Hosting Multiple Websites on Apache Using Virtual Hosting

- In my last project, I hosted a single website on the Apache server. For details, please refer to the previous project documentation.
- In this current project, I will host two websites on a single machine using the Apache server.
- I am using the same OS on which I hosted my last project.

Purpose and Outcome

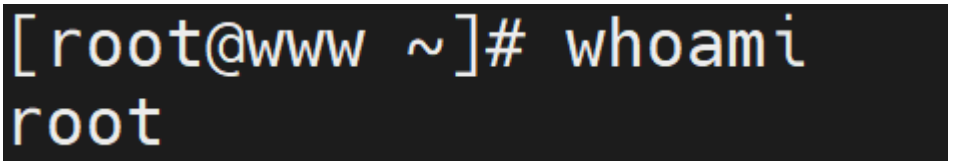
The purpose of this project is to demonstrate the ability to configure an Apache server to host multiple websites on a single machine. This is a valuable skill in web development and server administration, allowing for efficient resource utilization and management. By the end of this project, you will have two fully functional websites hosted securely using HTTPS on a single Apache server instance.

Step 1:

Checking the System Configuration and Settings:

1. The user logged in as `root`

```
whoami
```



2. The `current working dirrectory` is `/root`

```
pwd
```

```
[root@www ~]# pwd
/root
```

3. The `hostname of the machine` is `www.myproject.com` .

```
hostname
```

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

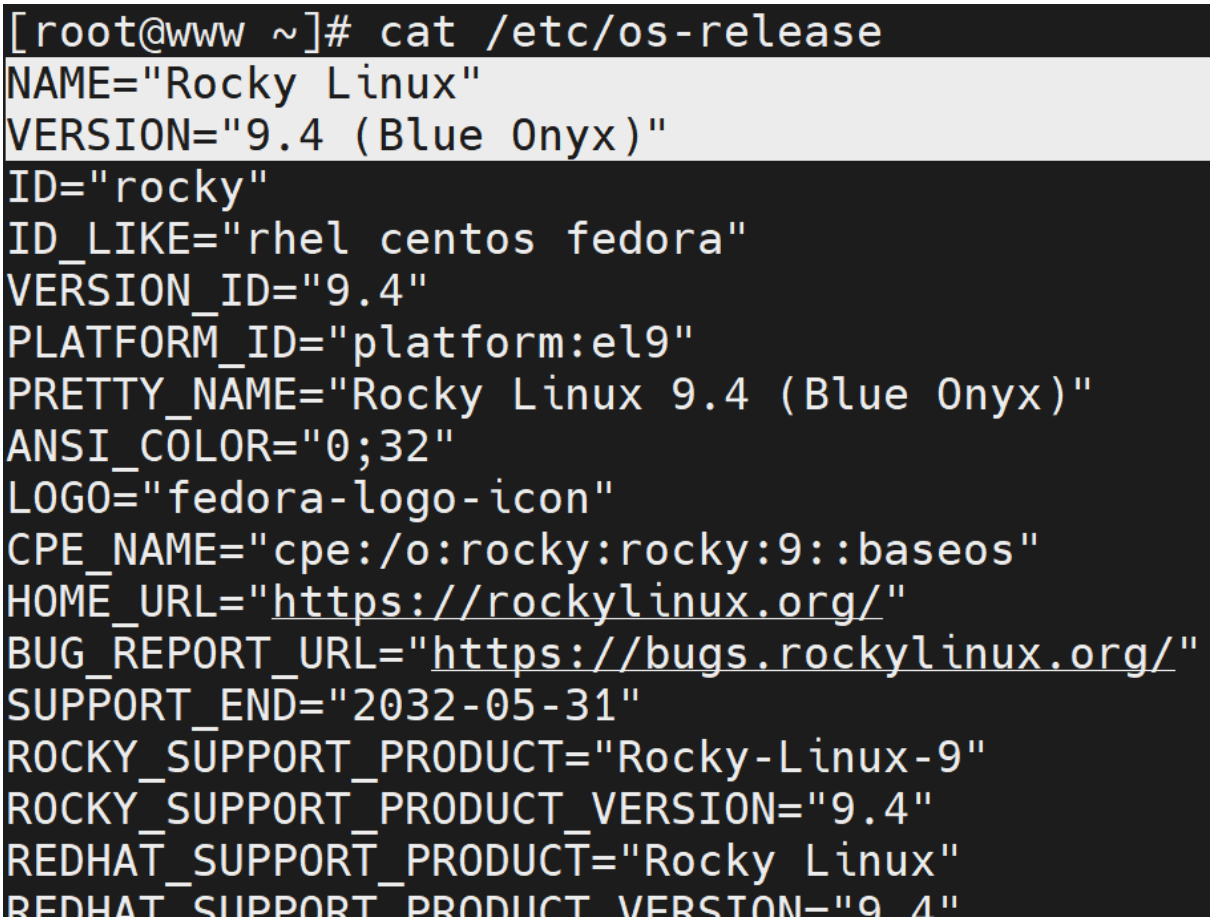
- We will change the `hostname` to `www.myproject2.com`

```
hostnamectl set-hostname www.myproject2.com
```

```
[root@www ~]# hostname
www.myproject2.com
```

