo dnf update
```

```
 xorg-x11-server-common                 x86_64        1.20.11-24.el9                appstream        35 k
 yum                                    noarch        4.14.0-9.el9                  baseos           88 k
Installing dependencies:
 grub2-tools-efi                        x86_64        1:2.06-77.el9                 baseos          540 k
 grub2-tools-extra                      x86_64        1:2.06-77.el9                 baseos          840 k
 kernel-core                            x86_64        5.14.0-427.16.1.el9_4         baseos           21 M
 kernel-modules                         x86_64        5.14.0-427.16.1.el9_4         baseos           39 M
 kernel-modules-core                    x86_64        5.14.0-427.16.1.el9_4         baseos           33 M
 pipewire-jack-audio-connection-kit-libs x86_64       1.0.1-1.el9                   appstream       134 k
Installing weak dependencies:
 evolution-data-server-ui               x86_64        3.40.4-9.el9                  appstream        82 k

Transaction Summary
================================================================================================================================
Install    8 Packages
Upgrade  431 Packages

Total size: 1.2 G
Total download size: 1.2 G
Is this ok [y/N]: S
```

If i press `y` new updates will be installed , which are in a total size of `1.2 G`

- But i wanted to install them directly without prompting me a confirmation question. So, i will use the command

```
#Running this command will directly install thepackages wi
sudo dnf update -y
```

```
  yum-4.14.0-9.el9.noarch
Installed:
  evolution-data-server-ui-3.40.4-9.el9.x86_64          grub2-tools-efi-1:2.06-77.el9.x86_64
  grub2-tools-extra-1:2.06-77.el9.x86_64                kernel-5.14.0-427.16.1.el9_4.x86_64
  kernel-core-5.14.0-427.16.1.el9_4.x86_64              kernel-modules-5.14.0-427.16.1.el9_4.x86_64
  kernel-modules-core-5.14.0-427.16.1.el9_4.x86_64      pipewire-jack-audio-connection-kit-libs-1.0.1-1.el9.x86_64

Complete!
[ali@www ~]$
```

- Installation complete

> ***Note:*** It is consider a good practice to take a `snapshot` of the machine before any update and upgrade. (snapshot means backup)

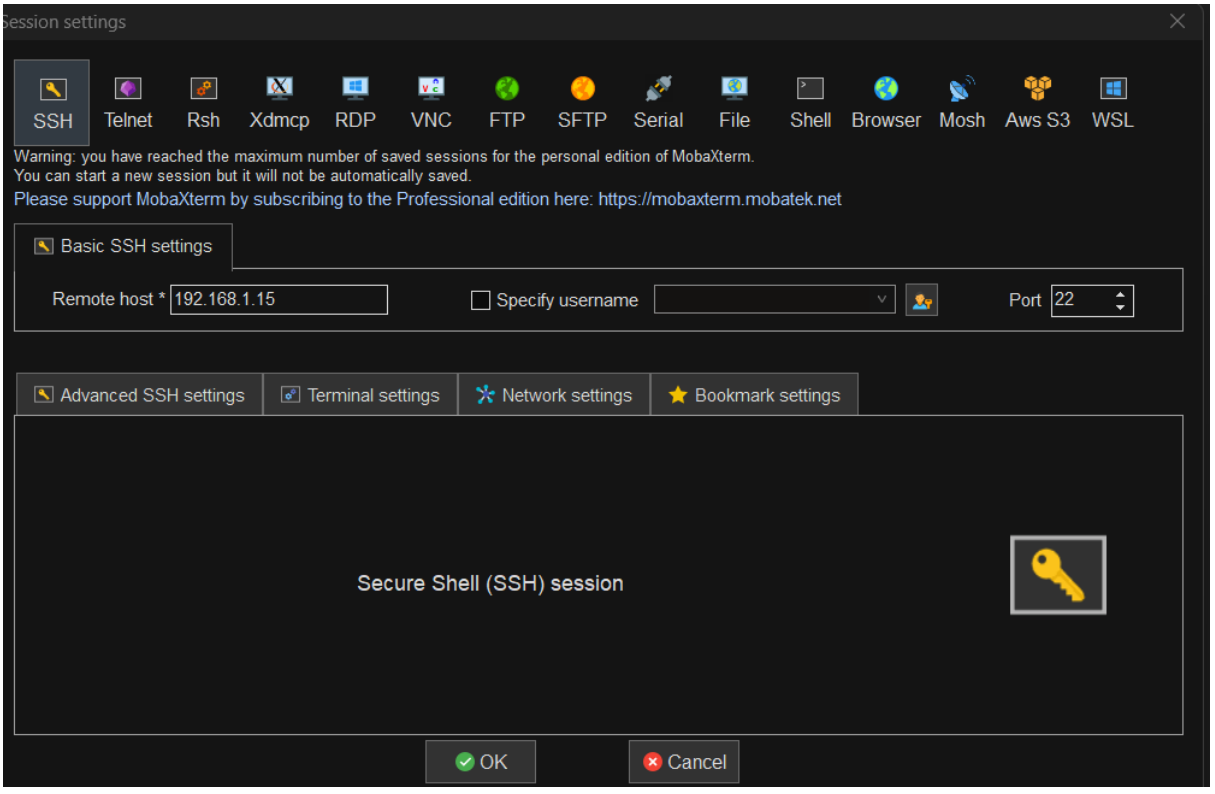
- I will do rest of my work in `mobaXterm` remote client. I have to create a `SSH session` `security shell session` . Secure Shell protocol use the `port 21`
- To establish a connection i must know the `ip address` of my machine.
- To check the `ip address` of my machine i can run the following commands :

