**WordPress installation with Varnish**

The following are the prerequisites to have WordPress installed-

**Nginx**

**Database like MariaDB or MySQL**

**php**

**Install Nginx and configure it to run on port no. 8080**

sudo apt update -y

sudo apt install nginx -y

**Change the default port no 80 to 8080 in the ‘default’ folder in the following path–**

cd /etc/nginx/sites-enabled

vi default (Change the port from 80 to 8080)

sudo systemctl start nginx

sudo systemctl status nginx

sudo systemctl enable nginx (it will start the service at the next reboot)

**Install the MariaDB and configure it.**

sudo apt-get install mariadb-server

sudo systemctl enable mariadb.service

sudo mysql\_secure\_installation

**Now configure the database**

mysql -u root -p

Enter password:

MariaDB [mysql]> CREATE DATABASE wordpress\_db;

Query OK, 1 row affected (0.00 sec)

MariaDB [mysql]> GRANT ALL ON wordpress\_db.\* TO 'wpuser'@'localhost' IDENTIFIED BY 'Passw0rd!' WITH GRANT OPTION;

Query OK, 0 rows affected (0.00 sec)

MariaDB [mysql]> FLUSH PRIVILEGES;

Query OK, 0 rows affected (0.00 sec)

MariaDB [mysql]> exit

**Troubleshooting**

After running < mysql\_secure\_installation> command, an error might appear after resetting the password

Failed! Error: SET PASSWORD has no significance for user 'root'@'localhost' as the authentication method used doesn't store authentication data in the MySQL server. Please consider using ALTER USER instead if you want to change authentication parameters.

Then use the following steps

1. Open the terminal application.

2. Terminate the mysql\_secure\_installation from another terminal using the killall command:

sudo killall -9 mysql\_secure\_installation

3. Start the mysql client:

sudo mysql

4. Run the following SQL query:

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'SetRootPasswordHere';

exit

5. Then run the following command to secure it:

sudo mysql\_secure\_installation

6. When promoted for the password enter the SetRootPasswordHere (or whatever you set when you ran the above SQL query)

7. That is all.

**Now the database server is successfully configured.**

**Now download Php and wordpress**

sudo apt install php7.4 php7.4-gd php7.4-mysql php7.4-zip php7.4-fpm -y

**Download wordpress**

wget https://wordpress.org/latest.zip

unzip latest.zip

cd wordpress

sudo cp -r \* /var/www/html/ (This is the path where nginx will access the files)

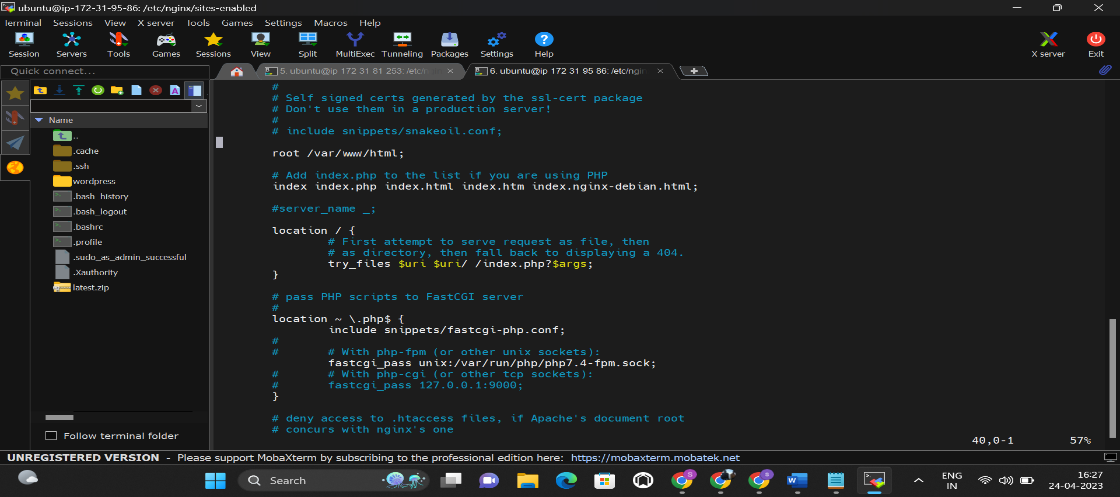
sudo chown -R www-data:www-data /var/www/html/ (need to give the user access )

sudo systemctl restart nginx

cd /etc/nginx/sites-enabled/

sudo mv default wordpress

sudo vim wordpress (Edit the file as below)

****

sudo nginx -t (to check the syntax of the file)

cd /var/www/html

mv wp-config-sample.php wp-config.php

vi wp-config.php

**Insert the database name and credentials here**

sudo systemctl restart nginx

**Now configure Varnish and make it run on port 80**

**Install the varnish service**

apt-get update

apt-get install varnish

**Verify the default.vcl configuration file**

cat /etc/varnish/default.vcl

{We can see in this file the backend server is configured at port 8080. Hence we configured Nginx to run on port 8080)