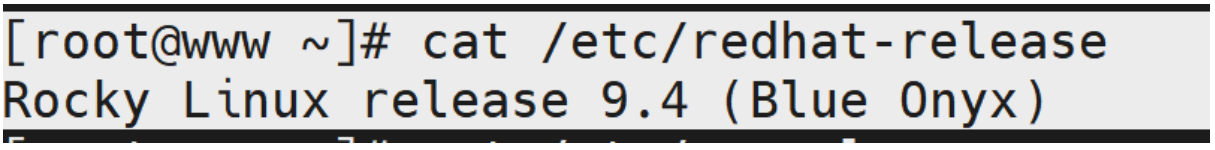
4. The the information about the `current OS relase and version` can be viewed using the command :

```
cat /etc/os-release
```



- To check just the `Red Hat release name and version`:

```
cat /etc/os-release
```



- The `kernal information` , `pc architecture` and other information can be viewed using :

```
hostnamectl
```

```
[root@www ~]# hostnamectl
 Static hostname: www.myproject2.com
       Icon name: computer-vm
         Chassis: vm
      Machine ID: 37507b7b90d24[redacted]
         Boot ID: 63745ad30f6f4[redacted]
  Virtualization: oracle
Operating System: Rocky Linux 9.4 (Blue Onyx)
     CPE OS Name: cpe:/o:rocky:rocky:9::baseos
          Kernel: Linux 5.14.0-427.16.1.el9_4.x86_64
    Architecture: x86-64
 Hardware Vendor: innotek GmbH
  Hardware Model: VirtualBox
Firmware Version: VirtualBox
```

5. As a good practice `checking any update` for the `current OS`

```
dnf check-update
```

```
[root@www ~]# dnf check-update
Rocky Linux 9 - BaseOS                                  3.3 kB/s | 4.1 kB     00:01
Rocky Linux 9 - AppStream                               3.1 kB/s | 4.5 kB     00:01
Rocky Linux 9 - Extras                                  2.3 kB/s | 2.9 kB     00:01
```

- Minor updates are available. Apply the updates

```
dnf update -y
```

```
[root@www ~]# dnf update -y
Last metadata expiration check: 0:00:16 ago on Mon 27 May 2024 10:16:21 AM WIB.
Dependencies resolved.
Nothing to do.
Complete!
```

6. Checking the `ip address` of the machine:

```
ip add
```

```
[root@www ~]# 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:f9:b7:c2 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.15/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s3
       valid_lft 84349sec preferred_lft 84349sec
    inet6 fe80::a00:27ff:fef9:b7c2/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

- To narrow down the ip address :

```
ip a | grep enp0s3
```

```
[root@www ~]# ip a | grep enp0s3
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
qlen 1000
    inet 192.168.1.15/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s3
```

- The `ip address` of the current machine is `192.168.1.15`

Step 2:

Checking The Necessary `packages` . Downloading new `packages` . Inspecting the `directories`

1. Now, we need some packages to deploy `two websites servers` on our single machine.

Package 1 , Apache HTTP:

1. We need to download and install the `apache server` `httpd` on our machine.
2. Checking the `repositories` if it is already installed or not:

```
rpm -qa | grep httpd
```

```
[root@www ~]# rpm -qa | grep httpd
httpd-tools-2.4.57-8.el9.x86_64
httpd-filesystem-2.4.57-8.el9.noarch
httpd-core-2.4.57-8.el9.x86_64
rocky-logos-httpd-90.15-2.el9.noarch
httpd-2.4.57-8.el9.x86_64
httpd-devel-2.4.57-8.el9.x86_64
httpd-manual-2.4.57-8.el9.noarch
```

- Turned out `httpd` already installed in my machine. Let's check for any update available for the package

```
dnf update httpd*
```

```
[root@www ~]# dnf update httpd*
Last metadata expiration check: 0:09:54 ago on Mon 27 May 2024 10:16:21 AM WIB.
Dependencies resolved.
Nothing to do.
Complete!
[root@www ~]#
```

- My `apache server` package is Updated to current version.

4. I can view my the directory of `httpd` at `cd /var/www/html`

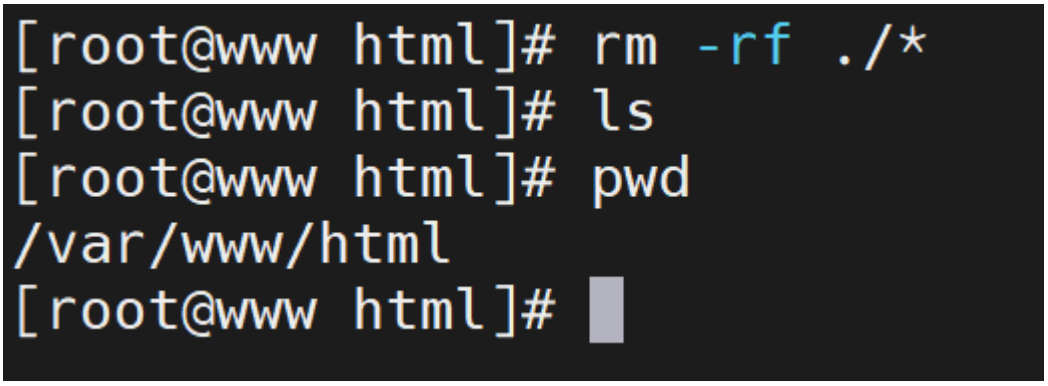
```
cd /var/www/html/