```
# ALl of the following commands will work fine
ifconfig

ip a
```

```
ip add
```

```
[ali@www ~]$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.15  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::a00:27ff:fef9:b7c2  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:f9:b7:c2  txqueuelen 1000  (Ethernet)
        RX packets 1081010  bytes 1426158189 (1.3 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 300421  bytes 18162194 (17.3 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 42  bytes 4494 (4.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 42  bytes 4494 (4.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- The `ip address` of my `machine` is `192.168.1.15`. I will login using the `192.168.1.15` in the `mobaXterm`.



- This time i logged in as a `root user`

```
[root@www ~]# pwd
/root
[root@www ~]# whoami
root
[root@www ~]# hostname
www.myproject.com
[root@www ~]#
```

Installation of Apache server:

- The apache server package in Linux is called `httpd` .
- To utlize the package we have to `install and configure` the `httpd` so we can host our website.
- Before installation we have to check either the `apache` is already installed in our machine or not. We will run the command :

```
# Checking any Apache httpd package is installed or not
rpm -qa | grep httpd
```

```
[root@www ~]# rpm -qa | grep httpd
[root@www ~]#
```

The image shows that there is no `httpd` apache package is installed.

- We can check the available `httpd` package by running the command:

```
# To check the package availability and version number
dnf list httpd
```

```
[root@www ~]# dnf list httpd
Last metadata expiration check: 0:21:31 ago on Fri 24 May 2024 12:04:40 AM WIB.
Available Packages
httpd.x86_64                                    2.4.57-8.el9
```

- The output shows that there is one package available. But, we have to keep ourself on safe side. Sometimes, a software package has other dependencies without it the main

package does not work. We can check the other packages and dependencies by running the command

```
dnf list httpd*
```

```
[root@www ~]# dnf list httpd*
Last metadata expiration check: 0:21:47 ago on Fri 24 May 2024 12:04:40 AM WIB.
Available Packages
httpd.x86_64                                    2.4.57-8.el9
httpd-core.x86_64                               2.4.57-8.el9
httpd-devel.x86_64                              2.4.57-8.el9
httpd-filesystem.noarch                         2.4.57-8.el9
httpd-manual.noarch                             2.4.57-8.el9
httpd-tools.x86_64                              2.4.57-8.el9
[root@www ~]#
```

- To install the `apache httpd` and it's related dependencies we run the command :

