**The LAMP stack (Linux, Nginx, MariaDB, and PHP) is a popular server configuration for developing and hosting web applications. The four components of the stack are not tightly coupled, making it possible to substitute your preferred technologies. The LEMP stack is a common variant in which the Apache web server is replaced by NGINX.**

**The LEMP software stack is a group of software that can be used to serve dynamic web pages and web applications. This is an acronym that describes a Linux operating system, with an Nginx (pronounced like “Engine-X”) web server. The backend data is stored in the MariaDb database and the dynamic processing is handled by PHP.**

**install Nginx web server from the Ubuntu repository.**

apt install nginx -y

**Run the UFW command below.**

ufw allow 'Nginx HTTP'

**MariaDB is a popular fork of MySQL, and its development is considered to be more open and transparentthan MySQL’s. MariaDB is administered with the same commands as MySQL.**

apt install mariadb-server mariadb-client

**To secure the installation, MySQL comes with a script that will ask whether we want to modify some insecure defaults.**

mysql\_secure\_installation

**Create a test database and user with access permission for testing.**

mysql -u root -p

CREATE DATABASE testdb;

CREATE USER 'abhi'@'localhost' IDENTIFIED BY 'temp';

GRANT ALL PRIVILEGES ON testdb.\* TO 'abhi'@'localhost';

**Install the PHP FastCGI Processing Manager, which includes the core PHP dependencies also additional**

**helper package, php-mysql, which will allow PHP to communicate with your database backend.**

**The installation will pull in the necessary PHP core files:**

apt install php-fpm php-mysql -y

systemctl start php7.2-fpm

**Now craete website and also insatll phpmyadmin for web interface**

mkdir -p /var/www/html/sam.com

vi /var/www/html/sam.com/index.html

<welcome to my website>

chmod -Rf 775 /var/www/html/sam.com/

**Add the following content, which was taken and slightly modified from the default server**

**block configuration file, to your new server block configuration file-**

vi /etc/nginx/sites-available/sam.com.conf

server {

listen 80;

root /var/www/html/sam.com;

index index.php index.html index.htm index.nginx-debian.html;

server\_name sam.com www.sam.com;

location / {

try\_files $uri $uri/ =404;

}

location ~ \.php$ {

include snippets/fastcgi-php.conf;

fastcgi\_pass unix:/var/run/php/php7.2-fpm.sock;

}

location ~ /\.ht {

deny all;

}

}

***\*****\* location ~ \.php$ — This location block handles the actual PHP processing by pointing Nginx to the*

*fastcgi-php.conf configuration file and the php7.2-fpm.sock file, which declares what socket is*

*associated with php-fpm.*

***\*\**** *location ~ /\.ht — The last location block deals with .htaccess files, which Nginx does not process.*

*By adding the deny all directive, if any .htaccess files happen to find their way into the document*

*root they will not be served to visitors.*

**A Nginx server block is like a virtual host in Apache. We will not use the default server block**

**because it’s inadequate to run PHP code and if we modify it, it becomes a mess. So remove the**

**default symlink in sites-enabled directory by running the following command.**

**(It’s still available as /etc/nginx/sites-available/default.)**

rm /etc/nginx/sites-enabled/default OR unlink /etc/nginx/sites-enabled/default

**Enable your new server block by creating a symbolic link from your new server block configuration file to the /etc/nginx/sites-enabled/**

ln -s /etc/nginx/sites-available/sam.com.conf /etc/nginx/sites-enabled/

**for check evrything is ok or not check syntax by using**

nginx -t

**make a entry on host file in window (machine)**

192.168.72.92 www.sam.com sam.com

**Now you can install phpMyAdmin**

apt install phpmyadmin -y

**do not select Apache or Lighthttp Because we are using Nginx as web server,**

**Press tab and then OK to advance to the next step.**

**we need to create a symbolic link from the installation files to Nginx's document root directory:**

ln -s /usr/share/phpmyadmin /var/www/html/sam.com/

**now you can access your web site**

http://sam.com/

http://sam.com/phpmyadmin