ls -ltr
```

```
[root@www html]# cd /var/www/html/
[root@www html]# ls -ltr
total 292
drwxr-xr-x. 2 root root    48 May 25 16:53 css
-rw-r--r--. 1 root root 14529 May 25 16:53 contact.html
-rw-r--r--. 1 root root 83545 May 25 16:53 bootstrap-restaurant-template.jpg
-rw-r--r--. 1 root root 14703 May 25 16:53 booking.html
-rw-r--r--. 1 root root 18013 May 25 16:53 about.html
drwxr-xr-x. 3 root root    45 May 25 16:53 scss
-rw-r--r--. 1 root root   549 May 25 16:53 READ-ME.txt
-rw-r--r--. 1 root root 35675 May 25 16:53 menu.html
-rw-r--r--. 1 root root  1456 May 25 16:53 LICENSE.txt
drwxr-xr-x. 9 root root   120 May 25 16:53 lib
drwxr-xr-x. 2 root root    21 May 25 16:53 js
-rw-r--r--. 1 root root 54922 May 25 16:53 index.html
drwxr-xr-x. 2 root root  4096 May 25 16:53 img
-rw-r--r--. 1 root root 13622 May 25 16:53 testimonial.html
-rw-r--r--. 1 root root 18668 May 25 16:53 team.html
-rw-r--r--. 1 root root 14869 May 25 16:53 service.html
[root@www html]#
```

5. Files from my `previous project website` is still here. I want to deploy new websites . So lets remove it. I am currently in directory `/var/www/html` . Run the command :

```
rm -rf ./*
```



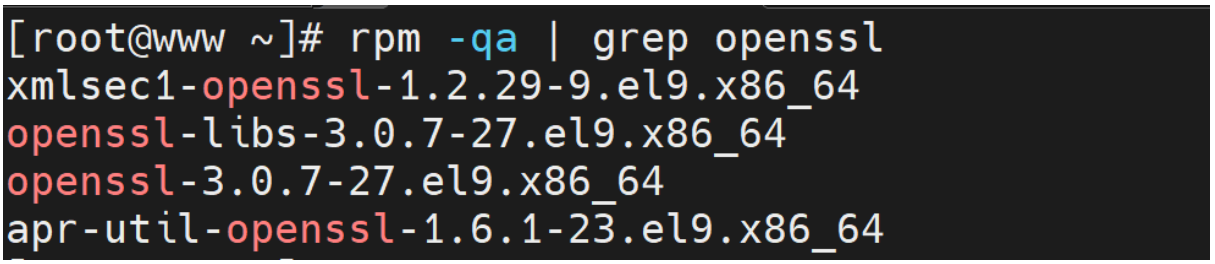
- All the files has been deleted from the directory.

Package 2 and 3 , `openssl` and `mod_ssl` :

1. For this project we are deploying two website on with `enable secure communication` in short `https` .
2. We need `SSL/TLS certificate` to use `https`

3. To achieve this we need to install two packages `openssl` and `mod_ssl`. They form `secure connection` between `web server` and `user browser`.
4. They both packages work together for `encryption` and `decryption`
5. We will create `certificate` using the package `openssl` while `mod_ssl` is a supportive package for `openssl`
6. First, check either the both packages are installed in our current system or not. Checking the `openssl` first and if installed updating it.

```
rpm -qa | grep openssl
```



-

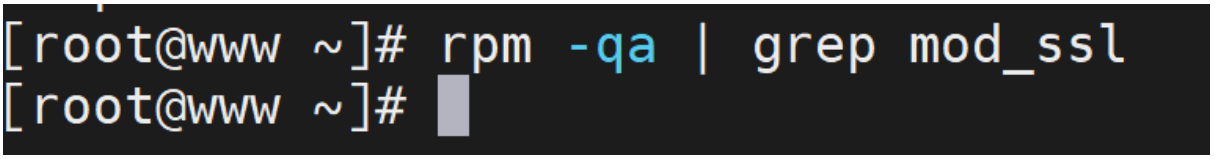
- The package `openssl` is already installed. Let's check updates:

```
    dnf update openssl
```

- The package is updated already.

6. Checking the package `mod_ssl` . Installing if needed .

