## **Installation of Magento2.4.5 and LEMP(Nginx) stack**

### **Run Update & Upgrade Command**

**sudo apt update -y**

### **Install Nginx**

**sudo apt install nginx -y**

### **Install mysql-server**

**sudo apt install mysql-server -y**

### **Install php8.1**

**sudo apt-get install software-properties-common -y**

**sudo add-apt-repository ppa:ondrej/php**

**sudo apt-get update**

**sudo apt install php8.1**

### Install php8.1-Modules

apt install php8.1-bcmath php8.1-common php8.1-curl php8.1-fpm php8.1-gd php8.1-intl php8.1-mbstring php8.1-mysql php8.1-soap php8.1-xml php8.1-xsl php8.1-zip

**sudo apt install** **php8.1-{amqp,inotify,rrd,fpm,interbase,smbclient,ast,intl, snmp,bcmath,ldap,soap,bz2,mailparse,cgi,maxminddb,sqlite3,cli,mbstring,ssh2,common,mcrypt,stomp,curl,memcache,swoole,dba,memcached,sybase,decimal,tideways,dev,msgpack,tidy,ds,mysql,uopz,enchant,oauth,uploadprogress,facedetect,odbc,uuid,opcache,vips,gd,pcov,xdebug,gearman,pgsql,xhprof,gmagick,phpdbg,xml,gmp,protobuf,xmlrpc,gnupg,ps,xsl,grpc,pspell,http,psr,yaml,igbinary,raphf,zip,readline,zmq,imap,redis,zstd} -y**

### Configure PHP

sudo chown -R $USER:$USER /var/www/html/

sudo nano /etc/nginx/sites-available/default

### Uncomment the lines as followed below:-

location ~ \.php$ {

include snippets/fastcgi-php.conf;

#

# # With php-fpm (or other unix sockets):

fastcgi\_pass unix:/run/php/php8.1-fpm.sock;

# # With php-cgi (or other tcp sockets):

# fastcgi\_pass 127.0.0.1:9000;

}

location ~ /\.ht {

deny all;

}

location = /favicon.ico { log\_not\_found off; access\_log off; }

location = /robots.txt { log\_not\_found off; access\_log off; allow all; }

location ~\* \.(css|gif|ico|jpeg|jpg|js|png)$ {

expires max;

log\_not\_found off;

}

### Changes

/etc/php/8.1/fpm/php.ini

/etc/php/8.1/cli/php.ini

memory\_limit = 1G

max\_execution\_time = 1800

zlib.output\_compression = On

### Check php

**sudo nano /var/www/html/info.php**

**<?php**

**phpinfo();**

**?>**

**#open Browser :- http://localhost/info.php**

### Install PhpMyAdmin

sudo apt install phpmyadmin -y

sudo ln -s /usr/share/phpmyadmin /var/www/html

### Download elastic Search

cd ~/Downloads

wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.17.0-amd64.deb

chmod +x elasticsearch-7.17.0-amd64.deb

sudo dpkg -i elasticsearch-7.17.0-amd64.deb

### Download Composer

cd ~/Downloads

wget https://getcomposer.org/download/2.2.18/composer.phar

mv composer.phar composer

mv composer /usr/local/bin

cd /usr/local/bin

sudo chmod -R 777 composer

composer

### Download magento2.4.5

cd ~/Downloads

wget https://github.com/magento/magento2/archive/refs/tags/2.4.5.tar.gz

tar -xvf 2.4.5.tar.gz

mv magento2-2.4.5 magento2

sudo mv magento2 /var/www/html/

cd /var/www/html/magento2

composer install

### Permissions

| cd /var/www/html/<magento install directory> |
| --- |

| find var generated vendor pub/static pub/media app/etc -type f -exec chmod g+w {} + |
| --- |

| find var generated vendor pub/static pub/media app/etc -type d -exec chmod g+ws {} + |
| --- |

| chown -R :www-data . # Ubuntu |
| --- |

| chmod u+x bin/magento |
| --- |

### Install Magento2

bin/magento setup:install --base-url=https://www.magento2ced.com/ --db-host=localhost --db-name=magento --db-user=magento --db-password=password --admin-firstname=shubham --admin-lastname=bauddh --admin-email=shubham@cedcommerce.com --admin-user=admin --admin-password=admin@123 --language=en\_US --currency=USD --timezone=America/Chicago --use-rewrites=1

| cd /var/www/html/magento2/ |
| --- |

| bin/magento deploy:mode:set developer |
| --- |

sudo nano /etc/nginx/sites-available/magento

upstream fastcgi\_backend {

server unix:/run/php/php8.1-fpm.sock;