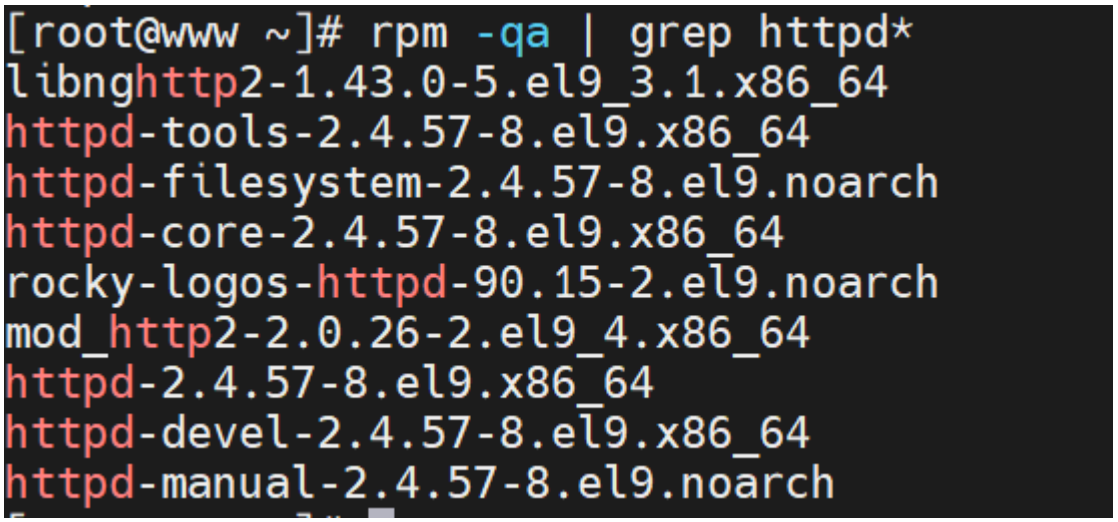
```
dnf install -y httpd*
```

```
[root@www ~]# dnf install -y httpd*
Last metadata expiration check: 0:30:02 ago on Fri 24 May 2024 12:04:40 AM WIB.
Dependencies resolved.
================================================================================
 Package                    Architecture          Version
================================================================================
Installing:
 httpd                      x86_64                2.4.57-8.el9
 httpd-core                 x86_64                2.4.57-8.el9
 httpd-devel                x86_64                2.4.57-8.el9
 httpd-filesystem           noarch                2.4.57-8.el9
 httpd-manual               noarch                2.4.57-8.el9
 httpd-tools                x86_64                2.4.57-8.el9
Installing dependencies:
 apr                        x86_64                1.7.0-12.el9_3
 apr-devel                  x86_64                1.7.0-12.el9_3
 apr-util                   x86_64                1.6.1-23.el9
 apr-util-bdb               x86_64                1.6.1-23.el9
 apr-util-devel             x86_64                1.6.1-23.el9
 cyrus-sasl                 x86_64                2.1.27-21.el9
 cyrus-sasl-devel           x86_64                2.1.27-21.el9
 expat-devel                x86_64                2.5.0-2.el9_4
 libdb-devel                x86_64                5.3.28-53.el9
 openldap-devel             x86_64                2.6.6-3.el9
 rocky-logos-httpd          noarch                90.15-2.el9
Installing weak dependencies:
 apr-util-openssl           x86_64                1.6.1-23.el9
 mod_http2                  x86_64                2.0.26-2.el9_4
 mod_lua                    x86_64                2.4.57-8.el9

Transaction Summary
================================================================================
Install  20 Packages
```

```
Installed:
  apr-1.7.0-12.el9_3.x86_64                 apr-devel-1.7.0-12.el9_3.x86_64          apr-util-1.6.1-23.el9.x86_64             apr-util-bdb-1.6.
  apr-util-devel-1.6.1-23.el9.x86_64        apr-util-openssl-1.6.1-23.el9.x86_64     cyrus-sasl-2.1.27-21.el9.x86_64          cyrus-sasl-devel-
  expat-devel-2.5.0-2.el9_4.x86_64          httpd-2.4.57-8.el9.x86_64                httpd-core-2.4.57-8.el9.x86_64           httpd-devel-2.4.5
  httpd-filesystem-2.4.57-8.el9.noarch      httpd-manual-2.4.57-8.el9.noarch         httpd-tools-2.4.57-8.el9.x86_64          libdb-devel-5.3.2
  mod_http2-2.0.26-2.el9_4.x86_64           mod_lua-2.4.57-8.el9.x86_64              openldap-devel-2.6.6-3.el9.x86_64        rocky-logos-httpd

Complete!
```