```
rpm -qa | grep mod_ssl
```



- The package is not installed on my machine. Let's install it:

```
dnf install mod_ssl
```

```
[root@www ~]# dnf install -y mod_ssl
Last metadata expiration check: 0:28:32 ago on Mon 27 May 2024 10:16:21 AM WIB.
Dependencies resolved.
================================================================================
 Package            Architecture      Version                   Repository            Size
================================================================================
Installing:
 mod_ssl            x86_64            1:2.4.57-8.el9            appstream            109 k

Transaction Summary
================================================================================
Install  1 Package

Total download size: 109 k
Installed size: 268 k
Downloading Packages:
mod_ssl-2.4.57-8.el9.x86_64.rpm                 210 kB/s | 109 kB     00:00
--------------------------------------------------------------------------------
Total                                            57 kB/s | 109 kB     00:01
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Installing       : mod_ssl-1:2.4.57-8.el9.x86_64                          1/1
  Running scriptlet: mod_ssl-1:2.4.57-8.el9.x86_64                          1/1
  Verifying        : mod_ssl-1:2.4.57-8.el9.x86_64                          1/1

Installed:
  mod_ssl-1:2.4.57-8.el9.x86_64

Complete!
```

Step 3:

Creating `self signed Certificate` Using `OpenSSL`

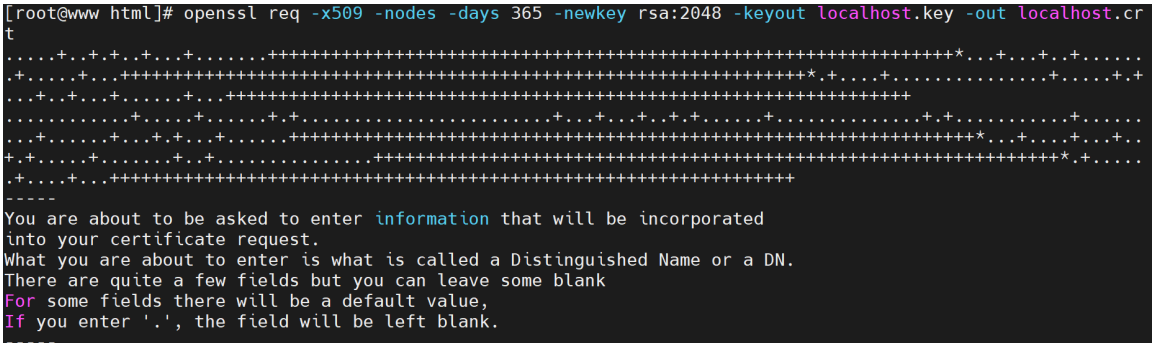
1. We have installed the package `openssl` and `mod_ssl`. Now , we will create `SSL/TLS` certificate.
2. Here we will call it `SAN` certificate `(Subject Alternative Names)` instead of `SSL/TLS` certificate `(Secure Sockets Layer/Transport Layer Security)` . Basically both are same , but with one difference in nature:
    a. SSL/TLS : This certificate is used to deploy one `single domain` on `one certificate only`
    b. SAN : This certificate is used to deploy `multiple domain names` in a `single certificate`

3. To `install and configure` the certificate use the command :

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout l

- openssl req = Requesting for Certificate
- -x509   =  Creates a self-signed certificate instead of a c
- `-nodes`: Omits private key encryption. **Note:** This is n
- `-days 365`: Sets the certificate validity period to 365 da
- `-newkey rsa:2048`: Generates a new RSA private key with a
- `-keyout localhost.key`: Specifies the filename for the pri
- `-out localhost.crt`: Specifies the filename for the genera
```

- Explanation of the command is :
  - openssl req = Requesting for Certificate
  - -x509 = Creates a self-signed certificate instead of a certificate signing request (CSR)
  - `-nodes`: Omits private key encryption. **Note:** This is not recommended for production environments as it stores the private key unencrypted.
  - `-days 365`: Sets the certificate validity period to 365 days (one year).
  - `-newkey rsa:2048`: Generates a new RSA private key with a size of 2048 bits. This key will be used for encryption and decryption by your server.
  - `-keyout localhost.key`: Specifies the filename for the private key (localhost.key in this case)
  - `-out localhost.crt`: Specifies the filename for the generated certificate (localhost.crt in this case)



```
-----
Country Name (2 letter code) [XX]:id
State or Province Name (full name) []:bali
Locality Name (eg, city) [Default City]:ubud
Organization Name (eg, company) [Default Company Ltd]:talha corp
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:www.myproject2.com
Email Address []:[redacted]@gmail.com
```

- You will be prompt for some information which you have to fill to obtain certificate.

> If you have put wrong information and want to change it itis not recommended. Best practice is to create a new certificate.