}

server {

listen 80;

server\_name www.magento-dev.com;

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

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

}

php bin/magento setup:upgrade

php bin/magento setup:static-content:deploy -f

php bin/magento indexer:reindex

php bin/magento cache:flush

### If CSS Error Occurred

sudo chown -R www-data:www-data \*

Change base url  
bin/magento setup:store-config:set --base-url="https://yourdomain.com/"

bin/magento setup:store-config:set --base-url-secure="<https://yourdomain.com/>"

============================================================================

### OpenSSL Command - SSL Setup

cd /etc/nginx/sites-available/

openssl req -newkey rsa:4096 -x509 -sha256 -days 3650 -nodes -out magento2ced.crt -keyout magento2ced.key

============================================================================

### Creating magento configuration file for domain mapping

sudo nano /etc/nginx/sites-available/magento

upstream fastcgi\_backend {

server unix:/run/php/php8.1-fpm.sock;

}

server {

listen 80;

listen listen 443 ssl;

server\_name www.magento2ced.com;

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

set $MAGE\_MODE developer;

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

ssl\_certificate /etc/nginx/sites-available/magento2ced.crt;

ssl\_certificate\_key /etc/nginx/sites-available/magento2ced.key;

}

============================================================================

### Changes in default configuration file

sudo nano /etc/nginx/sites-available/default

##

# You should look at the following URL's in order to grasp a solid understanding

# of Nginx configuration files in order to fully unleash the power of Nginx.

# https://www.nginx.com/resources/wiki/start/

# https://www.nginx.com/resources/wiki/start/topics/tutorials/config\_pitfalls/

# https://wiki.debian.org/Nginx/DirectoryStructure

#

# In most cases, administrators will remove this file from sites-enabled/ and

# leave it as reference inside of sites-available where it will continue to be

# updated by the nginx packaging team.

#

# This file will automatically load configuration files provided by other

# applications, such as Drupal or Wordpress. These applications will be made

# available underneath a path with that package name, such as /drupal8.

#

# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.

##

# Default server configuration

#

server {

listen 80 default\_server;

listen [::]:80 default\_server;

# SSL configuration

#

# listen 443 ssl default\_server;

# listen [::]:443 ssl default\_server;

#

# Note: You should disable gzip for SSL traffic.

# See: https://bugs.debian.org/773332

#

# Read up on ssl\_ciphers to ensure a secure configuration.

# See: https://bugs.debian.org/765782

#

# Self signed certs generated by the ssl-cert package

# Don't use them in a production server!

#

# include snippets/snakeoil.conf;

root /var/www/html;

# Add index.php to the list if you are using PHP

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

server\_name \_;

location / {

# First attempt to serve request as file, then

# as directory, then fall back to displaying a 404.

try\_files $uri $uri/ =404;

}

# pass PHP scripts to FastCGI server

#

location ~ \.php$ {

include snippets/fastcgi-php.conf;

#

# # With php-fpm (or other unix sockets):

fastcgi\_pass unix:/run/php/php8.1-fpm.sock;

# # With php-cgi (or other tcp sockets):

# fastcgi\_pass 127.0.0.1:9000;

}

location = /favicon.ico {

log\_not\_found off;

access\_log off;

}

location = /robots.txt {

log\_not\_found off;

access\_log off;

allow all;

}

location ~\* \.(css|gif|ico|jpeg|jpg|js|png)$ {

expires max;

log\_not\_found off;

}

# deny access to .htaccess files, if Apache's document root

# concurs with nginx's one

#

location ~ /\.ht {

deny all;

}

}

# Virtual Host configuration for example.com

#

# You can move that to a different file under sites-available/ and symlink that

# to sites-enabled/ to enable it.

#

#server {

# listen 80;

# listen [::]:80;

#

# server\_name www.magetno2ced.com;

#

# root /var/www/html;

# index index.html index.php nginx.conf.sample;

#

# location / {

# try\_files $uri $uri/ =404;

# }

#}

============================================================================

## **Configure Varnish in Magento2.4.5 with SSL**

sudo apt update

### Register Varnish Cache 7 repo on Ubuntu 22.04|20.04|18.04

sudo apt install debian-archive-keyring curl gnupg apt-transport-https -y

curl -fsSL https://packagecloud.io/varnishcache/varnish70/gpgkey|sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/varnish.gpg