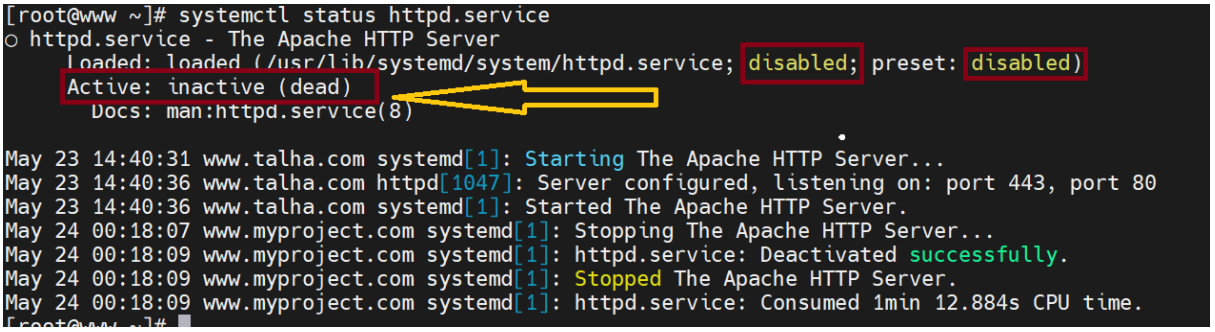
- After the installation is complete we can run the following command to check wither the package and its dependencies are installed completely:

```
rpm -qa | grep httpd*

OR

dnf install httpd*
```



- The image shows that the packages are installed on the system.
- Now , we will check the `status` of our `http server` by running the command:

```
systemctl status httpd.service
```

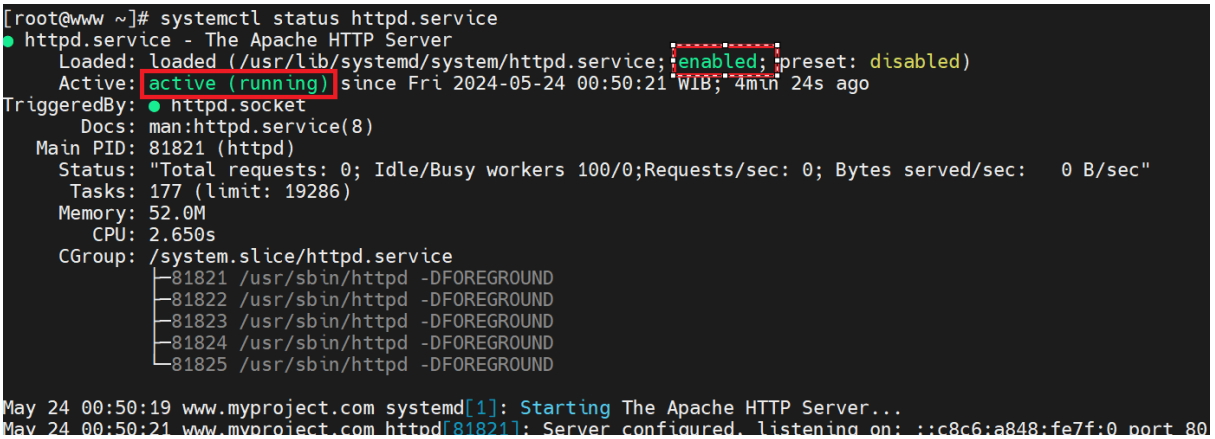


- To `activate` the `httpd server` we will run the command:

```
systemctl start httpd.service
```

- When you will run the `systemctl start httpd.servce` command and it does not show any error message. This shows that the service has started. To check it run the `status` command again:

```
systemctl status httpd.service
```



- The `httpd service` is up and running. To allow it to start as the system start we use the `enable` command.

```
systemctl enable httpd.service --now
```

- Similarly if we to `stop, restart, relaod and disable` a service we can do that using the following commands:

```
#To stop a service
systemctl stop httpd.service

#To restart a service
systemctl restart httpd.service

#To reload a service
systemctl restart httpd.service

#To disbale a service
systemctl diable httpd.service
```

- The website code after the installation of `apache server` will be placed in `cd /var/www/html`

Setting up firewall in machine and adding HTTP port to firewall:

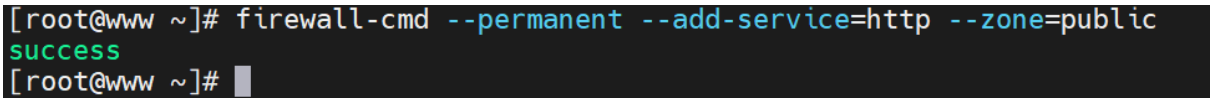
- The default port for the `http` is `80`. To make it running in our machine we have to add the `port 80` to our firewall.
- In Linux firewall is already installed and enabled. But there is a possibility that it might be turned of some reason. So, it is a good practice to always check the firewall. In Linux pre-installed firewall is called `firewalld`. Although there are some lighter version of firewall `e.g ufw` is available. But it is recommended to use the `firewalld`
- Check the current status of firewall by running the status command :

```
#These both comamnds will give same results
systemctl status firewalld

OR

 systemctl status firewalld.service
```

```
[root@www ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
     Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset: enabled)
     Active: active (running) since Thu 2024-05-23 14:40:21 WIB; 10h ago
       Docs: man:firewalld(1)
   Main PID: 895 (firewalld)
      Tasks: 4 (limit: 19286)
     Memory: 36.9M
        CPU: 7.483s
     CGroup: /system.slice/firewalld.service
             └─895 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --nopid

May 23 14:40:16 www.talha.com systemd[1]: Starting firewalld - dynamic firewall daemon...
May 23 14:40:21 www.talha.com systemd[1]: Started firewalld - dynamic firewall daemon.
[root@www ~]# systemctl status firewalld.service
● firewalld.service - firewalld - dynamic firewall daemon
     Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset: enabled)
     Active: active (running) since Thu 2024-05-23 14:40:21 WIB; 10h ago
       Docs: man:firewalld(1)
   Main PID: 895 (firewalld)
      Tasks: 4 (limit: 19286)
     Memory: 36.9M
        CPU: 7.483s
     CGroup: /system.slice/firewalld.service
             └─895 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --nopid

May 23 14:40:16 www.talha.com systemd[1]: Starting firewalld - dynamic firewall daemon...
May 23 14:40:21 www.talha.com systemd[1]: Started firewalld - dynamic firewall daemon.
```

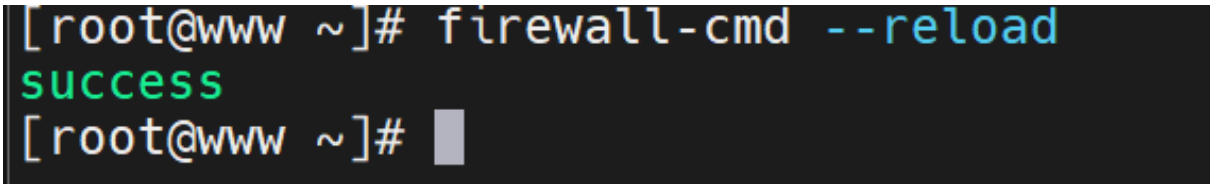
- As we have checked that the `firewalld` is up and running. Now we ill add the `http` to our firewall. The command is:

```
firewall-cmd --permanet --add-service=http --zone=public
```



- After successfully adding the `http` on the firewall. It is necessary to `reload` the firewall.

```
firewall-cmd --reload
```