Step 4

Copying the certificate generated files to proper destination

1. While we were `generating our `SAN certifictae`. We were in the directory name `/var/www/html` .

```
[root@www html]# pwd
/var/www/html
[root@www html]# ls -ltr
total 8
-rw-------. 1 root root 1704 May 27 11:14 localhost.key
-rw-r--r--. 1 root root 1415 May 27 11:16 localhost.crt
[root@www html]#
```

2. We will move the two generated files named `localhost.key` and `localhost.crt` to the directory `/root`. It is good to have a copy of keys.

```
mv localhost.key localhost.crt /root
```

```
[root@www ~]# ls -ltr
total 1708
-rw-r--r--. 1 root root 1552247 Jan  7  2022 restoran.zip
-rw-------. 1 root root     827 May 17 02:27 anaconda-ks.cfg
-rwxr-xr-x. 1 root root     752 May 17 14:36 demo
-rwxr-xr-x. 1 root root     615 May 17 14:42 step1
-rwxr-xr-x. 1 root root    1556 May 17 14:47 time
-rw-r--r--. 1 root root  163566 May 17 17:52 proxy.zip
-rw-------. 1 root root    1704 May 17 19:07 localhost.key
-rw-r--r--. 1 root root    1497 May 17 19:09 localhost.crt
drwxr-xr-x. 3 root root      61 May 17 20:24 website_code
drwxr-xr-x. 2 root root       6 May 18 14:40 directory_name
-rw-------. 1 root root    1704 May 27 11:14 localhost.key
-rw-r--r--. 1 root root    1415 May 27 11:16 localhost.crt
[root@www ~]# pwd
/root
```

3. We have to move the generated files to specific directories:
   a. `localhost.key` copy to `/etc/pki/tls/private/`
   b. `localhost.cert` copy to `/etc/pki/tls/cert/`

4. Before we move them we have to do two things:
   a. The directories are presnt
   b. We have to check the file named `ssl.conf` present in directory `cat /etc/httpd/conf.d/ssl.conf` that the path were we copying the file `localhost.key` and `localhost.cert` are mention exactly the same way `` `/etc/pki/tls/private/` and `/etc/pki/tls/cert/`

```
SSLCertificateFile /etc/pki/tls/certs/localhost.crt

#   Server Private Key:
#   If the key is not combined with the certificate, use this
#   directive to point at the key file.  Keep in mind that if
#   you've both a RSA and a DSA private key you can configure
#   both in parallel (to also allow the use of DSA ciphers, etc.)
#   ECC keys, when in use, can also be configured in parallel
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
```

5. Now we will move the file name `localhost.key` to the directory `/etc/pki/tls/private/`

```
[root@www ~]# cp localhost.key /etc/pki/tls/private/
[root@www ~]# cd ^C
[root@www ~]# cd /etc/pki/tls/private/
[root@www private]# ls
localhost.key
[root@www private]#
```

6. The filename `localhost.crt` will be moved to the directory `/etc/pki/tls/cert/`

```
[root@www ~]# cp localhost.crt /etc/pki/tls/c
cert.pem         certs/           ct_log_list.cnf
[root@www ~]# cp localhost.crt /etc/pki/tls/certs/
[root@www ~]# cd /etc/pki/tls/certs/
[root@www certs]# ls -ltr
total 4
-rw-r--r--. 1 root root 1415 May 27 11:43 localhost.crt
[root@www certs]#
```

7. Now to verify that the syantax of the configuration files is okay or not. Run the command:

```
httpd -t
```

```
[root@www ~]# httpd -t
Syntax OK
[root@www ~]#
```

- Since its shows the message `Sytanx OK` so everything is working perfectly.
- Sometime if there is any error, read the display message there could be a minor issue, like spelling mistake, spelling mismatch, wrong server name , or any colon or semi-colon error.

Step 5:

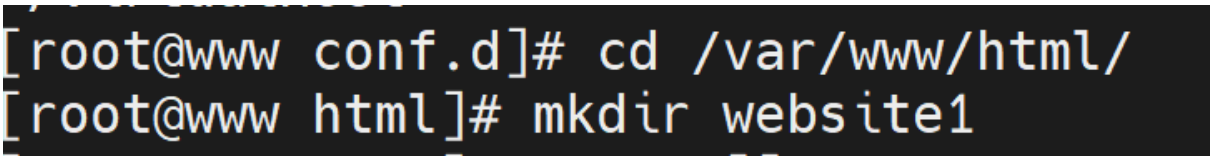
1. As we know we want to deploy two websites on our machine. For this reason we have to create two directories in the `/var/www/html
2. We will create two directories with name `website1` and `website2` . Where we will place our `website code.`.
3. Let's create the directory name `website1` in `/var/www/html/` and download the website code using `wget` `https://www.free-css.com/assets/files/free-css-templates/download/page284/mical.zip` .Then unzip it and remove the unnessary zipped file afterward.

```
cd /var/www/html/

mkdir website1

cd website1

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



- The website `zip` file is downloaded:

```
[root@www website1]# ls
'Mical Free Website Template - Free-CSS.com.zip'
[root@www website1]#
```

- Unzip the file using the command:

```
unzip Mical Free Website Template - Free-CSS.com.zip
```

```
[root@www website1]# unzip Mical\ Free\ Website\ Template\ -\ Free-CSS.com.zip
Archive:  Mical Free Website Template - Free-CSS.com.zip
   creating: mical-html/
  inflating: mical-html/about.html
  inflating: mical-html/contact.html
```

```
[root@www website1]# ls -ltr
total 5572
drwxr-xr-x. 7 root root     169 Dec 23  2019  mical-html
-rw-r--r--. 1 root root 5703509 May 27 12:52 'Mical Free Website Template - Free-CS
S.com.zip'
```

- Remove the downloaded zip file as we do not need it anymore.