### Ubuntu 20.04 / 18.04:

. /etc/os-release

sudo tee /etc/apt/sources.list.d/varnishcache\_varnish70.list > /dev/null <<-EOF

deb https://packagecloud.io/varnishcache/varnish70/$ID/ $VERSION\_CODENAME main

deb-src https://packagecloud.io/varnishcache/varnish70/$ID/ $VERSION\_CODENAME main

EOF

### Ubuntu 22.04:

sudo tee /etc/apt/sources.list.d/varnishcache\_varnish70.list > /dev/null <<-EOF

deb https://packagecloud.io/varnishcache/varnish70/ubuntu/ focal main

deb-src https://packagecloud.io/varnishcache/varnish70/ubuntu/ focal main

EOF

sudo apt update

### Install Varnish Cache 7 on Ubuntu 22.04|20.04|18.04

sudo apt install varnish

### Create Systemd service file for Varnish Cache 7

sudo cp /lib/systemd/system/varnish.service /etc/systemd/system/

### Configure the listening port and cache size

### sudo nano /etc/systemd/system/varnish.service

[Unit]

Description=Varnish Cache, a high-performance HTTP accelerator

After=network-online.target nss-lookup.target

[Service]

Type=forking

KillMode=process

# Maximum number of open files (for ulimit -n)

LimitNOFILE=131072

# Locked shared memory - should suffice to lock the shared memory log

# (varnishd -l argument)

# Default log size is 80MB vsl + 1M vsm + header -> 82MB

# unit is bytes

LimitMEMLOCK=85983232

# Enable this to avoid "fork failed" on reload.

TasksMax=infinity

# Maximum size of the corefile.

LimitCORE=infinity

ExecStart=/usr/sbin/varnishd \

-a :80 \

-a localhost:8443,PROXY \

-p feature=+http2 \

-f /etc/varnish/default.vcl \

-s malloc,2g

ExecReload=/usr/sbin/varnishreload

[Install]

WantedBy=multi-user.target

### Reload Daemon

sudo systemctl daemon-reload

### Restart Varnish

sudo systemctl restart varnish

## **Configure the Nginx to work with Varnish**

### sudo nano /etc/nginx/sites-available/magento

upstream fastcgi\_backend {

server unix:/run/php/php8.1-fpm.sock;

}

server {

server\_name www.magento2ced.com magento2ced.com;

listen 8080;

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

set $MAGE\_MODE developer;

access\_log /var/log/nginx/magento2ced.com-access.log;

error\_log /var/log/nginx/magento2ced.com-error.log;

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

}

server {

listen 443 default\_server ssl;

server\_name www.magento2ced.com magento2ced.com;

ssl\_certificate /etc/nginx/sites-available/magento2ced.crt;

ssl\_certificate\_key /etc/nginx/sites-available/magento2ced.key;

location / {

proxy\_pass http://192.168.2.182:80;

proxy\_set\_header Host $http\_host;

proxy\_set\_header X-Forwarded-Host $http\_host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header Ssl-Offloaded "1";

proxy\_set\_header X-Forwarded-Port 443;

proxy\_set\_header X-Forwarded-Proto https;

proxy\_set\_header X-Forwarded-Proto $scheme;

}

}

### sudo ln -s /etc/nginx/sites-available/magento /etc/nginx/sites-enabled/

### sudo nginx -t

### sudo systemctl restart nginx

## **Configure the VCL backend**

### sudo nano /etc/varnish/default.vcl

vcl 4.1;

backend default {

.host = "127.0.0.1";

.port = "8080";

}

sub vcl\_recv {

}

sub vcl\_backend\_response {

}

sub vcl\_deliver {

}

### sudo nano /lib/systemd/system/varnish.service

Description=Varnish Cache, a high-performance HTTP accelerator

After=network-online.target nss-lookup.target

[Service]

Type=forking

KillMode=process

# Maximum number of open files (for ulimit -n)

LimitNOFILE=131072

# Locked shared memory - should suffice to lock the shared memory log

# (varnishd -l argument)

# Default log size is 80MB vsl + 1M vsm + header -> 82MB

# unit is bytes

LimitMEMLOCK=85983232

# Enable this to avoid "fork failed" on reload.

TasksMax=infinity

# Maximum size of the corefile.

LimitCORE=infinity

ExecStart=/usr/sbin/varnishd \

-a :6081 \

-a localhost:8443,PROXY \

-p feature=+http2 \

-f /etc/varnish/default.vcl \

-s malloc,256m

ExecReload=/usr/sbin/varnishreload

[Install]