**Edit the configuration file named Varnish located inside the directory named default.**

vi /etc/default/varnish

**Here is the Varnish configuration file, before our configuration.**

START=yes

NFILES=131072

MEMLOCK=82000

DAEMON\_OPTS="-a :6081 \

-T localhost:6082 \

-f /etc/varnish/default.vcl \

-S /etc/varnish/secret \

-s malloc,256m"

**Here is the Varnish configuration file, after our configuration.**

DAEMON\_OPTS="-a :80 \

-T localhost:6082 \

-b localhost:8080 \

-u varnish -g varnish \

-S /etc/varnish/secret \

-s file,/var/lib/varnish/$INSTANCE/varnish\_storage.bin,256m"

**Next, edit the Varnish service configuration file.**

vi /lib/systemd/system/varnish.service

change the port number in front of -a to 80

**Now reload and restart the varnish service**

systemctl daemon-reload

service varnish restart

**Now we need to download Hitch for SSL connection** (The reason behind using Hitch is that Varnish doesn’t support SSL connection so we need to use Hitch for SSL termination)

**Install hitch**

apt install hitch

vi /etc/hitch/hitch.conf

frontend = "[\*]:443"

backend = "[127.0.0.1]:80"

pem-file = {

cert = "/etc/hitch/cert.pem"

private-key = "/etc/hitch/key.pem"

}

Exit the file

**Generate the SSL certificates**

## openssl req -newkey rsa:2048 -new -nodes -x509 -days 3650 -keyout key.pem -out cert.pem

Prefer generating the letsencrypt certificates. (Follow the link given in the sources section).

Follow the steps given in the link to generate the ssl certificates. (Webroot is /var/www/html)

**Now the cert file and key file are generated.**

**Check the permission on these two files. Use the following commands to set the appropriate permissions**

chmod 755 cert.pem

chmod 755 key.pem

systemctl restart hitch

systemctl status hitch (to check if the hitch is running)

{ # Additional info. Not necessary

To ask hitch to take a particular configuration file for running

hitch –config <path of hitch conf file>

}

**Now the following steps are used to redirect HTTP requests to HTTPS-**

**Make the following changes in the Varnish configuration file-**

First, add the line **import std;** just below vlc 4.0; Then write the below code in **sub vcl recv** section and add the following **sub vcl synth** section below it.

sub vcl\_recv {

if (std.port(server.ip) != 443) {

set req.http.location = "https://" + req.http.host + req.url;

return(synth(301));

}

}

sub vcl\_synth {

if (resp.status == 301) {

set resp.http.location = req.http.location;

set resp.status = 301;

return (deliver);

}

}

Also, need to make changes in vi /etc/default/varnish file   
Add **-a 127.0.0.1:8443,proxy** below -a :80

Also, need to make changes in vi /lib/systemd/system/varnish.service file

Add **-a 127.0.0.1:8443,proxy** ahead of -a :80

**Need to make the following changes in the hitch configuration file**

Need to change the backend to the **8443** port and need to add the following line

**write-proxy-v2 = on**

**Restart the hitch and varnish**

systemctl daemon-reload

systemctl restart hitch

systemctl restart varnish

-----------------------------------------------------------------------------------------------------

**The following is to check if the WordPress is configured properly or not before installing varnish and hitch**

**Link to access WordPress- http://<public-ip>/wp-admin/**

Note- As here the ip used is dynamic, the following steps need to be followed whenever the ip changes.

MYSQL Commands

Login- mysql -u root -p

**List Databases-** Show Databases;

**List tables-** Show tables;

**See table content-** Select \* from table\_name;

**Update-** update wp\_options set option\_value = "http://44.202.1.170:8080" where option\_id = "1";

{The above command replaces the existing IP with the new one which is being used}

---------------------------------------------------------------------------------------------------

**Challenges faced**

Faced difficulties while configuring hitch. Need to check the compatibility version of Hitch when using it on Ubuntu. The Hitch version used here is 1.5.2.

Also, need to check the configuration file of the hitch. The configuration file used here is simple and precise.

Had trouble with the redirection of the HTTP requests to HTTPS. The above steps ensured the proper redirection of HTTP requests to HTTPS. Need to make sure that the proxy is mentioned in the acceptance part of the varnish configuration file. Also, need to ensure the proxy protocol is on in the hitch configuration file.

-------------------------------------------------------------------------------------------------------------------------

**Link for references:**

<https://www.youtube.com/watch?v=vZjAhjqLakU>

<https://techexpert.tips/varnish/varnish-nginx-installation-ubuntu/>

<https://certbot.eff.org/instructions?ws=other&os=ubuntufocal> (for ssl generation)

<https://www.tecmint.com/enable-https-for-varnish-cache-on-centos/> (for redirection)

<https://wpaq.com/letsencrypt-nginx-certbot/> (letsencrypt ssl for nginx)