```
rm -r Mical\ Free\ Website\ Template\ -\ Free-CSS.com.zip
```

```
drwxr-xr-x. 7 root root     169 Dec 23  2019  mical-html
-rw-r--r--. 1 root root 5703509 May 27 12:52 'Mical Free Website Template - Free-CSS.com.zip'
[root@www website1]# rm -r Mical\ Free\ Website\ Template\ -\ Free-CSS.com.zip
rm: remove regular file 'Mical Free Website Template - Free-CSS.com.zip'? y
[root@www website1]# ^C
[root@www website1]# ls -ltr
total 0
drwxr-xr-x. 7 root root 169 Dec 23  2019 mical-html
[root@www website1]#
```

- Open the directory `cd mical-html/` and copy the whole content to the directory `website1` and afterward remove the directory `mical-html

```
#Enter in directory `mical-html/`
cd mical-html/
```

```
# Copy the enitire content to the parent directory whischis `
cp -r * ..


#Now remove the extra folder which is `mical-html/
rm -rf mical-html
```

Repeat the same procedure for the second website

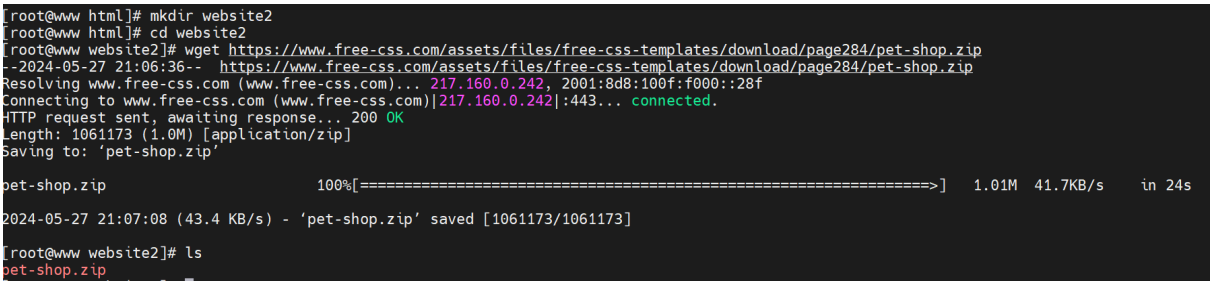
1. As we know we want to deploy two websites on our machine. For this reason we have to create two directories in the `/var/www/html
2. We will create two directories with name `website1` and `website2` . Where we will place our `website code.` .
3. Let's create the directory name `website2` in `/var/www/html/` and download the website code using `wget https://www.free-css.com/assets/files/free-css-templates/download/page284/pet-shop.zip` .Then unzip it and remove the unnecessary zipped file afterward.

```
cd /var/www/html/

mkdir website2

cd website1

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



4. The website `zip` file is downloaded as `pet-shop.zip`

5. Unzip the file using the command:

```
unzip pet-shop.zip
```

6. After unzipping remove the file `pet-shop.zip`

```
    rm -rf pet-shop.zip
```

7. Copy the entire content to the parent directory which is `var/www/html/website2`

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

8. Remove the extra directory

```
rm -rf pet-shop-website-template/
```

Step 6:

Creating Virtual Host , For Hosting two website on single Machine.

1. To make it possible that we will host two website . We have to use the concept of `virtual hosting`. In which we will create two file named `website1.conf` and `website2.conf` in the directory `cd /etc/httpd/conf.d`

```
touch website1.conf website2.conf
```

```
[root@www conf.d]# touch website1.conf website2.conf
[root@www conf.d]# ls -ltr
total 44
-rw-r--r--. 1 root root 8720 Apr 22 08:01 ssl.conf
-rw-r--r--. 1 root root  344 Apr 22 08:01 manual.conf
-rw-r--r--. 1 root root  653 Apr 22 08:01 welcome.conf
-rw-r--r--. 1 root root 1252 Apr 22 08:01 userdir.conf
-rw-r--r--. 1 root root  400 Apr 22 08:04 README
-rw-r--r--. 1 root root 2916 Apr 22 08:04 autoindex.conf
-rw-r--r--. 1 root root 8718 May 17 19:43 ssl.conf.rpmsave
-rw-r--r--. 1 root root    0 May 27 12:17 httpd.conf
-rw-r--r--. 1 root root    0 May 27 12:24 website2.conf
-rw-r--r--. 1 root root    0 May 27 12:24 website1.conf
```