WantedBy=multi-user.target

### sudo systemctl restart nginx varnish

## **Verify Varnish Cache 7 Server**

### curl -I domain\_or\_ip

magentolemp@magentolemp:~$ curl -I [www.magento2ced.com](http://www.magento2ced.com)

HTTP/1.1 302 Found

Server: nginx/1.18.0 (Ubuntu)

Date: Thu, 29 Sep 2022 12:39:32 GMT

Content-Type: text/html; charset=UTF-8

Set-Cookie: PHPSESSID=gsa2hqbn6ihlafr0aiuivdet8u; expires=Thu, 29-Sep-2022 13:39:32 GMT; Max-Age=3600; path=/; domain=www.magento2ced.com; HttpOnly; SameSite=Lax

Location: <https://www.magento2ced.com/>

Pragma: no-cache

Cache-Control: max-age=0, must-revalidate, no-cache, no-store

Expires: Wed, 29 Sep 2021 12:39:32 GMT

X-Magento-Debug: 1

Content-Security-Policy-Report-Only: font-src data: 'self' 'unsafe-inline'; form-action geostag.cardinalcommerce.com geo.cardinalcommerce.com 1eafstag.cardinalcommerce.com 1eaf.cardinalcommerce.com centinelapistag.cardinalcommerce.com centinelapi.cardinalcommerce.com pilot-payflowlink.paypal.com www.paypal.com www.sandbox.paypal.com 'self' 'unsafe-inline'; frame-ancestors 'self'; frame-src fast.amc.demdex.net \*.adobe.com bid.g.doubleclick.net geostag.cardinalcommerce.com geo.cardinalcommerce.com 1eafstag.cardinalcommerce.com 1eaf.cardinalcommerce.com centinelapistag.cardinalcommerce.com centinelapi.cardinalcommerce.com www.paypal.com www.sandbox.paypal.com pilot-payflowlink.paypal.com player.vimeo.com \*.youtube.com 'self' 'unsafe-inline'; img-src assets.adobedtm.com amcglobal.sc.omtrdc.net dpm.demdex.net cm.everesttech.net \*.adobe.com widgets.magentocommerce.com data: www.googleadservices.com www.google-analytics.com googleads.g.doubleclick.net www.google.com bid.g.doubleclick.net analytics.google.com www.googletagmanager.com t.paypal.com www.paypal.com www.paypalobjects.com fpdbs.paypal.com fpdbs.sandbox.paypal.com \*.vimeocdn.com i.ytimg.com \*.youtube.com validator.swagger.io data: 'self' 'unsafe-inline'; script-src assets.adobedtm.com \*.adobe.com www.googleadservices.com www.google-analytics.com googleads.g.doubleclick.net analytics.google.com www.googletagmanager.com geostag.cardinalcommerce.com 1eafstag.cardinalcommerce.com geoapi.cardinalcommerce.com 1eafapi.cardinalcommerce.com songbird.cardinalcommerce.com includestest.ccdc02.com www.paypal.com www.sandbox.paypal.com www.paypalobjects.com t.paypal.com s.ytimg.com www.googleapis.com vimeo.com www.vimeo.com \*.vimeocdn.com \*.youtube.com 'self' 'unsafe-inline' 'unsafe-eval'; style-src \*.adobe.com 'self' 'unsafe-inline'; object-src 'self' 'unsafe-inline'; media-src \*.adobe.com 'self' 'unsafe-inline'; manifest-src 'self' 'unsafe-inline'; connect-src dpm.demdex.net amcglobal.sc.omtrdc.net www.google-analytics.com www.googleadservices.com analytics.google.com www.googletagmanager.com geostag.cardinalcommerce.com geo.cardinalcommerce.com 1eafstag.cardinalcommerce.com 1eaf.cardinalcommerce.com centinelapistag.cardinalcommerce.com centinelapi.cardinalcommerce.com www.sandbox.paypal.com www.paypalobjects.com www.paypal.com pilot-payflowlink.paypal.com 'self' 'unsafe-inline'; child-src http: https: blob: 'self' 'unsafe-inline'; default-src 'self' 'unsafe-inline' 'unsafe-eval'; base-uri 'self' 'unsafe-inline';

X-Content-Type-Options: nosniff

X-XSS-Protection: 1; mode=block

X-Frame-Options: SAMEORIGIN

Vary: Accept-Encoding

X-Varnish: 655509

Age: 0

Via: 1.1 varnish (Varnish/7.0)

Connection: keep-alive