# Magento 2.4 with LEMP Stack on Ubuntu

| SOFTWARE DEPENDENCIES | 2.4.5 | 2.4.4 | 2.4.3-P2 | 2.4.3 | 2.4.2 | 2.4.1 | 2.4.0 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Composer | 2.1 | 2.1 | 1 | 1 | 1 | 1 | 1 |
| Elasticsearch | 7.17 | 7.16 | 7.16 | 7.10 | 7.9 | 7.7 | 7.6 |
| OpenSearch | 1.2 | 1.2 | 1.2 | -- | -- | -- | -- |
| MariaDB | 10.4 | 10.4 | 10.4 | 10.4 | 10.4 | 10.4 | 10.2, 10.3, 10.4 |
| MySQL | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 5.7, 8.0 |
| PHP | 8.1 | 8.1 | 7.4 | 7.4 | 7.4 | 7.4 | 7.3, 7.4 |
| RabbitMQ | 3.9 | 3.9 | 3.8 | 3.8 | 3.8 | 3.8 | 3.8 |
| Redis | 6.2 | 6.2 | 6.0 | 6.0 | 6.0 | 5.0 | 5.0 |
| Varnish | 7.0 | 7.0 | 6.5 | 6.5 | 6.4 | 6.2 | 6 |
| Apache | 2.4 | 2.4 | 2.4 | 2.4 | 2.4 | 2.4 | 2.4 |
| nginx | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 |
| AWS Aurora (MySQL) | -- | -- | -- | 5.7 | -- | -- | -- |
| AWS S3 | -- | -- | -- | ✔️ | ✔️ | -- | -- |
| AWS MQ | -- | -- | -- | 3.8.11 | -- | -- | -- |
| AWS ElastiCache | -- | -- | -- | Redis 6 | -- | -- | -- |
| AWS ElasticSearch | -- | -- | -- | 7.9 | -- | -- | -- |

sudo apt-get install nginx

apt-get install software-properties-common

apt-key adv --recv-keys --keyserver hkp://keyserver.ubuntu.com:80 0xF1656F24C74CD1D8

add-apt-repository 'deb [arch=amd64,arm64,ppc64el] http://mirror.lstn.net/mariadb/repo/10.4/ubuntu bionic main'

apt update

apt install mariadb-server | systemctl enable mariadb.service

### **Install Elasticsearch**

cd /opt

wget<https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.6.1-amd64.deb>

dpkg -i elasticsearch-7.6.1-amd64.deb | systemctl start elasticsearch

curl -XGET '[http://localhost:9200](http://localhost:9200/)'

### **Install PHP 7.4**

add-apt-repository ppa:ondrej/php

apt-get install php7.4

apt install php7.4-fpm php7.4-common php7.4-mysql php7.4-gmp php7.4-curl php7.4-intl php7.4-mbstring php7.4-xmlrpc php7.4-gd php7.4-xml php7.4-cli php7.4-zip php7.4-bcmath

apt-get install git curl software-properties-common

nano /etc/php/7.4/fpm/php.ini

Change the following settings:

file\_uploads = On

allow\_url\_fopen = On

short\_open\_tag = On

memory\_limit = 256M

cgi.fix\_pathinfo = 0

zlib.output\_compression = On

upload\_max\_filesize = 128M

max\_execution\_time = 600

max\_input\_time = 900

date.timezone = America/Chicago

mysql\_secure\_installation

mysql -u root -p

create database db;

CREATE USER 'user'@'localhost' IDENTIFIED BY ['Password@11](mailto:'Password@11)';

GRANT ALL PRIVILEGES ON db.\* TO 'user'@'localhost';

flush priviledge;

## **Install Magento 2**

mkdir -p /var/www/magento2/

cd /var/www/magento2/

git clone https://github.com/magento/magento2.git /var/www/magento2/

curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/bin --filename=composer

cd /var/www/magento2

composer install

bin/magento setup:install --base-url=http://192.168.2.121/ --db-host=localhost --db-name=db --db-user=user --db-password=Password@11 --admin-firstname=sugandh --admin-lastname=kumar --admin-email=sugandh@gmail.com --admin-user=magentoadmin --admin-password=Password@11 --language=en\_US --currency=USD --timezone=America/Chicago –use-rewrites=1

**Do not forget to replace the domain name, database password, admin username, password and admin email address accordingly.**

[Progress: 699 / 701]

Post installation file permissions check...

For security, remove write permissions from these directories: '/var/www/magento2/app/etc'

[Progress: 700 / 701]

Write installation date...

[Progress: 701 / 701]

[SUCCESS]: Magento installation complete.

[SUCCESS]: Magento Admin URI: /admin\_1k4pc0

Nothing to import.

sudo chown -R www-data:www-data /var/www/magento2/

## **Create an Nginx Configuration File**

nano /etc/nginx/sites-available/magento2

upstream fastcgi\_backend {

server unix:/run/php/php7.4-fpm.sock;

}

server {

server\_name yourdomain.com;

listen 80;

set $MAGE\_ROOT /var/www/magento2;

set $MAGE\_MODE developer; # or production

access\_log /var/log/nginx/magento2-access.log;

error\_log /var/log/nginx/magento2-error.log;

include /var/www/magento2/nginx.conf.sample;

}

rm -f /etc/nginx/sites-enabled/default

ln -s /etc/nginx/sites-available/magento2 /etc/nginx/sites-enabled/magento2

nginx -t

nginx: the configuration file /etc/nginx/nginx.conf syntax is ok

nginx: configuration file /etc/nginx/nginx.conf test is successful

Setup sample Data on Magento

cd */var /www /*

sudo git clone https://github.com/magento/magento2-sample-data.git

cd /var/www/magento2

sudo chown -R sugandh:www-data /var/www/magento2

sudo chmod -R 777 var pub

php -f /var/www/magento2-sample-data/dev/tools/build-sample-data.php -- --ce-source=/var/www/magento2

sudo chown -R www-data:www-data /var/www/magento2/

php bin/magento setup:upgrade

sudo systemctl restart nginx.service