2. Using the `vi editor` we will do some configuration in both files. Open the file `website1.conf` The configuration in the file name `website1.conf` will be :

```
<VirtualHost *:443>  # Defines a virtual host for incoming co
  SSLEngine on        # Enables the SSL engine for this virtu
  SSLCertificateFile /etc/pki/tls/certs/localhost.crt  # Path
  SSLCertificateKeyFile /etc/pki/tls/private/localhost.key  #
  ServerName www.carrepairing.id  # Specifies the server name
  DocumentRoot /var/www/html/website1  # The directory where
</VirtualHost>
```

```
[root@www conf.d]# vi website1.conf
[root@www conf.d]# cat website1.conf
<VirtualHost *:443>
SSLEngine on
SSLCertificateFile /etc/pki/tls/certs/localhost.crf
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
ServerName www.carrepairing.id
DocymentRoot /var/www/html/website1
```

3. Open the file `website2.conf` The configuration in the file name `website1.conf` will be :

```
<VirtualHost *:443>
SSLEngine on
SSLCertificateFile /etc/pki/tls/certs/localhost.crt
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
ServerName www.petshop.id
DocumentRoot /var/www/html/website2
</VirtualHost>
```

```
[root@www conf.d]# vi website2.conf
[root@www conf.d]# cat website2.conf
<VirtualHost *:443>
SSLEngine on
SSLCertificateFile /etc/pki/tls/certs/localhost.crf
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
ServerName www.petshop.id
DocymentRoot /var/www/html/website2
</VirtualHost>
```

Step 7:

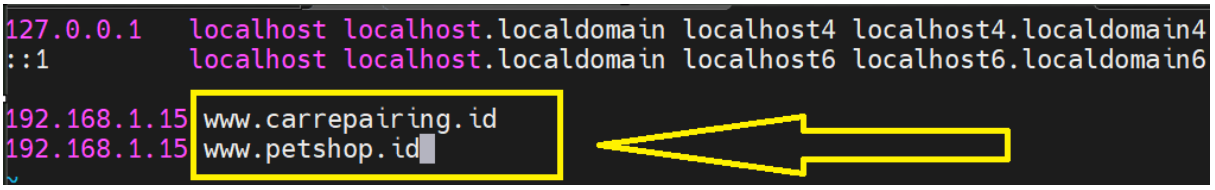
1. We have configured the directory, our website code is in their respective directory already.
2. In this step we will go to the directory `vim /etc/hosts` and define the `ip address` and `domain names`
3. Our `ip address` is `192.168.1.15`
4. The domain names are `www.carrepairing.id` and `www.petshop.id`
5. Open the file in vim editor:

```
vim /etc/hosts
```

6. Add the following details:

```
192.168.1.15 www.carrepairing.id
```

```
192.168.1.15 www.petshop.id
```



7. Save the file

Step 8:

Restart HTTPD service and Firewall setting:

- It is always a good practice to restart the `apache server` so run the following commands:

```
systemctl reset-failed httpd.service

systemctl start httpd.service

systemctl status httpd.service
```

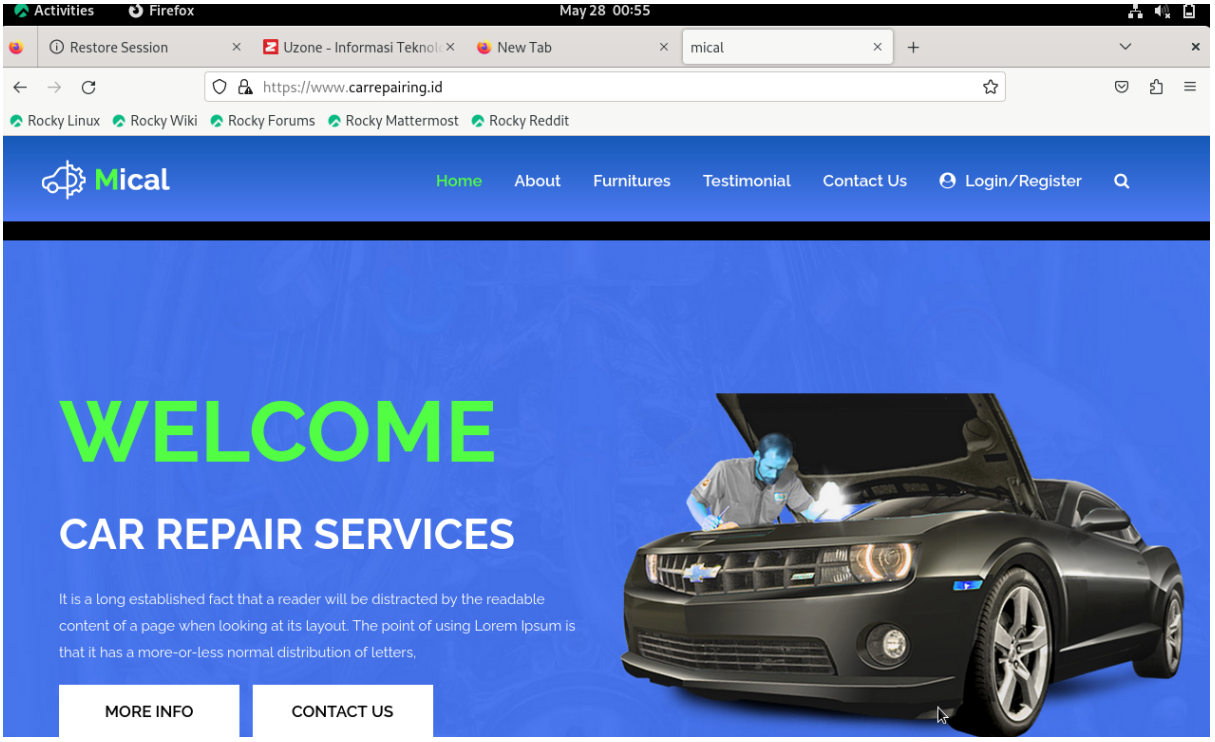
- **Firewall Configuration:** Ensure your firewall is configured correctly to allow HTTP and HTTPS traffic. So run these command one by one