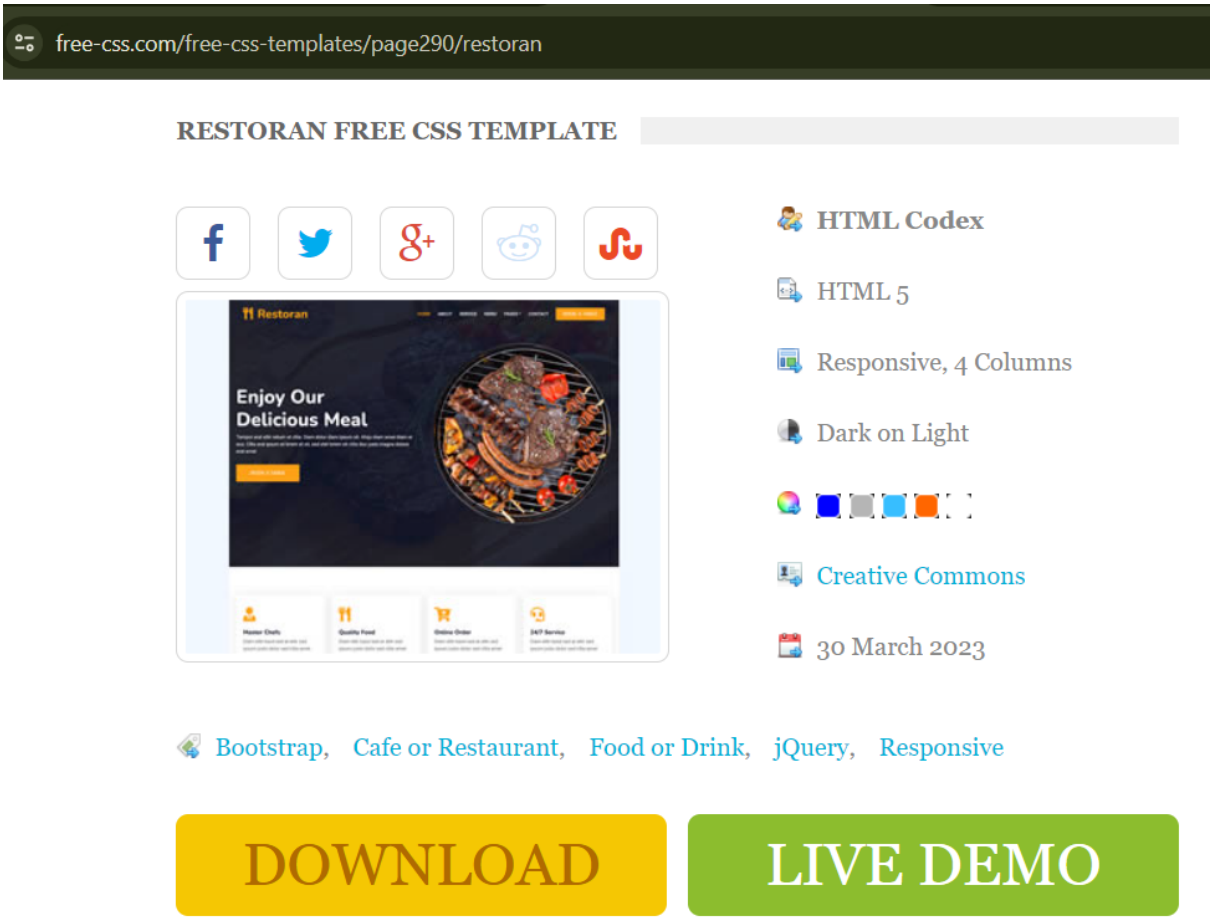
- To check the list of all the services added to firewall we run the following command :

```
firewall-cmd --list-all

# or to be specific

firewallcmd --list-services
```

```
[root@www ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpv6-client http https ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[root@www ~]# firewall-cmd --list-services
cockpit dhcpv6-client http https ssh
```

Download a Sample Website:

- For our project and practice we will dowload the code of the website from a website. Which we ill deploy on our `apache server.
- Some website offers free code of `websites` we can download the code from those website as per our choice.
- Let’s download a code from the website `https://www.free-css.com/` . I have decided to download the code of the following website:



The download `url` is `https://www.free-css.com/assets/files/free-css-templates/download/page290/restoran.zip`

- To download the `template` direct in my `Linux Machine` i will use the `wget` command. I will run the following command to download the `zip` file of the code:

```
wget https://www.free-css.com/assets/files/free-css-templates
```

```
[root@www ~]# wget https://www.free-css.com/assets/files/free-css-templates/download/page290/restoran.zip
--2024-05-25 16:37:47--  https://www.free-css.com/assets/files/free-css-templates/download/page290/restoran.zip
Resolving www.free-css.com (www.free-css.com)... 217.160.0.242, 2001:8d8:100f:f000::28f
Connecting to www.free-css.com (www.free-css.com)|217.160.0.242|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1552247 (1.5M) [application/zip]
Saving to: 'restoran.zip'

restoran.zip                    100%[=========================================================>]   1.48M  78.9KB/s    in 20s

2024-05-25 16:38:12 (74.5 KB/s) - 'restoran.zip' saved [1552247/1552247]

[root@www ~]# ls -ltr
total 1700
-rw-r--r--. 1 root root 1552247 Jan  7  2022 restoran.zip
```

- The code of the website is downloaded as file name `restoran.zip`

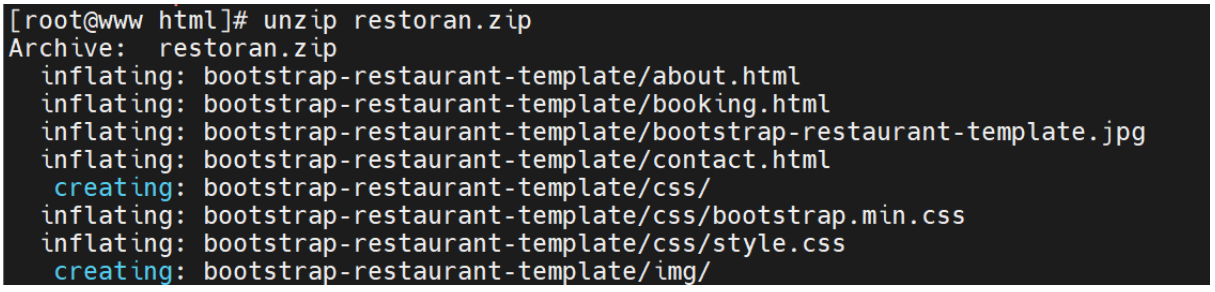
Moving to code to the `apache` file systema and unzipping:

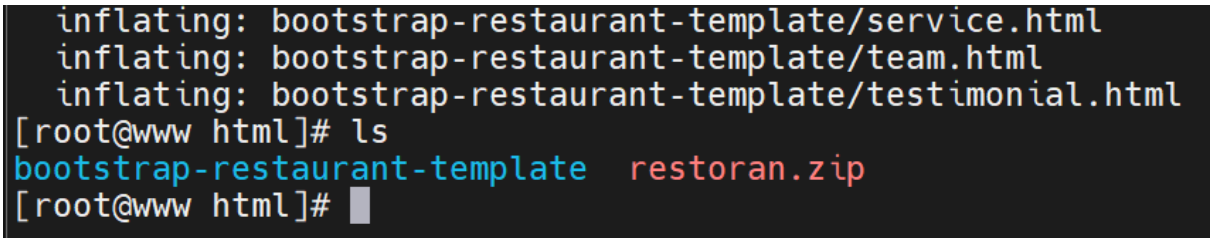
- To utilize the code for the `apache server` and host the website. We have to move the `downloaded file` from our `/root/home` to the directory `var/www/html`. We can use the `mv` to completely move the file. Also, we can use the `cp` copy command to keep the file on the `home directory` as well as in the desired directory. I will go for the `cp` command:

```
cp restoran.zip /var/www/html
```

- After copying and checking now i will `unzip` the code . First go to the directory `/var/www/html` . Now unzip the code using the command `unzip restoran.zip`

```
unzip restoran.zip
```





- We can remove the `zip file` which is named as `restoran.zip` . Run the command :

```
rm -rf restoran.zip
```

```
bootstrap-restaurant-template  restoran.zip
[root@www html]# rm -rf restoran.zip
```

- Now, we have to copy all the files present in the unzipped folder which is `cd /var/www/html/bootstrap-restaurant-template/`.
- When in folder` `/var/www/html/bootstrap-restaurant-template/` run the following copy command to copy all the content of the folder to its parent directory. Which in current stage will be `/var/ww/html` . The command is :

```
cp -r * ..
```

```
[root@www html]# cd bootstrap-restaurant-template/
[root@www bootstrap-restaurant-template]# cp -r * ..
[root@www bootstrap-restaurant-template]# cd ..
[root@www html]# ls
about.html                    bootstrap-restaurant-template.jpg  img         lib          READ-ME.txt   team.html
booking.html                  contact.html                       index.html  LICENSE.txt  scss          testimonial.html
bootstrap-restaurant-template  css                                js          menu.html    service.html
[root@www html]# cd
bootstrap-restaurant-template/ img/                          lib/
```

- We can now remove the directory `/bootstrap-restaurant-template/`. Run the command :

```
rm -rf /bootstrap-restaurant-template/
```

```
[root@www html]# ls
about.html                    bootstrap-restaurant-template.jpg  img         lib          READ-ME.txt   team.html
booking.html                  contact.html                       index.html  LICENSE.txt  scss          testimonial.html
bootstrap-restaurant-template  css                                js          menu.html    service.html
[root@www html]# rm -rf bootstrap-restaurant-template
[root@www html]# ls
about.html    bootstrap-restaurant-template.jpg  css  index.html  lib          menu.html    scss          team.html
booking.html  contact.html                       img  js          LICENSE.txt  READ-ME.txt  service.html  testimonial.html
[root@www html]#
```

> It is a good practice to restart the `http service` after deploying your website. Run the command `systemctl restart httpd.service`

- The `website` is `up and hosted` we can check it on our terminal first using the command

```
curl localhost
```

```
#OR use your IP address
curl 192.168.1.15
```

- We can open the `web browser` e.g `chrome or firefox` and use our `ip address` which is `192.168.1.15` to check the website is running.