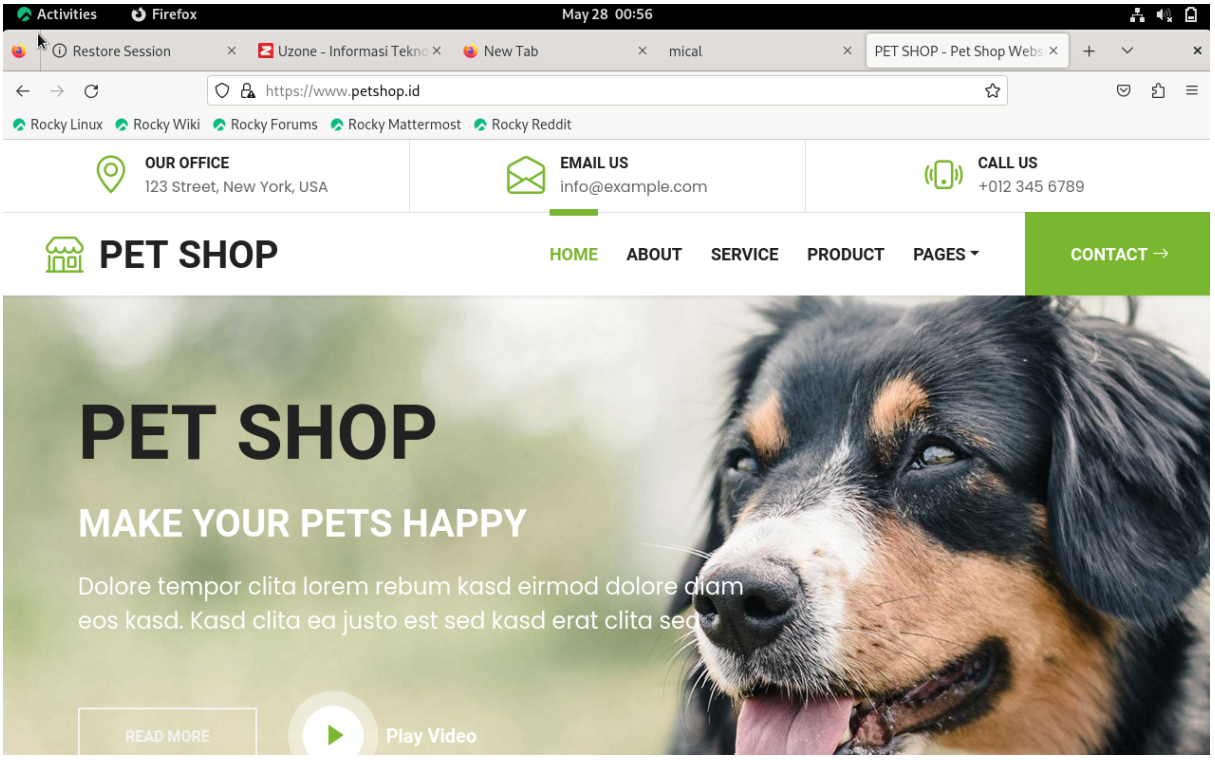
```
firewall-cmd --permanent --zone=public --add-service=http
firewall-cmd --permanent --zone=public --add-service=https
firewall-cmd --reload
```

Step 9:

Testing the website

- In your local machine open your web browser and enter url `www.carrepairing.id` and `www.petshop.id`





Conclusion

By following these steps, you will successfully host two websites on a single Apache server. This setup is efficient and demonstrates important skills in server configuration, virtual hosting, and secure communication using SSL/TLS.